

1 YEAR UPGRADE

BUYER PROTECTION PLAN

www.sharexxx.net - free books & magazines



CISCO® Security Professional's

Guide to

Secure Intrusion Detection Systems

Your Complete Guide to Cisco Enterprise IDS Management

- Complete Coverage of the Cisco Secure Policy Manager (CSPM)
- Step-by-Step Instructions for Installing, Configuring, and Using the Cisco Intrusion Detection Sensor
- Includes Coverage of the Cisco Secure Intrusion Detection Systems Exam (CSIDS 9E0-100)

C. Tate Baumrucker

James D. Burton

Scott Dentler

Ido Dubrawsky

Vitaly Osipov

Michael Sweeney Technical Editor

s o l u t i o n s @ s y n g r e s s . c o m

With more than 1,500,000 copies of our MCSE, MCSA, CompTIA, and Cisco study guides in print, we continue to look for ways we can better serve the information needs of our readers. One way we do that is by listening.

Readers like yourself have been telling us they want an Internet-based service that would extend and enhance the value of our books. Based on reader feedback and our own strategic plan, we have created a Web site that we hope will exceed your expectations.

Solutions@syngress.com is an interactive treasure trove of useful information focusing on our book topics and related technologies. The site offers the following features:

- One-year warranty against content obsolescence due to vendor product upgrades. You can access online updates for any affected chapters.
- “Ask the Author” customer query forms that enable you to post questions to our authors and editors.
- Exclusive monthly mailings in which our experts provide answers to reader queries and clear explanations of complex material.
- Regularly updated links to sites specially selected by our editors for readers desiring additional reliable information on key topics.

Best of all, the book you’re now holding is your key to this amazing site. Just go to **www.syngress.com/solutions**, and keep this book handy when you register to verify your purchase.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there’s anything else we can do to help you get the maximum value from your investment. We’re listening.

www.syngress.com/solutions

S Y N G R E S S[®]

CISCO SECURITY PROFESSIONAL'S
Guide to

Secure Intrusion Detection Systems

James Burton

Ido Dubrawsky

Vitaly Osipov

C. Tate Baumrucker

Michael Sweeney Technical Editor

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively “Makers”) of this book (“the Work”) do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable care, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, “Career Advancement Through Skill Enhancement®,” “Ask the Author UPDATE®,” and “Hack Proofing®,” are registered trademarks of Syngress Publishing, Inc. “Syngress: The Definition of a Serious Security Library”™, “Mission Critical™,” and “The Only Way to Stop a Hacker is to Think Like One™” are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY	SERIAL NUMBER
001	PK9H7GYV43
002	Q2UN7T6CVF
003	8J9HF5TX3A
004	Z2B76NH89Y
005	U8MPT5R33S
006	X6B7NC4ES6
007	G8D4EPQ2AK
008	9BKMUJ6RD7
009	SW4KP7V6FH
010	5BVF7UM39Z

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Cisco Security Professional's Guide to Secure Intrusion Detection Systems

Copyright © 2003 by Syngress Publishing, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0

ISBN: 1-932266-69-0

Technical Editor: Michael Sweeney
Acquisitions Editor: Mike Rubin
Cover Designer: Michael Kavish

Page Layout and Art by: Patricia Lupien
Copy Editor: Mike McGee
Indexer: Odessa & Cie

Distributed by Publishers Group West in the United States and Jaguar Book Group in Canada.



Acknowledgments

We would like to acknowledge the following people for their kindness and support in making this book possible.

Ralph Troupe and the team at Callisma for their invaluable insight into the challenges of designing, deploying and supporting world-class enterprise networks.

Karen Cross, Meaghan Cunningham, Kim Wylie, Harry Kirchner, Kevin Votel, Kent Anderson, Frida Yara, Jon Mayes, John Mesjak, Peg O'Donnell, Sandra Patterson, Betty Redmond, Roy Remer, Ron Shapiro, Patricia Kelly, Andrea Tetrick, Jennifer Pascal, Doug Reil, David Dahl, Janis Carpenter, and Susan Fryer of Publishers Group West for sharing their incredible marketing experience and expertise.

The incredibly hard working team at Elsevier Science, including Jonathan Bunkell, AnnHelen Lindeholm, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, and Rosie Moss for making certain that our vision remains worldwide in scope.

David Buckland, Wendi Wong, Daniel Loh, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, and Joseph Chan of Transquest Publishers for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

Jackie Gross, Gayle Voycey, Alexia Penny, Anik Robitaille, Craig Siddall, Darlene Morrow, Iolanda Miller, Jane Mackay, and Marie Skelly at Jackie Gross & Associates for all their help and enthusiasm representing our product in Canada.

Lois Fraser, Connie McMenemy, Shannon Russell, and the rest of the great folks at Jaguar Book Group for their help with distribution of Syngress books in Canada.

David Scott, Annette Scott, Delta Sams, Geoff Ebbs, Hedley Partis, and Tricia Herbert of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji Tonga, Solomon Islands, and the Cook Islands.

Winston Lim of Global Publishing for his help and support with distribution of Syngress books in the Philippines.



Contributors

Pieter J. Bakhuijzen (CCIE #11033, CCDP, JNCIA-M, MCSE) is the owner of iXio Networks, a Netherlands-based network security consulting and training company. He specializes in network and security implementation and design, based on Cisco, Nokia, and Check Point products. Before starting his own company he worked for companies in the service provider, financial and publishing industry, such as Demon Internet, TeliaSonera, Kluwer Academic Publishers, and Formus Communications. Pieter Jan currently resides in the city of The Hague in The Netherlands where he is preparing to take the CCIE Security Lab exam.

C. Tate Baumrucker (CISSP, CCNP, Sun Enterprise Engineer, MCSE) is responsible for leading engineering teams in the design and implementation of complex and highly available systems infrastructures and networks. Tate is industry recognized as a subject matter expert in security and LAN/WAN support systems such as HTTP, SMTP, DNS, and DHCP. He has spent eight years providing technical consulting services in enterprise and service provider industries for companies including American Home Products, Blue Cross and Blue Shield of Alabama, Amtrak, Iridium, National Geographic, Geico, GTSI, Adelphia Communications, Digex, Cambrian Communications, and BroadBand Office.

James D. Burton (CISSP, CCNA, MCSE) is a Colorado Springs-based Systems Security Engineer for Northrop Grumman Mission Systems. He currently works at the Joint National Integration Center performing information assurance functions. James has over eight years of security experience having started his career as a Terminal Area Security Officer with the United States Marine Corps. His strengths include Cisco PIX firewalls and IDSs, and freeware intrusion detection systems. James holds a Master's degree from Colorado Technical University. He is deeply appreciative of his wife Melissa whose support of his information security career has helped keep him focused.

Scott Dentler (CISSP, CCSE, CCSA, MCSE, CCNA) is an IT consultant who has served with companies such as Sprint and H&R Block, giving him exposure to large enterprise networks and corporate environments. He is currently providing systems support for a campus network at a medical center with national affiliations. Scott's

background includes a broad range of information technology facets, including Cisco routers and switches, Microsoft NT/2000/XP, Check Point firewalls and VPNs, Red Hat Linux, network analysis and enhancement, network design and architecture, and network IP allocation and addressing. He has also prepared risk assessments and used that information to prepare business continuity and disaster recovery plans for knowledge-based systems. Scott is a contributing author for *Snort 2.0 Intrusion Detection* (Syngress Publishing, ISBN: 1-931836-74-4).

Ido Dubrawsky (CCNA, SCSA) has been working as a UNIX/Network Administrator for over 10 years. He has experience with a variety of UNIX operating systems including Solaris, Linux, BSD, HP-UX, AIX, and Ultrix. He was previously a member of Cisco's Secure Consulting Service providing security posture assessments to Cisco customers and is currently a member of the SAFE architecture team. Ido has written articles and papers on topics in network security such as IDS, configuring Solaris virtual private networks, and wireless security. Ido is a contributing author for *Hack Proofing Sun Solaris 8* (Syngress Publishing, ISBN: 1-928994-44-X) and *Hack Proofing Your Network, Second Edition* (Syngress, ISBN: 1-928994-70-9). When not working on network security issues or traveling to conferences, Ido spends his free time with his wife and their children.

Vitaly Osipov (CISSP, CCSA, CCSE) is a Security Specialist who has spent the last five years consulting various companies in Eastern, Central, and Western Europe on information security issues. Last year Vitaly was busy with the development of managed security service for a data center in Dublin, Ireland. He is a regular contributor to various infosec-related mailing lists and Syngress publications, and recently co-authored *Check Point NG Certified Security Administrator Study Guide*. Vitaly has a degree in mathematics. He lives in Australia.



Technical Editor, Contributor and Technical Reviewer

Michael Sweeney (CCNA, CCDA, CCNP, MCSE) is the owner of the network consulting firm Packetattack.com. His specialties are network design, network troubleshooting, wireless network design, security, and network analysis using NAI Sniffer and Airmagnet for wireless network analysis. Michael's prior published works include *Cisco Security Specialist's Guide to PIX Firewalls* (Syngress Publishing, ISBN: 1-931836-63-9). Michael is a graduate of the University of California, Irvine, extension program with a certificate in communications and network engineering. Michael resides in Orange, California with his wife Jeanne and daughter Amanda.

Contents

Foreword	xxiii
Chapter 1 Introduction to Intrusion Detection Systems	1
Introduction	2
Understanding the AVVID Architecture	3
Understanding the SAFE Blueprint	6
The Network Campus Area	7
The Small Campus Module	8
The Medium Campus Module	8
The Enterprise Campus	8
The Network Edge Area	10
The Remote User Network Edge	10
The Small Network Edge	11
The Medium Network Edge	12
The Enterprise Network Edge	12
The Internet Service Provider Area	13
SAFE Axioms	14
The Cisco Security Wheel	15
Corporate Security Policy	16
Secure	17
Access Control	17
Encryption	18
Authentication	18
Vulnerability Patching	18
Monitor and Respond	19
Test	19
Manage and Improve	20
Threats	20
Unstructured Threats	21

Structured Threats	21
External Threats	22
Internal Threats	22
Network Attacks	22
Reconnaissance Attacks	22
Access Attacks	23
Data Retrieval	23
System Access	24
Privilege Escalation	24
DoS Attacks	24
Anatomy of an Attack	25
Overview of IDS	25
Types of IDS	26
Network IDS	26
Host IDS	27
Others	28
How Does IDS Work?	28
Signature-Based IDS	30
Anomaly-Based IDS	31
Defeating an IDS	32
Summary	34
Solutions Fast Track	35
Frequently Asked Questions	37
Chapter 2 Cisco Intrusion Detection	39
Introduction	40
What Is Cisco Intrusion Detection?	41
Cisco's Network Sensor Platforms	42
Cisco IDS Appliances	43
4210 Sensor	45
4215 Sensor	45
4230 Sensor	45
4235 Sensor	46
4250 Sensor	46
4250 XL Sensor	46
The Cisco IDS Module for Cisco 2600, 3600, and 3700 Routers	46

The Cisco 6500 Series IDS Services Module	47
Cisco's Host Sensor Platforms	49
Cisco Host Sensor	50
Managing Cisco's IDS Sensors	51
Cisco PostOffice Protocol	53
Remote Data Exchange Protocol	55
Deploying Cisco IDS Sensors	56
Understanding and Analyzing the Network	57
Identifying the Critical Infrastructure and Services	58
Placing Sensors Based on Network and Services Function	59
Case Study 1: Small IDS Deployment	60
Case Study 2: Complex IDS Deployment	62
Summary	69
Solutions Fast Track	70
Frequently Asked Questions	72
Chapter 3 Initializing Sensor Appliances	75
Introduction	76
Identifying the Sensor	76
Initializing the Sensor	79
What Is the root User?	81
What Is the netrangr User?	83
What Is sysconfig-sensor?	83
Configuring the Sensor	83
The Display	93
Using the Sensor Command-Line Interface	94
cidServer	95
idsstatus	95
idsconns	96
idsvers	97
idsstop	97
idsstart	98
Configuring the SPAN Interface	98
Spanning Ports	99
Spanning VLANs	99
Recovering the Sensor's Password	100
Reinitializing the Sensor	102

Downloading the Image	102
Using the CD	102
Using the Recovery Partition	103
Uninstalling an Image	107
Upgrading a Sensor from 3.1 to 4.0	107
Upgrading a Sensor BIOS	108
Initializing a Version 4.0 Sensor	109
Summary	113
Solutions Fast Track	114
Frequently Asked Questions	117
Chapter 4 Cisco IDS Management	119
Introduction	120
Managing the IDS Overview	121
Using the Cisco Secure Policy Manager	123
Installing CSPM	123
Logging In to CSPM	128
Configuring CSPM	129
Adding a Network	130
Adding a Host	132
Adding a Sensor	135
The Properties Tab	137
The Sensing Tab	138
The Blocking Tab	139
The Filtering Tab	142
The Logging Tab	145
The Advanced Tab	146
The Command Tab	148
The Control Tab	149
Signature Updates	150
Configuring IPSec	151
Viewing Alarms	152
Using the CSID Director for Unix	155
Installing and Starting the Director	155
How to Configure the CSID Director	157
Adding a New Sensor	157
Event Processing	159

Using the IDS Device Manager	160
How to Configure IDS Device Manager	161
Logging In	162
Configuring the IDS Device Manager	164
The Device Tab	165
The Configuration Tab	168
The Monitoring Tab	172
The Administration Tab	175
Using the Cisco Network Security Database	178
Summary	180
Solutions Fast Track	180
Frequently Asked Questions	183
Chapter 5 Configuring the Appliance Sensor	185
Introduction	186
Configuring SSH	186
Cisco IDS Software v3	190
Cisco IDS Software v4.0	192
Configuring SSH Using IDM	198
Compatible Secure Shell Protocol Clients	200
Configuring Remote Access	201
Terminal Server Setup	202
BIOS Modifications for IDS 4210/4220/4230 Sensors	203
The IDS-4210 Sensor	203
The BIOS Setup for the	
IDS-4220 and IDS-4230 Sensors	204
Applying the Sensor Configuration	204
Cisco Enabling and Disabling Sensing Interfaces	205
Adding Interfaces to an Interface Group	207
Configuring Logging	208
Configuring Event Logging (IDS version 3.1)	208
Exporting Event Logs	209
Configuring Automatic IP Logging	211
Configuring IP Logging	212
Generating IP Logs	214
Upgrading the Sensor	216
Upgrading from 3.1 to 4.x	216

Updating Sensor Software (IDS 4.0) from the Command Line	219
Updating Sensor Software (IDS 4.0) with IDM	219
Updating Sensor Software (IDS 4.0) Using the IDM	221
Upgrading Cisco IDS Software from Version 4.0 to 4.1	222
Updating IDS Signatures	222
Updating Signatures (IDS 3.0)	223
Automatic Updates	223
Updating Signatures (IDS 4.0)	225
How to Restore the Default Configuration	226
Summary	227
Solutions Fast Track	228
Frequently Asked Questions	231
Chapter 6 Configuring the Cisco IDSM Sensor	233
Introduction	234
Understanding the Cisco IDSM Sensor	234
Configuring the Cisco IDSM Sensor	236
Setting Up the SPAN	244
Setting Up the VACLs	244
Configuring Trunks to Manage Traffic Flow	246
Verifying the Configuration	246
Updating the Cisco IDSM Sensor	247
Booting the IDSM Sensor from Partition 2	247
Upgrading the IDSM Sensor	250
Verifying the IDSM Sensor Upgrade	254
Shutting Down the IDSM Sensor	256
Updating the IDSM Sensor Signatures and Service Packs	258
Troubleshooting the Cisco IDSM Sensor	259
Summary	265
Solutions Fast Track	266
Frequently Asked Questions	268
Chapter 7 Cisco IDS Alarms and Signatures	271
Introduction	272
Understanding Cisco IDS Signatures	272
Signature Implementation	274
Signature Classes	275

Signature Structure	275
Signature Types	276
Cisco IDS Signature Micro-Engines	277
The ATOMIC Micro-Engines	281
The SERVICE Micro-Engine	286
The FLOOD Micro-Engine	289
The STATE.HTTP Micro-Engine	293
The STRING Micro-Engine	296
The SWEEP Micro-Engine	302
The OTHER Engine	311
Understanding Cisco IDS Signature Series	314
Configuring the Sensing Parameters	315
TCP Session Reassembly	315
No Reassembly	316
Loose Reassembly	316
Strict Reassembly	316
Configuring TCP Session Reassembly	316
IP Fragment Reassembly	317
Configuring IP Fragment Reassembly	317
Internal Networks	319
Adding Internal Networks	319
Sensing Properties	320
Configuring Sensing Properties	320
Excluding or Including Specific Signatures	321
Excluding or Including Signatures in CSPM	321
Excluding or Including Signatures in IDM	322
Creating a Custom Signature	323
Creating Custom Signatures Using IDM	324
Creating Custom Signatures Using CSPM	326
Working with SigWizMenu	326
Starting SigWizMenu	327
Tune Signature Parameters	328
Adding a New Custom Signature	330
Understanding Cisco IDS Alarms	334
Alarm Level 5 – High Severity	334
Alarm Level 4 – Medium Severity	335

Alarm Level 3 – Low Severity	335
Sensor Status Alarms	335
Identifying Traffic Oversubscription	337
Summary	338
Solutions Fast Track	339
Frequently Asked Questions	345
Chapter 8 Configuring Cisco IDS Blocking	347
Introduction	348
Understanding the Blocking Process	349
What Is Blocking?	351
Access Control Lists	351
Device Management	357
Understanding Master Blocking	358
Using ACLs to Perform Blocking	360
General Considerations for Implementation	361
Where Should I Put My Access Control Lists?	365
Configuring the Sensor to Block	366
Configuring a Router for a Sensor Telnet Session	366
Configuring the Sensor	368
The Never Block IP Addresses Setup	370
Using the Master Blocking Sensor	371
Manually Blocking and Removing a Block	372
Determining the Status of the Managed Device and Blocked Addresses	373
Summary	376
Solutions Fast Track	377
Frequently Asked Questions	380
Chapter 9 Capturing Network Traffic	383
Introduction	384
Switching Basics	385
Configuring SPAN	388
Configuring an IOS-Based Switch for SPAN	388
Configuring 2900/3500 Series Switches	389
Configuring a 4000/6000 Series IOS-Based Switch	393
Configuring a SET-Based Switch for SPAN	395
Configuring RSPAN	401

Configuring an IOS-Based Switch for RSPAN	403
Source Switch Configuration	403
Destination Switch Configuration	403
Configuring a SET-Based Switch for RSPAN	404
Source Switch Configuration	404
Destination Switch Configuration	405
Configuring VACLs	406
Using Network Taps	411
Using Advanced Capture Methods	415
Capturing with One Sensor and a Single VLAN	415
Capturing with One Sensor and Multiple VLANs	417
Capturing with Multiple Sensors and Multiple VLANs	418
Dealing with Encrypted Traffic and IPv6	419
Summary	423
Solutions Fast Track	424
Frequently Asked Questions	427
Chapter 10 Cisco Enterprise IDS Management	429
Introduction	430
Understanding the Cisco IDS Management Center	431
IDS MC and Security Monitor	431
The IDS MC and Sensors	432
IDS MC and Signatures	433
IDS MC and Security Policy	433
Installing the Cisco IDS Management Center	435
Server Hardware Requirements	435
CiscoWorks Architecture Overview	436
IDS MC Installation	438
IDS MC Processes	439
VMS Component Compatibility	439
Client Installation Requirements	440
Installation Steps	441
Getting Started	442
Authorization Roles	443
Installation Verification	444
Adding Users to CiscoWorks	445
The IDS MC	446

Setting Up Sensors and Sensor Groups	447
The IDS MC Hierarchy	448
Creating Sensor Subgroups	449
Adding Sensors to a Sensor Group	450
Deleting Sensors from a Sensor Group	453
Deleting Sensor Subgroups	454
Configuring Signatures and Alarms	455
Configuring Signatures	455
Configuring General Signatures	455
Configuring Alarms	457
Tuning General Signatures	458
How to Generate, Approve, and Deploy IDS Sensor	
Configuration Files	460
Reviewing Configuration Files	460
Generating Configuration Files	461
Approving Configuration Files	461
Deploying Configuration Files	462
Configuring Reports	464
Audit Reports	464
The Subsystem Report	465
The Sensor Version Import Report	465
The Sensor Configuration Import Report	465
The Sensor Configuration Deployment Report	465
The Console Notification Report	465
The Audit Log Report	466
Generating Reports	466
Viewing Reports	467
Exporting Reports	467
Deleting Generated Reports	467
Editing Report Parameters	468
Example of IDS Sensor Versions Report Generation	468
Security Monitor Reports	470
Administering the Cisco IDS MC Server	471
Database Rules	471
Adding a Database Rule	471
Editing a Database Rule	473

Viewing a Database Rule	473
Deleting a Database Rule	473
Updating Sensor Software and Signatures	474
Defining the E-mail Server Settings	474
Summary	475
Solutions Fast Track	476
Frequently Asked Questions	478
Appendix A Cisco IDS Sensor Signatures	513
IP Signatures 1000 Series	514
ICMP Signatures 2000 Series	516
TCP Signatures 3000 Series	518
UDP Signatures 4000 series	540
Web/HTTP Signature 5000 Series	546
Cross Protocol Signature 6000 series	582
ARP Signature 7000 Series	588
String Matching Signature 8000 Series	589
Back Door signature Series 9000 Series	590
Policy Violation Signature 10000 Series	595
Sensor Status Alarms	596
IDS Signatures Grouped by Software Release Version	598
Index	631

Foreword

The Internet used to be a place of shared access and shared ideas. In recent years, however, the Internet has taken on more of a Wild West personality, with general users, hackers, crackers, troublemakers, and information thieves using it for both business and pleasure. With such a mix of personalities online, it has become much more difficult to sort out who is safe and who is a threat. At the same time, the threats have become much more difficult to detect and protect against. Like the old west, network managers, administrators, and anyone else with a vested interest in protecting their data have built forts on the Internet to protect that data (now called “intellectual property”). People have finally awoken to the understanding that information is power and a significant amount of monetary value is often attached to information. So, in response to the threats, they have built walls that limit network access and have implemented gatekeepers in the form of firewalls. But, the malcontents have also been active. They have learned how to subvert the TCP/IP three-way handshake and use TCP’s own rules against itself in the form of Denial-of-Service (DoS) attacks. They have also learned how to generate and send spoofed packets with bits set to cause the IP stack to fail and, in some cases, give the attacker access to the computer. Indeed, the barbarians have become stealthy and masquerade their attack by using a normal port such as port 80 to launch attacks against DNS servers, web servers, or SQL servers with Unicode attacks and SQL injection attacks. And as one side raises the bar, the other side will match and raise the bar of network protection.

How does one begin to protect their network against such a determined enemy who can sneak in past the firewall by using traffic that, by all accounts, looks to be perfectly acceptable according to the firewall? By using a Cisco Intrusion Detection Sensor, that’s how. The Cisco IDS looks at traffic more deeply than the firewall and operates proactively by blocking or changing access-lists on the PIX firewall or Cisco routers on the fly. In order for the Cisco IDS sensor to do its job, the IDS sensor and management software must be installed and configured properly. This is what we are

striving to accomplish in this book—the correct way to install, configure, and use the Cisco IDS sensor and management tools provided to you.

To this end, we have organized this book to take you from IDS basics to the configuration of your own custom IDS sensor signatures. The following contains an overview of each chapter.

- **Chapter 1: Introduction to Intrusion Detection Systems** This chapter explains intrusion detection as well as Cisco’s spin on the process. We cover basic threats and types of attacks and provide an overview of the various types of intrusion detection, such as Network-based and Host-based IDSs. The basics of TCP connection theory and how an attack might evade the IDS are also discussed.
- **Chapter 2: Cisco Intrusion Detection** This chapter explores the nuts and bolts behind a Cisco-based IDS system, covering both Cisco’s “Active Defense” and “Defense in Depth” methodologies. Afterward, various platforms from Cisco are discussed, including how to use the Cisco Post Office Protocol and how to effectively deploy the IDS sensors in your network.
- **Chapter 3: Installing Sensor Appliances** Hands-on learning begins here with instruction on how to install the Cisco IDS appliances on your network. Password recovery is discussed as well as various commands like *idsstatus* and *idsconns*.
- **Chapter 4: Cisco IDS Management** All the IDS sensors in the world won’t do you a bit of good if you can’t manage them effectively. In this chapter, we start with a review of Cisco IDS management and show how to install the Cisco Secure Policy Manager (CSPM). Then we move on to the new Web-based management tool set that handles the Cisco sensor. The IDS Event Viewer is also covered, as well as Cisco’s Network Security Database.
- **Chapter 5: Configuring the Sensor Appliance** Now that the appliance is installed on your network, how the heck do you configure it? Chapter 5 answers these and other burning questions. We look at configuring the sensor in detail, explain how to configure SSH, how to configure event logging, and how to restore the defaults in case of trouble. Updating your signature files is also a major topic of discussion. A Cisco IDS sensor with old and out-of-date signature files is just another pretty boat anchor and we want to help you avoid that fate for your Cisco IDS sensor.

- **Chapter 6: Configuring the IDSM Sensor** Along with the appliance sensor, there is the black box of Cisco IDS sensors, the IDSM module or blade, which resides on the Cisco Catalyst 6500 series switch. This powerful but relatively unknown IDS sensor is explained in this chapter. We explore the installation, configuration, and management of the sensor when installed in the Cisco 6500 series switch chassis.
- **Chapter 7: IDS Signatures and Alarms** All the sensors in the world are pretty but useless paperweights unless there is some way of distributing the alarms. By the same token, if every alarm were dispatched, you would be quickly overwhelmed. Chapter 7 therefore explains how the signatures work and how to tune the type of alarms they generate. We also explore Cisco signatures in detail and explain the relevance of the various signature series. You'll learn how to configure signature parameters and how to build a custom signature. Lastly, we'll discuss how to tune the signatures to your network and explain why the effort of tuning is so very important to your network security and peace of mind.
- **Chapter 8: Configuring Cisco Blocking** This chapter explores Cisco blocking, yet another way the Cisco IDS can help protect the network by proactively blocking threats to your network security in real time. Along the way, the blocking process is explained, as well as how it works with Cisco IDS sensors and other Cisco products, such as the Cisco PIX. Finally, we explore how access-lists carry out blocking and how to configure the Cisco IDS sensor to perform the blocking actions.
- **Chapter 9: Capturing Network Traffic** In this chapter, we learn how to configure the switch to provide the mirrored traffic that the IDS sensor needs to watch over the network. We show you the hows and whys of switching and explain how to sniff traffic in a switched network. Specifically, we demonstrate how to configure your Cisco switches to use SPAN or VACLs to get access to the traffic your IDS sensor needs to see. We also explain why you might want to consider using network taps instead of just SPAN.
- **Chapter 10: Cisco Enterprise IDS Management** So, you have more than a couple of sensors? You, my friend, are why we wrote this chapter. We explain what the Cisco IDS Management Center is all about and how to install the Management Center, as well as how to configure the Cisco IDS

sensors and add them to the Management Center so you can manage all of the sensors from a single source. You'll learn how to configure reporting so you can justify to the boss all the money spent on these expensive tools, and how to administer the Cisco IDS Management Center server and keep it happy with the proper care and feeding.

- **Chapter 11: Cisco Firewall/IDS IOS** You say you don't have a sensor? That you're just a poor system administrator on a shoestring budget, but you do have a Cisco router? You may be in luck! Cisco offers a version of IDS software on the IOS router code, and in this chapter we teach you about the Cisco IDS IOS and how to configure the Cisco IDS IOS code on the router. You'll learn how to configure the IDS signatures and find out the limitations of the IOS-based version of IDS. We also show you how to verify that your IOS IDS installation actually works and how to get it to do what you want.

Introduction to Intrusion Detection Systems

Solutions in this Chapter:

- Understanding the AVVID Architecture
 - Understanding the SAFE Blueprint
 - Threats
 - Network Attacks
 - Overview of IDS
 - Defeating an IDS
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

The Internet can be a dangerous and costly place. Since its inception, there has been a consistent and steady rise in network and systems security incidents in every existing business and government sector. And, in a world where the number of computers and networks attached to the Internet grows by the hour, the number of potential attack targets has grown proportionally, and now includes a large concentration of home users who are experiencing “always on” broadband connectivity for the first time.

At first glance, the numbers related to Internet security breaches can be staggering, both in terms of sheer frequency and financial impact. Market researcher TruSecure estimates that losses from computer crime in 2003 could total over 2.8 billion. The Code Red worm in 2001 alone caused an estimated \$2 billion in damages and cleanup costs. Shortly thereafter, the Nimda worm was unleashed, with estimates of over \$2.5 billion in damage.

In the eighth annual CSI/FBI Computer Crime and Security Survey, 251 of 530 companies surveyed reported combined losses of nearly \$202 million, most of which stemmed from proprietary information theft and Denial-of-Service attacks. A bright spot in the 2003 CSI/FBI report indicated that reported losses of the companies surveyed dropped for the first time since the initial 1995 survey. This drop in costs occurred even though the number of attempted attacks did not diminish. Could this savings be attributed to increased corporate vigilance and attention to network security?

Perhaps most troubling of these figures, however, is the fact that many security incidents go undetected and most go unreported. Companies and governments readily admit they don't report incidents to avoid competitive disadvantage and negative publicity. Furthermore, the CSI/FBI report also indicates that a majority of known attacks occur from within an organization, proving that it is no longer adequate to “lock the front door.”

A new scourge has become a reality as well; the threat of electronic terrorism is widely recognized as a real motivation for attack. Governments and terrorist organizations alike practice overt and covert techniques aimed at disrupting the very network and systems infrastructure on which we so heavily depend.

What can be done to combat these threats? And upon what can we rely as prevention in the face of this constant and genuine danger?

This book presents a combination of intrusion detection systems (IDS) and security theory, Cisco security models, and detailed information regarding specific Cisco-based IDS solutions. The concepts and information presented in this book

are one step towards providing a more secure working and living network environment. This book also exists as a guide for Security Administrators seeking to pass the Cisco Secure Intrusion Detection Systems Exam (CSIDS 9E0-100), which is associated with CCSP, Cisco IDS Specialist, and Cisco Security Specialist 1 certifications.

Cisco has developed two primary and dynamic components that form their security model, the Architecture for Voice, Video, and Integrated Data (AVVID) and the Secure Blueprint for Enterprise Networks (SAFE), that are intended as tools for network and security architects to assist in the efficient, modular, and comprehensive design of today's modern networks.

Along with AVVID and SAFE, Cisco has developed a Security Wheel to provide a roadmap for implementing enterprisewide security and a foundation for effective and evolving security management. Within these security models, Cisco has identified four security threat categories and three attack categories. Administrators should understand each of these categories to better protect their network and systems environments.

In addition to Cisco security theory, there exist many different types of IDS functions such as Network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). We'll examine each of these and other types throughout this chapter and describe in detail how IDS actually function to detect potential security events.

Finally, we'll discuss the potential issues and shortcomings of an IDS so that administrators can understand the limitations of their security devices. Hopefully, armed with this information, white hat security professionals can provide their organizations and governments proper, comprehensive, and forward-thinking security capabilities.

Understanding the AVVID Architecture

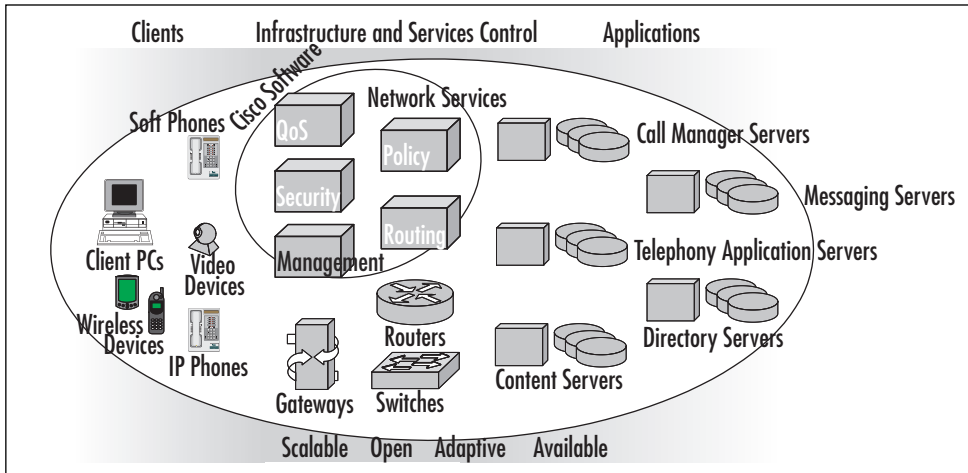
Today's networks transport an increasingly wide array of services such as voice and video, and application traffic including critical e-business and communication services. To assist network architects in the proper design of capable networks, Cisco created the Architecture for Voice, Video, and Integrated Data (AVVID). The AVVID architecture is based on an open, multiservice model and is composed of four interrelated, yet distinct layers as follows:

- Network Infrastructure Layer
- Services Control Layer

- Application Intelligence Layer
- Client Layer

The Cisco AVVID end-to-end architectural model is shown in Figure 1.1.

Figure 1.1 The AVVID Architectural Model



The Network Infrastructure Layer provides the groundwork for the AVVID architecture and is composed of switches, firewalls, IDS, VPN and security appliances, gateways, and routers. These are the devices and services that provide the foundational transport mechanisms for the network. It is in the Network Infrastructure Layer that intelligent logic is functionally applied, providing QoS, security, wire speed switching, and appropriate routing. Specific examples in the Network Infrastructure Layer might include Cisco Catalyst 6500 switches, Cisco PIX firewalls, Cisco 4200 Series IDS, and Cisco 7500 Series routers.

The Services Control Layer provides management of mechanisms applied in the Network Infrastructure Layer such as QoS and policy control, content distribution control, wireless access control, and call control, among others. This layer is composed of control consoles uniquely suited to assist in the management of the complexities present in the Network Infrastructure Layer. For instance, the CiscoWorks management modules and the PIX Device Manager are both examples of systems that could be present in the Services Control Layer.

These components provide reliable and efficient communication between the Client Layer, composed of AVVID appliances such as IP phones, wireless devices, PCs, and video equipment and the Application Layer. The Client Layer has

become increasingly sophisticated in recent years to fully leverage the growing list of advanced applications that promote enhanced business functionality. This sophistication places demands on the Network Infrastructure Layer for increased throughput, reduced latency, and more focused services. For example, the network capabilities delivered to the IP Telephone switch port might be different than those provided to a typical desktop workstation switch port. This could be provided by ingress port QoS classification and marking in the Network Infrastructure Layer and controlled via the Services Control Layer, which proves the need for holistic and comprehensive AVVID design.

The Application Layer provides the tools and logic that promote more efficient and capable business processing. The Application Layer includes functionality such as telephony application, messaging, video content distribution, and e-commerce services. Each of these services relies on the proper implementation of the Network Infrastructure Layer. An example of an Application Layer component is Cisco Call Manager. This application provides the functionality and logic behind the IP phones within the enterprise. It relies on other applications such as Directory Services to provide authentication and unique services to each IP Phone user. Along with the Client Layer IP Phones, it also relies on a well-built and functional network over which it can provide services.

The overarching theme of the AVVID architecture is the use of a single converged IP network for voice, video, and data traffic. Doing so facilitates gains in operational and technical efficiency, and reduces total cost of ownership for those migrating from traditional separation of services across multiple infrastructures. AVVID also incorporates centralized control and management of the infrastructure for increased administrative productivity.

The benefits of AVVID are

- **Integration** By using the Cisco AVVID architecture and applying the network intelligence imbedded within IP, companies can develop comprehensive tools to improve productivity.
- **Intelligence** AVVID promotes the prioritization of traffic and delivers intelligent network services to maximize network efficiency and performance.
- **Innovation** Cisco customers can adapt quickly to a changing business environment.
- **Interoperability** Standards-based APIs enable integration with third-party developers.

With the increased dependence on the IP network infrastructure comes amplified requirements for network capacity, QoS, resiliency, and security, however. These critical network attributes are imbedded throughout the Cisco AVVID architecture. For additional information regarding Cisco AVVID, go to www.cisco.com/go/avvid. To address the need for security, Cisco developed the SAFE blueprint, which augments the AVVID architecture.

Understanding the SAFE Blueprint

Another powerful tool available from Cisco for security administrators is SAFE, a security blueprint for enterprise networks. The SAFE blueprint builds on the Cisco AVVID architecture by incorporating best practices and comprehensive security functionality throughout the infrastructure. Fundamentally, the SAFE blueprint reinforces the absolute need for security in modern enterprise networks and details the management protocols and functions necessary to administer the security infrastructure.

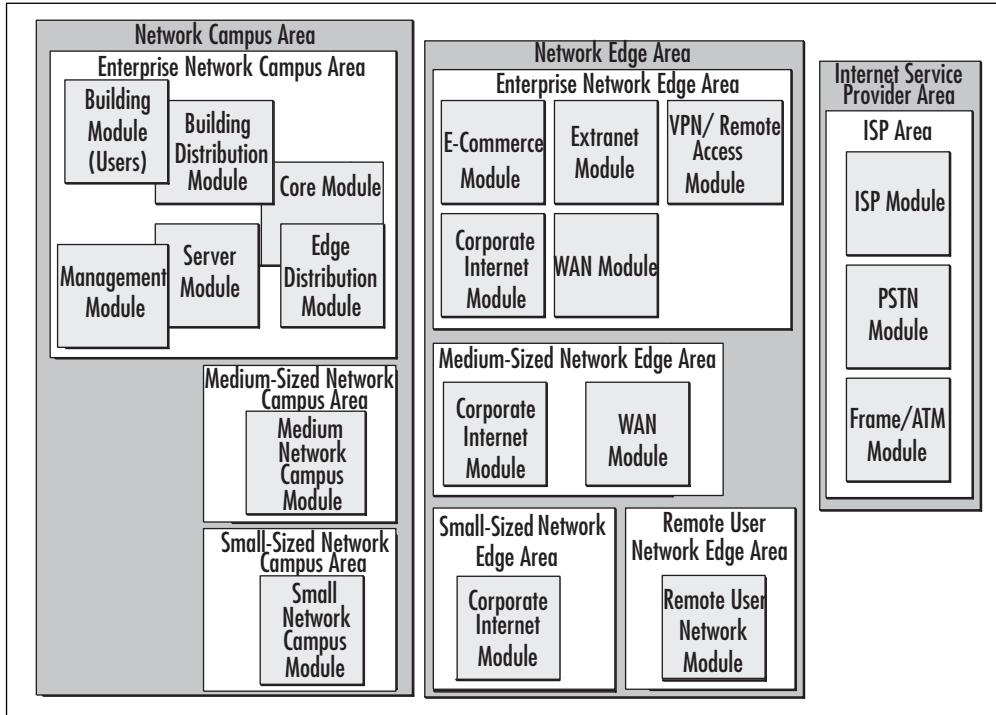
The benefits of SAFE are

- SAFE provides a detailed blueprint to securely compete in today's Internet and interconnected economy.
- SAFE provides a solid foundation for migrating to a secure and cost-effective network.
- SAFE, by being modular in design, enables companies to stay within their budgets.
- SAFE provides protection at each access point to the network using best-in-class security products and services.

SAFE is organized by network area as follows:

- Network Campus Area
- Network Edge Area
- Service Provider Area

Each area is modular for consistent and rapidly deployable security throughout the enterprise, when and where it is needed. When network managers use SAFE to design their security, the security architecture does not need to be redesigned each time a new service is added to the network. Each area has several modules addressing size and site-specific security functionality. The SAFE blueprint is depicted at a high level in Figure 1.2.

Figure 1.2 The SAFE Blueprint

Each of these modules incorporates designs for maximum performance, yet ensures security and integrity. SAFE modules are designed to address several network attributes including, but not limited to, security and threat response, secure management, availability, scalability, QoS support, and voice support.

Additionally, Cisco has updated the SAFE blueprint with new modules that incorporate Wireless LAN and IP Telephony security. Both address small-, medium-, and enterprise-sized environments and include design topics similar to those listed earlier.

Let's look at these areas in more detail.

The Network Campus Area

The SAFE blueprint includes security architectural information specific to the size of the networks and includes details for small, medium, and enterprise-sized networks. Regardless of size, however, the Campus Area includes security services directed primarily to the internal, corporate user. Common security infrastructure within the Campus Area includes packet filtering and VLAN-capable switch

devices, virus scanning systems, intrusion detection, and security management solutions to name a few.

Let's look a little closer at what each sized campus module provides within the SAFE blueprint.

The Small Campus Module

The Small Campus Module provides security infrastructure sized appropriately for budget-conscious and small organizations. Included within the Small Campus Module are intrusion detection systems, virus scanning servers, proxy devices, and security management systems. Within the Small Campus Module design, users are trusted more internally due to budget and size. For example, internal firewalls to separate Accounting from Engineering may not be practical based on cost.

The Medium Campus Module

The Medium Campus Module is similar to the Small Campus Module, yet includes more security infrastructure to provide protection for an increased number of people and services. For instance, in addition to the security implemented in the Small Campus Module, the Medium Campus Module includes switches capable of separating users via VLANs and filtering based on Layer 3 and 4 attributes. Critical services such as Call Management or Accounting Servers are separated by stateful inspection firewalls. Intrusion detection systems are more capable in the Medium Campus Module and can provide focused analysis in Layers 4 through 7. As in the Small Campus Module, the Medium Campus Module includes network management systems, virus scanning gateways, and proxy devices.

The Enterprise Campus

The Enterprise Campus Area within the SAFE blueprint is targeted at large organizations that may span several geographical locations and provide a multitude of user-focused internal services. The Enterprise Campus is large enough to warrant the creation of several modules, each addressing specific security requirements within the Campus. Let's look at these modules, starting from the user edge and working towards the services.

The Building Module

The Building Module might best be thought of as the Access Layer in the traditional tiered network architecture model. It is where the users are connected to

the network and includes virus scanning software, personal firewalls, and VLAN-separated user space.

The Distribution Module

Within the SAFE blueprint, there are two types of Distribution Modules, a Building Distribution Module and an Edge Distribution Module. As they both contain similar security infrastructure and largely provide the same type of network services, we'll discuss both of them in this section.

From the Building Module, the user traffic is directed through the Building Distribution Module. This module acts as a transport area to quickly provide access to the core networks. Within the Building Distribution Module, security features include RFC 2827 filtering to prevent DoS attacks and address spoofing and continued VLAN separation. Layer 3 separation may also exist if routing occurs in the Building Distribution Module.

The Edge Distribution Module serves as the security handoff to the Network Edge Area, which we'll discuss in a moment. Like the Building Distribution Module, the Edge Distribution Module also includes RFC 2827 filtering and, potentially, Layer 3 access control.

The Core Module

As is traditional in core networks, very little security infrastructure is included so as to not impede high-speed transport across the campus. While the Core Module does not call for security features, there are an increasing number of security devices, such as IDS and firewalls, that can potentially exist within the Core based on their high-speed performance.

The Server Module

The Server Module specifically addresses the needs of server farm or other service areas. Many security capabilities are present in the Server Module to protect enterprise assets such as directory services, messaging servers, DHCP, VoIP Call Management services, and the like. Included within the Server Module are stateful inspection firewalls and packet-filtering devices, IDS in the form of HIDS and NIDS, and VLAN-capable switches.

The Management Module

The Management Module exists as the command and control module for the entire SAFE blueprint. It is within this module that security support infrastruc-

ture resides. The Management Module can include the following services and capabilities:

- AAA services such as Cisco Secure ACS for network device access control
- SNMP-based network monitoring and control services, such as CiscoWorks
- Syslog servers for comprehensive error and event data capture
- Out-of-band (OOB) network access and infrastructure
- Two-factor authentication systems such as SecurID servers
- Device configuration management systems for revision control
- VPN termination systems for remote, secure management

In addition to these services, the Management Module is itself protected by focused Layer 4–7 IDS analysis, various traffic filtering mechanisms such as router filters and stateful inspection firewalls, and, as in other modules, VLAN-capable switches for Layer 2 separation.

The Network Edge Area

Similar to the Network Campus Area, the Network Edge Area consists of security architectural information specific to the size of the networks that includes details for small-, medium-, and enterprise-sized networks. The Network Edge Area also includes a Remote User Network Module focusing on home office and remote access networks. Furthermore, each specifically sized Network Edge Area addresses security regarding the more publicly available services a company may provide. This Area also includes the security features necessary to safeguard an organization's connection to the Internet.

Let's look more closely at the Network Edge Area as it applies to differently sized companies.

The Remote User Network Edge

The Remote User Network Edge Module provides security for users working from external locations such as home offices or small remote offices. There exist four connectivity options within the Remote User Network Edge Module as follows:

- **Software Access Option** Users connect to the central office via VPN and authentication software installed on their computer workstation. Users may have broadband connectivity, but most likely rely on dialup access for remote connectivity. This is the simplest option for remote connectivity.
- **Remote Site Firewall Option** A firewall device is used in this option for more permanent and robust secure remote connectivity. This option infers a broadband connection and provides stateful inspection and/or Layer 7 packet filtering. VPN access and authentication services can be located at the firewall or on the user's computer workstations in this option.
- **Hardware VPN Client Option** Similar to the Remote Site Firewall Option, the Hardware VPN Client Option uses broadband network connectivity and provides VPN and authentication services on behalf of the user. This option relies on user workstation personal firewall software for perimeter security, however.
- **Remote Site Router Option** Nearly identical to the Remote Site Firewall Option, this option uses a router with firewall capabilities to provide perimeter packet filtering and may include stateful inspection and/or Layer 7 filtering capabilities.

Regardless of the connectivity options, the Remote User Network Edge Module includes security infrastructure typical of user network areas such as virus scanning systems, HIDS, and personal firewalls.

The Small Network Edge

The Small Network Edge combines economical and appropriate security measures to protect smaller organizations. The Small Network Edge includes one module, the Corporate Internet Module.

The Corporate Internet Module

The Small Network Corporate Internet Module acts as the demarcation between the company's assets and the ISP Area. It also serves to protect the application systems that the company provides to the public, such as web, database, and mail servers.

The security infrastructure present in the Small Network Corporate Internet Module includes perimeter stateful inspection firewalls, Layer 7 filtering capabili-

ties, and IDS in the form of NIDS and HIDS. The Small Network Corporate Internet Module also includes Remote Authentication services, VPN termination devices, and VLAN-capable switches.

The Medium Network Edge

The Medium Network Edge includes more advanced and comprehensive security mechanisms to protect the larger asset and employee base of the medium-sized company. It includes two modules, as discussed next.

The Corporate Internet Module

Like the Small Network Edge Corporate Internet Module, the Medium Network Edge Corporate Internet Module includes perimeter stateful inspection firewalls and Layer 7 filtering capabilities. These serve to protect the corporate internal networks and services. This module has more focused IDS capabilities, however, and also includes content inspection for mail services, more robust VPN termination, and scalable authentication services.

The WAN Edge Module

The Medium Network Edge has a second module to address WAN connectivity needs. This module may include packet-filtering capabilities, but most likely it simply provides reliable and secure transport to remote office locations.

The Enterprise Network Edge

The Enterprise Network Edge Area within the SAFE blueprint is targeted at large organizations with various customer-focused, publicly available services in several locations. The Enterprise Network Edge necessitates the creation of several modules, each addressing specific security requirements within the Edge Network. We'll discuss these modules in the following pages.

The E-Commerce Module

The E-Commerce Module is intended to house and protect the business-driving public infrastructure of the organization and includes database, application, and web services components, among others. To provide a comprehensive defense, the SAFE blueprint calls for focused Layer 4–7 IDS analysis and Host IDS capabilities. Furthermore, multitiered stateful inspection firewalls and packet-filtering devices are included for perimeter defense. Wire speed switching on VLAN-

capable switches provides server connectivity in the E-Commerce Module for fast, efficient server access.

The Corporate Internet Module

The Corporate Internet Module provides secure connectivity for internal corporate users to the Internet. It also offers logical space for inbound and outbound services such as SMTP, web proxy, and content inspection servers. This business functionality is protected with stateful inspection firewalls, Layer 7 filtering, spoof mitigation, and other basic filtering. It also includes advanced and focused Network IDS analysis and host-based detection systems.

The VPN/Remote Access Module

Due to the potential size and scaling requirements of Enterprise-sized VPN solutions, the Enterprise Network Edge Area includes a VPN/Remote Access module. This module contains the required encryption, VPN termination points, and authentication mechanisms for the Enterprise environment. Included in this module are various IDS components that are placed at the encryption endpoint to inspect inbound and outbound VPN traffic. Stateful inspection firewalls are also integrated into the VPN/Remote Access Module for perimeter security from, and to, remote connections.

The Extranet Module

The Extranet Module is similar to the E-Commerce Module in that it houses application and web-based services. Extranets are typically intended to facilitate access by semi-trusted users such as partners or other remote entities. Like the E-Commerce Module, the Extranet Module includes NIDS and HIDS, as well as stateful inspection firewalls. It also includes authentication and VPN termination services for remote use.

The WAN Module

The Enterprise Network Edge WAN Module includes sparse security features to facilitate efficient network transport. The WAN Module may include Layer 3 access control mechanisms for secure transport.

The Internet Service Provider Area

The Internet Service Provider Area as described by the SAFE blueprint provides companies and organizations with a secure and high-speed transit network to the

public Internet. While the ISP Area is outside the enterprise-, small, and medium-sized business network demarcation, it too includes security features to protect customers and the ISP network itself.

The ISP Area contains the following three modules:

- The ISP Module
- The PSTN Module
- The Frame/ATM Module

Of these modules, the PSTN and Frame/ATM Modules do not include many security mechanisms other than self-protective ACLs and filters on network equipment to protect the ISP routers, switches, and telephony infrastructure.

The ISP module, however, typically includes spoof mitigation, DoS limiting features, and some limited Layer 4 filtering capabilities. These are typically intended to protect the ISP itself, yet as network-based attack frequency and sophistication rises, ISPs face increased pressure to help combat security incidents through additional security mechanisms.

SAFE Axioms

The SAFE blueprint includes key devices to be deployed in each module along with design guidelines and alternatives, and potential threats mitigated by the solution. All of this design information is predicated on several SAFE axioms that follow:

- Routers are targets
- Switches are targets
- Hosts are targets
- Networks are targets
- Applications are targets
- Intrusion detection systems are necessary
- Secure management and reporting are necessary

In the blueprint, each of these axioms has comprehensive mitigation techniques and implementation guidelines.

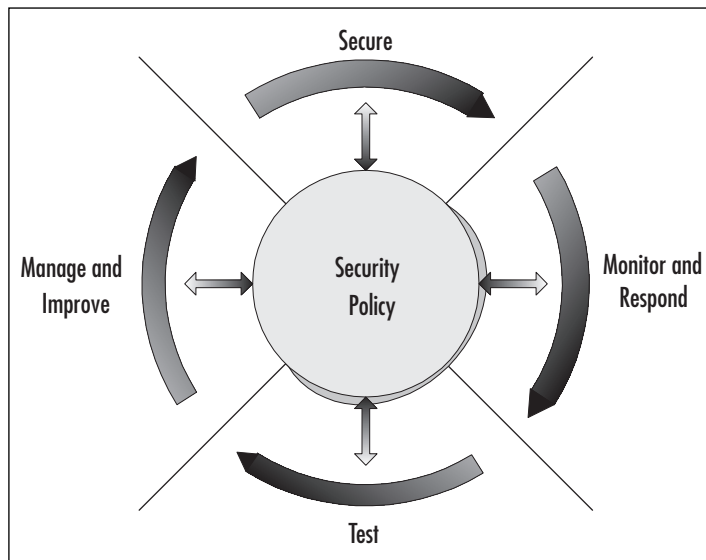
The SAFE blueprint is a detailed and holistic approach to securing the enterprise. It includes in-depth defense strategies and multidisciplined approaches for

security. Security administrators should be familiar with the SAFE design. For additional information regarding Cisco SAFE, go to www.cisco.com/go/safe.

The Cisco Security Wheel

Implementing a comprehensive security stance is critical in successfully defending one's network and services infrastructure. To do so, Cisco recommends a cyclical, evolutionary approach depicted by a wheel (as shown in Figure 1.3) known as the Cisco Security Wheel.

Figure 1.3 The Cisco Security Wheel



This approach incorporates the following repetitive methodology:

1. **Corporate Security Policy** Instantiate a solid security policy.
2. **Secure** Secure all existing networks and systems.
3. **Monitor and Respond** Monitor the infrastructure and respond accordingly to events.
4. **Test** Regularly test security systems, including human response capabilities.
5. **Manage and Improve** Effectively manage and continually improve the security stance.

Use of this methodology can help provide a holistic and evolving security plan that keeps pace with the ever-changing threats present in today's technical environment. Each of these steps is discussed in detail in this section.

Corporate Security Policy

All effective security measures start with a good, comprehensive security policy. Developing a written and well-defined policy must be the first step in addressing an organization's security needs. Indeed, all efforts, both tactical and strategic, should flow from the policy. Furthermore, as a company practices the methodology ascribed by the Security Wheel, the security policy should become an integral feedback mechanism to measure success and failure and should be updated as the need arises.

The security policy should contain a complete set of proactive and reactive measures that an organization should take to prevent, or react to, security events. The security policy should also address the following items: roles and responsibilities, clear delineation of acceptable behavior, and definition of data sensitivity classification. The repercussions of breaching security policy should also be documented. Other considerations within the security policy include the delineation of:

- The incident response team
- The security team
- Response procedures
- Communication procedures
- Logging procedures
- Training/rehearsal plans

Once a clear, balanced policy has been constructed, it must be approved by an organization's stakeholders, such as Executive managers, Human Resources Staff, IT and Security Staff, Legal personnel, and others. With this buy-in, the policy can be universally and consistently enforced rather than being relegated to a shelf in the document library.

There are many resources regarding policy formation available to the security administrator. Good starting points include *RFC 2196 – The Site Security Handbook* (www.ietf.org/rfc/rfc2196.txt) and the SANS "Design and Implementation of the Corporate Security Policy" document (www.sans.org/resources/policies). Ample time should be dedicated to developing a good security policy. Above all, the policy should be realistic, flexible, and should be easily understandable by all within the organization.

Secure

Securing the network involves the intelligent placement of security devices such as firewalls, IDS, and other systems. Before doing so, however, the security team should have a detailed knowledge of the network in which they work. This involves gathering and understanding attributes such as overall network size and topology, ingress and egress points, service locations, and general application flow parameters. Understanding the traffic and how it flows across the network is an essential step in security implementations.

Securing the network also involves the security policy established in the first step of the methodology. Each network and organization differs in their needs, which is why a tuned security policy is necessary. Security administrators will find that the following security solutions are required:

- Access Control
- Encryption
- Authentication
- Vulnerability Patching

Access Control

Access control mechanisms can take many forms. Perimeter barrier devices are often first considered when securing a network. Firewalls in the form of packet filters, proxies, and stateful inspection devices are all helpful agents in permitting or denying specific traffic through the network. Access controls also exist on end systems in the form of a privilege level for access to resources, configuration files, or data.

NOTE

Securing the enterprise requires intimate knowledge of your infrastructure including network design, services locations, and data traffic flow attributes, among others. Knowing these details allows you to place IDS and perimeter security devices such as firewalls in the most effective locations to prevent unwanted intrusions. Without this knowledge, administrators will waste corporate resources by over-deploying security infrastructure, or worse, missing unseen attack avenues into the enterprise.

Encryption

Encryption in the form of IPSec, PPTP, or other protocols can help ensure confidentiality of data transport within networks and between networks. Virtual Private Networks (VPNs) are often cost-effective measures to facilitate private communication across a shared network infrastructure.

Authentication

After thorough planning, security support infrastructure such as authentication, authorization, and accounting (AAA) systems can be implemented to provide verification for access and privilege control through firewalls and VPNs to services. Cisco offers Secure Access Control (ACS) as a means of implementing AAA. Several varying degrees of authentication can be integrated with AAA such as clear-text passwords, Microsoft CHAP, S/Key and SecurID. Administrators should set up logging capabilities for historical and forensic data analysis and monitoring.

Vulnerability Patching

Securing the network also means securing the systems on which services reside. Staying current with patches, operating systems, and application software revisions can mitigate commonly used attack vectors. Policy should dictate regular and systematic upgrades to organizations' software-based systems.

Administrators should regularly check for security patch updates on vendor web sites and newsgroups. Some examples of vendor patch and security advisory web sites are:

- **Microsoft** <http://windowsupdate.microsoft.com>
- **Sun Microsystems** <http://sunsolve.sun.com>
- **Red Hat Linux** www.redhat.com/apps/support/errata/
- **Cisco** www.cisco.com/warp/public/707/advisory.html

Finally, securing the network includes the implementation of physical security measures. The best network security methods can prove meaningless without solid security to protect against physical access to servers, firewalls, and other network equipment. Cipher systems, and identity cards and verification systems are all examples of ways to improve physical security.

Monitor and Respond

Once the environment is secure, the next step in the Cisco Security Wheel is realization of comprehensive monitoring and response techniques. This means the use of documented and policy-directed software and human practices to ensure full awareness of potential security events.

Software systems include well-tuned alert thresholds and logging mechanisms on the devices used to secure the network, such as firewalls, IDS, and AAA servers. It is absolutely critical that the reporting mechanisms are properly configured, however. Otherwise, security administrators will be overwhelmed with false-positive data and will be rendered ineffective in actual security situations. Furthermore, in large enterprise environments, it is quite impossible for humans to keep pace with copious logs and alert messages, even with well-configured devices; there is simply too much data to analyze. In these situations, additional software to perform event aggregation and correlation proves necessary to alleviate data overload.

In addition to well-constructed software mechanisms, security administrators must practice proper and methodical monitoring techniques. Administrators should baseline and understand the normal attributes of the network so as to recognize anomalous events. Regular and repeated practices in log and alert monitoring can reduce the chances of missing the precursory events of security attacks and stave off damaging situations before they occur.

With good human and software monitoring techniques, most security issues can be detected. It is at the point of detection that defined and practiced response measures must be implemented. Some responses may be automated, such as automatic shunning or filtering based on an IDS signature detection. Most responses will likely be manual, however. In these situations, administrators should have clear roles and responsibilities to mitigate the effects of an attack and alert upstream authorities, both inside and outside of the organization. Well-developed security policies are often helpful in delineating such roles, responsibilities, and actions.

Finally, administrators should also be prepared to react dynamically in atypical and new security situations. Again, security policy can aid in these situations by defining the realm of the administrators' authority and obligation.

Test

Through the use of the Cisco Security Wheel, an organization may have developed a strong security policy, secured the network properly, and implemented compre-

hensive monitoring and response techniques. The next step is to thoroughly and regularly test these constructs to ensure validity, accuracy, and effectiveness.

Testing can take the form of scanning across firewalls, servers, and IDS to ensure correct configuration. Oftentimes, an organization will seek external audits of the infrastructure for objectivity. Testing should also include assessment of administrative responses through mock events and practice drills. Doing so not only helps identify areas of weakness, but provides training and rehearsal time to finely tune the security team's responses.

Testing should be regular and repetitive, and should be clearly defined in the security policy.

Manage and Improve

Finally, as a security team practices the methodology of the Cisco Security Wheel, they should seek to continually improve their capabilities through proper management. This involves not only the cyclical actions associated with the Security Wheel, but also the unrelenting defense against new and unknown threats.

Good security management includes continuous education through training, practice, and reading. Administrators should keep pace with security newsgroups and publications and should appreciate potential vulnerabilities as they are discovered and before they are automated.

Postmortem sessions after security events should be conducted to investigate lessons learned and reveal places for improvement. Administrators should develop education systems for the employees of the organization who may not be well-informed of good everyday security practices.

The threats against and consequences of participating in the networked environment have not, and will not, stop changing and challenging those who seek to protect an organization's assets. Above all, the security team and infrastructure of an organization must continually evolve to defend against such threats.

Threats

The threats against an organization's networks and systems can be categorized into four general types, as follows:

- Unstructured
- Structured

- External
- Internal

Intuitively, these categories are not necessarily exclusive of each other; security events may be characterized by a combination of the threats previously listed. We will discuss each of these threats in this section.

Unstructured Threats

Unstructured threats are characterized by attacks often based on well-known vulnerabilities and scripted vectors. Generally, such threats emanate from less-competent attackers or hackers known as script kiddies or newbies who may be motivated less by malicious intent and more by curiosity and intellectual challenge. The attacker usually does not understand the actual mechanisms of the exploit attempted, nor the full ramifications of his/her actions.

Oftentimes, good security practices that effectively keep pace with the latest known attack methodologies and vulnerabilities prove capable in defending against unstructured threats; by the time an attack vector is scripted by a knowledgeable miscreant, distributed, and finally deployed by the many script kiddies, it should be preventable by alert security staff and, therefore, relatively ineffectual.

This does not diminish the potential impact such threats pose to organizations, however. For instance, certain Denial-of-Service (DoS) attacks triggered by script kiddies can be difficult to defend against and could cause serious harm to an organization's operation.

Structured Threats

Structured threats are often far more serious and potentially damaging to an organization than unstructured threats. These threats are characterized by directed and specific attempts to do harm, gather information, and, disrupt business and operations. Those engaged in structure threats are often erudite assailants with detailed knowledge of network functionality and application logic. Furthermore, the attackers are often motivated by achieving a specific outcome such as fraud, theft, or industry- or state-sponsored intelligence gathering and may focus on specific targets. Oftentimes, the perpetrators of structured threats are those creating the tools and scripts used by script kiddies in unstructured threats.

Structured threats can be challenging to security administrators who may not understand their network infrastructure and systems as well as the attacker. While

there are far fewer individuals engaged in structured threats than unstructured threats, it is arguable that these few are the most dangerous elements.

External Threats

External threats, intuitively, are those originating outside the secured organization. These threats are from trespassing individuals not authorized to use an organization's systems and networks. External threats could be composed of unstructured or structured threats and could emanate from industry competitors or rival nation states, among others.

Internal Threats

Internal threats are those instigated within an organization and are far more common than external threats, counter to conventional wisdom. Internal threats are initiated by someone with some authorized access to an organization's infrastructure. Classic cases of internal threats might be those triggered by disgruntled employees seeking to do damage to an organization, or employees recently dismissed that wish to steal proprietary assets.

Network Attacks

While there are many specific ways to attack a network or the systems on a network, there are three general types of attack, as follows:

- Reconnaissance attacks
- Access attacks
- DoS attacks

Like the different types of threats previously discussed, these attack types are not discrete and may be used in combination to meet the goals of a malicious attacker. Each of these network attack types are described in this section.

Reconnaissance Attacks

Reconnaissance attacks are used to gather information about a target network or system. Such attacks may seem harmless at the time and may be overlooked by security administrators as “network noise” or pestering behavior, but it is usually the information gained through reconnaissance attacks that is used in subsequent Access or DoS attacks.

Several means may be used to gather information about an organization and could include automated and manual technological attacks as well as human social attacks. Examples might include ICMP ping sweeps against a network or SNMP walking techniques to gather network map and device configuration data. Likewise, application-level scanners could be used to search for vulnerabilities such as web server CGI or ASP weaknesses.

No specific damage may be caused by the reconnaissance attack, but it is akin to burglars staking out a neighborhood, watching for times of inactivity, and occasionally testing windows and doors for access.

Reconnaissance attacks are quite common and should be considered a serious threat to an organization as they may give potential attackers the information required to perform access or DoS attacks.

Access Attacks

Access attacks, as the name implies, are those involving the unauthorized use of a target machine or machines. The means by which an intruder gains access to infrastructure are typically specific to the exploitable vulnerabilities present in operating systems, application software, or physical protection mechanisms. Often these vulnerabilities are discovered by hackers during previous reconnaissance attacks.

Access attacks can be manual or automated and may be composed of unstructured or structured threats. Generally, access attacks can be categorized into three forms of unauthorized activity, as follows:

- Data Retrieval
- System access
- Privilege escalation

The sophistication of access attacks has increased as hackers have become more proficient with tools and more knowledgeable about vulnerabilities. Often, these forms of attack are combined to enlarge the scope and severity of an assault. We discuss each of these attacks in this section.

Data Retrieval

The first form of access is unauthorized data retrieval in which information is read, copied or moved on a system. Data retrieval access attacks are common from internal threats and are largely the result of poorly configured file and directory permissions. For instance, world readable Windows file shares or Unix NFS

directories are relatively simple ways unauthorized users can gain access to potentially sensitive data such as accounting or human resources information. In this example, use of proper mounting or access permissions and even encryption could prevent such access.

System Access

System access occurs when an intruder has operating system level or actual login access to a device. Such unauthorized access could be achieved through weak or non-existent passwords or through known exploits against operating system vulnerabilities. Many secondary attacks could result from unauthorized system access. For example, compromised machines could be used to target other machines on the network. Or, once a hacker obtains system access, he or she could attempt privilege escalation.

Privilege Escalation

Attaining higher privileges on a system allows hackers to perform far more dangerous actions. Once an intruder has system access as previously described, they often seek super user or root privileges to install Trojan code or create backdoors for future covert access. Privilege escalation is often acquired via operating system or application vulnerabilities such as buffer overflow attacks. Once a system has been compromised in this manner, it is completely at the control of an attacker.

DoS Attacks

A third form of network attack is known as denial of service, where the attacker seeks to prevent legitimate use of a service or system. Oftentimes, this is accomplished by overwhelming an infrastructure with bogus requests for service. DoS attacks can also be caused by corrupted data or configurations. For instance, a DoS attack could be the result of an intentionally corrupted Border Gateway Protocol (BGP) routing configuration. If an attacker changed the network advertisement, authentication attributes, or Autonomous System Number (ASN) parameters on an organization's routing equipment, that organization could simply disappear from the Internet or, worse yet, traffic destined to that organization could be routed to an illegitimate remote location on the Internet.

DoS attacks can also be dispersed so that numerous compromised machines launch a DoS attack simultaneously on the same target service or host. Known as a Distributed Denial of Service (DDoS) attack, such events are extremely difficult to combat since it is often impossible to ascertain the difference between legitimate and illegitimate traffic.

Anatomy of an Attack

Now that we've discussed the various forms and methods of attack, let's look at an example involving a combination of what we've learned.

Let's assume a bank, the ACME Bank, has an online account system by which bank patrons access their accounts and assets. Sally, a fairly knowledgeable hacker, wants to create some trouble via a DoS attack on the bank. She's upset that her mother's account was accidentally closed and wants to teach the bank a lesson. This makes Sally an external and structured threat.

Sally begins by slowly performing reconnaissance attacks on the bank's network and system infrastructure. Using a series of readily available hacking software tools, she determines the bank's IP network address ranges and critical systems including web, mail, and Domain Name Servers (DNS). From her reconnaissance attacks, Sally determines that the weakest link at the bank appears to be the DNS; the DNS servers are poorly configured to allow unrestricted zone transfers and report that they are running outdated and vulnerable code.

From an anonymous dialup account, Sally uses a script to perform a DoS attack based on the "zxfi" bug. She remotely causes the DNS servers to repetitively crash by requesting compressed zone file transfers using commonly available tools. Because of the DoS attack, bank customers without cached DNS information effectively cannot "find" all of the bank's services, including web, e-mail, and other vital customer support functions.

Had the DNS administrators properly restricted zone transfers or maintained recent revisions of code, this incident could have been prevented. Had security administrators positioned IDS sensors near the DNS servers, they might have been alerted to the situation. Are your systems and network properly secured? Could this happen to you? How would you react should this situation occur?

Overview of IDS

Intrusion detection systems come in many shapes and sizes. Some are small, one rack unit appliances that tuck neatly into your server rack while others are modules, such as the Cisco IDSM, that insert directly into active network components. Some IDS are simply software applications that run on servers or workstations. Their general purpose is to monitor events on systems and networks and notify security administrators of an event that the sensor determines is worthy of alert. An IDS weighs these situations using a variety of means. Some IDS compare network conversations they "hear" to a list of known attack sequences or signatures. When the network traffic matches a known exploit sig-

nature, they trigger an alert. These IDS are known as Signature-based IDS. Other IDS collect a baseline of “normal” network operations over time. They then continue to monitor the network for situations that don’t match what they’ve determined as normal. If this happens, they trigger an alert. These IDS are called anomaly-based IDS.

Some IDS can perform automated actions beyond simply sending alerts, such as resetting malicious connections by using a technique called TCP Reset, blocking offending source addresses, or shunning the IP address. Some of the more advanced IDS sensors can even reconfigure ACLs on routers and firewalls dynamically.

On today’s busy networks, a lot of information and data is transferred between clients and servers. While most of this communication is legitimate and beneficial, some of it might not be. But how could you possibly determine which is which? How are you to know if a reconnaissance attack or data retrieval attack is underway, while hidden among the normal, good network traffic? Such knowledge is simply not possible without an IDS. In this section, we’ll discuss the various types of IDS and some of the ways in which these devices function.

Types of IDS

There are several types of IDS that can be deployed to aid security administrators in their endeavors. Two types, network-based intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) are most prevalent in modern security deployments. There are other types of IDS, however, which include file integrity and log file checkers, and decoy devices known as honeypots. Additionally, there exist hybrid systems that combine some of the different functionalities mentioned earlier. We’ll discuss each of these IDS in this section.

Network IDS

Network-based intrusion detection systems (NIDS) are devices intelligently distributed within networks that passively inspect traffic traversing the devices on which they sit. NIDS can be hardware or software-based systems and, depending on the manufacturer of the system, can attach to various network mediums such as Ethernet, FDDI, and others. Oftentimes, NIDS have two network interfaces. One is used for listening to network conversations in promiscuous mode and the other is used for control and reporting.

With the advent of switching, which isolates unicast conversations to ingress and egress switch ports, network infrastructure vendors have devised port-mir-

roring techniques to replicate all network traffic to the NIDS. There are other means of supplying traffic to the IDS such as network taps. Cisco uses Switched Port Analyzer (SPAN) functionality to facilitate this capability on their network devices and, in some network equipment, includes NIDS components directly within the switch. We'll discuss Cisco's IDS products in the next chapter.

While there are many NIDS vendors, all systems tend to function in one of two ways; NIDS are either signature-based or anomaly-based systems. Both are mechanisms that separate benign traffic from its malicious brethren. Potential issues with NIDS include high-speed network data overload, tuning difficulties, encryption, and signature development lag time. We'll cover how IDS work and the difficulties involved with them later in this section.

Host IDS

host-based intrusion detection systems (HIDS) are systems that sit at service endpoints rather than in the network transit points like NIDS. The first type of IDS that's widely implemented, Host IDS, is installed on servers and is more focused on analyzing the specific operating system and application functionality residing on the HIDS host. HIDS are often critical in detecting internal attacks directed towards an organization's servers such as DNS, mail, and web servers. HIDS can detect a variety of potential attack situations such as file permission changes and improperly formed client-server requests.

File Integrity and Log File Checkers

File integrity and log file checking agents are a form of HIDS that focus on the operating systems binary files and the log files normally produced by OS-based security mechanisms such as login logs. File integrity software systems are best installed immediately after operating system installation. The software creates a local database and MD5 hashes of operating system binaries and configuration files. Should system binaries or other files change in any way, nightly processes that compare current hashes against original file hashes will detect the change and alert administrators.

Log file checkers run regularly as well and parse system and application logs to search for signature-based alerts. For instance, multiple failed logins on a server would typically be detected and reported by log-checking software.

Others

While Host IDS and Network IDS are the most commonly deployed forms of IDS, other types of IDS such as Hybrid IDS and honeypots can be useful tools in detecting potential security situations.

Hybrid IDS

Hybrid IDS are systems that combine both Host IDS and limited Network IDS functionality on the same security platform. A Hybrid IDS can monitor system and application events and verify a file system's integrity like Host IDS, yet because the monitoring network interface runs in a non-promiscuous mode, the Network IDS functionality only serves to analyze traffic destined for the device itself. A Hybrid IDS is often deployed on an organization's most critical servers.

Honeypots

Another form of IDS are honeypots. These systems differ from the other forms of IDS in that they act as service endpoints, yet have no actual production services. The honeypots simply appear to run vulnerable services and capture vital information as intruders attempt unauthorized access. Using a honeypot, you can effectively place a doorbell on the network for hackers to ring and let you know they are there. In some honeypot designs, not only does the honeypot alert administrators regarding hacking attempts, but they capture all keystrokes and any files such as a rootkit that might have been used in the intrusion attempt. There are several commercial and open source honeypot software applications available for use on various operating system platforms. Some of the commonly used applications include:

- **Honeyd** www.citi.umich.edu/u/provos/honeyd/
- **Deception Toolkit** www.all.net/dtk/dtk.html
- **Specter** www.specter.ch/
- **Symantec Mantrap** <http://enterprisesecurity.symantec.com>
- **Honeynets** <http://project.honeynet.org/>

How Does IDS Work?

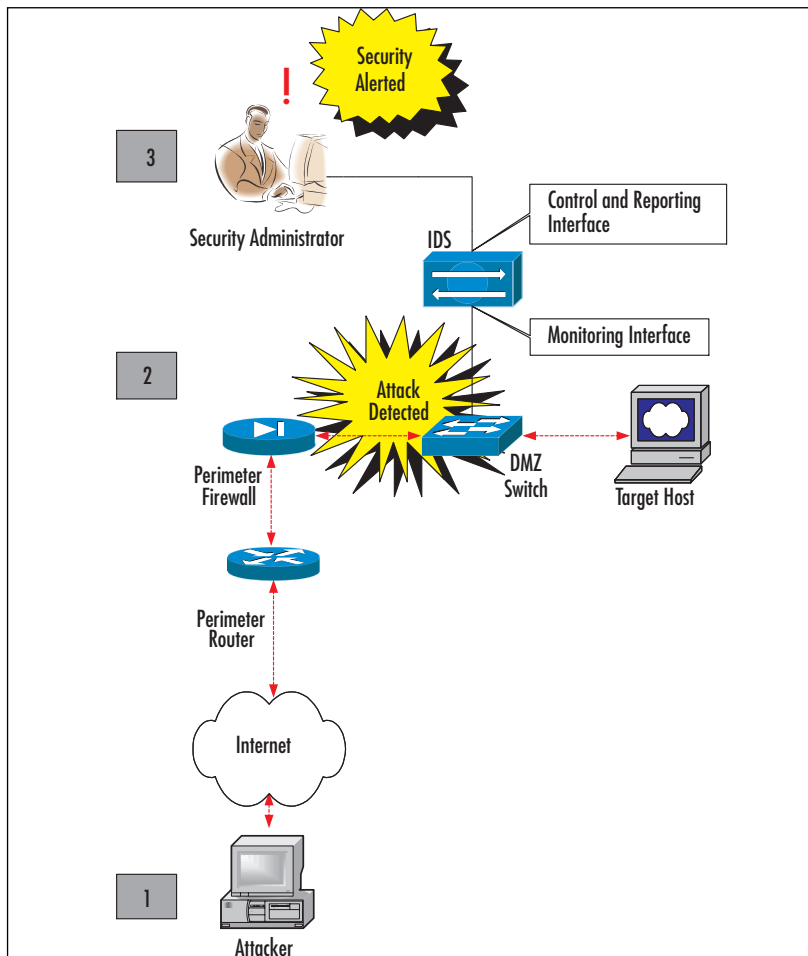
All IDS are monitoring tools that serve an essential service—they detect potential security events and alert administrators. Some IDS can even help perform auto-

mated actions such as issuing Access Control List (ACL) updates to firewalls. IDS, however, should not be confused with perimeter security devices such as firewalls since they do not directly prevent security events through blocking and authentication mechanisms themselves.

As we've previously discussed, an NIDS should be intelligently distributed in the network and an HIDS installed on critical systems. While we'll discuss detailed deployment strategies and best practices for deploying IDS in the enterprise in the next chapter, let's briefly cover basic NIDS configurations and functionality.

NIDS are deployed where services or important traffic traverses network devices. A typical IDS deployment, attack, and response sequence is shown in Figure 1.4.

Figure 1.4 A Typical NIDS Deployment



In this NIDS deployment, all network traffic that traverses the DMZ switch is inspected by the IDS via its Monitoring Interface. The NIDS could obtain data from the DMZ switch via a Cisco RSPAN port or a network tap. Alternatively, the NIDS could be an internal component of the switch such as a Cisco IDS Switch Module. We'll discuss these possibilities further later in the book. Regardless, should an attacker initiate a series of malicious actions against servers on the DMZ network, it might unfold this way:

1. A hacker working from a remote workstation on the Internet begins an attack on the DMZ-located target host. Perhaps the hacker is using freely available software to scan for open Windows file shares. Since the connection between the attacker and the target host traverses the DMZ switch, the NIDS “hears” the attacker’s attempts to scan and mount unprotected Windows file shares via its Monitoring Interface.
2. Since the NIDS is listening on the network and actively comparing all traffic against predefined attack signatures, it detects the attacker’s scanning attempts. Depending upon how the NIDS is configured, several outcomes could result at this point. The IDS could simply send an alert to administrators via its Control and Reporting Interface. Alternatively, the NIDS could automatically reset the attacker’s connection or add rule sets to the firewall or router to deny the attacker further access.
3. After the NIDS sends an alert via its Control and Reporting Interface, administrators can take action based on security policy. This may mean manually placing deny statements in the firewall rule set to deny the attacker, or reporting the attack to management and the proper authorities depending upon its severity.

We have discussed several varieties of IDS, all with different ways of accomplishing the same goal. The means by which these systems actually function can also be different. Typically, IDS are either signature-based detection devices or anomaly-based detection devices. We'll discuss each in this section.

Signature-Based IDS

The most prevalent form of intrusion detection is through signature matching. Referred to as signature-based IDS, these systems monitor the network or server and match packet traffic attributes against a set of predetermined attack lists or signatures. Should a particular network conversation match a signature configured on the IDS, the system alerts administrators or takes other pre-configured action.

Signature-based IDS can be quite effective in security monitoring, yet they have several drawbacks. To detect most potential attacks, the signature database on the IDS must be large. As the speed of networks increases, it is difficult for signature-based IDS to keep pace with network traffic. Typically, signature-based IDS must be de-tuned by removing some of the signatures from the active database before use. While this permits the IDS to function properly, it does so at the risk of missing potential attacks. Similarly, because these IDS only alert administrators as to potential attacks for which it has a signature, new vulnerabilities and exploits will not be detected until the vendors or administrators develop new signatures.

NOTE

Intrusion detection systems must be properly tuned once they're in the network environment. Because each signature within an IDS consumes system resources, it may not be advisable to load all signatures based on your network requirements and services. For instance, if you don't run a specific service or block access to the service at perimeter security devices, it might not be necessary to monitor for potential attacks against that service.

Anomaly-Based IDS

Anomaly-based IDS do not use static signatures to detect potential security events. Rather, these IDS use network traffic baselines to determine a “normal” state for the network and compare current traffic to that baseline. If network anomalies occur, the IDS alerts security administrators.

Two types of anomaly-based systems exist, behavior anomaly and protocol anomaly IDS. Both use the same type of statistical calculations to determine whether current traffic deviates from “normal” traffic, yet they specifically track different attributes. Behavior anomaly systems tend to monitor network resources using timing, volume, and similar resource characteristics while protocol anomaly IDS typically monitor application-level traits such as RFC compliancy and other operational protocol content attributes.

As compared to a signature-based IDS, an anomaly-based IDS has the potential to detect new attack vectors as they occur. Anomaly IDS, however, can suffer from numerous false positives as security administrators attempt to determine the dynamic definition of “normal” network operations.

Defeating an IDS

Intrusion detection systems are extremely helpful tools that aid security administrators in the ever-evolving task of securing the network. Using a variety of techniques previously discussed, these systems can monitor and alert the security team in many potentially harmful situations. This does not imply, however, that IDS are invincible. The art of managing intrusion detection systems is not simple and requires constant effort and attention.

We have already discussed several limitations of each type of intrusion detection system. All varieties can suffer from information overload in bandwidth intensive networks and most IDS require constant tuning and support. For instance, if signature-based IDS are not updated with the latest, most prevalent attack signatures, they will be ineffective against newly discovered vulnerabilities. Likewise, should new network applications be added or altered on the network, anomaly-based IDS must again run baselines against the new “normal” network state. Even if IDS are properly maintained and updated, the security team must respond properly and quickly to security events, otherwise the IDS is useless.

Network IDS must be positioned properly in the network and the network infrastructure must be appropriately configured to deliver traffic to the IDS. In most modern networks and certainly in large network environments, one IDS will not suffice. Multiple IDS (and oftentimes, multiple types of IDS) are therefore required for effective detection coverage, which necessitates good management practices and potentially, the use of IDS event correlation and aggregation servers.

There also exist methods by which an attacker may render IDS ineffective. These include DoS attacks directed at IDS infrastructure and other more focused attacks. For instance, if a hacker overloads a network with decoy attack signatures, he or she may be able to secretly exploit other code simultaneously and remain undetected by the IDS.

Another way attackers may elude IDS is through an act known as session slicing. This can occur when a malicious payload is successfully delivered over multiple packets and may defeat simple pattern- or signature- matching mechanisms. Oftentimes, this payload can be delivered over long time periods using various means, which leads to another vulnerability of IDS; slow scanning. Many IDS do not recognize attacks that occur over extended periods of time. If an attacker is patient enough, he or she may be able to elude IDS simply by working slowly.

IDS can also be bypassed by changing the default manner in which applications or network communications operate. For instance, if a signature-based system is looking for Back Orifice connections on TCP port 31337, a hacker

might simply change the TCP port to avoid detection. Similarly, if an attacker changes the sequence of exploit events, he or she may not trigger common network signature alert routines.

Finally, proxy attacks and spoofing are ways in which attack traffic may appear from internal, trusted hosts and may, therefore, be ignored by IDS.

Summary

We discussed several important security concepts and designs, and IDS types and functions in this chapter. The idea that network and systems security is essential in the modern enterprise environment is of utmost importance. Industry and government financial losses themselves merit the inclusion of good security practices, let alone the lost productivity and effort spent combating security events.

To assist security architects, Cisco created two comprehensive guides for the secure, modular, and efficient design and deployment of network security. Cisco AVVID provides the notion of a single, IP-based, resilient network infrastructure that acts as the foundation for all enterprise e-business and mission-critical operations. SAFE, when used in combination with AVVID, provides a complete security overlay that addresses all enterprise security options in a modular format. These guides, Cisco AVVID and Cisco SAFE, both provide indispensable and detailed information and should be fully understood by those charged with building today's network infrastructure.

Cisco also created a methodology known as the Cisco Security Wheel to aid in thorough and dynamic security management and operations. The Security Wheel is a cyclical process involving security policy, network security, monitoring and response, regular testing, and evolving management.

Through use of the security wheel, security teams can effectively respond to the four Cisco-identified primary threats and three attack types. The four threats include structured and unstructured threats, and internal and external threats. The three types of attacks are reconnaissance, access, and DoS attacks.

Finally, we examined the types and functions of intrusion detection systems. Host and Network IDS are the most deployed in networks, but there are other types of IDS, such as honeypots and Hybrid IDS. Most of these devices alert administrators about potential attacks via signature matching or anomalous event detection.

In preparation for the Cisco Secure Intrusion Detection Systems Exam, you should be well versed in these topics. Specifically, you should know and understand the underlying principles of network security in terms of Security Policy and practice. You should also be very familiar with intrusion detection terminology and the various means by which IDS function. Regardless of IDS functionality, it is important to know that no IDS is invincible. There are several means by which hackers can elude IDS, which makes a holistic security stance extremely important in the enterprise.

Solutions Fast Track

Understanding the AVVID Architecture

- ☑ The AVVID architecture central principal entails using a single converged IP network for voice, video, and data traffic.
- ☑ The three central components of the AVVID architecture are network infrastructure, application intelligence, and capable clients.
- ☑ The importance of security within the AVVID architecture increases dramatically as more services rely on the network infrastructure.
- ☑ AVVID is a standards-based enterprise architecture that can accelerate the integration of business requirements and technology

Understanding the SAFE Blueprint

- ☑ The SAFE blueprint augments the AVVID architecture by adding a security infrastructure that is modular and flexible enough to fit any environment from small- to enterprise-class networks.
- ☑ SAFE includes the key devices to be deployed in each module, and provides design guidelines and alternatives necessary to diffuse potential threats mitigated by each security solution.
- ☑ Because SAFE is modular, it can be rapidly implemented in most networks and can evolve as technologies and environments change. The recently developed SAFE IP Telephony and Wireless Network Security Modules are an example of SAFE's flexibility.

Threats

- ☑ There are four basic types of threats: unstructured, structured, internal, and external.
- ☑ There are three basic types of attacks: reconnaissance, access, and DoS attacks
- ☑ Internal threats are possibly the most detrimental to organizations since attackers are often armed with insider information.

Network Attacks

- ☑ Network attacks generally come in three forms: reconnaissance attacks, access attacks, and DoS attacks.
- ☑ Though not considered a highly dangerous attack form, reconnaissance attacks are often precursors to other types of attack.
- ☑ There are three types of access attacks: unauthorized data retrieval, unauthorized system access, and unauthorized privilege escalation.

Overview of IDS

- ☑ IDS come in various forms including Network IDS, Host IDS, Hybrid IDS, and honeypots, among others.
- ☑ Intrusion detection mechanisms are generally either based on signature-matching or anomaly-recognition routines.
- ☑ Regardless of the IDS type, form, and function, all IDS require regular maintenance and monitoring to act as a beneficial security tool.

Defeating an IDS

- ☑ IDS must be intelligently placed within the network and/or on critical servers to be effective.
- ☑ Signature databases and anomaly baselines must be constantly updated or IDS will create an increasing number of false positive and false negative alerts over time.
- ☑ Attack methods such as session splicing and DoS techniques may elude different IDS types. Don't forget the "slow" strike where the attacks are so slow they do not trip the scanner alarms.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: I already have a firewall. Why do I need IDS too?

A: Firewalls and IDS traditionally perform different roles in helping to security a network environment. Firewalls are typically used to prevent or allow access to systems and resources while IDS are generally used to monitor the actions and events on the network and systems. An increasing number of products perform both roles, but unless you have firewalls deployed in all critical areas of the network, you may not achieve a comprehensive view of potential security risks.

Q: Are there additional countermeasures other than IDS that I can deploy to safeguard my network?

A: At this time, some of the more deployed security countermeasures include firewalls, VPNs, encryption, and authentication, to name a few. Technology changes rapidly, however, so you should participate in the security community as much as possible to keep pace with new tools and concepts.

Q: Where should an IDS be deployed on my network?

A: Typically, an IDS is deployed near network ingress and egress points and by network-based services locations. Specifically, IDS may be deployed on critical hosts, the network perimeter, LAN and WAN backbones, and in server farms such as DMZ and extranets. We'll cover this in more detail in the next chapter.

Q: How do I test my IDS?

A: There are commercial and open source software products available to administrators to aid in testing IDS. Often, the tools used to test an IDS are also used to perform reconnaissance attacks. Examples of tools include scanning software such as nmap (www.insecure.org/nmap) and nessus (www.nessus.org).

- Q:** What is the difference between a false positive and a false negative?
- A:** A false positive is when normal and legitimate traffic or actions trigger an IDS alert. False negatives occur when malicious traffic does *not* trigger an alert. Properly tuned IDS should have a minimum of both types of situations.
- Q:** Where can I find more information regarding security policy development?
- A:** There are many resources available to security administrators regarding policy development and use. Good places to start include *RFC 2196 – The Site Security Handbook* (www.ietf.org/rfc/rfc2196.txt), the SANS’ “Design and Implementation of the Corporate Security Policy” document (www.sans.org/resources/policies), and the Computer Security Institute (www.gocsi.com).

Cisco Intrusion Detection

Solutions in this Chapter:

- What Is Cisco Intrusion Detection?
 - Cisco's Network Sensor Platforms
 - Cisco's Host Sensor Platforms
 - Managing Cisco IDS Sensors
 - Deploying Cisco IDS Sensors
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

In Chapter 1, we learned the fundamental principals and theory of security and intrusion detection systems. We also looked at Cisco-centric security mechanisms such as Cisco AVVID and SAFE. Cisco focuses on two primary types of IDSs, Host IDSs, and Network IDSs. Within each of these systems, Cisco develops products that promote an “active defense” to secure the network environment. Cisco Active Defense focuses on three points:

- **Detection** The ways and means to identify malicious attacks on networks and resources.
- **Prevention** How to stop detected attacks from being executed.
- **Reaction** How to immunize the systems from future attacks and provide real-time alerts.

We’ll learn that Cisco IDS sensors provide Active Defense detection using several methods, including signature detection and other hybrid techniques. We’ll also discuss the ways Cisco IDS can stop an attacker in his footsteps by sending TCP resets or dynamically manipulating firewall rule sets to prevent unwanted access. Finally, we’ll see how Cisco IDS solutions, such as the Host IDS sensor, can protect your resources, thwarting attacks through intelligent integration with application services and operating systems.

But, just what is Cisco Intrusion Detection? In this chapter, we’ll answer that question as we look closely at the specific Network and Host IDS platforms that comprise the Cisco IDS solution. We’ll discuss the 4200 IDS Sensor product line, the new IDS modules available for the Cisco Catalyst 6500 and Cisco 2600, 3600, and 3700 routers, and the Cisco Host IDS software.

Next, we’ll examine how to effectively manage the Cisco intrusion detection systems by using tools like Cisco IDS Event Viewer (IEV), IDS Device Manager (IDM), Cisco Secure Policy Manager (CSPM), and CiscoWorks VPN/Security Management Solution (VMS). Each of these tools has benefits for different environments and uses different mechanisms and protocols to communicate with Cisco IDSs in the network. We will discuss two protocols that Cisco has used to facilitate communication between the management stations and the sensors, the Cisco PostOffice Protocol and Cisco Remote Data Exchange protocol.

Finally, we’ll discuss in detail where Cisco IDS may be best deployed in the network. While each network environment requires different security approaches, there are several guiding principals regarding the intelligent and effective deployment of Cisco IDS.

Let's begin by defining Cisco Intrusion Detection.

What Is Cisco Intrusion Detection?

Cisco Intrusion Detection is a complete security approach that provides a wide range of intrusion detection capabilities to help administrators secure and monitor their network environments against threats and security breaches. Cisco Systems IDS solutions are based on four concepts:

- Accurate threat detection
- Intelligent threat investigation and mitigation
- Ease of management
- Flexible deployment options

Cisco delivers each of these concepts through flexible Network IDS hardware, host-based IDS software, Cisco IDS sensor software, and scalable Cisco IDS management software.

At the heart of the Cisco Intrusion Detection System is the Cisco Network and Host IDS software, which provides accurate threat detection, intelligent threat investigation and mitigation, and simplified management. The software imparts comprehensive threat detection, delivering a hybrid system that uses methods including stateful pattern recognition, protocol analysis, traffic anomaly detection, and protocol anomaly detection. With the software, unauthorized exploits, DoS activity, reconnaissance attempts, and other malicious actions are quickly detected.

Accurate detection leads to threat investigation and mitigation. When an attack is detected, Cisco's Threat Response technology works with Cisco IDSs to eliminate false alarms and escalate authentic attacks. This is accomplished using a three-step process involving:

- Basic investigation of target vulnerability
- Advanced investigation of targets
- Forensic data capture

Cisco IDSs are capable of several means of protecting a company's assets. Whether dropping an offending packet, terminating an attacker's session by using the TCP reset feature, dynamically reconfiguring Access Control Lists (ACLs) on routers and switches, or automatically modifying firewall policies, Cisco IDS offers an array of immediate response actions to stop attacks in near-real time.

Cisco understands the potential difficulties involved with managing network and security infrastructure. To alleviate management impediments, Cisco provides a series of management options that offer ease of use and centralized management. With tools like the Cisco IDS Event Viewer, IDS Device Manager, Secure Policy Manager, and the CiscoWorks VPN/Security Management Solution, administrators have many powerful options at their fingertips.

The Cisco Network IDS solution set includes appliance-based intrusion detection through the Cisco 4200 line of sensors. Ranging from performance options between 45 Mbps to 1 Gbps, the 4200 series offers multiple options for security administrators and can be quickly and easily integrated into network environments. Cisco also helps companies leverage existing switching and routing infrastructures through use of the Cisco Catalyst 6500 IDSM and Cisco IDS Module for 2600, 3600, and 3700 routers. These modules integrate seamlessly into existing hardware to provide additional network security. And last but certainly not least, network IDS functionality is available in routers through an integrated but limited IOS functionality.

Cisco Host IDS works on the service endpoints in the network. Installed on hosts such as web and mail servers, the host sensor software protects operating systems and application-level functionality through tight integration. This is accomplished by inspecting all interaction with the operating system and comparing the requests for service against a database of known attacks. Should the request match a known exploit, the request for service will be terminated by the sensor software. Along with preventing known attacks, the Host sensor can also protect against generic or unknown exploits by preventing dangerous situations such as buffer overruns, a typical result of hacker exploits. Finally, the Host IDS software acts as a shield against intentional file corruption attempts, such as Trojan code insertion attacks. This is performed by “fingerprinting” executables and configuration files during baseline operations. This fingerprint or checksum is then regularly compared to the current version to protect system resources such as Registry keys, password files, and executables against unwanted manipulation.

Cisco's Network Sensor Platforms

As part of their flexible deployment strategy, Cisco offers several different Network IDS platforms to meet the varying needs of enterprise environments. Included in the Network IDS suite of products are the Cisco IDS 4200 Series

sensors, the Cisco Catalyst 6000 IDS Modules, Cisco IDS Modules for 2600, 3600, and 3700 routers, and the Cisco router and firewall-based sensors. All of these devices represent the cost-effective, comprehensive security solutions Cisco can provide for custom-tailored network performance needs.

From the affordable Cisco IDS 4210 to the high performance IDS 4250XL, the Cisco 4200 Series devices provide an appliance-based detection system. Refer to Table 2.1 for details regarding the Cisco IDS platforms.

Cisco IDS Appliances

At the core of Cisco's IDS solution are the dedicated IDS sensors that compose the 4200 series. These appliance-based products are available in five performance levels as follows:

- Cisco IDS 4210—45 Mbps
- Cisco IDS 4215—80 Mbps
- Cisco IDS 4230—100 Mbps
- Cisco IDS 4235—250 Mbps
- Cisco IDS 4250—500 Mbps
- Cisco IDS 4250 XL—1000 Mbps

Each specific sensor incorporates the same richly featured functionality of Cisco IDS 4.0 software, yet has different interface and internal hardware that imposes varied traffic processing limitations. The flexibility of these small form factor devices facilitates easy integration into different environments from SOHO to enterprise to service provider networks.

Cisco rates the performance of their devices based on specific traffic variables such as new and concurrent TCP or HTTP sessions and average packet size. For instance, the performance rating of all the 4200 Series IDS sensors, except the 4250 XL, is based on an average packet size of 445 bytes. The 4250 XL Gigabit performance is based on 595 bytes packets. In general, smaller packet sizes add an increased overhead as devices must process more header information per number of packets vs. a smaller number of larger packets with less header overhead which will result in reduced performance.

Table 2.1 The Cisco Sensor Capability Matrix

Sensor	Throughput	Monitoring Interface	Control Interface	Optional Interfaces	RU
Cisco IDS 4210	45 Mbps	1 10/100 Base-TX	1 10/100Base-TX	N/A	1
Cisco IDS 4215	80 Mbps	1 10/100 Base-TX	1 10/100Base-TX	Four 10/100 BaseTX sniffing interfaces	1
Cisco IDS 4230	100 Mbps	1 10/100 Base-TX	1 10/100Base-TX	N/A	4
Cisco IDS 4235	250 Mbps	1 10/100/1000 Base-TX	1 10/100/1000 Base-TX	Four 10/100 BaseTX sniffing interfaces	1
Cisco IDS 4250	500 Mbps	1 10/100/1000 Base-TX	1 10/100/1000 Base-TX	Four 10/100Base-TX One 1000Base-SX	1
Cisco IDS 4250XL	1 Gbps	2 1000Base-SX (MTRJ)	1 10/100/1000 Base-TX	One 1000Base-SX	1
Cisco IDS Module for 2600, 3600, and 3700 Router	2600: 10 Mbps 3600: 45 Mbps 3700: 45 Mbps	Router internal bus	1 10/100/1000 Base-TX	N/A	1 Network Module Slot
Cisco IDS Module for 6500 Switch	600 Mbps	Switch backplane	Via Switch or direct Telnet	N/A	1 Slot

4210 Sensor

The Cisco 4210 Sensor is the newest member to the 4200 series lineup. It is a rack mountable, 1RU device that can deliver up to 45 Mbps of traffic analysis. The 4210 has two fixed ports, both 10/100Base-TX (Fast Ethernet) to be used for monitoring and control. Due to its processing capabilities, the Cisco 4210 is optimized to monitor multiple T1/E1, T3, or Ethernet environments. The 4210 could also function as a sensor in partially loaded Fast Ethernet environments.

The Cisco 4210 is ideally suited for SOHO, remote office locations, and other low bandwidth demand environments.

4215 Sensor

Similar to the 4210, the Cisco 4215 Sensor is a sensor designed for network infrastructure running at less than Fast Ethernet speeds. The 4215 could perform adequately in a typical partially loaded 100 Mbps environment. Capable of 80 Mbps, the 4215 improves upon the 4210 in throughput capability and in potential maximum interfaces. Instead of only one monitoring interface like the 4210, the 4215 has four additional (and optional) monitoring interfaces. This means that with the primary monitoring interface, the 4215 is able to provide intrusion detection on five different interfaces.

Because of the improved interface density, the 4215 is well suited for monitoring multiple, discrete network segments such as internal, external, and DMZ networks. Like most of the 4200 Series devices, the 4215 is 1 rack unit in height, making it a good fit for tight equipment rooms and closets.

4230 Sensor

The 4230 Sensor is one of the older models in the 4200 series. In fact, the Cisco IDS 4230 sensor was end-of-sale (EOS) as of July, 2002. While software and hardware support will continue for a limited time, this device is no longer available from Cisco. Instead, Cisco recommends the use of the 4235 sensor based on improved performance, size, and port density. We'll discuss the 4230 sensor in this chapter because the hardware is still included in the CSIDS 9E0-100 exam.

The 4230 sensor is a dual Pentium III-based sensor with two fixed 10/100Base-T ports. Like the 4210, one is reserved for monitoring, while the other is intended for command and control access. The 4230 is capable of handling 100 Mbps, which makes it a good choice for Fast Ethernet environments. At four RU, the 4230 is a larger device than the other 4200 series sensors.

4235 Sensor

As the replacement of the 4230 sensor, the 4235 improves on size, performance, port density, and port capacity. The 4235 offers performance up to 250 Mbps and due to its 10/100/1000-capable TX monitoring interface, the 4235 can be used in partially loaded gigabit environments. Ideally, the 4235 is suited for multiple T3 networks or high-speed switched environments.

The 4235 sensor, like the 4215, has the option of four additional 10/100Base-TX interfaces enabling IDS capabilities on multiple networks with one device. The 4235 is one RU in height and has a gigabit-capable control interface.

4250 Sensor

The Cisco IDS 4250 sensor incorporates many of the features of the 4235 sensor, but with increased performance of 500 Mbps. The 4250 is also the only 4200 series sensor that is scalable via a simple hardware upgrade for full line-rate gigabit performance. At one RU, the 4250 has a 10/100/1000Base-TX control and monitoring port. The 4250 also has the option of four additional 10/100Base-TX interfaces or one additional 1000Base-SX SC fiber interface. This flexibility enables the use of the 4250 in various environments including gigabit subnets and on switches used to aggregate traffic from numerous subnets.

4250 XL Sensor

The most capable of the Cisco 4200 IDS series, the 4250 XL performs at gigabit speeds and is ideal for fully or partially saturated gigabit network environments. Like the other sensors, the 4250 XL is one RU, but accommodates dual 1000Base-SX monitoring interfaces with MTRJ connectors. The 4250 XL also has a 10/100/1000Base-TX control interface and an additional and optional 1000Base-SX SC monitoring interface.

The Cisco IDS Module for Cisco 2600, 3600, and 3700 Routers

With the recent addition of the Cisco IDS Module for the 2600XM, 3600, and 3700 Cisco routers, Cisco provides affordable and capable intrusion detection services in small office and branch office environments. The module provides security on WAN links and reduces operational costs through integration with existing equipment.

The IDS module fits on a single network module on the router. It has a 20GB onboard IDE hard disk for event storage and logging and provides a single 10/100 Fast Ethernet port for command and control. Because it monitors data directly from the router bus, the module does not require a monitoring port. In a 2600XM, the IDS module can process 10 Mbps of data. In the 3600 and 3700, it can process 45 Mbps. Only one IDS module can function in the routing device.

The IDS module runs the same Cisco IDS 4.0 software that the 4200 series IDS sensors do giving the router full IDS capabilities. Furthermore, the module provides the ability to inspect traffic traversing the router on any interface and, given an attack signature detection, can either shutdown router interfaces or send TCP resets to terminate the offending TCP session

NOTE

The IDS router module requires the IOS FW/IDS feature set and Cisco IOS 12.2(15)ZJ or later.

The Cisco 6500 Series IDS Services Module

Like the IDS Module for Cisco routers, Cisco also offers a module for the Cisco 6500 series switch. Referred to as the IDSM, the module occupies one or more slots in the 6500 chassis, making it an excellent IDS sensor choice in networks where the 6500 platform is already deployed. There are two revisions of the IDSM, the IDSM-1 and the IDSM-2. The IDSM-2 is a far more capable device offering five times the performance of the IDSM-1. The IDSM-1 has been EOL and is no longer supported either with service packs or signature updates. Some of the other differences in functionality between the revisions are highlighted in Table 2.2.

Table 2.2 IDSM-1 vs. IDSM-2 Comparison

Functionality	IDSM-1	IDSM-2
Performance	250 Mbps	600 Mbps
SPAN/RSPAN	X	X
VACL Capture	X	X
Shunning	X	X
IEV	X	X

Continued

Table 2.2 IDSM-1 vs. IDSM-2 Comparison

Functionality	IDSM-1	IDSM-2
VMS	X	X
IDM		X
TCP Resets		X
IP Logging		X
CLI		X
Signature Micro Engines		X
Same Code as Appliances		X
Fabric Enabled		X
SNMP		
Unix Director	X	
CSPM	X	
Event retrieval method	PostOffice	RDEP
Slot Size (form factor)	1 RU	1RU
Local Event Store	100,000 Events	N/A, retrieved

As can be seen, the IDSM-2 module has far greater capabilities. Indeed, because it runs the Cisco IDS 4.0 software, it incorporates all of the functionality of the Cisco 4200 IDS series appliances while delivering 600 Mbps of performance. The benefit of the IDSM is that it takes data directly from the switch backplane and can monitor any traffic sent across the switch. Data to be monitored can be specified by SPAN and RSPAN or by VLANS via VACL capture mechanisms.

Besides performance, noteworthy differences between the two revisions include more management capabilities and more security features. For instance, the IDSM-2 module facilitates management via the Cisco VPN/Security Management Solution (VMS), Cisco IDS Device Manager (IDM), IDS Event Viewer (IEV), and via the CLI. Additionally, the IDSM-2 supports advanced IDS features such as TCP Resets, IP Logging, and Signature Micro Engines while the IDSM-1 does not. Also, the new IDSM supports Cisco's new method of event retrieval, Remote Data Exchange Protocol (RDEP) whereas IDSM-1 supports PostOffice Protocol only.

On the IDSM-2 there is no limit to the number of VLANs monitored on the module and no impact to traffic traversing the switch. Furthermore, the only

limit to the number of IDS modules in a Catalyst 6500 is the number of free slots in the chassis. Finally, it should be noted that Cisco no longer sells the IDSM-1 as of April, 2003. All of this information and more will be discussed in detail in Chapter 6, which focuses on the IDSM solution specifically.

Cisco's Host Sensor Platforms

Cisco also offers Host IDS to protect the service endpoints distributed in the network. The Cisco HIDS solution is based on Enterscept functionality and augments Cisco's NIDS capabilities as proscribed in the AVVID architecture and SAFE blueprint. Two forms of the sensor are available, the Standard Agent and the Web Edition Agent. While both lend critical, focused functionality to the protection of host systems, the Web Edition includes all Standard Agent functionality and adds protective measures specifically for web servers. We'll discuss both of these agents next.

The software is distributed to the critical systems on the network, yet is controlled via a centralized, secure console for ease of management. From the Cisco IDS Host Sensor Console, administrators can configure and manage all sensors in the network. For instance, as new attack signatures are regularly made available by the Cisco Countermeasures Research Team (C-CRT), security administrators simply download the new signatures to the console, then upload them to the various NIDSs via a centralized process. Additionally, the Cisco VMS software can be used should administrators already be running CiscoWorks to manage other NIDS and security devices in the network. The Cisco IDS Host Sensor software is capable of protecting the following platforms:

- **Standard Agent:**
 - Windows 2000 Server and Advanced Server (up to Service Pack 2)
 - Windows NT v4.0 Server and Enterprise Server (Service Pack 4 or later)
 - Solaris 2.6 SPARC architecture 4u (32-bit kernel)
 - Solaris 7 SPARC architecture 4u (32- and 64-bit kernel)
 - Solaris 8 SPARC architecture 4u (32- and 64-bit kernel)
- **Web Edition Agent (includes all Standard Agent functionality):**
 - All Standard Agent OS platforms

- Web servers as follows:
 - Microsoft IIS v4.0 and v5.0
 - Apache v1.3.6 through v1.3.24 for Solaris SPARC (Apache on Windows NT/2000 and LINUX is not supported)
 - Planet Web Server v4.0 and v4.1 and v6 for Solaris SPARC
 - Netscape Enterprise Server v3.6 for Solaris SPARC
- **Console Agent:**
 - Windows 2000 Server and Advanced Server (SP1 and SP2)
 - Microsoft Windows NT Server (SP6a)

Cisco Host Sensor

Capable of running on various operating systems such as Windows or Solaris, the Cisco IDS Host Sensor integrates into the host OS to protect it from malicious intent. The Host Sensor not only inspects inbound traffic destined for the server, but also intercepts system calls, adding an extra and complete layer of security. This capability allows the sensor to understand the processes and users triggering the system call as well as the resources required for the call. Armed with this information, the sensor applies a combination of behavioral rules and attack signatures to determine whether the system activity is benign or malicious. Should abnormal activity be detected, the sensor has the power to terminate the system call and alert security administrators.

Due to the software design, the Host Sensor Standard Agent can prevent malicious activity in several ways. As we've discussed, the sensor uses known attack signatures to distinguish normal and harmful activity. Because Cisco maintains dedicated resources for the development of timely attack signatures, the Cisco Host Sensor will always be ready and able to detect the latest threats.

From Chapter 1, we know that signature-based detection systems are vulnerable during the time between new exploit discovery and protective signature development. To combat this issue, Cisco provides an additional layer of protection via behavior anomaly detection capabilities on the sensor. This helps detect and prevent previously unknown attacks until a signature can be developed. Should a call or action on a server violate predefined and normal behavioral patterns, the sensor can block the malicious activity and alert the security team.

Because the sensor software is fully integrated with the host operating system, the software can also prevent arbitrary code execution, possibly due to buffer

overflow exploits. This functionality is critical since over 60 percent of Computer Emergency Response Team (CERT) security advisories result from buffer overflow exploits.

The tight integration also permits the host sensor to protect the operating system's critical resources and files such as configuration files, Registry settings, and binaries that are often the focus of an attack. Similarly, the sensor also prevents unauthorized privilege escalation by securing user permissions and configurations.

The Web Edition Agent includes all Standard Agent functionality, yet includes additional protective mechanisms to prevent web server-specific attacks. When installed, the Web Edition Agent automatically determines and adapts to the existing Apache, iPlanet, or IIS web server. It can then act as a protective element that parses HTTP streams, inspecting the TCP conversations for malicious logic and blocking potential attacks before they reach the server. Because the Agent sits on the server, it can examine web requests without obfuscation by application-level encryption techniques such as Secure Sockets Layer (SSL) thereby adding additional security that Network IDS cannot provide.

Managing Cisco's IDS Sensors

In conjunction with Cisco's flexible approach to security management, Cisco has developed several means of managing IDS platforms in the network. Each has different intents and benefits to better address the varying needs of security administrators. Some of the methods by which security professionals can manage their Network IDS infrastructure include

- Command Line Interface (CLI) via console, Telnet, or SSH access
- Cisco IDS Event Viewer (IEV)
- Cisco IDS Device Manager (IDM)
- Cisco Secure Policy Manager (CSPM)
- CiscoWorks VPN/Security Management Solution (VMS)

Of these management techniques, all but CSPM and CiscoWorks VMS are provided as part of the Cisco IDS 4.0 Sensor software. Cisco Host IDS sensors can also be managed by VMS or, for smaller environments, by the Cisco IDS Host Sensor Console software. While we'll briefly examine each of these methods in this section, these administrative tools will be covered in detail in subsequent chapters.

As the most simple and perhaps quickest method of management, the CLI is available on all NIDS products, including the IDS modules for Cisco routers and switches. The CLI is accessible from the device console, but also from remote terminals via Telnet and Secure Shell (SSH). Using the CLI enables administrators to efficiently monitor and maintain their devices from virtually anywhere in the network.

The Cisco IEV and IDM are both graphical interface tools that allow administrators less experienced in CLI operations to manage the IDS infrastructure. IEV is a Java-based event viewer that runs on Windows platforms and includes MySQL as a backend database for event storage. Using IEV, administrators can view event and alert data from up to five IDS sensors in the network. The Cisco IDM is a browser-based configuration tool from which the security team can view and manipulate any number of IDS devices in the network. Although the IDM can be used to manage over 1000 IDS devices, Cisco typically recommends a ratio of 20 to 25 sensors per management console. Additional sensors can overwhelm administrators with high volumes of information that they may be required to analyze. For deployments larger than 25 sensors, multiple IDM consoles should be installed to manage the sensors. Both Cisco IEV and IDM provide secure management of the IDS infrastructure through Secure Sockets Layer-based (SSL) access.

Alternatively, large enterprise administrators may choose to implement Cisco VMS. Cisco VMS can run on either a Windows/Intel platform or on a Sun SPARC running Solaris. The Cisco VMS software is part of the CiscoWorks Suite of products and is intended as a large-scale, enterprise solution for security management. Using the VMS, an organization can manage all of their security devices including

- Cisco Network IDS sensors
- Cisco Switch IDSM sensors
- Cisco IDS Network Module for routers
- Management Center for Cisco Security Agents
- Cisco PIX Firewalls
- Cisco Firewall Services Modules
- Cisco IOS Routers
- Cisco Host IDS

As can be seen, the Cisco VMS enables an enterprise wide view of security. It includes all of the IDS management capabilities available via IEV, IDM, and the CLI, but facilitates management of a far greater scale of devices. VMS has several modules itself. Among those, the Cisco VMS 2.1 Security Monitor and IDS Management Center v1.1 are required from IDS management.

Because security devices (such as IDS) transport potentially sensitive data, secure techniques, such as SSH, IEV, or IDM, should be used to monitor and maintain the security infrastructure. Cisco has also developed two protocols by which IDS equipment can be managed, PostOffice Protocol and Remote Data Exchange Protocol (RDEP). We'll discuss both of these protocols next.

Cisco PostOffice Protocol

To manage and maintain the Cisco IDS devices, Cisco first developed a proprietary protocol known as PostOffice Protocol. It is now being replaced by RDEP, which we'll describe later. The PostOffice Protocol is not to be confused with the Post Office Protocol POP3 (TCP port 110) commonly used by mail clients to retrieve Internet mail. Rather, the Cisco PostOffice Protocol is a UDP service that functions, by default, over port 45000 to provide messaging between the management console and IDS sensors. After Cisco IDS Software Version 2.2.1, this default port is configurable. The PostOffice Protocol provides messaging for:

- Command data
- Error and alarm messages
- Command and IP logs
- Redirects
- Device heartbeats

The PostOffice Protocol is primarily a “push” technology as opposed to the “pull” mechanism of RDEP. Because PostOffice Protocol was the primary means of communication between security devices, Cisco developed reliability, redundancy, and fault-tolerance schemes within the protocol to ensure messaging success.

While a UDP-based service, PostOffice Protocol requires acknowledgement of alarm message delivery. This promotes reliability since the IDS sensor will continue to send alert messages until it receives acknowledgement from the console.

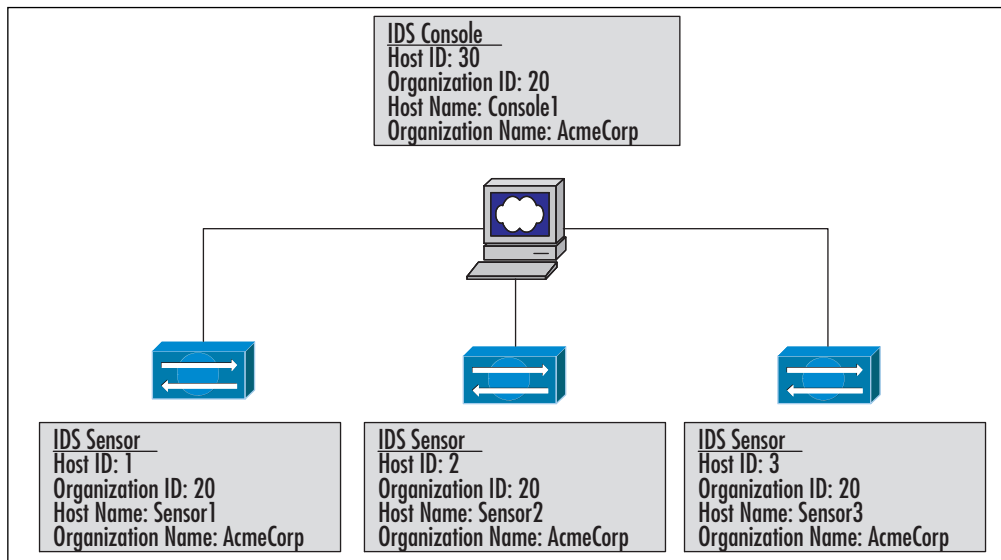
Redundancy and fault tolerance are enabled via multiple IDS console devices configured to service the same group of sensors. The PostOffice Protocol permits sensors to propagate messages up to 255 destinations, which allows for redundant alarm notifications and ensures the appropriate personnel are notified when an alarm is received. Similarly, up to 255 addresses can be specified for a single console host. This facilitates fault tolerance; should one route to a console address fail, another could easily initiate connectivity.

With PostOffice, administrators must assign each IDS sensor a unique identifier composed of some of the following attributes:

- **Host ID** The Host ID must be a unique numeric value greater than zero, such as 30.
- **Organization ID** The Organization ID must be a numeric value greater than zero, such as 100. This number can be the same for multiple sensors.
- **Host name** The Host name is an alphanumeric string that identifies the host, such as Sensor1B.
- **Organization name** The Organization name is an alphanumeric string that identifies the company or organization, such as AcmeCorp.

An example of the PostOffice naming convention is shown in Figure 2.1.

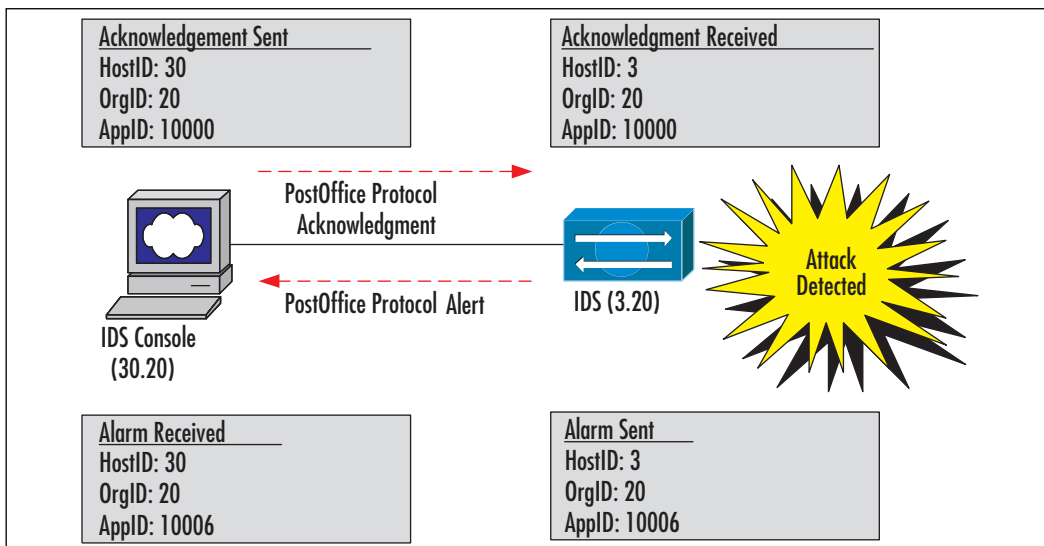
Figure 2.1 PostOffice Protocol Addressing



This helps the security team identify sensors in large environments, but it is also required for the PostOffice Addressing scheme, which is composed of three components. The host and organization identifiers signify the first two components of the addressing scheme, while the third component is a unique application identifier. All three of these unique identifiers are used by the protocol to route command and control communications.

For example, in Figure 2.2, a sensor with Host ID 3 and Org ID 20 issues a PostOffice Protocol alert using Application ID 10006 destined for an IDS console with Host ID 30 and Org ID 20. Upon receiving the alert, the Console acknowledges it via Application ID 10000 to the sensor.

Figure 2.2 PostOffice Addressing Scheme



Remote Data Exchange Protocol

As of the Cisco IDS 4.0 software, PostOffice Protocol is no longer used for communication between console and IDS sensor devices. Instead, Cisco implements the Remote Data Exchange Protocol (RDEP), which is a proprietary HTTP and XML-based configuration and event generation messaging system. It employs “pull” mechanisms for event collection and analysis.

With Cisco IDS 4.0 Sensors, management and control functionality used an SSL-based XML messaging format for communication. Alarm notification from sensors still requires acknowledgement as it did with PostOffice Protocol. The RDEP protocol is TCP-based however, so it employs the reliability routines pre-

sent in TCP as well. Because the transport uses Secure Socket Layer to encrypt communications, the protocol is secure.

The RDEP protocol is simpler and easier to manage than the PostOffice Protocol. It uses well-known TCP port 443 by default for quick firewall rule set modification. When configuring RDEP communications, administrators will need to provide a device name for the sensor, whether they intend to use encryption for communication, and on what port they wish to run the service.

Deploying Cisco IDS Sensors

In the first chapter, we briefly discussed some of the best practices related to planning and managing the implementation of IDS sensors. In general, security architects will find that IDS is best deployed near the ingress/egress points of the network. This could include locations such as the following:

- **Internet-connected Networks** An IDS connected near the Internet/Corporate demarcation point provides insight into all traffic destined to and from the corporate network.
- **Extranet Networks** IDSs near vendor and partner portals or gateways provide visibility into these mixed zone, semi-trusted networks.
- **Intranet Networks** IDSs at the gateway routers and firewalls between divisions such as Accounting, Human Resources, and other sensitive internal groups.
- **Remote Access Networks** Don't forget the alternative points of entry and exit to your network. Remote Access Networks could include traditional dialup RAS network, broadband VPN demarcation points, or Wireless Access Points.

We also covered security policy generation through the Cisco Security Wheel methodology and studied the Cisco AVVID architecture and SAFE blueprint. All of these resources can help security architects and administrators decide the most effective locations to place IDS in the infrastructure.

Intelligent deployment of Cisco IDS sensors involves at a minimum, three steps. These include

1. Understanding and analyzing the network
2. Identifying the critical infrastructure and services
3. Placing sensors based on network and services function

We'll discuss each of these steps in this section.

NOTE

Securing the network is part of the Secure step in the Cisco Security Wheel process, which comes after building security policy. If administrators are in the process of deciding where to deploy IDS, it is assumed they have generated a comprehensive and solid security policy complete with security zone definition and other critical attributes of the policy.

Understanding and Analyzing the Network

Intelligent IDS deployment requires detailed knowledge and analysis of the network as a whole. As we discussed in Chapter 1, this involves gathering and understanding attributes such as overall network size and topology, ingress and egress points, service locations, and general application flow parameters. In small environments this may be simple, but in large enterprise networks, a comprehensive appreciation of the routing and content switching foundation can be quite a task.

You should start with a map of the network, examining the topology from a routed or Layer 3 perspective. You need to gain an understanding of the routed environment first. As part of the audit, you should scrutinize active/active, redundant networks. Since asynchronous routing and switching can create havoc on IDS systems; the IDS sensor needs to inspect the entire dataflow or conversation to be effective. Understand the perimeter security devices where access may be permitted or denied. Also, you should understand the impact of IP version 6 and VPN encryption—both of these can defeat IDS. It may also be necessary to learn the Layer-2 design of the network, especially in large ATM or MPLS clouds, since communities of interest are often aggregated on the same physical network platform.

After full comprehension of the Layer-3 environment, you should work up the OSI model to Layer 7, the application layer. Make an overlay of the Layer-3 network map by placing services flow information on the routed links. This will help you understand which links in the network carry the most critical application traffic such as web or e-mail requests. It will also help you understand the next step, Identifying the Critical Infrastructure and Services.

Finally, using the previously developed security policy, verify that the security zones are properly defined and examine how they interact with the routed and application environment. Understanding the traffic and how it flows across the network is an essential step in planning IDS implementations.

Identifying the Critical Infrastructure and Services

As part of the network analysis, security administrators should identify the critical components both in terms of networks and service. After all, the network exists only to get people and machines to application services! On the network map, place symbols near the endpoints of critical services remembering the function of IDS and the Cisco SAFE axioms:

- **Routers are targets** As an active element in the network, hackers can direct attacks towards routers to disrupt a large number of services and network connections with one strike. For instance, in July of 2003, a vulnerability in Cisco IOS (CERT Advisory CA-2003-15) was discovered affecting Cisco devices. By sending specially crafted IPv4 packets to an interface on a vulnerable device, an intruder could cause the device to stop processing packets destined to that interface. By targeting routers with this vulnerability, a hacker could effectively shut down a Cisco-based network. Cisco quickly released fix code for the vulnerability.
- **Switches are targets** Similar to routers, switches serve as an active element in the network. Disrupting their functionality through a DoS attack or by manipulating their configuration could impact large groups of people. Some Cisco switches were affected by the vulnerability as discussed earlier.
- **Hosts are targets** One of the most dangerous evolutions in hacking involves using compromised hosts as unwitting attackers in a large scale Distributed DoS (DDoS) attack. This type of attack was used in the well-known Nimda worm. Oftentimes, hosts are used in “blended threats” where a combination of worms, Trojan horses, and other malicious code is instantiated on hosts for use in a secondary attack such as a DDoS.
- **Networks are targets** Networks are only functional with the cooperative interaction of many router, switches, and other active elements. Large-scale attacks or blended threats can disrupt networks as a whole. A

good example occurred when the Slammer Worm was unleashed (CERT Advisory CA-2003-04) and many Internet-connected networks ground to a halt under the load of UDP worm traffic.

- **Applications are targets** Application functionality is the primary reason networks exist—we all connect to the network to access some form of application. It may be a file share or a web site or perhaps a database to which we seek access. Regardless, applications are a traditional favorite of hackers since they contain vital information and can, when compromised, affect such a large community.

In a well-developed network and systems architecture, services should be aggregated in high bandwidth, manageable farms. Often, these are in DMZs, extranets, or intranets. Regardless, it is most likely that the map will highlight the following locations as critical:

- Internet ingress/egress points
- Server farm ingress/egress points
- Remote Access networks
- Wireless access points

Because wireless access points can involve encryption such as WEP, they, and VPNs in general, present a challenge for IDS systems. The encryption prevents IDS sensors from gaining cleartext access to the payload, and in some instances, the packet header and payload. Since IDS cannot decrypt these datastreams, the traffic passes without IDS inspection. This is precisely why it is beneficial to place IDS at the point of decryption in networks so that you may gain insight into the traffic passing through the tunnel.

In most instances, the critical network and services locations will be near existing security infrastructures such as firewalls. Once the critical infrastructure has been mapped, it's time to select the placement of sensors.

Placing Sensors Based on Network and Services Function

With technological changes and new threats, the placement of intrusion detection systems has evolved over time. Initially, IDSs were typically deployed only at the Internet ingress/egress point, outside the company firewall. With the understanding that perhaps most malicious activity emanates from within an organization, this

approach proved inadequate in monitoring all security threats. Now, with cost-effective, more advanced management techniques and software, an increased number of IDSs can typically be supported.

NOTE

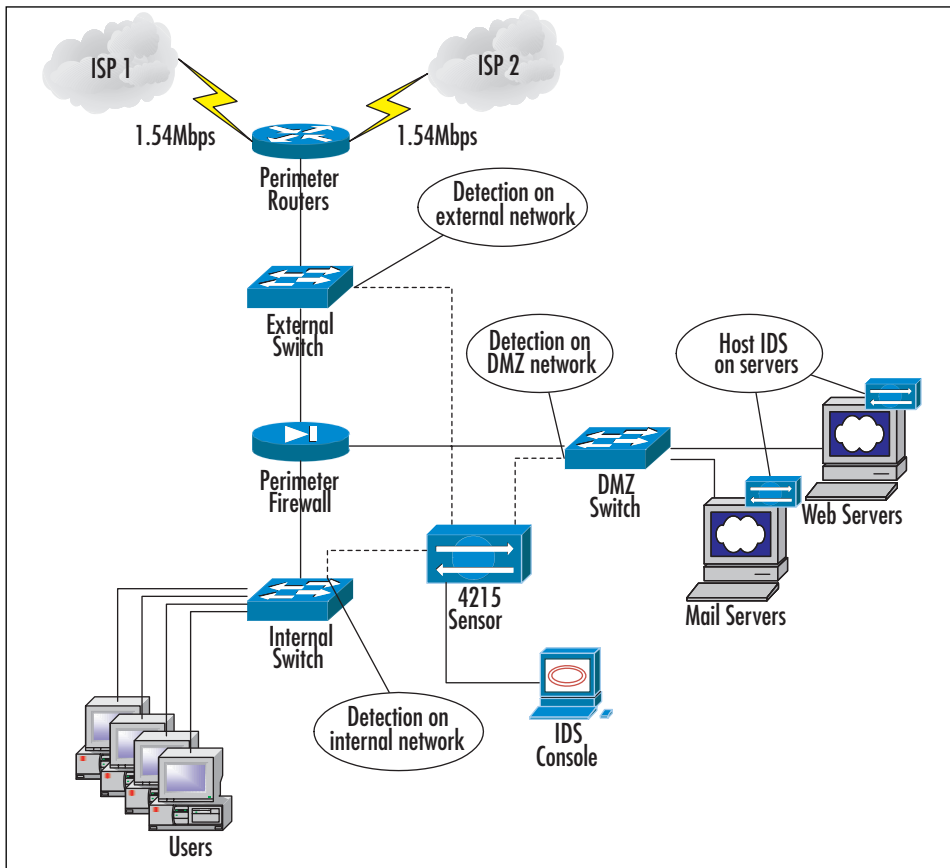
When placing an IDS, don't forget to consider how to connect to the devices for management purposes once they are placed in the network. Security architects should design and build efficient and reliable networks over which to manage the security infrastructure.

With the Cisco IDSM sensor modules and 4250 XL sensors, it is often possible to place IDS in core network environments. In many ways, this makes good sense, since a lot of traffic traverses the core network in many network architectures and it is simply not feasible to position IDS in every distribution and/or access device. If the IDS deployed in an organization can handle the core network speeds, it is generally recommended to place equipment there.

IDSs should also be positioned near the areas considered as critical in the previous steps. This may mean that IDSs are deployed on DMZs, above or below firewalls, and near alternative network access locations such as RAS or WAP segments. Let's look at a couple examples that illustrate the placement of an IDS.

Case Study 1: Small IDS Deployment

Our first example (Figure 2.3) involves the Nittany Corporation, who has a small internal network and a server farm DMZ that houses all internally and externally accessed services. The organization relies heavily on its e-commerce web site and e-mail server for business success.

Figure 2.3 Simple IDS Deployment

After fully investigating the network architecture, the security administrator knows that a lot of potentially dangerous network traffic flows from the Internet to the DMZ. She makes this network her first priority for IDS. She also knows that the web and e-mail servers are absolutely critical to business, so she chooses to deploy host sensors on these servers for extra application layer protection. Finally, the security administrator knows, based on firewall alerts and log files, that a lot of attacks are directed towards the internal network of her company.

The Nittany Company is small, however, and is restricted to a fairly tight budget. Thus, it cannot afford multiple IDS sensors.

While the primary intent of the IDS deployment may be to safeguard the company's critical servers, the company can get the added benefits of multinet-work coverage by selecting the Cisco 4215 IDS Sensor. By using the optional 10/100Base-TX interfaces, the security administrator can simultaneously monitor

the external, internal, and DMZ networks as shown earlier. Since the 4215 is capable of performing at 80 Mbps, it is a good choice—the company’s internal network is only 100 Mbps and the dual Internet connections provide roughly 3 Mbps maximum combined throughput.

Furthermore, because she’s selected to install Cisco Host IDS sensors on the critical servers, the Nittany Corporation will have extra protection at the service endpoints operating systems and at the application layer.

From a cost perspective, this solution allows the company to deploy IDS in multiple network segments without the cost of additional IDS sensors.

Case Study 2: Complex IDS Deployment

The second example involves a larger, more complex network and services environment with high bandwidth requirements. In this example, the ACME Company is a large defense contracting organization with a headquarters campus network and remote offices in seven cities. While each location has its own security infrastructure, headquarters contains most internally and externally sought services. Network and services operations are centrally managed from the headquarters office.

As a consultant, you have been asked to review the ACME Company security stance with specific regards to Intrusion Detection. ACME has a very limited deployment of IDS, but, because of recent hacking and worm attack problems, seeks to deploy an enterprise-wide IDS solution.

So, where do you start? Based on what we’ve discussed so far, you should remember that intelligent deployment of Cisco IDS sensors involves, at a minimum, three steps as follows:

1. Understanding and analyzing the network
2. Identifying the critical infrastructure and services
3. Placing sensors based on network and services functions

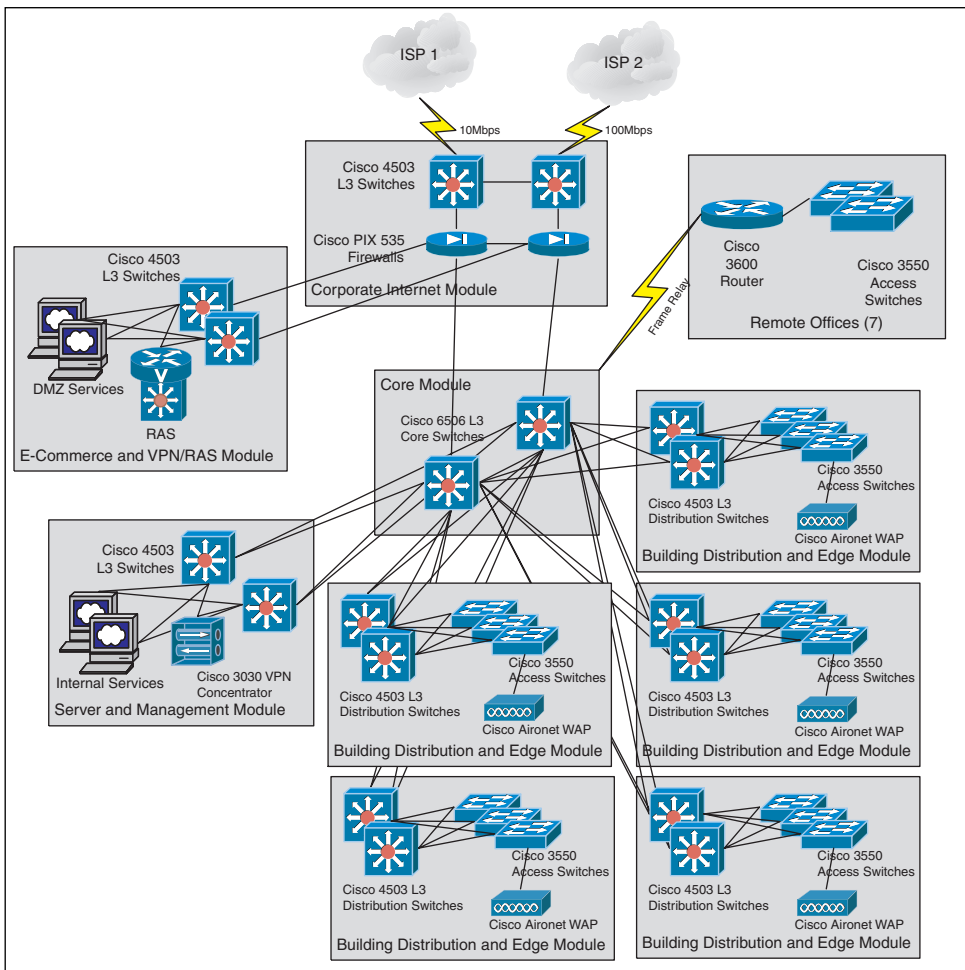
You should also remember the Cisco AVVID and SAFE information from Chapter 1. Your first step is to map the network to understand how routing, switching, and traffic flow occurs in the ACME Company. While you’re drawing, you add the SAFE modular design to the map for reference.

NOTE

To simplify the network map, some SAFE modules are combined where possible in Figure 2.4.

When finished, your map should look like Figure 2.4.

Figure 2.4 Complex IDS Deployment Network Map



In your research, you determine ACME is using BGP in the Corporate Internet Module to provide redundant and load-balanced access to the Internet.

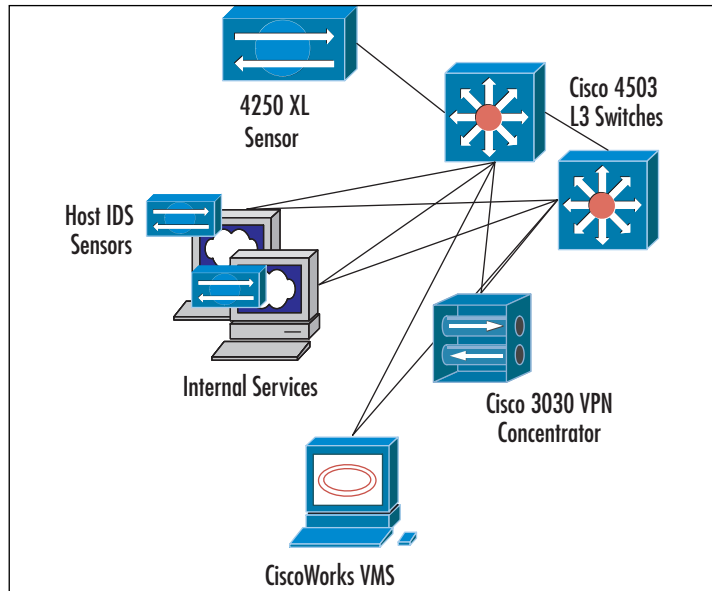
You also realize that, internally, ACME uses OSPF and routes down to the Distribution Cisco 4503 switches in the Building Distribution and Edge Modules. It's important to note that OSPF and BGP are providing active/active network connectivity where possible since this can disrupt an IDS, as we've previously discussed. Use the remote offices route into the Campus Core for connectivity.

The next step is to determine the critical services and application layer flows across the network. From Figure 2.4, it's apparent that the E-Commerce and VPN/RAS Module contains Internet accessible services. A lot of critical services, such as DNS, E-mail, and E-commerce web sites are located in this module and, therefore, require extra security. VPN and remote access services are provided in this module as well. There's also an internal server farm in the Server and Management Module. Since many of the network management systems (NMS), databases, and other critical applications reside here, it's important to protect this area as well. Finally, you've made note of the wireless access that ACME has recently installed in each building. To ensure security in the wireless deployment, they provide force clients to authenticate and tunnel wireless connections to the VPN concentrator in the Server and Management Module.

So, now that you've gained a good appreciation for the network and critical services at the ACME Company, it's time to determine where the best locations are for an IDS. In your discussions with ACME managers, you've determined that budget, while not infinite, probably won't be a limiting factor in your design. Based on the SAFE architecture, you choose to focus on network areas other than the distribution and edge networks.

Let's have a look at your IDS implementation by focusing on each area in which you've selected to place IDS. The Server and Management Module is shown in Figure 2.5.

The Service and Management Module is an essential part of the network to protect. Therefore, you've decided to install the Cisco Host IDS sensor on the critical servers. You'll also need to inspect the traffic coming and going from the SAFE module. Don't forget that it includes VPN traffic from all of your wireless clients in the access layer of the network. Because this part of the network has high-bandwidth requirements, you select the Cisco 4250 XL Sensor, which provides gigabit performance, to inspect traffic. Finally, this is the network from which you'll be managing the entire Cisco-based IDS infrastructure. Because you're working in an enterprise-sized network with multiple IDS sensors, you select CiscoWorks VMS, which will provide management capabilities for all your IDSs. For each IDS deployment, you'll configure the Control and Reporting IDS

Figure 2.5 Server and Management Module IDS

sensor interface in a private VLAN that communicates securely back to the VMS server.

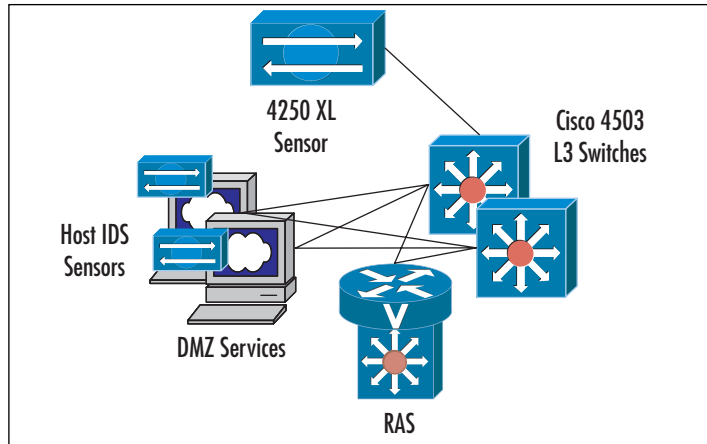
NOTE

As previously discussed, the routed environment in the ACME Company provides for active/active network flow across redundant platforms. To accommodate this design, IDSs need special provisions at the switch so that they may inspect traffic flowing across either of the ingress/egress paths. This could be accomplished via trunks configured between the switch devices over which RSPAN data is shared.

Like the Services and Management Module, the E-Commerce and VPN/RAS Module contains critical servers and services that require extra security protection. It's also a high-speed network environment, with gigabit attached servers and switching devices. This type of computing environment requires a similar solution to that in the Services and Management Module. You load servers with the Cisco Host IDS software and install another Cisco 4250XL Sensor connected to the Cisco 4503 switches. This way, you'll be able to inspect traffic at

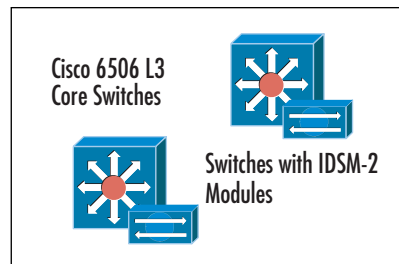
speeds of up to 1 Gbps and you'll have host-based inspection and protection for your servers. The E-Commerce and VPN/RAS Module is shown in Figure 2.6.

Figure 2.6 E-Commerce and VPN/RAS Module IDS



So far, you've done a good job of protecting the services in the organization. But what about the security of the users and general network infrastructure? As we discussed earlier, the SAFE architecture doesn't include IDS at the distribution and edge networks. So where is a good location to inspect user traffic? Since the ACME Company uses the Cisco 6506 switch platform in the core, you can most likely deploy the Cisco IDSM-2 Module in the 6506 chassis. This decision will depend on the interface speeds and utilization of the Core switches. If you're using less than 1 Gbps, the IDSM-2 Module will work well. Again, the active/active network design in the core is something you'll need to consider. Like the other modules we've discussed so far, you'll need something like RSPAN to trade traffic between the core switches. This will ensure your IDS can inspect entire network flows, regardless of which network device they traverse. The Core Module is shown in Figure 2.7.

Figure 2.7 Core Module IDS

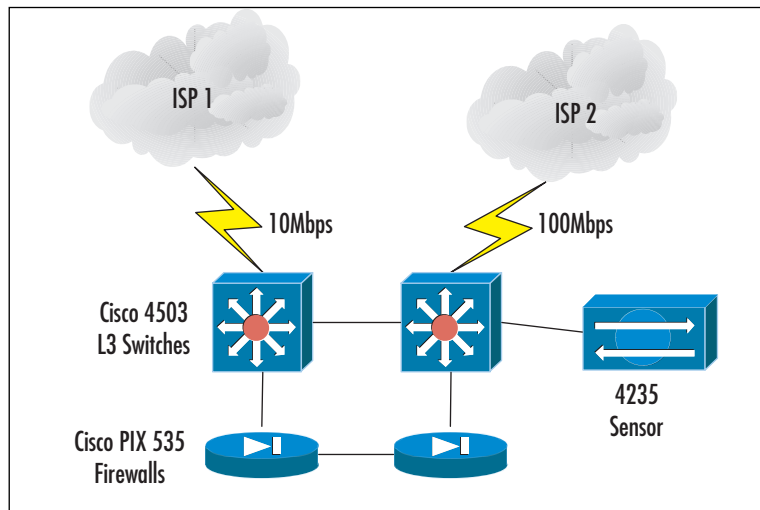


Now the ACME Enterprise appears to be secure on the inside. Don't forget about the front door! Of course, you've considered that as well. The ACME Company currently has two internet connections to two different ISPs for redundancy. They're fortunate to have Ethernet handoffs to their providers and use BGP attributes to distribute network traffic accordingly across the 10- and 100-Mbps connections. Since the redundant PIX firewalls are operating in an active/passive mode, all traffic will traverse the active firewall under normal circumstances. Your IDS, therefore, can be implemented above the active PIX, but will become useless if the PIX firewalls fail-over. Again, you could use the RSPAN solution discussed previously.

The ISP connections are high speed, but not so fast as the internal networks. Based on this information, choose the Cisco 4235 IDS Sensor since it will perform at speeds up to 250 Mbps and will easily support the maximum combined connection speed of 110 Mbps. You position these sensors above the firewalls (and possibly above the routed interface on the Cisco 4503 switches) to inspect all traffic to and from the ACME Company.

The Corporate Internet Module is depicted in Figure 2.8.

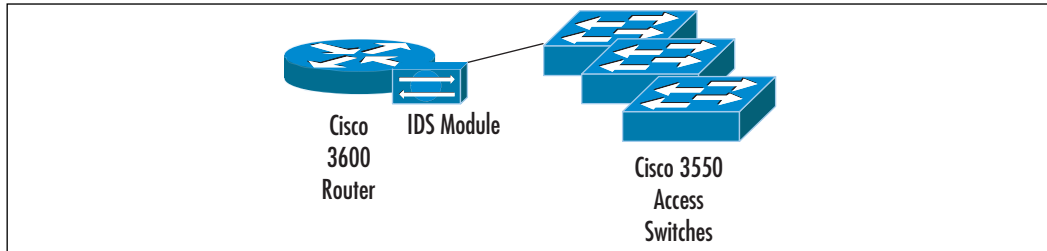
Figure 2.8 Corporate Internet Module IDS



Finally, you realize that one remaining ingress/egress point exists in the ACME network, the frame-relay links to the seven remote office locations. Each office is configured very similarly, so you can design the remote office solution once and replicate it to each site. Luckily, the remote sites already each have a

Cisco 3640 router that provides WAN connectivity back to the ACME core network Cisco 6506 switches. It makes sense then to implement the Cisco IDS Module for the 3600 series router. With this module, you'll be able to provide IDS services in each remote location without requiring additional rack space, cabling, or power, since the module inserts directly into the 3640 chassis. The Remote Site IDS solution is depicted in Figure 2.9.

Figure 2.9 Remote Site IDS



At this point, the ACME Company appears to have a fairly comprehensive IDS architecture. What other locations in the network would be good candidates for IDS? What processes should you use to tune the IDS infrastructure? What other security devices could be deployed to increase the security of the ACME network? Finally, how do we actually manage all of these security devices? These are all excellent questions and bring us back to the Cisco Security Wheel concepts we discussed in Chapter 1. As a security professional, you need to consider how policy, monitoring, response, testing, and management all tie into your IDS deployment.

Summary

Building upon Chapter 1, we've covered Cisco's vision and implementation of comprehensive intrusion detection. After reading Chapter 2, you should be familiar with Cisco's conceptual approach to IDSs, which includes precise threat detection, effective mitigation techniques, efficient management, and adaptable deployment capabilities. You should also be quite familiar with Cisco's Network IDS product line, which includes the Cisco 4200 Series appliance IDS sensors, and the switch and router modules for the Cisco Catalyst 6500 switch and 2600, 3600, and 3700 routers. While all the devices run the same standard and powerful software, each sensor has different capabilities and performance characteristics lending to Cisco's flexible deployment strategy. Be certain you understand each platform's specific capabilities.

We covered Cisco's Host IDS offering as well, which includes the Entercept-powered Cisco Host IDS Sensor and offers both Standard Agent and Web Edition Agent protection. These software components add an extra layer of protection to the service endpoints in the network, such as e-mail servers and web servers. They also allow the security administrator to gain insight into encrypted traffic flows destined to the servers as they inspect traffic after the service requests are decrypted.

All of these IDSs are manageable via various Cisco-based software solutions that offer a range of scalability for everything from small office networks to global enterprise environments. These include Cisco IVE and IDM, which are part of Cisco IDS 4.0 software, and CSPM and CiscoWorks VMS, which are additional and optional tools for managing IDSs in large networks. All of these tools facilitate simplified, secure, and holistic management of a Cisco IDS. We looked in depth at the protocols Cisco developed to communicate with IDS sensors deployed in networks, including the PostOffice Protocol and Remote Data Exchange Protocol. It is critical that you understand the PostOffice Protocol addressing scheme and the types of messaging that the protocol provides.

Finally, we discussed some of the underlying principals of deploying Cisco IDS products to effectively and securely protect networks and services. Beginning with a complete, detailed knowledge of the network and services environment, security administrators should map critical infrastructure and servers in order to select the most optimal location of intrusion detection devices. This can be a time-consuming process since you need to understand many network attributes that span the entire OSI model. Most importantly, you'll need to identify the network routing design and the way in which application traffic traverses the entire network infrastructure.

To illustrate this process, we examined two IDS deployments, one small and one large. In both, we saw how different Cisco IDS platforms can be used in appropriate network locations such as at Internet connections, and intranet and extranet networks. We also discussed some of the challenges encryption can present to IDS sensors and emphasized the need for IDS at ingress/egress points such as VPN, RAS, and wireless network demarcation points.

Now that we've looked at the big picture, let's focus on the configuration of Cisco IDS devices starting in Chapter 3 with Initializing Sensor Appliances.

Solutions Fast Track

What Is Cisco Intrusion Detection?

- ☑ Cisco Intrusion Detection is a holistic approach to security based on accurate threat detection, intelligent threat investigation and mitigation, ease of management, and flexible deployment options.
- ☑ Cisco delivers each of these concepts through flexible Network IDS hardware and Host IDS software, well-crafted Cisco IDS software, and powerful, scalable Cisco IDS management software.
- ☑ Cisco's Intrusion Detection approach is backed by the power of Cisco Support and by the Cisco Countermeasures Research Team (C-CRT) for up-to-date network defense and expertise.

Cisco's Network Sensor Platforms

- ☑ Cisco offers a wide range of IDS performance capability starting at 45 Mbps with the Cisco 4210 IDS Sensor and ending at 1 Gbps with the Cisco 4250 XL Sensor.
- ☑ Organization can leverage existing infrastructure by deploying IDS Modules in Catalyst 6500 switches and in 2600, 3600, and 3700 routers.
- ☑ All of Cisco Network Sensors run Cisco IDS 4.0 software, providing a holistic and easily managed IDS infrastructure.

Cisco's Host Sensor Platforms

- ☑ Cisco provides Host IDS sensors for Sun Solaris and Microsoft Operating Systems that incorporate signature detection and behavior anomaly analysis functionality.
- ☑ The Host Sensor is available in two forms: the Standard Agent and the Web Edition Agent.
- ☑ The Cisco Web Edition Agent host sensor provides additional, web-server focused protection for Apache, iPlanet, and Microsoft web server software.

Managing Cisco IDS Sensors

- ☑ Cisco Network IDS sensors can be managed via CLI, IVE, and IDM. These are all provided as part of the Cisco IDS 4.0 software.
- ☑ Larger, enterprise environments can use CiscoWorks VMS instead of IVE and IDM to provide more centralized, scalable management capabilities.
- ☑ The Cisco Host IDS Sensors can be managed by the Cisco IDS Host Sensor Console software or by the CiscoWorks VMS.

Deploying Cisco IDS Sensors

- ☑ Before deploying IDS sensors, security administrators should have a well-developed security policy and comprehensive understanding of the network and services infrastructure.
- ☑ IDS sensors are typically deployed near critical services and network infrastructures such as server farms, ingress/egress points, and alternative access network locations.
- ☑ Because IDS sensor performance is capable of gigabit speeds, it may be advisable to place IDS in the core of some networks.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Where can I get the latest signature updates for my IDS sensors?

A: Cisco IDS Signatures are produced regularly by the Cisco Countermeasures Research Team (C-CRT) and are available for download from www.cisco.com/kobayashi/sw-center/ciscosecure/ids/crypto/.

Q: What’s the difference between Cisco’s PostOffice Protocol and RDEP?

A: Both are proprietary and secure mechanisms Cisco uses to control IDS sensors. The PostOffice Protocol, which is a pull mechanism, is currently being replaced by RDEP, a more efficient push messaging protocol.

Q: How many IDS modules can I deploy in the Catalyst 6500 and Cisco routing platforms?

A: The Catalyst 6500 supports multiple IDSM blades, limited only by the number of free slots in the chassis. The Cisco 2600, 3600, and 3700 routers only support one IDS module at this time.

Q: Is there one software tool I can use to manage all of my IDS devices, including Host IDS?

A: Yes. CiscoWorks VMS is an enterprise tool that can be used to manage not only your IDS infrastructure, but also nearly all other Cisco-based security products.

Q: What happens if a Cisco IDS sensor becomes oversubscribed by high utilization of a network?

A: If the performance of an IDS sensor cannot keep up with the throughput of the network which it monitors, the IDS will issue an alert to administrators indicating that it is oversubscribed.

Q: If I deploy a lot of IDS sensors in my network, will I overload the network with alert messages and other IDS report traffic?

A: The Cisco IDS sensors only issue an alarm when they detect a potentially malicious situation. Even then, they do not replicate the offending traffic back to the IDS console. They simply report the event in an efficient and quickly transferred UDP flow. It is extremely unlikely that an IDS deployment will cause negative or noticeable effects on the production network.

Q: I read that Cisco IDS can inspect encrypted traffic. Is this true?

A: Not exactly. Cisco Network IDS sensors cannot decrypt and inspect traffic as it traverses their monitoring interfaces. You should certainly keep this in mind as you design your IDS deployment. However, by deploying the Cisco Host IDS sensor on your critical servers, you can inspect the encrypted traffic after it is decrypted on the service endpoint. This will provide insight into the encrypted stream.

Initializing Sensor Appliances

Solutions in this Chapter:

- Identifying the Sensor
- Initializing the Sensor
- Using the Sensor Command-Line Interface
- Configuring the SPAN Interface
- Recovering the Sensor's Password
- Reinitializing the Sensor
- Upgrading a Sensor from 3.1 to 4.0

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Sensor initialization is the first step in deploying a Cisco IDS sensor. Cisco also refers to this as bootstrapping the sensor. Once you have decided where the sensor will be placed on your network (in front of, or behind, your firewall) and racked the sensor, you will need to perform a few configuration steps. In this chapter, we show you how to identify the sensors and the interfaces on each. You will also learn different methods of connecting to the sensor during the initialization process.

There are only a handful of commands that can be used with the *command-line interface* but they do everything you need to get the system up and online. We will explain each of those commands and what they are used for. One of the most important commands we'll explore is *sysconfig-sensor*. It provides you with a menu that allows you to configure the sensor name, IP address, network mask, default route, some basic access controls, and the communications infrastructure of the sensor. You will also learn about the two user accounts on the Cisco IDS—root and netranger—what they are, and why you want to log in as one or the other.

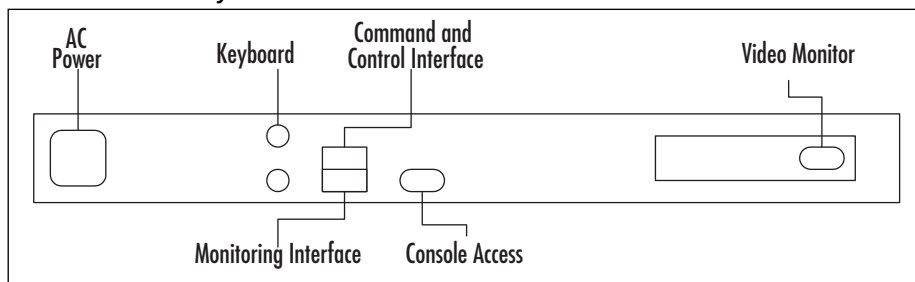
Once the initialization process is complete and you have become familiar with the accounts and commands they perform, we will take you through the process of how to recover the sensor using a recovery partition CD-ROM. You may even need to know how to recover passwords. Thus, we'll discuss how to get past the dreaded unknown password of a used sensor. These two processes are important to know, considering frequent personnel turnover and how often hardware changes hands. As a result, information like passwords is not always passed on to the next responsible party. You may also experience a situation that requires you to change the interface configuration between the control port and the snooping port. We'll explain why and walk you through the process.

Identifying the Sensor

Technically speaking, there are two types of sensor platforms available: the 4200 series sensors and the Catalyst 6000/6500 series IDS Module (or IDSM), both of which we cover in detail in Chapter 6. Within the 4200 series, there are four different sensor appliances offered in the Cisco product line. Depending on your budget, organizational needs, and the number of external connections to the Internet, multiple sensors or a single sensor could be the answer. It is important to be able to identify which sensor you will be working with considering there are

some subtle differences between the models. The old Netranger sensors, 4220 and 4230, were bulky 7-inch, four-rack-unit (RU) models. The introduction of the newer blade-style models streamlined the chassis into a 1U format for all models, including 4210 (as shown in Figure 3.1), 4215, 4235, 4250, and 4250-XL. For the purpose of this chapter, we will focus on the model 4230 since it is one of the most commonly available and is still used on the Cisco IDS certification test..

Figure 3.1 4210 Layout of Back Panel



Each of the sensors has two ports: a monitoring or sniffing interface which captures the traffic to analyze, and a control port that provides access to the sensor via Telnet, CSPM, and so on. The control port is the only port on the sensor that will actually be assigned an IP address on the network. Some modules have a console port that can be a DB9 connector, such as the 4230, or an RJ45 console cable jack.

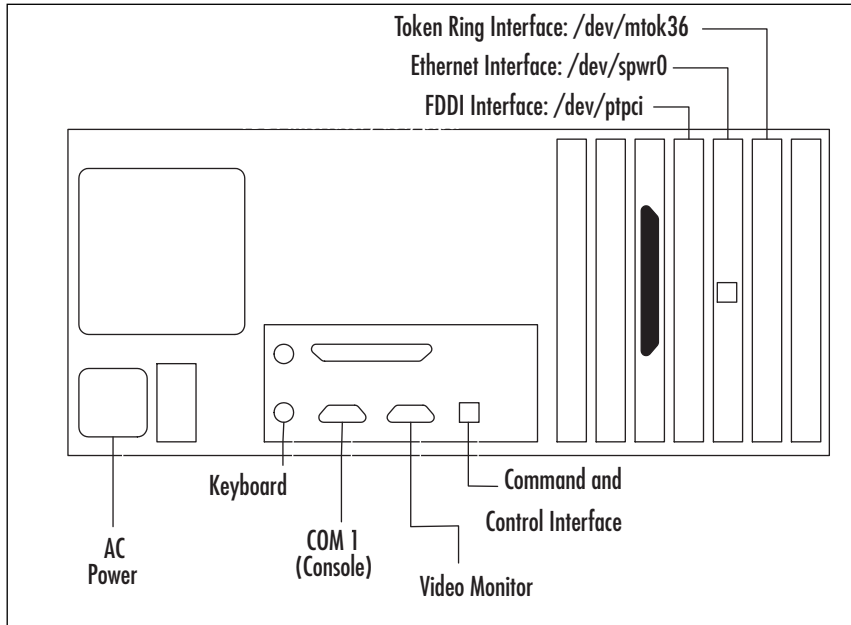
NOTE

Cisco best practices tell you the control port should be placed on an isolated network or out-of-band management network that routes traffic for management purposes on another network other than the enterprise. Cisco documentation refers to this type of network as the *Command and Control Network*.

It is critical that we can identify the monitoring or sniffing port on the IDS. On the 4210, the device name is `/dev/iprb0`. The 4210 sensor has two built-in ports directly on top of one another. The monitoring interface is the lower port, `iprb0`. The control port is `iprb1`, which is located above the sniffing port (refer to Figure 3.2). The 4220 and 4230 sensors have expandable slots. One of the ports is built-in, and the other is located on the expansion slot. That is, `iprb0` can be

found on the sensor, while, `/dev/spwr0` is physically located in slot 5 in order to capture packets.

Figure 3.2 4230 Layout of Back Panel



The 4230 and 4220 sensors have the ability to be configured in different manners to accommodate different networks. `iprb0` is used for control in each configuration. For a token ring network, use `/dev/mtok36`, which is located in slot 6 to capture packets. For a FDDI network, `/dev/ptpci`, is used. It's located in slot 4.

NOTE

The sniffing port and control port on the 4230 can be swapped under certain circumstances to sniff multicast traffic. We will discuss that process later in the chapter.

Initializing the Sensor

Initializing the sensor is where the rubber meets the road, so to speak. Besides physically installing the sensor into a rack and cabling, this is the basic process for getting your sensor up and running. Two accounts are created by default when the sensor software is installed: *root* and *netrangr*. These should be the only accounts needed to log in to the sensor and perform administrative tasks. In fact, certain commands can only be performed as *root* or *netrangr*, so it's best to become familiar with them.

Listed next are several ways to gain management access of the sensor:

- **Console Port** This requires a RS232 cable and a program such as Hyperterm or Teraterm.
- **Monitor and keyboard** This requires connecting a monitor and keyboard directly to the sensor.
- **Telnet** This requires an IP address to be configured to the command and control interface.
- **Secure Shell (SSH)** This also requires an IP address and an SSH client application.
- **Cisco IDS Device Manager (IDM)** This requires an IP address and uses a Web browser.
- **Cisco Secure Policy Manger (CSPM)** An IP address is required along with a mail server for the PostOffice protocol to communicate with.
- **Cisco IDS Director for Unix** The Director requires HP OpenView and runs on an HP or Solaris platform.

The easiest way to initialize the sensor for the first time is either through a directly connected keyboard and monitor, or by using the COM port connected to a workstation via a null-modem cable.

Configuring & Implementing.....

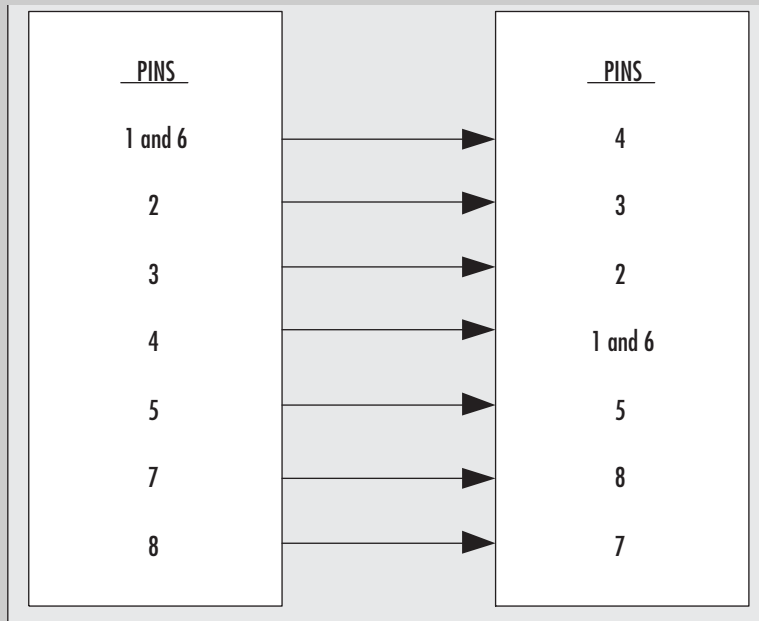
How to Use the Console Port and Why to Use It?

There are several ways to connect to the IDS sensor: Telnet to the sensor, connect a keyboard and monitor directly to the sensor, or connect a workstation to the sensor utilizing the COM ports. Some functions—such as Updating signatures, troubleshooting, and even running the *sysconfig-sensor* utility—cannot be performed via a management application and thus require you to log in directly..

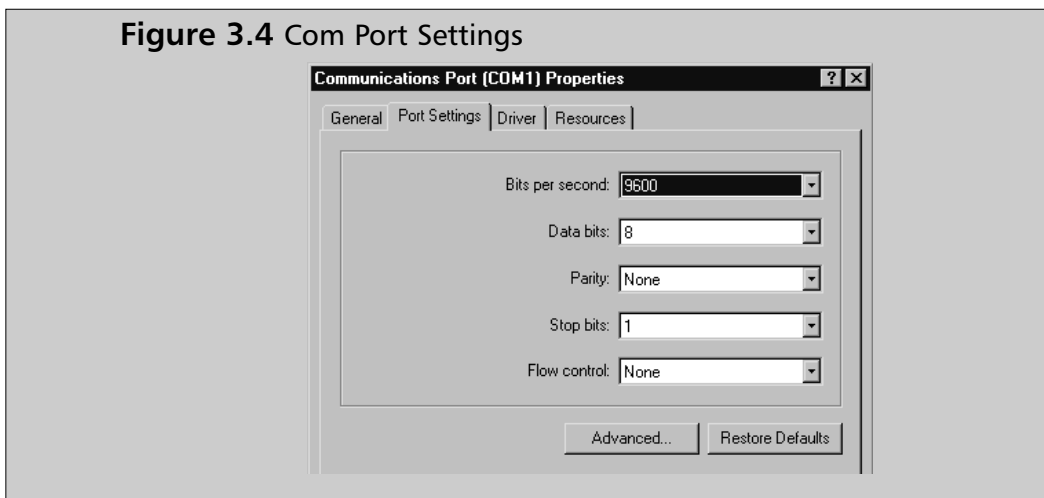
In order to connect, you will need a null-modem cable with DB-9 female connectors on both ends. If you have a sensor right out of the box, a cable should accompany the sensor. In the event it is not provided, you can either purchase one or make it yourself. (The pinout for this cable is shown in Figure 3.3.) Connect the cable to the COM port of the IDS sensor and to the workstation using the COM 1 port. The COM port settings on the workstation should look like those in Figure 3.4.

Using your communications software, attempt to call the sensor. If you are successful, log in as **netrangr** and **su** to root if needed.

Figure 3.3 Null Modem Pinout Chart



Continued



What Is the root User?

The user *root* on the sensor is used strictly for configuration of the operating system. It is not used for daily administrative tasks. The main function *root* is used for *sysconfig-sensor*, explained in detail later in this chapter. The *root* account is used for the following tasks:

- Bootstrapping the sensor by executing the *sysconfig-sensor* command
- For certain system level commands, such as *snoop*
- When installing signature updates or service packs
- When starting or stopping the IDM web services

The *snoop* command is a handy one to remember since you can use it to verify that the sensor can see the traffic you are interested in. *snoop* captures packets from the network and displays their contents to the screen. It can be saved to a file if needed. If *snoop* cannot see the traffic, neither can the IDS sensor monitor interface. *snoop* examines the raw traffic on your network and can be executed to look at any interface.

For example:

```
# snoop -d spwr0 port 45000
```

Using device `/dev/spwr` (promiscuous mode)


```
10.0.0.8 -> 10.0.0.4 UDP D=45000 S=45000 LEN=52
```

```
10.0.0.8 -> 10.0.0.4 UDP D=45000 S=45000 LEN=52
```

```
10.0.0.8 -> 10.0.0.4 UDP D=45000 S=45000 LEN=52
```

```
10.0.0.8 -> 10.0.0.4 UDP D=45000 S=45000 LEN=52
```

The preceding output is an example of the sensor 10.0.0.8 sending packets on UPD 45000, but no packets are received. If the two devices were communicating properly, the *snoop* output would look like the following:

```
# snoop -d spwr0 port 45000
```

```
Using device /dev/iprb (promiscuous mode)
```

```
10.0.0.4 -> sensor1          UDP D=45000 S=45000 LEN=56
```

```
sensor1 -> 10.0.0.4 UDP D=45000 S=45000 LEN=56
```

```
172.18.124.142 -> sensor1      UDP D=45000 S=45000 LEN=56
```

```
sensor1 -> 172.18.124.194 UDP D=45000 S=45000 LEN=56
```

Notice traffic is flowing on UDP 45000 in both directions.

NOTE

If both sides are sending and receiving UDP 45000 packets and the output of the command *idsconnns* says that a connection has not been established, go back and troubleshoot the *postoffice* parameters on the sensor and the management device.

The architecture is set up in a way that certain commands work specifically with root but not for user *netrangr*. root is used to initialize the sensor and make configuration changes as needed down the road. Even when telneting into the sensor, *netrangr* is used and the user must *su* to root in order to perform root tasks for configuration, or to modify a setting or permission in the Unix architecture.

What Is the netrangr User?

To perform administrative of IDS-level functions on the sensor, you will need to log in as **netrangr**. All the commands discussed later in this chapter are executed using this account, with the exception of *sysconfig-sensor*. They include *idsstatus*, *idsvers*, *idsstop*, *idsstart*, and *cidServer*.

NOTE

An important item to remember with all the Cisco IDS operating systems is that the user IDs of *root* and *netrangr* both have a default password of *attack*. The first login into the system forces you to change the password.

What Is *sysconfig-sensor*?

Once you have logged into the sensor as *root* and changed the password, *sysconfig-sensor* is the next command performed in order to configure the sensor. This is commonly known as *bootstrapping* the sensor.

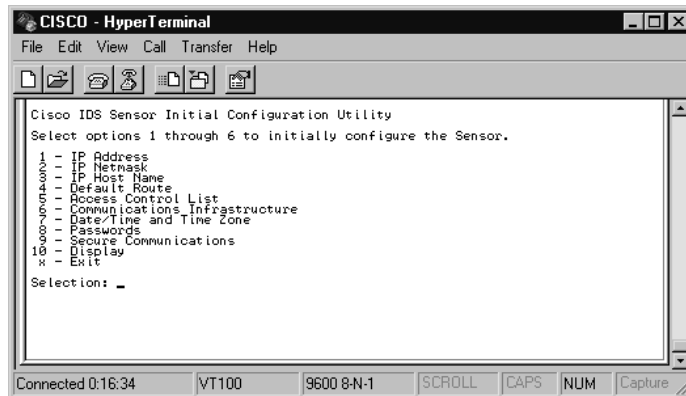
NOTE

Passwords are case-sensitive and can be up to eight characters in length. Any password longer than eight characters is truncated.

Configuring the Sensor

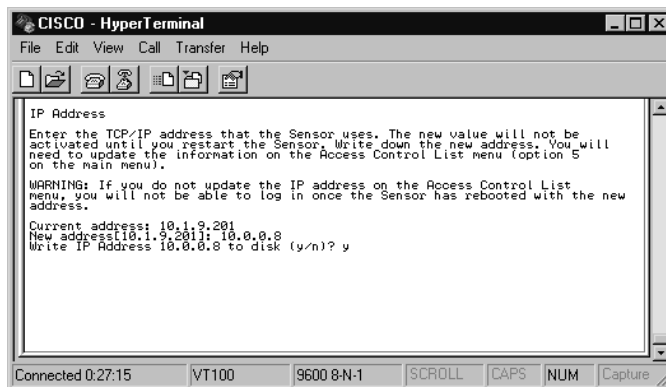
Configuring the sensor is a fundamental step in deploying an IDS infrastructure. The first step in configuring the sensor is running the *sysconfig-sensor* command and going through each option, filling in the required information along the way. Any of the options that pertain to addressing—options 1–5—will require a reboot if modified.

1. Execute the command *sysconfig-sensor*. The configuration utility menu is shown in Figure 3.5

Figure 3.5 *sysconfig-sensor*

2. Select option 1, **IP Address**. Figure 3.6 shows the screen for entering the sensor's IP address. The sensor comes out of the box with a default IP address of 10.1.9.201. Change this address to reflect your network. Remember the address! You will be prompted to write the information. Check your entry and select **yes** or **no**.

Figure 3.6 The IP Address

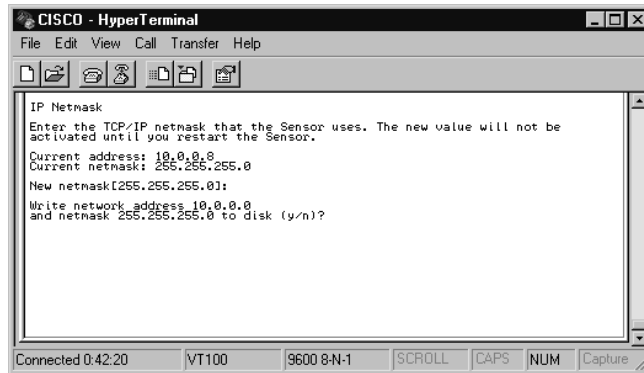
**NOTE**

Options 1–6 must be configured for the sensor to communicate properly.

3. Select option 2, **IP Netmask**. For option 2 of the *sysconfig-sensor* menu, you must enter the subnet mask of your network, as shown in

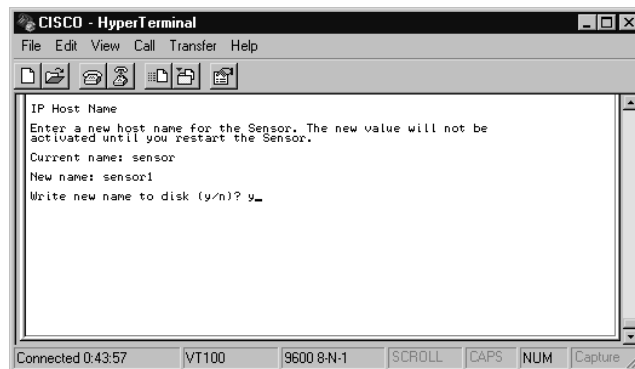
Figure 3.7. The subnet mask defines boundaries that can vary depending on the extent of the subnetting being implemented. If it is entered incorrectly, the sensor may not communicate properly with the management host or the rest of the network. Check with the network engineers in your organization.

Figure 3.7 The IP Netmask



4. Select option 3, **IP Host Name**. The default host name for the sensor is *sensor*, shown in Figure 3.8. Add a unique name for your sensor here. It would be wise to enter a name that can easily be identified on the network. The example shows *sensor1*. If you add other sensors, increment the number to reflect the number of sensors on the network.

Figure 3.8 The IP Host Name

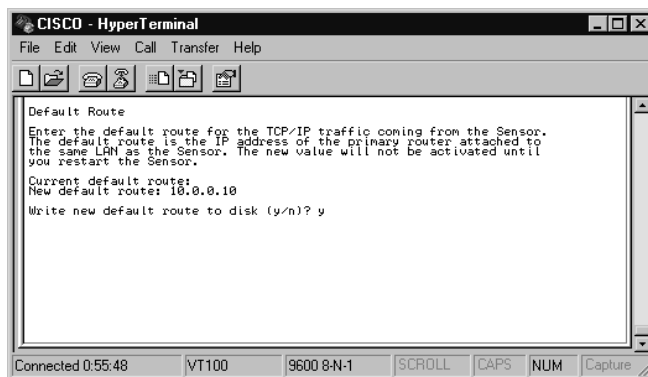


NOTE

The IP Host Name of the sensor can be up to 256 alphanumeric characters in length with no spaces. "-" and "_" are valid special characters, and case is important.

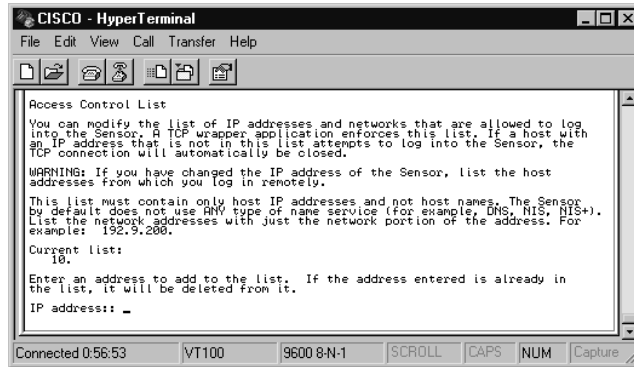
5. Select option 4, **Default Route**. The default route tells the sensor what path to take to reach other hosts in the IDS infrastructure. This setting is usually a router or a firewall interface. Enter the default route for your network, as shown in Figure 3.9. Again, check with your network engineers to verify your route.

Figure 3.9 The Default Route



6. Select option 5, **Access Control List**. The Access Control List (ACL) is imperative if you are not able to physically access the sensor. Figure 3.10 shows a default access list of the entire 10. network. Enter a network address or the individual IP addresses of hosts that should have access to the sensor. The ACL works via a standard TCP wrapper application. The TCP connection is automatically closed if a host attempts to log in to the sensor without the host's IP in the ACL.

Figure 3.10 The Access Control List



NOTE

Cisco's best practices tell us that we should be as specific as possible and only enter the IP addresses that will be able to connect to the sensor.

7. Select option 6, **Communications Infrastructure**. The configurations in option 6 (shown in Figure 3.11) are critical for proper communication between the sensor and IDS Manager. Make sure to verify each setting and document them. Table 3.1 shows the values for each field. Several of the settings have already been configured in previous steps. Those settings are in brackets and can be kept by pressing **Enter** during that specific configuration step.

Figure 3.11 Communications Infrastructure

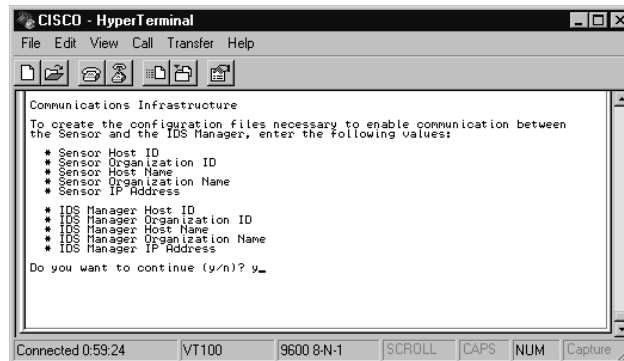


Table 3.1 Communications Infrastructure Values

Field	Input
Sensor Host ID	1–65535
Sensor Organization ID	1–65535
Sensor Host Name	256 alphanumeric characters; no spaces; "-" and "_" are okay
Sensor Organization Name	256 alphanumeric characters; no spaces; "-" and "_" are okay
Sensor IP Address	Valid IP address
IDS Manager Host ID	1–65535
IDS Manager Organization ID	1–65535
IDS Manager Host Name	256 alphanumeric characters; no spaces; "-" and "_" are okay
IDS Manager Organization Name	256 alphanumeric characters; no spaces; "-" and "_" are okay
IDS Manager IP Address	Valid IP address

The sensor host ID and the IDS manager host ID must be unique, as well as all subsequent sensors and devices added to the IDS infrastructure. This number can be any number between 1 and 65535. The organization ID should be the same for all devices in the infrastructure. This organization ID is used to group sensors and management devices together and can be between 1 and 65535. The organization name should also be the same for all devices. This is typically the location where you work, or where it is installed. Once all settings have been made, the sensor will prompt you to create the configuration file.

Cisco says you should use only lowercase letters to define organization names. The host and organization name are case-sensitive with regards to how postoffice processes audit events on the local host. Host and organization names are not passed between different postoffice clients, only the host and organization IDs. The `/usr/nr/etc/hosts` file is where this information is listed for the Cisco IDS infrastructure. The syntax is as follows:

```
[host ID]. [organization ID] [host name].[organization name]
```

```
8.100 localhost
```

```
8.100 sensor1.security
4.100 ids-mgr.security
```

The preceding sample is what the hosts file entries look like on the sensor. Notice there are two entries for the sensor itself: localhost and sensor1.security.

- You now need to write the configuration. Verify all of the settings on the screen (shown in Figure 3.12) and type **y** to accept the settings. If any of the settings are incorrect, type **n** to discard the settings and repeat the configuration steps. A message is displayed once all the configuration files have been written to successfully. Once the configuration files have been written, you should get the message, shown in Figure 3.13, telling you that all configuration files were written successfully.

Figure 3.12 *sysconfig-sensor* Settings

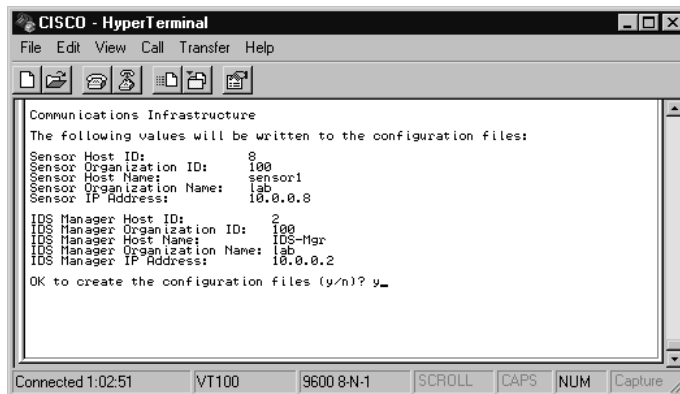


Figure 3.13 The *sysconfig-sensor* Success Message



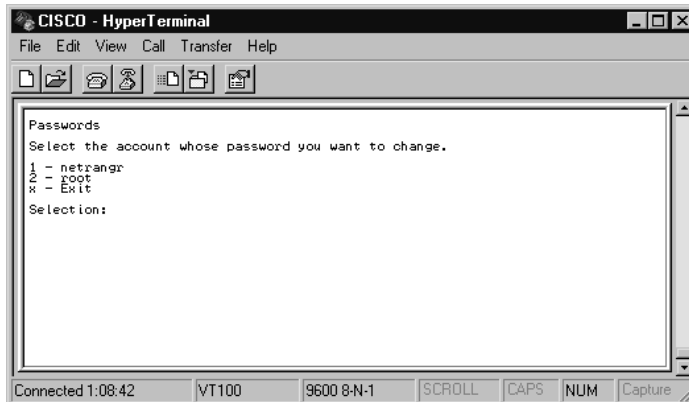
9. You now need to set the date, time, and time zone for your organization. Use the Date/Time and Time Zone section, shown in Figure 3.14, to synchronize the sensor with the rest of the network. You have the option of entering a specific date, time, and time zone, or entering a host to synchronize with, such as a time server on the network.

Figure 3.14 Date/Time and Time Zone



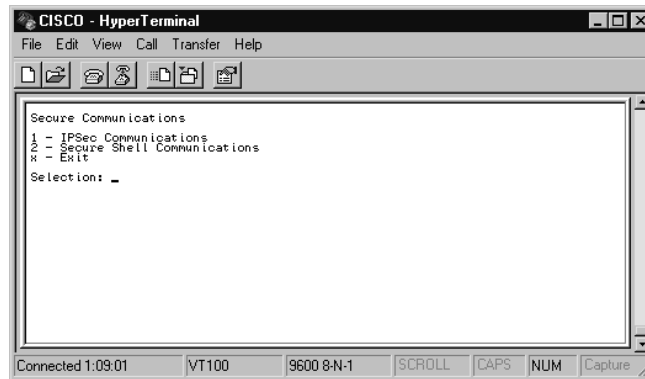
10. You can now change/set your passwords as needed. Use the Passwords option, shown in Figure 3.15, to change the root or netrangr passwords. Once the password has been changed the old password is not saved on the system anywhere. It is important to remember the new passwords.

Figure 3.15 The Passwords



11. You now need to configure your Secure Communications. The secure communications menu shown in Figure 3.16 is used to define configuration settings for encrypted communications between the sensor and the IDS Management device. Before we get into configuring the secure communication, it is important to remember IPsec is used mainly for connection to CSPM or the Director, and not the IDS Event Viewer, which is used with the IDS Manager.

Figure 3.16 Secure Communications



NOTE

IPsec is supported on sensors running version 2.5 or later, Unix Directors running version 2.2.2 or later, and version 2.3.i or later for CSPM. SSH is supported on sensors running version 3.0.

Using SSH provides a command and control (C & C) interface that allows you to administer the Sensor remotely without exposing plain-text usernames and passwords to the network connecting you to the Sensor.

To configure secure communications, you have two options:

- IPsec Communications
- Secure Shell Communications

Option 1: IPsec Communications

To use IPsec communications, follow these steps:

1. Select suboption **1** in Secure Communication, IPSec Communications. You will see the IPSec Communications window.
2. Select option 1, **Session Keys**. Here you have the option of accepting the default keys or creating custom keys. For default keys, proceed with step 3. For custom keys, skip to step 5.
3. Select option **1** to access the **Set Up Default Keys** screen.
4. Enter the inbound and outbound Security Port Index (SPI) values. (Refer to Table 3.2 for SPI values.) Once this is done, proceed to step 7.
5. Select option **2** to access the **Set Up Custom Keys** screen.
6. Enter the key values. (Refer to Table 3.2 for key value options.)
7. Exit back to the IPSec Communications screen.

Table 3.2 IPSec Communications Field Values

Key	Value
Cipher	8-byte hexadecimal string
Authentication	16-byte hexadecimal string
SPI	Value 0x100 - 0xffff ffff (numeric)

The other option in the IPSec Communications menu is to enable NAT. Only use this if NAT is set up between the management device and the sensor. If NAT is enabled, the result is encryption of IDS traffic only. TCP traffic is unencrypted. For added security, use SSH communications when NAT is enabled.

Some of the benefits from using IPSec are as follows:

- Secure communications between the management device and the sensor
- Traffic is encrypted between the nodes
- Authentication, confidentiality, and nonrepudiation protection for IDS communications

Option 2: Secure Shell Communications

To use Secure Shell Communications, follow these steps:

1. Select option **2** on the IPSec Communications menu to access the Secure Shell Communications screen. This screen allows you to select up to three levels of security.
2. Select options **1-High** (Telnet and FTP disabled), **2-Medium** (Telnet disabled), or **3-Low** (insecure services available).
3. Exit to return to the Secure Shell screen.
4. Select option **2** to access the Manage Secure Shell Known Hosts screen. The SSH client keeps a list of hosts it has connected to. Keys become invalid when keys are moved to different IP address or regenerated. The invalid information needs to be purged for further communication.
5. Choose a user with a known_hosts file.
6. Exit to return to the Secure Shell screen.
7. Select option **3** to access the Host Key Operations screen.
8. Here you have two options, 1-Delete host key and generate a new one, or 2-Delete host key. Make your selection. With key pairs, you need a public and a private key. The encrypted messages can only be decrypted with the proper key. The server generates the host key pair when the server is first started. Regenerate the key if it becomes suspect. If the key is regenerated, all the hosts that have communicated with the server will have to have the old key pair cleared from their cache in order to communicate in the future.
9. Exit and reboot.

NOTE

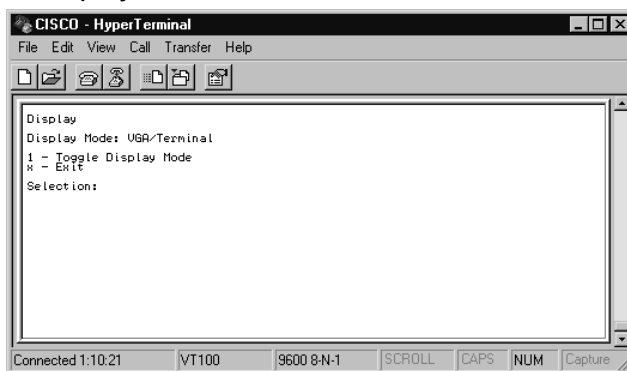
IPSec is resource-intensive on the sensor. Excessive processing due to IPSec can cause intrusion alarms to go undetected and ultimately unreported.

The Display

The display in Figure 3.17 allows you to toggle back and forth between VGA and terminal settings. This is a setting that everybody needs to get familiar with because inevitably it will be forgotten. VGA/Terminal mode allows the VGA and

terminal port to be used to log in to the sensor. Boot messages are suppressed on the terminal port and sent only to the VGA port.

Figure 3.17 The Display



The Terminal mode disables the VGA display and only displays output to a terminal connected to the COM1 port. This sends boot messages to the terminal port COM1. With the Terminal only mode configured, you cannot log in to the sensor by connecting a keyboard and monitor.

Once the *sysconfig-sensor* options have been completed, the system will need to be rebooted. Modifying options 1 through 5 require a reboot. Options 6 through 8 do not require a reboot. One exception is changing the time zone in option 7. If the time zone is changed, a reboot is required.

Using the Sensor Command-Line Interface

When using the command-line interface you need to be aware of all the pertinent commands that are used to initialize the sensor and which ultimately can be used to administer the IDS Sensor. Many of these commands will be used for troubleshooting purposes only. The daemons start automatically when the sensor is started. Many of the commands are used to verify the status of those daemons and/or services.

To execute commands specific to the administration of the sensor, make sure you are logged in as *netrangr*. For all commands, with the exception of *cidServer*, the `/usr/nr/bin` directory has an *nr* command that does the same as the *ids* commands.

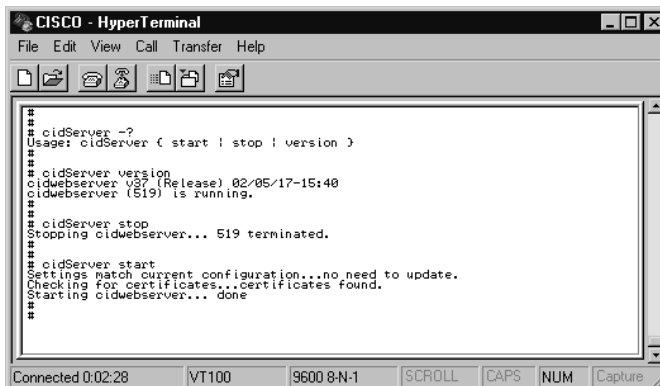
cidServer

cidServer is the IDS Web server itself and enables the administrator to connect via IDM. The server automatically begins during system startup. You must be logged in as root to execute this command. *cidServer* has three parameters that can accompany the command:

- version
- stop
- start

Figure 3.18 demonstrates the output.

Figure 3.18 *cidServer*



```
#####
cidServer -?
Usage: cidServer ( start | stop | version )
#####
cidServer version
cidwebserver v37 (Release) 02/05/17-15:40
cidwebserver (S19) is running.
#####
cidServer stop
Stopping cidwebserver... S19 terminated.
#####
cidServer start
Settings match current configuration...no need to update.
Checking for certificates...certificates found.
Starting cidwebserver... done
#####
```

NOTE

If you are having trouble communicating with the sensor via Telnet, SSH, or IDM, use the command *cidServer version* to check the status of the sensor.

idsstatus

idsstatus is used to check the status of the IDS sensor daemons and services. This can be accomplished with the *unix ps* command, but the *idsstatus* only shows IDS services. Figure 3.19 shows the expected output.

Figure 3.19 *idsstatus*

```

CISCO - HyperTerminal
File Edit View Call Transfer Help
netrangr@sensor1:/usr/nr
>idsstatus
netrangr 407 1 0 12:05:54 ? 0:00 /usr/nr/bin/nr.fileXferd
netrangr 354 1 0 12:05:46 ? 0:00 /usr/nr/bin/nr.postofficed
netrangr 408 1 0 12:05:55 ? 0:00 /usr/nr/bin/nr.sspd
netrangr 406 1 0 12:05:52 ? 0:00 /usr/nr/bin/nr.loggerd
netrangr@sensor1:/usr/nr
^_
Connected 0:04:32 VT100 9600 8-N-1 SCROLL CAPS NUM Capture

```

idsconns

The *idsconns* command assists in troubleshooting connections between the sensor and the management device. If the connection is up, you will see *[established]*, as shown in Figure 3.20. The output shown in Figure 3.21 shows that a SYN packet has been sent *[SynSent]* but also says *syn NOT rcvd!* This means that communication between the management device and the sensor has not been established.

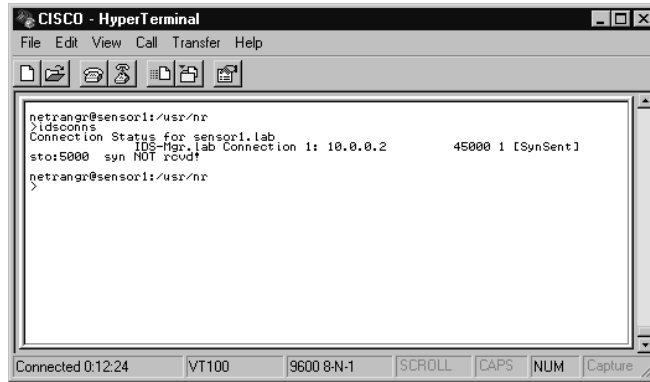
Figure 3.20 *idsconns established*

```

telnet - HyperTerminal
File Edit View Call Transfer Help
netrangr@sensor1:/usr/nr
>idsconns
Connection Status for sensor1.lab
IDS-Mgr. Lab Connection 1: 10.0.0.4 45000 1 [Established]
stor:0002 with Version 1
netrangr@sensor1:/usr/nr
^_
Connected 0:00:13 Auto detect TCP/IP SCROLL CAPS NUM Capture

```

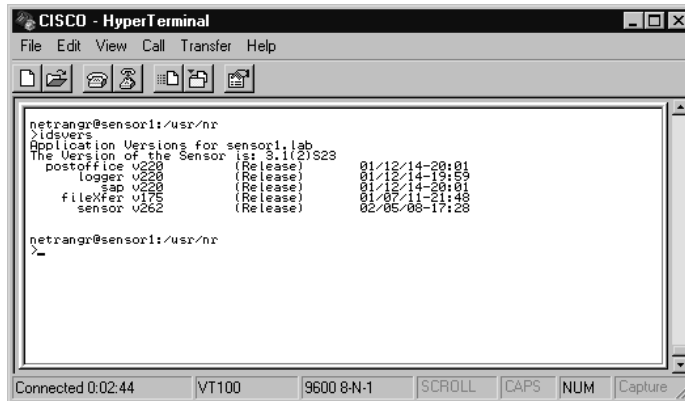
Figure 3.21 *idsconns* syn NOT rcvd!



idsvers

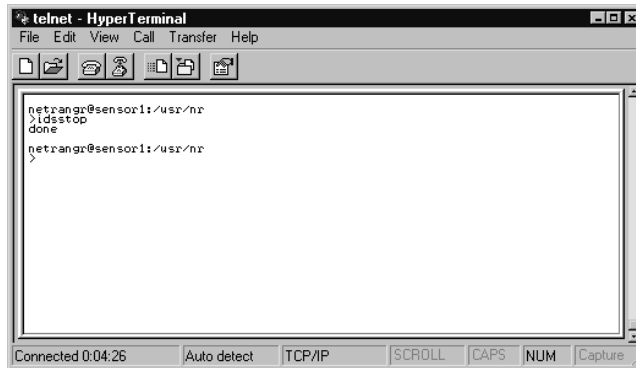
idsvers is used to verify the version of Cisco IDS software that is currently running on the sensor. Figure 3.22 shows the expected output.

Figure 3.22 The command *idsvers* results



idsstop

Another command that you may encounter is *idsstop*. It is pretty self-explanatory. No switches or additional parameters are needed for this command. In most cases, you don't need to use this command because the IDS services are started when the sensor is booted. If you are troubleshooting a problem, however, you may need to use this command. The output on the screen is limited, as shown in Figure 3.23.

Figure 3.23 *idsstop*


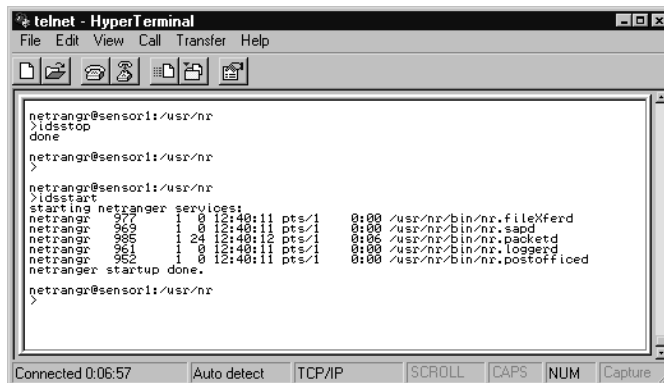
```

telnet - HyperTerminal
File Edit View Call Transfer Help
netrangr@sensor1:/usr/nr
>idsstop
done
netrangr@sensor1:/usr/nr
>
Connected 0:04:26 Auto detect TCP/IP SCROLL CAPS NUM Capture

```

idsstart

Of course, if the services are stopped, you need a method to restart them. Use the command *idsstart* to do so. This command only starts the services listed in */usr/nr/etc/daemons*. See Figure 3.24.

Figure 3.24 *idsstart*


```

telnet - HyperTerminal
File Edit View Call Transfer Help
netrangr@sensor1:/usr/nr
>idsstop
done
netrangr@sensor1:/usr/nr
>
netrangr@sensor1:/usr/nr
>idsstart
starting netranger services:
netrangr 977 1 0 12:40:11 pts/1 0:00 /usr/nr/bin/nr.fileXferd
netrangr 969 1 0 12:40:11 pts/1 0:00 /usr/nr/bin/nr.sapd
netrangr 985 1 24 12:40:12 pts/1 0:06 /usr/nr/bin/nr.packetd
netrangr 961 1 0 12:40:11 pts/1 0:00 /usr/nr/bin/nr.loggerd
netrangr 952 1 0 12:40:11 pts/1 0:00 /usr/nr/bin/nr.postofficed
netranger startup done.
netrangr@sensor1:/usr/nr
>
Connected 0:06:57 Auto detect TCP/IP SCROLL CAPS NUM Capture

```

Configuring the SPAN Interface

If you have worked with switches much, you are already familiar with Switched Port Analyzer (SPAN). SPAN is used to capture network traffic in the shape of packets for the purpose of analysis. It is especially beneficial to the IDS to have all traffic passed across a single port on a switch or an entire VLAN.

Spanning Ports

The following is an example of configuring SPAN on a 2900 series switch.

Configuring & Implementing...

Configuring SPAN

The SPAN interface can be any interface on the switch as long as it's a static-access port. The SPAN port also has to reside in the same VLAN as the ports being monitored. To configure SPAN, follow these steps:

1. Once you are in exec mode, enter global configuration mode by typing **configure terminal**.
2. Enter **interface <interface>** to configure the interface that will act as the monitoring port.
3. Enter **port monitor <interface>** to enable port monitoring.
4. Enter **end**.
5. Enter **show running config** to verify your configuration.
6. To disable SPAN, follow the preceding steps but at step 3 put a **no** in front of the command.

Spanning ports can be configured three different ways:

- **Ingress** The destination port configured for SPAN gets copies of all the traffic received by the source ports.
- **Egress** The destination port configured for SPAN gets copies of all the traffic transmitted by the source ports.
- **Both** The destination port configured for SPAN gets copies of all traffic transmitted and received by the source ports.

Spanning VLANs

Spanning VLANs is another way to configure SPAN. This is commonly known as VLAN-based SPAN. Instead of specific source ports, a source VLAN may be

useful. Your organization may have specific VLANs established that should be monitored more closely than others. All traffic destined to or originating from the VLAN can be copied to the destination port configured as the SPAN port. There are also three configuration options here almost identical to SPAN Ports.

- **Ingress** The destination port configured for VSPAN gets copies of all the traffic received by the source VLAN.
- **Egress** The destination port configured for VSPAN gets copies of all the traffic transmitted by the source VLAN.
- **Both** The destination port configured for VSPAN gets copies of all traffic transmitted and received by the source VLAN.

Recovering the Sensor's Password

Recovering the password on any device is of significant importance. This procedure should be documented early in the deployment of the sensor. Once the default password on a Solaris-based Cisco Secure IDS Sensor is changed from the default *'attack'*, it is up to the administrator to maintain the passwords. In the event of a lost or forgotten password, password recovery procedures may need to be performed.

Designing & Planning...

Before You Begin

In order to carry out the password recovery procedures, you will need the following:

- Solaris for Intel CD-ROM.
- Solaris Device Configuration Assistant disk (boot disk). This can be downloaded from the Sun support Web site. http://soldc.sun.com/support/drivers/dca_diskettes/. Cisco does not maintain this diskette on the Cisco Web site and cannot guarantee the whereabouts of the information on the Sun Web site.
- Console access to the workstation.

To recover your password, perform the following steps:

1. Insert the boot disk.
2. Insert the CD into the CD-ROM and power off the sensor.
3. After ten seconds, power on the sensor. The sensor will boot from the boot disk and display the Configuration Assistant screen.
4. Press **F3** to scan the system for boot devices. A list of boot devices is displayed.
5. Select the **CD-ROM** drive and put an **X** next to it using the **Spacebar**.
6. Press **F2** to continue. The sensor boots from the CD-ROM.
7. A display for selecting the install type appears. Select **Option 2, Jumpstart**.
8. When prompted, choose **Option 0** for English language.
9. The next screen is an additional prompt for English ANSI. Choose **Option 0**.
10. The sensor boots and the Solaris Installation screen appears.
11. Press **Ctrl + C** to stop the installation script and be dropped to a prompt.
12. Type **mount -F ufs /dev/dsk/c0t0d0s0 /mnt**.
13. The “/” partition is now mounted at the “/mnt” mount point. At this time, the “/etc/shadow” file can be edited to remove the root password. Type **cd /mnt/etc**.
14. In order to read the data correctly, set the shell environment by Typing **TERM=ansi**.
15. Type **export TERM**.
16. To edit the shadow file, type **vi shadow**.
17. The line to edit is **root:gNyrpgZhdfxPQ:9078:::**
18. The encrypted password is the second field separated by “:”.
19. Delete that second field by moving the cursor to the beginning of the encrypted password and use the “x” to delete each character. The finished record should look like this: **root:: 9078:::**
20. Type **:wq!** to write the file and quit the editor.

21. Remove the disk and CD-ROM from the drives.
22. To reboot, type **init 6**.
23. At the login screen, login as root. When prompted for a password, press **Enter**. You are logged in to the sensor as root. Set a new password.

Reinitializing the Sensor

Reinitializing the sensor is an important process to understand. The OS of the sensor can become corrupt simply from shutting down incorrectly. The sensor is an Intel-based Solaris platform and proper procedures must be followed. Also, if the sensor is used, or the password and configuration is unknown, you may want to reinitialize to get a fresh build. There is basically three ways to reinitialize the sensor or recover the image. You can download the image from Cisco.com, use the recovery CD supplied with the sensor, or uninstall/roll back to a previous version. Regardless of the method you use, make sure you have documented the configuration thoroughly. This will expedite the recovery or reinitialization process.

Downloading the Image

To download the appropriate image, follow these steps:

1. Download the binary file from Cisco.com at the following address:
www.cisco.com/cgi-bin/tablebuild.pl/ids-appsens.
2. Copy the binary file to the /tmp directory on the target sensor. Make sure you maintain the same filename shown on the Web site.
3. Log on to the sensor as root.
4. Make sure the file attributes allow the file to be executed. Use the command **chmod +x <filename>**.
5. Execute the command by typing the filename with the **-I** (install) switch, like this: **./filename -I**.
6. Review /usr/nr/sp-update/output.log to ensure the installation was successful.

Using the CD

Using the recovery/upgrade CD is the preferred method for recovering or reinitializing a sensor. Make sure you have it on hand, and then perform the following:

1. Insert the upgrade/recovery CD into the CD-ROM drive.
2. Attach a keyboard and monitor directly to the sensor or connect with a workstation via a null-modem cable to the COM port.
3. Log into the sensor as root.
4. Reboot the sensor. Type **init 6**.
5. During the reboot, break out of the process by pressing **F2** to enter the Setup menu. Here, you are verifying that the boot sequence is floppy drive, CD-ROM, and hard drive. You want to make sure the sensor boots from the CD. If this has been done previously, it should not need to be done again but it is good to check.
6. Save any changes and exit the menu.
7. Once the sensor boots completely, the first prompt asks if you will install from the console (option c) or from a remote/serial terminal connection (option t). Depending on how you are connected, console (keyboard and monitor), or remote/serial terminal connection (COM port) select the appropriate option. The CD defaults to option t after ten seconds if no selection is made.
8. After the re-imaging is complete login as root. All previous configurations have been overwritten at this time, so the password is once again 'attack'.
9. The next command is `sysconfig-sensor` at the prompt.
10. If you had previously configured the sensor you should have documented the configuration before re-imaging. Enter the appropriate settings, save, and exit.

Using the Recovery Partition

In version 4.0 of the IDS sensor software, administrators have the option of re-imaging the sensor from a recovery partition. This procedure works for all versions of the 4200 series sensors, provided you have the correct image file with your sensor model. Image file `IDS-42XX-K9-r-1.2-a-4.1-1-S47.tar.pkg` is for all 4200 series models except model IDS-4215. Image file `IDS-4215-K9-r-1.1-a-4.1-1-S47.tar.pkg` is specifically for the model IDS-4215 sensor. Neither image file will work for the Catalyst 6000 IDS Module. The sensor must also be version 4.0(1)S37 or later. This process cannot be used to upgrade a 3.x or earlier sensor.

If you need to upgrade the recovery partition, follow steps 1–6. To recover the application partition using the recovery partition, skip down to steps 7–9. Once the recovery process has been completed, you will need to initialize the sensor by following steps 10–32.

1. Download the Recovery Partition Image File to your Secure Copy Protocol (SCP) Network Server or your FTP Server from Cisco's Software Center at www.cisco.com/kobayashi/sw-center/sw-ciscosecure.shtml. You need a CCO account to access these downloads.
2. Log on to the sensor's CLI via the console port or Telnet session.
3. Type **configuration terminal** to enter config mode.
4. Type **upgrade scp://user@server_ipaddress//upgrade_path/image_filename**.
5. Enter the password for the SCP or FTP server. Once the image has been downloaded, you are prompted whether to continue with the upgrade:

```
Warning: Executing this command will re-image the recovery
partition.
```

```
The system may be rebooted to complete the upgrade.
```

```
Continue with upgrade?
```

6. Type **yes** to continue. The recovery partition has now been re-imaged with the latest image.
7. From the CLI, type **configuration terminal** to enter config mode.
8. Type **recover application-partition**. You are prompted whether to continue with the recovery and warned that all changes except the network settings will be reset to the default settings.
9. Type **yes** to continue. After the partition has been recovered, the sensor has to be initialized using the *setup* command.

NOTE

Version 4.0 adds the look and feel of the Cisco CLI to its configuration. If you are familiar with configuring routers and firewalls, these commands and syntax should be comfortable to use.

10. Type **setup** to initialize the sensor. The System Configuration Dialog screen, shown next, is displayed. Press the **Spacebar** to continue.

```
--System Configuration Dialog--
```

```
At any point you may enter a question mark '?' for help.
```

```
Use ctrl-c to abort configuration dialog at any prompt.
```

```
Default Settings are in square brackets '['].
```

```
Current Configuration:
```

```
networkParams
ipAddress
netmask
defaultGateway
hostname
telnetOption
accessList 10.0.0.0 255.0.0.0
exit
timeParams
summerTimeParams
active-selection
exit
exit
service webServer
general
ports
exit
exit
```

11. You are prompted whether to continue with the configuration dialog. Type **yes** or press **Enter**. Any default answers are in the square “[]” brackets.
12. Type the hostname of the sensor.
13. Type the IP address.
14. Type the IP netmask.
15. Type the default gateway.

16. Enter the Telnet Server status. The server is disabled by default
17. Enter the Web server port. The port is 443 by default.
18. Save the configuration by typing **yes** or **no** to reconfigure.
19. Do not reboot at this point. Type **no** when asked to continue the reboot.
20. Enter configuration terminal mode. Type **configure terminal**.
21. Enter host configuration mode. Type **service host**.
22. Enter network parameters configuration mode. Type **networkParams**.
23. To show the current settings, type **show settings**. The expected output should be similar to the following:

```
networkParams
-----
ipAddress: 10.0.0.8
netmask: 255.255.255.0 default: 255.255.255.0
defaultGateway: 10.0.0.10
hostname: sensor1
telnetOption: disabled default: disabled
accessList (min: 0, max: 512, current: 1)
-----
ipAddress: 10.0.0.0
netmask: 255.0.0.0 default: 255.255.255.255
```

24. Remove the 10. network from having complete access. The command syntax is as follows:

```
no accessList ipAddress 10.0.0.0 netmask 255.0.0.0
```

25. Enter the IP addresses of hosts or networks that will have access to the sensor. If you can afford to do it, only specify individual host addresses that will have access. Do not give entire networks access unless absolutely necessary.

The syntax for a single host is as follows:

```
accessList ipAddress 10.0.0.4
```

The syntax for an entire network is as follows:

```
accessList ipAddress 10.0.0.0 netmask 255.255.255.0
```

Repeat the command as necessary depending on the number hosts or networks being added.

26. Exit the parameters configuration mode. Type **exit**.
27. Set the System clock settings. Type **timeParams**. When done, exit back to configure terminal mode.
28. Type **yes** to apply settings. Type **no** to keep the system from rebooting, then exit configure terminal mode. Type **exit**.
29. Set the clock. Type **clock set hh:mm month day year**.
30. At this point, you need to generate the X.509 by typing **tls generate key**. Record the results. You will need to verify the authenticity of the certificate when you connect via a Web browser.
31. Reboot the sensor. Type **reset**, then **yes**.
32. Once you have rebooted, you will need to upgrade to the latest signature updates and set the interfaces.

Uninstalling an Image

Uninstalling is fairly easy. The `uninstall` uses the `-U` parameter, which should be familiar to most Unix people. The `-U` means uninstall a binary that has previously been installed. If you remember earlier, the downloaded image was installed using the `-I` parameter. The command would resemble this:

```
./filename -I
```

It is fairly straightforward and should roll the sensor back to the version previous to the one being uninstalled. This is not very common though. In most cases, the sensor is reloaded completely and not rolled back to earlier versions unless you are troubleshooting.

Upgrading a Sensor from 3.1 to 4.0

Upgrading your IDS sensor to version 4.0 from 3.1 is very similar to re-imaging the sensor using the Cisco IDS 4.0(1) Upgrade/Recovery CD. There are a few considerations before upgrading that need to be addressed. If your IDS sensor is model IDS-4235 or IDS-4250, you must upgrade the BIOS before you can install version 4.0 on either platform. The other consideration is that if your IDS sensors are models IDS-4220-E or IDS-4230-FE, you must swap the interface

cables on the two ports. The PCI card that is normally used for sniffing on the IDS-4220-E and the IDS-4230-FE does not support monitoring of *dot1q trunk packets* or the tracking of alarm 993, *Dropped Packet*. The performance of the PCI card is also lower than the integrated NIC. If you do not swap the cables on the IDS-4220-E or IDS-4230-FE, there is a chance you will not be able to connect to your appliance over the network.

Prior to upgrading, make sure you record the configuration information before reinitializing the sensor using the *run Diagnostics* command.

Upgrading a Sensor BIOS

To upgrade the Bios on the IDS4235 or IDS4250, follow these steps:

NOTE

You have to use a directly connected keyboard and monitor for this procedure. A console connection will not work.

1. Copy the file Bios_A04.exe from the Cisco IDS 4.0(1) Upgrade/Recovery CD located off of the root in /BIOS to a temp folder on your Windows workstation. If you do not have the CD, you can download it from Cisco.com.
2. Insert a 1.44MB floppy diskette into your workstation.
3. Execute the BIOS update file. Double-click **BIOS_A04.exe**. This creates the BIOS update diskette.
4. Take the new BIOS diskette and insert it into your IDS-4235 or IDS-4250.
5. Boot the sensor from the BIOS diskette and follow the instructions displayed. Do not reboot or power off until this process has completed.
6. With the upgrade finished, remove the diskette and reboot the sensor.

Initializing a Version 4.0 Sensor

Once you have met all the necessary requirements for older sensor models, you need to initialize the sensor. If the sensor is a newer model, no additional considerations need to be made.

To initialize a sensor with software version 4.0, follow these steps.

1. Power up the appliance.
2. Insert the Cisco IDS 4.0(1) Upgrade/Recovery CD.
3. When the boot menu appears, type either a **k** to use a directly connected keyboard and monitor, or type **s** to use a serial connection while installing the image. It will take several minutes for the files to copy to the sensor.
4. Log on to the sensor. The default username and password for version 4.0 are the same: *cisco*. You will be prompted to change the password on the first login.
5. At the prompt, type **setup** to initialize the sensor. The System Configuration Dialog screen, shown next, is displayed. Press the **Spacebar** to continue.

```
---System Configuration Dialog---
```

```
At any point you may enter a question mark '?' for help.
```

```
Use ctrl-c to abort configuration dialog at any prompt.
```

```
Default Settings are in square brackets '['].
```

```
Current Configuration:
```

```
networkParams
ipAddress
netmask
defaultGateway
hostname
telnetOption
accessList 10.0.0.0 255.0.0.0
exit
timeParams
summerTimeParams
```

```

active-selection
exit
exit
service webServer
general
ports
exit
exit

```

6. You are prompted whether to continue with the configuration dialog. Type **yes** or press **Enter**. Any default answers are in the square “[]” brackets.
7. Type the host name of the sensor.
8. Type the IP address.
9. Type the IP netmask.
10. Type the default gateway.
11. Enter the Telnet server status. The server is disabled by default
12. Enter the Web server port, which is 443 by default.
13. Save the configuration by typing **yes** or **no** to reconfigure.
14. Do not reboot at this point. Type **no** when asked to continue with the reboot.
15. Enter configuration terminal mode. Type **configure terminal**.
16. Enter host configuration mode. Type **service host**.
17. Enter network parameters configuration mode. Type **networkParams**.
18. To show the current settings, type **show settings**. The expected output should be similar to the following:

```

networkParams
-----
ipAddress: 10.0.0.8
netmask: 255.255.255.0 default: 255.255.255.0
defaultGateway: 10.0.0.10
hostname: sensor1
telnetOption: disabled default: disabled
accessList (min: 0, max: 512, current: 1)

```

```
-----  
ipAddress: 10.0.0.0  
netmask: 255.0.0.0 default: 255.255.255.255
```

19. Remove the 10. network from having complete access. The command syntax is as follows:

```
no accessList ipAddress 10.0.0.0 netmask 255.0.0.0
```

20. Enter the IP addresses of hosts or networks that will have access to the sensor. If you can afford to do it, only specify individual host addresses that will have access. Do not give entire networks access unless absolutely necessary.

The syntax for a single host is as follows:

```
accessList ipAddress 10.0.0.4
```

The syntax for an entire network is as follows:

```
accessList ipAddress 10.0.0.0 netmask 255.255.255.0
```

Repeat the command as necessary depending on the number hosts or networks being added.

21. Exit the parameters configuration mode. Type **exit**.
22. Set the System clock settings. Type **timeParams**. When done, exit back to configure terminal mode.
23. Type **yes** to apply settings. Type **no** to keep the system from rebooting, then exit configure terminal mode. Type **exit**.
24. Set the clock. Type **clock set hh:mm month day year**.
25. At this point, you need to generate the X.509 by typing **tls generate key**. Record the results. You will need to verify the authenticity of the certificate when you connect via a Web browser.
26. Reboot the sensor. Type **reset**, then **yes**.
27. Once you have rebooted, you will need to upgrade to the latest signature updates and set the interfaces.

Configuring & Implementing...

Switching Interfaces for Multicast Traffic

Multicast Media Access Control (MAC) traffic is becoming more prominent on enterprise networks. More employees have a need for, or want to have access to, television feeds, stock tickers, broadcast news, and radio. In order to monitor this type of traffic on the 4220-E or 4230-FE sensors, the sniffing ports need to be changed. Follow these five simple steps:

1. Log in to the sensor as root.
2. Change directories to the `/usr/nr/etc/` directory.
3. Open the `packetd.conf` file for editing.
4. Change the `NameOfPacketDevice` token to `/dev/iprb0`.
5. Save and exit.
6. Type `mv /etc/hostname.iprb0 /etc/hostname.spwr0` to reconfigure the `spwr` interface for command and control.
7. Swap the network cables between the two interfaces, `iprb0` and `spwr0`.
8. Reboot the sensor for changes to take place.

Summary

Initializing the sensor is essential in getting your IDS infrastructure up and running. Without the proper settings, the sensor may not communicate with the management devices or the network in general. There are basically two types of sensors available:

- 4200 series sensors (4210, 4220, 4230, and 4235)
- Catalyst 6000 IDS Module

We have only discussed the 4200 series sensors and how to bootstrap them. The Catalyst 6000 IDS Module will be discussed in a later chapter. The sensor port or the sniffer port is important to be able to identify for proper configuration. The sniffing port on the 4210, `/dev/iprb0`, is physically located directly above the control port.

The 4220 and 4230 sensors have expansion slots. One of the ports is built in (a control port) and the other is located on the expansion slot. The sniffing port for Ethernet, `/dev/spwr0`, is physically located in slot 5. Depending on the type of network, different cards and slots are used. For token ring, use `/dev/mtok36`, located in slot 6. An FDDI network utilizes `/dev/ptpci`, which can be found in slot 4.

`sysconfig-sensor` is the utility used to initially configure the sensor. Options 1–6 must be done in order to get the sensor up on the network and talking.

The sensors have two accounts associated with them, `root` and `netrangr`. `Root` is used to bootstrap the sensor and perform OS-level functions on it, while `netrangr` (remember, no “e”) is used to administer the sensor. The commands `netrangr` can utilize on the sensor include: `cidServer`, `idsstart`, `idsstop`, `idsvers`, `idsconns`, and `idsstatus`.

The PostOffice protocol utilizes UDP45000 for communications, and can send the same messages to as many as 255 devices. It can also be configured to send messages to multihomed devices in the event of a segment failure on your network. Thus, it will continue to send the same message until an acknowledgment is received from the management device.

A SPAN port, or SPAN VLAN (VSPAN), needs to be configured in order for the sensor to capture packets. The sensor should be placed on the destination port in the configuration. The source ports or VLANs are configured to copy packets to the destination port the sensor resides on.

When reinitializing or recovering, the CD is quickest. Insert it and reboot. The whole process takes about an hour to get back to the `sysconfig-sensor`

screen. Downloading images from Cisco.com is another option, but if you keep up with the notifications from Cisco, you should probably already have the image on file and thus can reinstall it. Rolling back to a previous image/version is also an option, but as I mentioned before, I have never seen this used for any reason other than just to do it. If you have already upgraded, chances are the management software has been upgraded too. You may as well start off with a fresh install if you have to back up.

Solutions Fast Track

Identifying the Sensor

- ☑ 4210 is a single RU.
- ☑ 4210 ports are on top of each other. The sniffing port, `/dev/prb0`, is located on the bottom. The control port `prb1` can be found on top.
- ☑ The 4220 and 4230 have expansion slots. The control port is built in, while the sniffing ports occupy one of the slots (which slot depends on the network used).
- ☑ The Ethernet sniffing port `/dev/spwr0` occupies slot 5.
- ☑ For token ring, use `/dev/mtok36`. The card occupies slot 6.
- ☑ An FDDI network utilizes `/dev/ptpci`, which occupies slot 4.

Initializing the Sensor

- ☑ You must be root to initialize the sensor.
- ☑ Execute the command `sysconfig-sensor` and complete options 1–6 to get the sensor online.
- ☑ The host IDs must be unique for each device in the IDS infrastructure.
- ☑ The organization name and ID should be the same for all devices in a single infrastructure.

Using the Sensor Command-Line Interface

- ☑ When troubleshooting the sensor, utilize *idsconns* to check connectivity with the management device.
- ☑ *idsstatus* will tell you what services are up.
- ☑ *cidServer* version will tell you what versions of the daemons are being used.
- ☑ *idsstart* and *idsstop* do just what they say.
- ☑ *idsvers* verifies the version of sensor software.
- ☑ Don't forget to be logged in as *netrangr* to use these commands!

Configuring the SPAN Interface

- ☑ Configure SPAN ports or VSPAN for either Egress, Ingress, or both.
- ☑ Egress is the SPAN port (or VSPAN) receiving and copying to the destination port.
- ☑ Ingress is the SPAN port (or VSPAN) transmitting and copying to the destination port.
- ☑ Both copies transmit and receive traffic to the destination port.
- ☑ The destination port is where the sensor resides.

Recovering the Sensor's Password

- ☑ Don't even attempt to recover the sensor's password unless you have a Solaris for Intel CD-ROM, Solaris Device Configuration Assistant disk (boot disk).
- ☑ You need console access to the workstation for password recovery.
- ☑ The Solaris Device Configuration Assistant boot disk can be downloaded from Sun, not from Cisco.
- ☑ You will be editing the shadow file in the OS that contains accounts and passwords. If you are not familiar or comfortable with the process, find a Unix person and have them do it for you.

Reinitializing the Sensor

- ☑ Use the accompanying Upgrade/Recovery CD to reinitialize the sensor.
- ☑ If you have the image downloaded from Cisco.com, use that to save a minute or two.
- ☑ Once you reinitialize the sensor, everything is overwritten, including passwords. You are starting from scratch.
- ☑ Don't forget to document your settings before going this route.

Upgrading a Sensor from 3.1 to 4.0

- ☑ To upgrade sensor models IDS-4220-E or IDS-4230-FE, swap the cables for the sniffing interface as well as for the command and control interface.
- ☑ Before you can upgrade a sensor model IDS-4235 or IDS-4250, you have to upgrade the BIOS in order to install version 4.0.
- ☑ The default username and password to log in to the CLI for version 4.0 are both *cisco*.
- ☑ The command to initially configure the sensor is *setup*.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: How many ports does each sensor utilize and what are they for?

A: Each sensor utilizes two ports. One is used to sniff traffic (packets), while the other is for command and control.

Q: What command is utilized to verify connectivity between the sensor and management device?

A: *idsconns*. It will show you a *[established]* connection or a *[syn sent]* with a *syn NOT rcvd!*

Q: What port and protocol does the PostOffice protocol utilize?

A: UDP and 45000.

Q: Which options in *sysconfig-sensor* must be completed for initial deployment of a sensor in your IDS infrastructure?

A: Options 1–6, IP Address, IP Netmask, IP Host Name, Default Route, Access Control List, and Communications Infrastructure.

Q: Which account do you use to bootstrap the sensor?

A: root.

Q: In order to use the command-line interface, what account must you be logged in as?

A: netrangr.

Q: What interface must be configured on the 4220-E and 4230-FE sensors in order to monitor multicast traffic?

- A:** `iprb0` must be reconfigured from the command and control interface to the monitoring interface.
- Q:** What does the command `cidServer` do and what user must you be in order to execute it?
- A:** `cidServer` can start and stop the Web server for IDM and also show the version. You must be root to execute the command.
- Q:** What configuration options require a reboot in `sysconfig-sensor`?
- A:** Options 1–5, IP Address, IP Netmask, IP Host Name, Default Route, and Network Access Control.
- Q:** If you are upgrading sensor models IDS-4220-E or IDS-4230-FE, what must you do before you can upgrade to version 4.0?
- A:** You have to swap the interface cables on the two ports. The PCI card that is normally used for sniffing on the IDS-4220-E and the IDS-4230-FE does not support monitoring of *dot1q trunk packets* or the tracking of alarm *993, Dropped Packet*. The performance of the PCI card is also lower than the integrated NIC. If you do not swap the cables on the IDS-4220-E or IDS-4230-FE, there is a chance you will not be able to connect to your appliance over the network.
- Q:** Before you can upgrade to software version 4.0 on a sensor model IDS-4235 or IDS-4250, what has to be done first?
- A:** You must upgrade the BIOS before you can install version 4.0.

Cisco IDS Management

Solutions in this Chapter:

- Managing the IDS Overview
 - Using the Cisco Secure Policy Manager
 - Using the CSID Director for Unix
 - Using the IDS Device Manager
 - Using the Cisco Network Security Database (NSDB)
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

There is so much more to intrusion detection than just putting a sensor out on a network and then never addressing it again. Someone has to take the time and manage the sensors. It would not be very efficient to have to go to each of the sensors on a network and look at them on an individual basis. What if you saw something suspicious? Then you would have to go to the others and try and correlate the events. That is not the most efficient way to manage a group of security sensors. Luckily, we have a central management solution to help us manage our Cisco IDS sensors.

There are several items that need to be addressed when managing the IDS sensors on the network:

- How secure is the network going to be? Are we looking at everything or looking for specific events driven by our security policy?
- How many people will have access to the management console and who can modify the configuration?
- How much logging is going to take place? Do we log everything or only the events we care about?
- How often do we generate reports?
- Will alarms be sent to e-mail/pagers?
- Do I shun or carry out TCP resets?

Designing & Planning...

Shunning and Resets

Shunning is the process of blocking traffic from a certain host or network. To most, this sounds like a great idea, but if you have a Web presence for the purpose of e-commerce or marketing, you may be denying customers or potential ones the ability to do business with your organization. Shunning should be done with extreme caution, or not at all. Make sure you get the okay from management and explain the situation carefully to them before shutting someone out.

Continued

The other option is to do TCP resets. The name of "TCP reset" itself should be a clue to you that this only applies to TCP traffic. When an attack is detected, the sensors send out TCP reset messages to both the source and the destination of the attack. In order to properly use TCP resets in a switched network, a SPAN port must be configured for bidirectional traffic. The SPAN configuration must support bidirectional traffic and on the SPAN port, MAC learning must be disabled.

This only scratches the surface of planning your management solution. Depending on your business needs, you may find some solutions suit your business better than others. No matter what the solution though, IDS management is a full-time job with or without the central management solution. The central management solution just makes it much easier. You will find yourself constantly tuning signatures to reduce the amount of traffic that is generated. Be warned that the initial traffic can seem overwhelming, but in the end it's manageable. In fact, having any of these management solutions in place makes life easy, letting you implement one change at one location that affects all the sensors simultaneously.

In this chapter, we cover all the IDS management applications in depth. Cisco has three different methods: Cisco Secure Policy Manager (CSPM), IDS Device Manager (IDM), and Cisco IDS Director. After covering management solutions, we take a look at the Cisco Network Security Database (NSDB). Like most management solutions, initial deployment and configuration is the toughest. So it is our intent to cover these steps thoroughly.

Managing the IDS Overview

Many organizations often struggle with intrusion detection solutions. The solutions are not always as straightforward as you might think. One of the major drawbacks of IDS solutions is experience with intrusion analysis and what exactly is being protected. IDS sensors have to be tuned to the organization and each organization is different. Different types of traffic and traffic flow can set off alarms, even though it may be considered normal traffic for a particular organization. As always, Cisco has graced us with multiple ways to manage the IDS sensors, CSPM, Unix Director, and IDM. The goal of any of the Cisco IDS management applications is to provide a method for configuring certain features of the IDS, configuring logging and to generate reports from the IDS. With the

management application, it is possible to manage more than one IDS sensor without much difficulty, greatly reducing your workload, and allowing you to do it all from one centralized location. In the past, IDS sensors did not work very well unless there was an administrator in front of the IDS sensor scrutinizing every little record or alarm. The administrator had to be careful to tune signatures precisely in order to filter out the false positives and false negatives. But Cisco—and its tools—has taken a lot of the work out of IDS monitoring.

Up to now, one of the most common tools for managing Cisco IDS sensors has been CSPM. CSPM is a very scalable solution for centralized management of IDS sensors. CSPM does not only support Cisco IDS sensors but also other components within your enterprise, such as IP Security (IPSec), virtual private networks (VPNs), PIX firewalls, and IOS firewalls. CSPM allows you, the security administrator, to implement, enforce, and audit a security policy from a central location. CSPM provides a friendly graphical user interface (GUI) that gives administrators the ability to tune signatures for all the sensors in the enterprise or a single signature on one sensor. The ability to generate reports on demand or schedule them is also a benefit of having CSPM. If incidents are not being reported, the sensors may as well not even be on the network.

Another enterprise level management solution for multiple security components is the Cisco IDS Director. It runs on a Unix platform in the flavor of HP-UX or Sun Solaris. Another feature of the Director is the fact that it also has to run on top of HP OpenView. As you can tell right away, this solution is a very costly one. But, if you already have OpenView deployed in your enterprise, it might not be a bad solution to look into. Provided you have a robust enough system, the Director software can be loaded on an already existing OpenView platform running other OpenView applications.

Unlike CSPM and the Director, IDM is a web-based management solution that only allows you to configure and manage your IDS sensors on your network. IDM Web-based management is quickly becoming the management tool of choice for the Cisco IDS sensor. You can access your sensor right from your desktop or through a remote connection via a secure session. Both Netscape and Internet Explorer can be used to access the Web server. The Web server process runs locally on each IDS sensor. The best thing about IDM is it is FREE! It comes with 4.x and later IDS sensor software. It also comes with an Event Viewer to let you peruse alarms without having to parse through the log files, and allows you the luxury of viewing them from multiple sensors. The drawback to IDM is that you can only configure one sensor at a time.

There are different approaches with each of these, and thus some tips that will make your life easier. Currently, the push is towards Web-based management with the Cisco IDS device manager. Future trends show even more of a push towards a management solution that ties together almost all functionality from the different tools for Cisco's entire product line. Expect the functionality of all of these security management solutions to be integrated into VMS VPN/Security Management Solution in the near future.

Using the Cisco Secure Policy Manager

Even though there is a huge push for ease of use technology, such as Web-based interfaces like IDM, CSPM is still the prominent application in the industry for administrators tasked with managing Cisco IDS sensors. This section will take you through the installation of CSPM, configuration, and management.

For most administrators, CSPM is what we look for in an administration tool, a Windows-based product designed specifically to manage security policies not only for sensors but also for the PIX firewall, IOS routers, and VPN software. The focus here is strictly on managing the sensors. CSPM allows us to manage multiple sensors from a single location without having to perform any administration at the devices themselves.

Installing CSPM

Before installing CSPM, make sure the following software requirements have been met to save yourself from having to backtrack and install/configure them:

- Windows NT 4.0
- Service Pack 6a for NT
- Internet Explorer 5.5
- TCP/IP Protocol Stack
- HTML Help 1.32 Update
- Microsoft's XML Parser 3 (MSXML3)
- NTFS
- TAPI/MAPI for email
- DHCP should be disabled
- NT Startup time set to zero

NOTE

The autostart utility does a check for NT 4.0, Internet Explorer 5.5, HTML Help 1.32 Update, and MSXML3 during setup. The installation application does not know what any Windows version later than NT 4 is, or any browser version later than 5.5, so it will not continue. It will run nicely in a Connectix Virtual PC session, which in turn runs very well on Windows 2000 or XP.

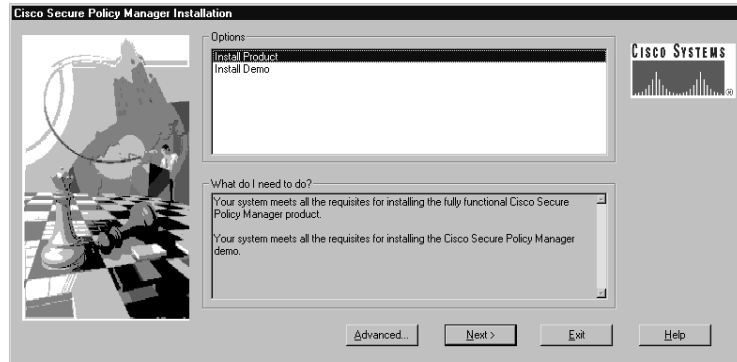
Due to the sensitivity of intrusion detection it is recommended that you install the CSPM as a stand-alone system. The CSPM system is designed to be in a location like a Security Operations Center (SOC). It allows all of the security personnel to look at the same interface and only those personnel with access to the SOC can access the system. The client/server installation allows administration to take place from different locations. This is not always a best practice and auditing, traceability, and nonrepudiation become an issue.

1. Insert the CSPM installation CD. The autostart utility will automatically initiate the installation.
2. The first thing you will see is a warning to disable any antivirus software during installation. Next, you will get the notice in Figure 4.1, Cisco Secure VPN client Not Installed on Host.

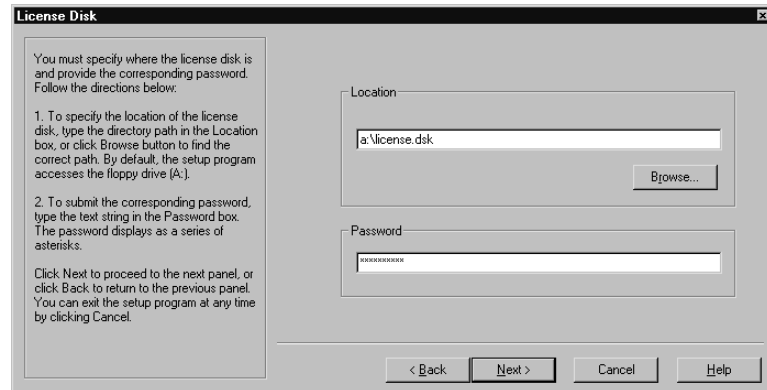
Figure 4.1 Cisco Secure VPN Client Warning Message



3. If you plan on installing the VPN client, do that before you install CSPM. Otherwise, press **Continue**.
4. Select **Install Product** in the **Options** box as seen in Figure 4.2, and then click **Next**.

Figure 4.2 Cisco Secure Policy Manager Installation

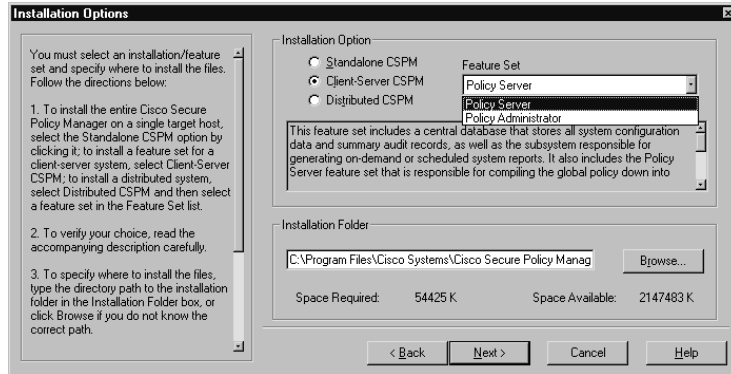
5. At this point, if the applications listed previously have not been installed, the installation cannot proceed. The Options box will display any required components that are not present.
6. At the **License Agreement panel**, accept the terms of the license and click **Next**.
7. Specify the location of the CSPM license disk, usually on the accompanying diskette, by entering the directory path.
8. You will also have to enter the password that corresponds with the license disk. The password is usually on the diskette label. Click **Next**. See Figure 4.3

Figure 4.3 CSPM License Disk

9. If you have downloaded the software, the password will be in the readme file.

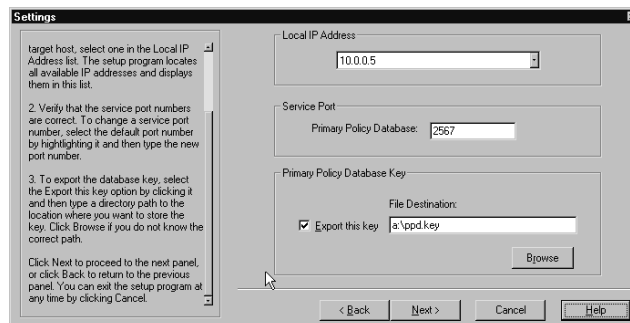
10. Select the type of system you want to install: **Standalone** or **Client/Server**. CSPM does not support the Distributed CSPM option. See Figure 4.4.

Figure 4.4 Installation Options



11. If you are installing a client/server system, select **Policy Server**. This needs to be installed before **Policy Administrator** in the **Feature Set** list. The Policy Administrator Feature Set is for Remote Administration. The Feature Set drop-down box is disabled for the Standalone option.
12. Specify the installation path in the **Installation Folder** box and click **Next**.
13. You will be prompted to enter the password for the Windows NT user-name detected during setup. Click **Next**.
14. Select the IP address configured on the local host for the stand-alone system and enter the port the Primary Policy Database will communicate on. The default port is **2567**. See Figure 4.5.

Figure 4.5 Settings



NOTE

When setting the IP address for CSPM, do not think that you can change it later. You can not change it without reinstalling CSPM, so make sure you get it right the first time. Don't ask how we know this.

15. Specify the Policy Database key location in the File Destination box. If you are doing a stand-alone system, it is not mandatory to export the key. The client/server system installation requires you to export the database key. Click **Next**.

NOTE

It is recommended that you export the database key to a diskette that is readily available and can be stored in a secure location. Exporting the database key to a network share is discouraged. If the network resources become inaccessible, the database key cannot be retrieved.

16. In the **Configure Communication Properties**, shown in Figure 4.6, enter your CSPM system's host ID, organization ID, the IP address (if it is not already displayed), the host name, and organization name.

Figure 4.6 Configure Communication Properties

Configure Communication Properties

Communication Infrastructure

You must define the properties used to establish communication between the host and the Cisco Secure Intrusion Detection Sensor (IDS) device. Follow the directions below.

1. To submit the Host ID, type in the host ID value specified during the initial sensor setup.
2. To submit the Organization ID, type in the organization ID value specified during the initial sensor setup.
3. To submit the IP Address, type the IP address used to manage the sensor object.
4. To submit the Host Name, type in the host name for the sensor.
5. To submit the Organization Name.

Fill in the information to enable the base communication. To continue, click Next.

Host ID: 5

Organization ID: 100

IP Address: 10.0.0.5

Host Name: cspm

Organization Name: lab

< Back Next > Cancel Help

17. Verify your settings. If a setting is incorrect, you can use the Back button to back up and make changes. If everything is correct, click **Copy Files**.
18. Once the installation has completed, click **Finish** to close the setup program.

If you are performing a stand-alone system installation, you will only have to do the installation procedures once. If you are implementing a client/server CSPM system, you need to repeat the preceding steps to install the Policy Administrator feature set on all additional hosts that will serve as clients for remote administration.

Once you have finished the installation, you will need to log in to start configuring.

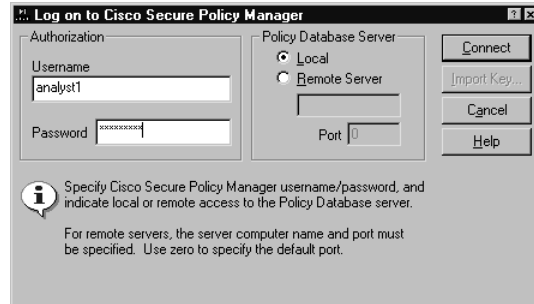
NOTE

A stand-alone system can be converted to a client/server system without having to uninstall and reinstall CSPM. The stand-alone system will act as the Policy Server. Once you have exported the database key from the stand-alone system, you can install the Policy Administrator feature set on multiple hosts for remote administration using that database key during the installation of the Policy Administrator feature set.

Logging In to CSPM

To log in to CSPM, follow these steps:

1. Open the **Log on to Cisco Secure Policy Manager** dialog box by maneuvering to the CSPM executable by clicking **Start | Programs | Cisco Systems**. Click **Cisco Secure Policy Manager**.
2. Use the account that was specified during the installation to log in. Enter the account name and password.
3. In a client/server system configuration when logging in from the Policy Server, click **Local** under Policy Database Server. When logging in from a remote server, click **Remote Server**, and then enter the IP address or DNS name in the box. Click **Connect**. See Figure 4.7.

Figure 4.7 Log on to Cisco Secure Policy Manager

If you are having trouble logging on to the CSPM, verify that the ORGID and ORGNAME on the CSPM match what is defined on the sensor. This is essential to communicate properly.

NOTE

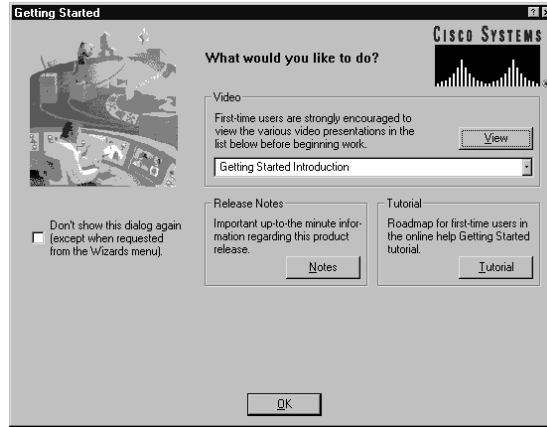
If the default port number of 2567 is still the communication port, you do not need to specify a port value.

Configuring CSPM

Now we are going to go through the configuration process for CSPM. The sensors need to be added to the topology in CSPM to start managing them. But before that happens, networks need to be defined and your CSPM host needs to be defined also. One thing that needs to be addressed up-front is that the postoffice configuration settings that include HOSTID, ORGID, HOSTNAME, and ORGNAME are correct and communication has been established between the sensors and management device. If the sensor is on the outside of a firewall, rules need to be put in place for postoffice communication to occur.

Once you log on to the CSPM, you will be greeted by the Getting Started pop-up window. The Getting Started window allows you to view different video tutorials that walk you through different procedures you will encounter while using CSPM. If you are a first-time user, it would be wise to take a moment and go through these videos. See Figure 4.8.

Figure 4.8 Getting Started



NOTE

The newest CSPM (3.1) does not support IDS sensors. For more details, see www.cisco.com/en/US/products/sw/secursw/ps2133/prod_software_versions_home.html.

CSPM v2.3.3i is the last version of CSPM that supports Cisco's IDS.

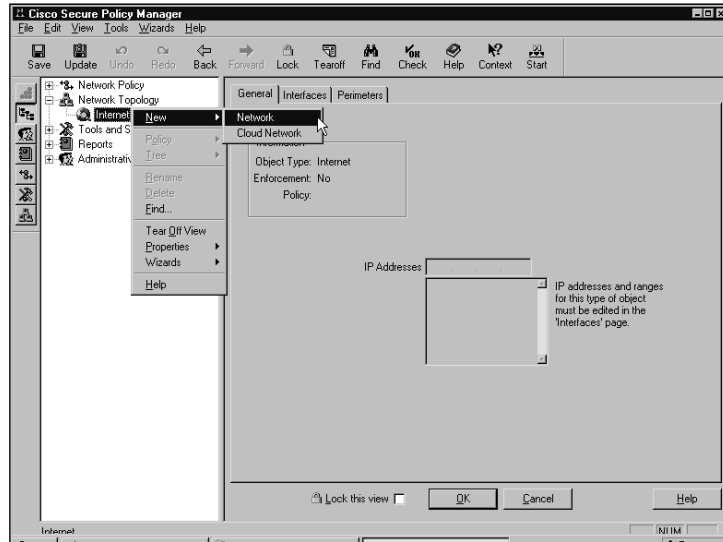
The first thing you need to do in configuring a topology in CSPM is to define the network upon which the control interface of the sensor will reside, and the network where the CSPM host will reside. If you do not have a command and control network, they may possibly be on the same subnet, hence only one network will need to be defined in the topology. So follow these steps to define a network for CSPM.

Adding a Network

Adding a network is the first step in defining a topology in CSPM. Without it, you will not be able to add any hosts. This is a logical map and does not necessarily need to be totally accurate, but it does need to be done.

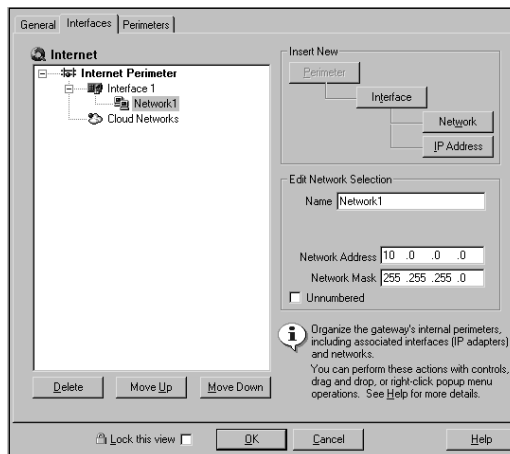
1. You will right mouse-click the Internet icon in the topology map and select **New**, then **Network** to create a new network. (Refer to Figure 4.9.)

Figure 4.9 Adding a Network



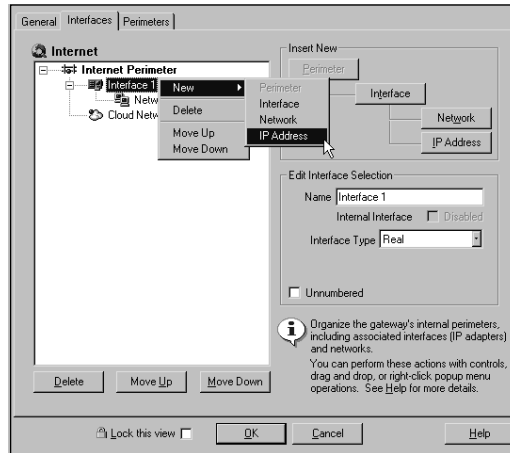
2. In the Network screen, add the name of the network, the network address, and the subnet mask that will be used. Notice in Figure 4.10, the name of the network can be whatever you want it to be. I recommend you name it something that makes sense to your organization (for instance, out-of-band network, command network, and so on). You have the option of simply identifying a network here without supplying any of the addressing by checking the **Unnumbered** box at the bottom of the window.

Figure 4.10 Network Parameters



3. Click the **IP Address** button or right-click the **interface icon**, select **New** then **IP Address**, as shown in Figure 4.11 and enter the IP address that the network will use to access the Internet. This should be your network's Default Gateway. Then click **OK**.

Figure 4.11 Interface IP Address



NOTE

Since you already defined these IP addresses on the sensor, they do not have to be correct on the topology map. This is for your benefit. The network will still be added to the topology map.

This topology map is more or less eye candy for you to know where your components are located in your IDS infrastructure. Since the IP addresses have already been defined on the sensors, they do not have to be correct

You have now defined your network. Now you need to add the CSPM host onto that network. We show how to add a CSPM host to your newly defined network in the next section.

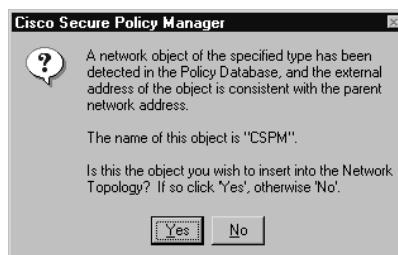
Adding a Host

In order to control a sensor with CSPM, you have to configure CSPM to communicate with the sensor. Configuration parameters are required to manage the

sensor. These procedures take you through the specific settings that have to be configured before the sensors can be managed with CSPM. Think PostOffice Protocol while setting up communications between CSPM and the sensors. The postoffice settings will also allow for the distribution of audit event messages.

1. Right-click the network icon you have just defined and select **New | Host**.
2. The **Cisco Secure Policy Manager** dialog box (shown in Figure 4.12) should appear, stating that a network object has been detected in the Policy Database. The dialog box will also display the name of the device. If you do not get a screen similar to this, you are not on the correct network.

Figure 4.12 Network Object Detection



3. Click the **Yes** button to install the CSPM host into the topology map.
4. To verify that the information for the CSPM host is correct, use the General screen, as shown in Figure 4.13. The SMTP Server will usually be your e-mail server in most cases. This should be defined as an object in your topology map also. If there is more than one IP address for your CSPM host, add them here.

Figure 4.13 The Host General Information Tab

- To configure the postoffice settings on the CSPM host, click the **Policy Distribution** tab shown in Figure 4.14. Each of the settings in the right pane have to be filled in correctly for CSPM to distribute policy changes. The Network Service field should be set to the PostOffice Protocol.

Figure 4.14 Host Policy Distribution Tab

- Once you have entered and verified the settings, click **OK**. The CSPM host icon will show up in the topology map under the network defined earlier.

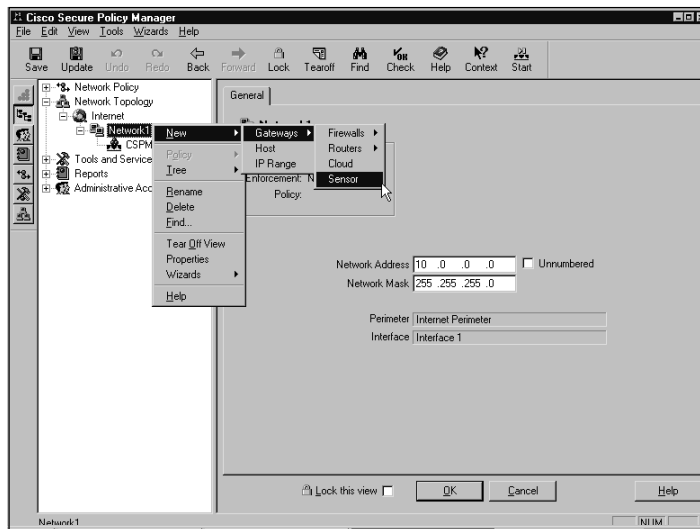
NOTE

If you modify the postoffice settings, audit events will not be forwarded or received until you save and update the configuration. A sensor must also be defined in order for events to be generated.

Adding a Sensor

After you have added your CSPM host, you will need to define the sensors that you will manage with CSPM. The procedure to define the sensors is similar to adding a host to your topology map. You can either right-click your network icon, click **New | Sensor** (as shown in Figure 4.15), or right-click your network icon and then click **Wizards | Add Sensor**. Whichever method you choose, the results will be the same. The wizard just helps take some of the work out of it.

Figure 4.15 Add Sensor

**NOTE**

If you have previously configured the sensor signatures, you will want to capture that configuration so you do not have to repeat the process. Use the wizard and check the box in the bottom-left corner of the first screen to capture that configuration.

The Identification tab for the sensor needs to be filled in for initial setup. You will enter the Sensor Name, Organization Name, choose the sensor version, verify the IP address, enter the host ID, and organization ID (refer to Figure 4.16). Do not worry about any of the other tabs at this moment. You just want to get the sensor added to your topology map.

Figure 4.16 Sensor Parameters

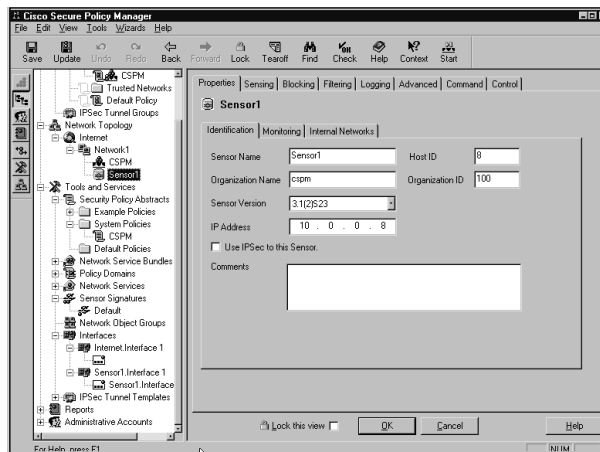
The screenshot shows the 'Sensor1' configuration window in the Cisco Secure Policy Manager. The 'Identification' tab is active, showing the following fields:

- Sensor Name: Sensor1
- Host ID: 8
- Organization Name: cspm
- Organization ID: 100
- Sensor Version: 3.112523
- IP Address: 10.0.0.8
- Use IPsec to this Sensor.
- Comments: (empty text area)

At the bottom of the window are buttons for 'Lock this view', 'OK', 'Cancel', and 'Help'.

In Figure 4.17, you see all of the tree structure that has been populated to the left pane of the CSPM screen. Notice under **Tools and Services | Sensor Signatures** the **Default** icon. This is the default set of signatures created for your sensors. You may actually have one of these for each sensor, or use only one to push the signatures to all sensors on your network.

Figure 4.17 CSPM Tree Structure



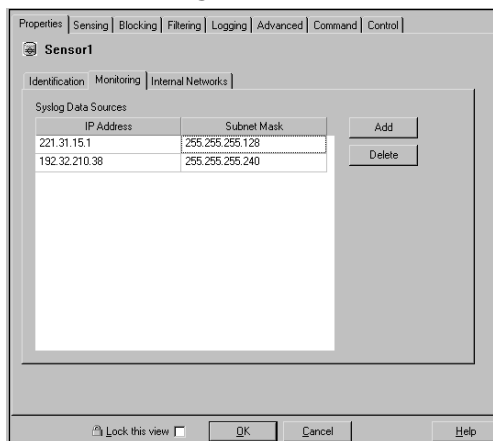
Once you have added all of your sensors and your CSPM host, you can begin configuring and optimizing/tuning the sensors and the sensor signatures. The sensor must be set up to sniff the traffic on the correct interface and log the events. Going through each of the configuration tabs on the sensor, we will configure your sensor.

The Properties Tab

The Properties tab allows you to set a few specific parameters to help identify your sensor, define internal and external networks, and also SYSLOG data streams via three subtabs: Identification, Monitoring, and Internal Networks.

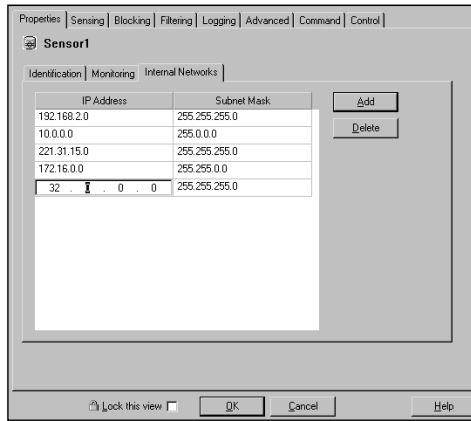
1. Select the sensor you are going to configure in the topology map. The first tab is the Properties tab. The Identification tab should already be filled in correctly. Verify the information on this tab is correct. Pay close attention to the Sensor Version. Also, utilize the comments box to enter important information regarding the network segment that is being monitored by this sensor.
2. To monitor SYSLOG data sources, select the **Monitoring** tab under the **Properties** tab (see Figure 4.18). The monitoring parameters allow you to add multiple SYSLOG data sources. Click **Add** and add the IP address and subnet mask for each data source. This is from the interface an IOS router is sending its SYSLOG traffic.

Figure 4.18 The Monitoring Tab



3. Select the **Internal Networks** tab (see Figure 4.19). In this section, you will define your Internal Protected networks that the sensor is protecting. CSPM uses this to parse the events in the Event Viewer. Any address space that is not identified in this section is considered an external address designated as “OUT.” The internal addresses are designated as “IN.”

Figure 4.19 The Internal Networks Tab

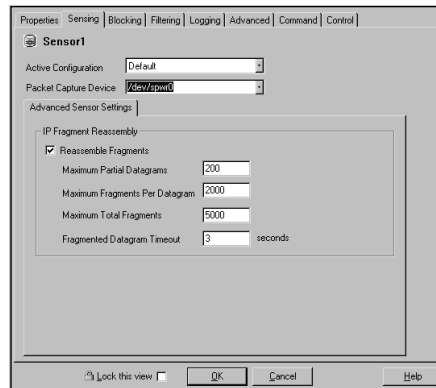


4. Click **Add** and add all of your internal address space that this sensor is protecting.

The Sensing Tab

The Sensing tab allows you to configure what signature configuration file the sensor is using, what Packet Capture Device (Interface) it's employing, and how to handle IP Fragment Reassembly.

1. Click the **Sensing** tab on the sensor you are going to configure (see Figure 4.20).
2. In the Active Configuration field, select the Sensor Signature file template the sensor will be using to monitor the network. It is not uncommon to have a different Sensor Signature file template for each sensor. Some signatures may be disabled or tuned differently depending on the positioning on the network.

Figure 4.20 The Sensing Tab

The Packet Capture device is the interface that is doing the sniffing. Refer to Chapter 3 for help with the different interfaces on a sensor.

Enabling IP Fragment Reassembly causes your sensor to reassemble a fragmented IP packet first, then compare that packet with a signature. This can be a resource hog depending on your network traffic patterns. Unless you are very familiar with the traffic patterns on your network, do not modify the default settings.

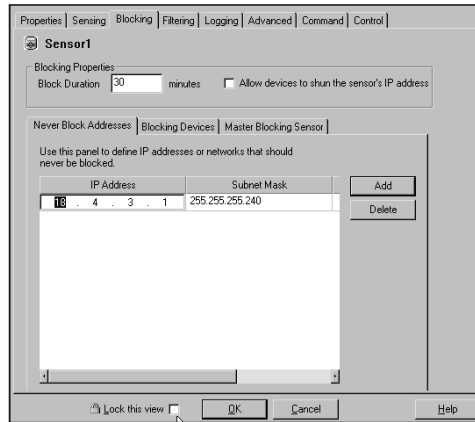
The Blocking Tab

Configuring blocking by the sensor on a network can be a difficult topic. Your networking team may not support your efforts to enable blocking because the sensor will automatically log in to a device and modify the configuration for a period of time when suspicious activity is detected. Some security policies make this a prohibited practice and not all sensor models support this feature. At present, only the 4200 series sensors support this configuration option. The Catalyst 6000 IDSM-1 module does not support blocking but the new IDSM-2 module does.

1. Click the **Blocking** tab on the sensor you are configuring for blocking. Within that tab are three subtabs:
 - Never Block Addresses
 - Blocking Devices
 - Master Blocking Sensor

There are also two fields, Block Duration and Cisco ACL Number (see Figure 4.21). You will add any addresses that will not be blocked to the list.

Figure 4.21 The Blocking Tab



The Never Block Address tab lets you specify IP addresses that should never be blocked. This is an important thing to consider when you do business online. If you have clients and customers with trusted business relationships, you may want to enter all of those addresses in this tab. This will prevent them from being blocked inadvertently by a false positive.

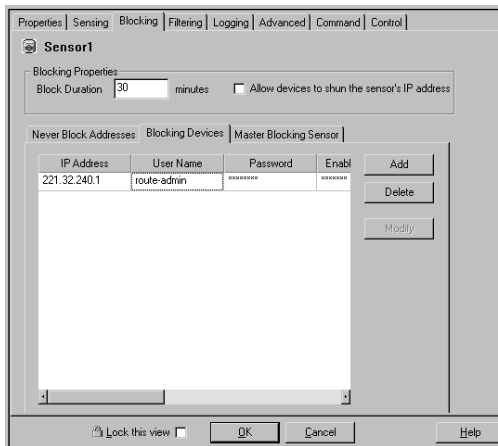
NOTE

Hackers can spoof IP addresses of clients, customers, and business partners and trigger alarms that prompt the sensor to block traffic. This can cause a denial of service to your resources.

2. Select the **Blocking Devices** tab. Here you define the parameters the sensor will use to access a device and modify an ACL. The information needed is
 - The Telnet IP address
 - The Telnet username

- The Telnet password
 - the enable password
 - The blocking interface
3. You can tell from the list of required information why the network personnel may be reluctant to support this feature. Click **Add**. See Figure 4.22. Add the information from the preceding list. Repeat as needed. Click **OK** to continue.

Figure 4.22 Blocking Device Properties



4. Specify the length of time the blocking will last in minutes in the Block Duration field. Also, specify the ACL number that will be modified. Without getting into the different types of ACLs, I will simply list them. Refer to Cisco.com for further information regarding ACLs.
- **Number 1–99** The IP Standard access list
 - **Number 100–199** The IP Extended access list
 - **Number 1300–1999** The IP Standard access list Expanded range
 - **Number 2000–2699** The IP Extended access list Expanded range
- Remember when the block duration has ended that the sensor will log back in to the device and remove the configuration used to block.
5. Access the **Master Blocking Sensor** tab. Select the sensor name that will act as the Master, then click **OK**.

NOTE

A Master Blocking sensor needs to be defined if you have multiple entry points into your network. What happens is, if a sensor blocks traffic at a certain entry point router, that sensor tells the Master Blocking Sensor to also block the other entry point(s).

The Filtering Tab

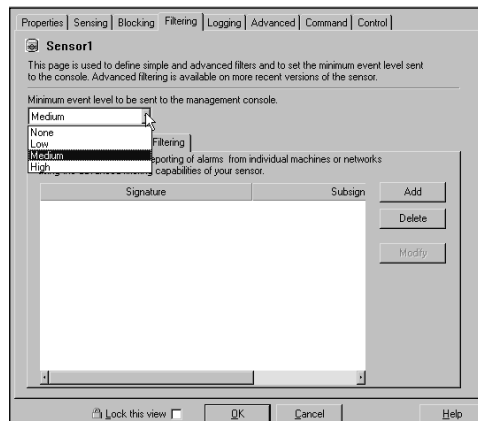
The Filtering tab helps you reduce the size of your database by filtering out certain signatures from hosts that you have determined to be false positives. There are three ways to filter alarms: minimum event level, simple filtering, and advanced filtering. To configure filtering, see the following sections.

Minimum Event Level

The Minimum Event Level drop-down menu allows you to choose the minimum severity level of alarms that will be sent to the management console. This helps with log reduction in that you can select Medium or High and not have to worry about sorting through low-level alarms.

1. Click the **Filter** tab on the sensor you are configuring.
2. The main screen shows the Minimum Event Level field at the top. Select the minimum level of alarms that will be sent to the CSPM console (see Figure 4.23).

Figure 4.23 Minimum Event Level Filtering



NOTE

You may not be interested in low severity alarms and only want Medium severity and above. This keeps you from having to sort through large amounts of minor alarms. This is a huge log reducer.

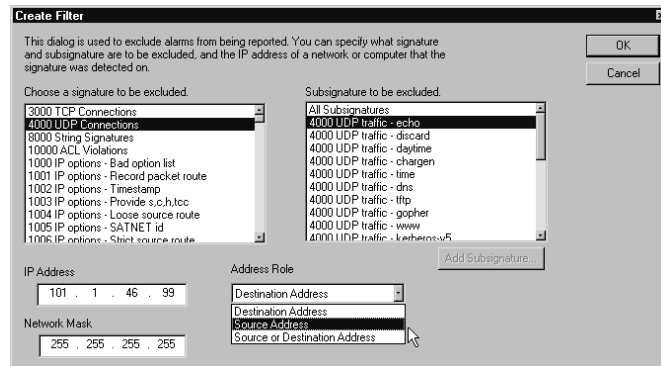
3. Save and Update your CSPM configuration.
4. Download the new sensor configuration to the target sensor.

Simple Filtering

Simple Filtering takes log reduction further than simply not receiving lower level alarms that might not interest you. With Simple Filtering, you can actually filter out signatures that you consider benign on your network to or from specific addresses. This helps reduce your logs even further, thus allowing you to spend more time on the important alarms. Follow these steps to configure Simple Filtering:

1. Click the **Filter** tab on the sensor you are configuring.
2. On the **Simple Filtering** subtab, click **Add**.
3. Select the Signature ID, any subsignatures, the IP address to exclude, and the address role. The address role tells the sensor if the IP address is the source or the destination address for the signature or both (see Figure 4.24).

Figure 4.24 Simple Filtering



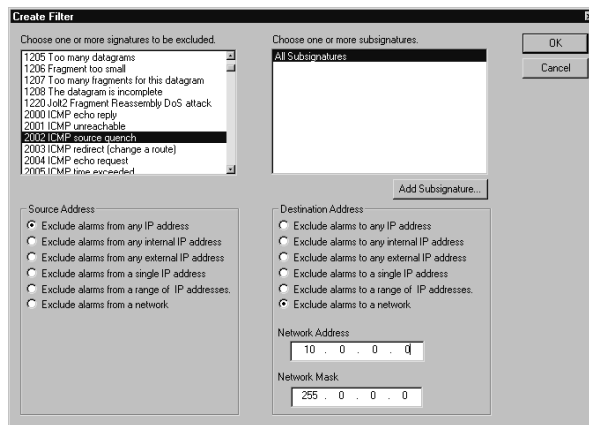
4. Once you have completed the information, click the **OK** button.
5. Save and update your CSPM configuration.
6. Download the new sensor configuration to the target sensor.

Advanced Filtering

Advanced Filtering goes even further to reduce your logs and help you focus on what's important. The difference in the Advanced Filtering tab is that, instead of just excluding signatures and associated subsignatures from a network or specific host, you can include and exclude the same to and from hosts. Certain hosts may generate an alarm based on a signature, but analysis may show that this is normal traffic for the host. In contrast, you may have configured the signature to be excluded in the Simple Filter tab and want to include or monitor a specific host or network based on the signature. Follow these steps to configure Advanced Filtering:

1. Click the **Filter** tab on the sensor you are configuring.
2. Click the **Advanced Filtering** sub-tab and click **Add**. This is similar to the Simple Filtering tab, with some added functionality.
3. Select the Signature ID and any subsignatures.
4. For IP addresses, you can specify single, multiple, or ranges of IP addresses for the source and destination. It is perfect for those noisy signatures that generate tons of alarms in your Event Viewer (see Figure 4.25).

Figure 4.25 Advanced Filtering



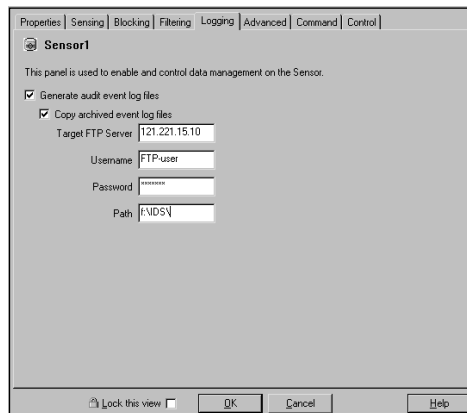
5. Once you have entered all of the required information, click **OK**.
6. Save and update your CSPM configuration.
7. Push the sensor configuration to the sensor.

The Logging Tab

By enabling logging on your sensors, you are creating log files for future use. It may be required in your industry to maintain logs for a period of time. By enabling logging, you can have the sensor do the work for you by creating the log and then FTPing it to a location for safe-keeping (see Figure 4.26). To enable logging, follow these steps:

1. Select the **Logging** tab on the sensor you are configuring.
2. Select **Generate audit event log files**.
3. Either have the log file saved to the sensor or have it FTP'd to another location. Although not mandatory for logging, you may have a requirement to archive the log files. In this same window, you can point the sensor to an FTP server and have the logs saved off to a logging server for archival and backup purposes. Click **OK**.
4. Save and update your CSPM configuration.
5. Download the new sensor configuration to the target sensor.

Figure 4.26 Logging



The Advanced Tab

The Advanced tab allows you to configure additional PostOffice features such as Watchdog Properties and Additional Destinations. Watchdog queries the PostOffice services running on the local host and the sensors. If Watchdog detects that a service is not running the parameters defined here, tell the sensor how to treat the situation and how it is reported (see Figure 4.27). To specify additional destinations that the sensor will forward alarms to, use the Additional Destinations subtab (see Figure 4.28).

Figure 4.27 Advanced PostOffice Settings

The screenshot shows the 'Advanced' tab of the 'Sensor1' configuration window. The 'Postoffice Settings' subtab is selected. The 'Watchdog Properties' section includes:

- Watchdog Interval: 30
- Number of Restarts: 3
- Watchdog Timeout: 240 (with a note: 'Must be at least twice as large as the Watchdog Interval + 1')
- Postoffice Heartbeat Interval: 5

 The 'Daemon Errors' section includes:

- Daemon Down Alarm Level: High
- Daemon Unstartable Alarm Level: High

 A 'Set to Defaults' button is located at the bottom right of the settings area. The bottom of the window has 'Lock this view', 'OK', 'Cancel', and 'Help' buttons.

Figure 4.28 Additional Destinations

The screenshot shows the 'Additional Destinations' subtab of the 'Sensor1' configuration window. It contains a table for defining destinations:

Name.OrganiZation	Host ID	Organization ID	Serv	
admin.lab	45	100	smid	<input type="button" value="Add"/> <input type="button" value="Delete"/>

The bottom of the window has 'Lock this view', 'OK', 'Cancel', and 'Help' buttons.

PostOffice Settings (Watchdog)

To configure the additional PostOffice settings (Watchdog) follow these steps:

1. Select the **Advanced** tab on the sensor you are configuring.
2. In the **Watchdog Interval** field, enter the number of seconds between each query Watchdog will perform on the services to see if they are running.
3. In the **Number of Restarts** field, enter the number of restart attempts PostOffice makes for downed services. If PostOffice cannot start the service in the number of times specified, a Daemon Unstartable alarm is fired. The default is three attempts.
4. In the **Watchdog Timeout** field specify the number of seconds Watchdog will wait for a response to a query. If Watchdog does not receive a response in the allotted time, a Daemon Down alarm is fired. The default is 240 seconds.

For the **PostOffice Heartbeat Interval** field, specify the number of seconds that PostOffice should wait after querying remote PostOffices. If the query does not generate a response, a Route Down alarm is fired. The default is five seconds.

6. To the right is the **Damon Down Alarm Level** field and the **Daemon Unstartable Alarm Level** field. Select the level of the alarm that will be sent to the console, High, Medium, or low. The default for both fields is High.
7. Save and update your CSPM configuration.
8. Push the sensor configuration to the sensor by clicking the **Approve Now** button on the **Command** tab for the sensor.

Additional Destinations

To configure the additional destinations, follow these steps:

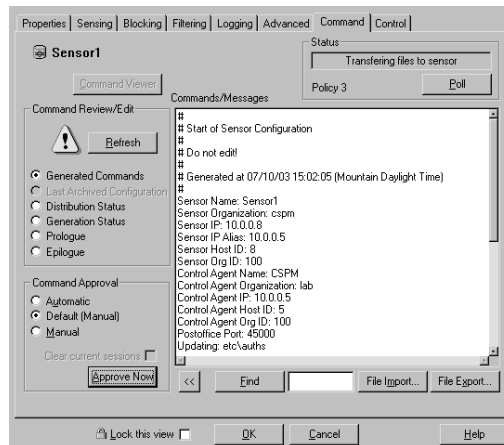
1. On the **Advanced** tab, select the **Additional Destinations** subtab.
2. Click **Add**.
3. Enter the sensor name, organization name, organization ID, sensor ID, service name, minimum event level, IP address, heartbeat timeout, and port.

4. Click **OK**.
5. Save and update your CSPM configuration.
6. Push the sensor configuration to the sensor.

The Command Tab

The Command tab allows you to update your sensors with updated configuration files (see Figure 4.29). The **Approve Now** button at the bottom of the screen starts the update process. The Approve Now button is enabled when configuration files are ready to be sent to the sensors. If no changes are available, the button is grayed out.

Figure 4.29 The Command Tab



In the Command Review/Edit pane, you can view Pending Command, Current Configuration, Distribution Status, Generation Status, Prologue, and Epilogue. Select the one you want to view the status of and press the **Refresh** button in the same pane.

NOTE

The sensor only utilizes two of the options: Pending Commands and Distribution Status.

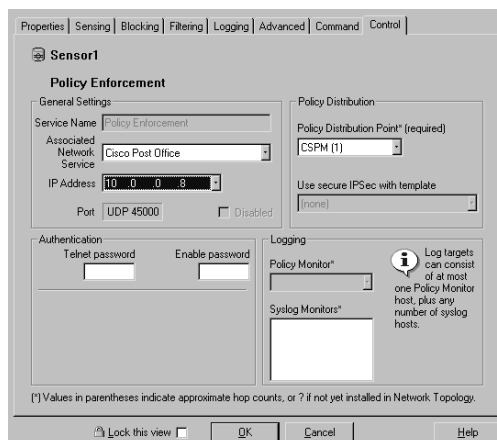
The Poll button located in the upper-right corner of the Command tab checks the status of your sensor. The window above the Poll button shows the current status.

The Control Tab

On the Control tab, you can specify the Policy Distribution Point and the Associated Network Service. There are other options listed in this window but the only ones that are available are these two. The Policy Distribution Point is the device sending updates to the policy. This is the CSPM server that generates and publishes command sets to the selected sensor(s). Remember, you can have multiple CSPM servers in your architecture so it is important to make sure you select the correct one. Follow these steps to select the CSPM server that will generate and publish the commands for your selected sensor:

1. Once you have selected the sensor, you want to specify a CSPM server or click the **Control** tab in the View pane. The Control tab, as shown in Figure 4.30, appears.
2. Click the drop-down menu to select the CSPM server you will use. Only CSPM servers that have already been defined in the network topology will be displayed.
3. Make sure the **Associated Network Service** is set to *Cisco Post Office*. This is the mode in which communication occurs. We are using the PostOffice Protocol.
4. Click **OK**, then save and update the configuration.

Figure 4.30 The Control Tab

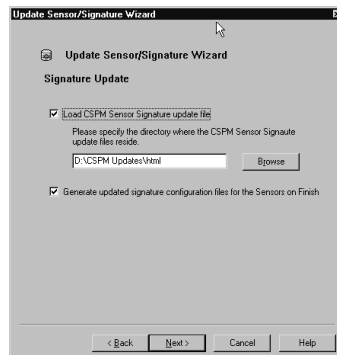


Signature Updates

Chances are that your initial setup of CSPM and the sensor are going to be out-of-date. The signature files that come with the CSPM software and the sensor itself will remain behind the current signatures to some degree. Remember that one of the rules of good network security is to stay current with patches and signatures, therefore we need to update the sensor and CSPM to the latest level. In order to update the signatures, we need to follow the steps listed here:

1. Go out to Cisco.com and download the current signature files from the following Web site: www.cisco.com/cgi-bin/tablebuild.pl/ids3-app. This requires you to have a SMARTnet maintenance contract number and a Cisco Connection Online (CCO) account to request software upgrades from CCO.
2. Download the CSPM signature update file(s) needed.
3. Back up your current CSPM topology and database. Export your topology by clicking **File | Export to file**. Back up your **data** directory from the CSPM Install Directory.
4. Load the CSPM signature update. Unzip the signature update file to a local folder. Select **Signature Update | Update Sensor** from the wizards list.
5. Check **Load CSPM Sensor Signature Update file**.
6. Specify the path to the **html** directory from the update file you previously unzipped (see Figure 4.31) and select **Next**. You do not need to check the box for Generate Updated Signature Configuration Files For The Sensors On Finish unless you intend to update the sensors also.

Figure 4.31 The Update Sensor/Signature Wizard



8. After the process is complete, save your changes by choosing **File | Save changes**.
9. Exit CSPM and reboot the system.
10. When the system finishes rebooting, start CSPM and log in.

Configuring IPSec

IP Security (IPSec) provides security features such as confidentiality, integrity, and authentication via a protocol suite into IP. CSPM can be used to create encrypted tunnels between devices that support IPSec. IPSec tunnels enable peer-to-peer secure transmission of data over a public, untrusted IP network. In this scenario it is used for communication between CSPM and the sensors. It cannot be used between the sensors and blocking devices. Refer to the IPSec Tunnel Implementation, v2.0, which can be found at the following address: www.cisco.com/en/US/products/sw/secursw/ps2133/products_user_guide_book09186a008010703e.html

Before you can configure the IPSec tunnels, the Cisco Secure VPN client must be installed on the CSPM server. Sensors that will be managed by CSPM using IPSec tunnels must be running IDS software version 2.5(1)S0 or later. The CSPM server and all sensors must be defined in the topology. The following steps walk you through configuring IPSec:

1. Verify that the sensor(s) supports IPSec and select the appropriate IPSec tunnel template. Use a manual template for CSPM server-to-sensor tunnels. IKE is not supported by the sensors. Do this for all of the sensors. The IPSec Tunnel Groups branch of the Network Policy tree will be populated with an IPSec tunnel group, which consists of the CSPM server and the sensors that will communicate via the IPSec tunnel.
2. Next, you need to configure Manual Keys for each of the sensors and the CSPM server. You must specify a key for each protocol/stage/transform present for each sensor and the CSPM server in the IPSec tunnel group.
3. Generate the Command Sets. This happens when you save and update the configuration in CSPM. The default for publishing command sets is set to manual. You can set CSPM to publish the command set automatically when you save and update.

**WARNING**

You have to disable the setting to automatically update while configuring the IPSec tunnel. If you do not disable the automatic update setting, CSPM will attempt to publish the configuration data to the sensor through the IPSec tunnel before the tunnel configuration is complete on both the CSPM end and the sensor end, generating a publishing error.

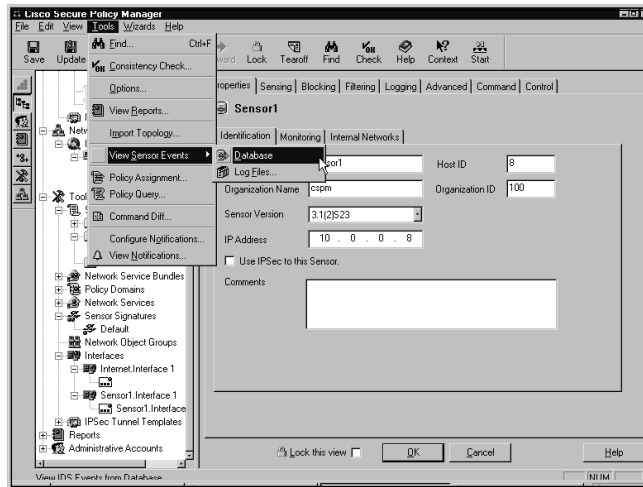
4. Two things can happen here. You either have to restart the Cisco Secure VPN Client, or if the VPN Client has been running during the IPSec tunnel configuration, don't do anything. If the VPN Client is running, the tunnel will not be displayed, even though it is still functioning. If this is the case, stop and then restart the VPN Client for it to be displayed.
5. Next, bootstrap the sensor(s) that will be communicating via the IPSec tunnel. Run through the bootstrapping process and select option **9**, Secure Communications, to configure the sensor for IPSec. Once the sensor is configured for IPSec, you can send data to CSPM and receive signature updates.
6. After the sensor has been bootstrapped and rebooted, you can then publish the command sets to the sensor from CSPM.

Viewing Alarms

Now that you have your sensors and CSPM at the current signature update level, you might want to see what is going on as far as alarms. Cisco is pretty good about tuning some pretty obvious signatures and turning off old signatures that are superseded by newer signatures. But chances are, alarms may abound with a new implementation. Alarms can run into the hundreds and thousands if they are not tuned correctly. So let's take a look at the CSPM Event Viewer and see what is going on.

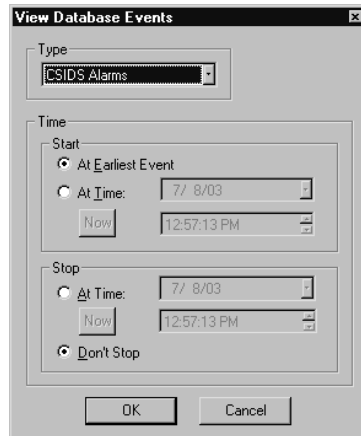
1. Select **Tools | View Sensor Events | Database**. You also have the option to choose Log Files instead of Database if you need to look at some archived records (refer to Figure 4.32).

Figure 4.32 Event Viewer Database



2. Choose **CSIDS Alarms** and click **OK**, as shown in Figure 4.33. Notice you can select certain time frames with a specific start and stop time and date, or have it be continuous.

Figure 4.33 View Database Events

**NOTE**

If you choose to have the alarms logged while you are looking at the event viewer, depending on the amount of alarms being generated, it may be hard to work with. The event viewer continuously refreshes when alarms are generated.

When the Event Viewer opens, it may take a minute depending on how many records are in the database. The event viewer has a default limit of 100,000 records. If the database receives more than that amount, the viewer will only display the first 100,000. You can change the settings on this to increase the limit, but I would not recommend it. With proper tuning of the signatures and alarms, and regular archiving to reduce the logs to a usable size, you should be able to stay under that amount. The viewing screen should look like Figure 4.34 when it opens.

Figure 4.34 Event Viewer

Count	Name	Source Address	Dest Address	Details
5	Net sweep-echo	+		
1	PostOffice Initial Notification	<none>	<none>	postofficed initial notification msg
3	Route Down	<none>	<none>	+
5	Route Up	<none>	<none>	+
1	Traffic Status Up	<none>	<none>	Traffic flow has started for fastethernet interf

Even after the initial install activity is completed, alarms are already being generated. Notice the color coating to the left. You can probably ascertain from the colors the importance of the different alarms. CSPM displays alarms in three categories, low: green, medium: yellow, and high: red. The columns are collapsed initially. To expand the alarms for the different signatures, you can either double-click the count or right-click the row you want to expand and select **Expand | All Columns**. Notice that for the signature Net sweep-echo there is a “+” symbol in the source address column. That tells you there are multiple source addresses for that signature. The expanded view should look like Figure 4.35. Also notice the other alarms are more informational to the administrator and are not associated with intrusion detection signatures. Those can be turned off in the configuration.

Figure 4.35 Event Viewer Expanded View

Count	Name	Source Address	Dest Address	Details	Source Loc	Dest Loc	SubSig ID	Severity
1	Net sweep-echo	10.10.200.4	143.56.31.32	<none>	OUT	OUT	0	Medium
1		10.10.205.23	<none>	-Interval Summary: 1 of total 2 alarms	OUT	OUT	0	Medium
1				-Interval Summary: 1 of total 3 alarms	OUT	OUT	0	Medium
1				-Interval Summary: 1 of total 6 alarms	OUT	OUT	0	Medium
1				-Interval Summary: 2 of total 6 alarms	OUT	OUT	0	Medium
1		24.221.192.5	<none>		OUT	OUT	0	Medium
1	PostOffice Initial Notification	<none>	<none>	postofficed initial notification msg	OUT	OUT	0	Low
1	Route Down	<none>	<none>	5.100 route 1 down	OUT	OUT	0	High
1				8.100 route 1 down	OUT	OUT	0	High
1	Route Up	<none>	<none>	5.100 route 1 up	OUT	OUT	0	Low
1								
1				8.100 route 1 up	OUT	OUT	0	Low
1								
1	Traffic Status Up	<none>	<none>	Traffic flow has started for fastethernet interface	OUT	OUT	1	High

Other viewing options include expanding one column, collapsing one or all columns, moving and deleting columns, selecting columns to be displayed, and also setting event expansion boundaries.

Using the CSID Director for Unix

What is the Cisco Secure Intrusion Detection (CSID) Director for Unix? CSID Director for Unix is another application that you can use to manage your IDS sensors. CSID Director runs on a Solaris or HP-UX platform and has hooks into HP OpenView Network Node Manager (NNM). Without the NNM software, the installation will not succeed. This section assumes you have NNM installed on either a Solaris or HP-UX platform.

Installing and Starting the Director

Very little about working with the CSID Director is simple. You will find that most of the initial setup and commands require a firm grasp of Unix.

To install the Director, follow these steps:

1. Log on to the system you plan to install the CSID Director software onto. You must be root to run this install.
2. Insert the CSID Director install CD into the CD-ROM. Mount the CD-ROM device.
3. Run the install script by typing `/cdrom/cdrom0/install`.

If you are downloading the image, you must first uncompress the downloaded file and then untar the file to a temp directory. After that, you can initiate the install script by typing `./install`.

5. When prompted, enter a password for the netrangr account. The netrangr account is created by default during the installation.
6. Once you have set the password, you will be required to run the `sysconfig-director` utility. Enter `y` when prompted to run the script. The `sysconfig-director` utility has to be run and the configuration completed before running the NNM. The settings in the `sysconfig-director` utility are the same as those for the `sysconfig-sensor` utility discussed in Chapter 3. The settings are shown in Table 4.1.

Table 4.1 *sysconfig-director* Parameters

Field	Input
Director Host ID	1-65535
Director Organization ID	1-65535
Director Host Name	256 alphanumeric characters, no spaces, "-" and "_" are okay.
Director Organization Name	256 alphanumeric characters, no spaces, "-" and "_" are okay.
Director IP Address	Valid IP address
HTML Browser Location	Enter the path to Netscape if the Director does not find it. The install path is /opt/netscape/netscape.

7. The major differences here are that there is no option to add IDS Manager information and you must specify the location of Netscape. Remember, you are on the CSID Director and not the sensor! Once you have entered the required information, type **y** to create the configuration files. You are then prompted to reboot. Type **y** to reboot the system. Once the system reboots, log on to the CSID Director as *netrangr*.
8. From here, you need to start up and configure HP OpenView and configure. First though, make sure all the daemons are running.

Remember in Chapter 3 when we discussed all the commands you can execute from *netrangr*? Specifically, *idsstatus* was used to verify the daemons were running. With the Director, the command is *nrstatus*. Once the *sysconfig-director* utility is run, the following daemons are started:

- *nr.loggerd*
- *nr.postofficed*
- *nr.sapd*
- *nr.configd*
- *nr.filexferd*
- *nr.smid*

Starting the NNM is fairly simple. Execute the following command:

```
ovw &
```

This is one of those times where Unix familiarity comes in handy. The “&” forces NNM to run in the background.

How to Configure the CSID Director

In order to configure the Director, use the NetRanger Configuration File Management Utility, better known as *nrConfigure*. In OpenView, you can launch *nrConfigure* from the Security drop-down menu. This is used to manage the configuration of the Director and sensors. It is similar to CSPM in that you can update configuration files for the Director and sensors, and add and delete sensors and basically manage all aspects of your IDS infrastructure. Once you get *nrConfigure* open, you see the local Director and any sensor that the Director has identified. Each item listed displays three categories of information:

- Organization and Host Name
- Configuration last modified date
- A description of the host

Adding a New Sensor

To add a new sensor use, the Add Host Wizard from the *nrConfigure* menus. Follow these steps:

1. Start the Add Host Wizard from the *nrConfigure* menus.
2. Enter the following Sensor Identification Parameters. Once you have done so, click **Next**:
 - Organization Name
 - Organization ID
 - Host Name
 - Host ID
 - Host IP Address
3. Select the Host Type and click **Next**. You have three options here:
 - Initialize a newly installed Sensor
 - Connect to a previously configured Sensor.
 - Forward alarms to a secondary Director.

For a new sensor, select the first option, **Initialize a newly installed Sensor**. If you are connecting to a sensor that has already been configured, select **Connect to a previously configured Sensor**.

4. Since this is a new sensor, select **Initialize a newly installed Sensor**.
5. Enter the duration for IP blocking and session logging. The defaults are ten minutes. Click **Next**.
 - Number of minutes to log on an event: 1–1440 minutes
 - Number of minutes to shun an event: 1–1440 minutes
 - Network Interface Name.
6. Select the sniffing interface. The different interface types are discussed earlier in Chapter 3.
7. Define the characteristics for blocking/shunning and click **Next**. These include:
 - Router's username/password
 - Router's enable password
 - Router's NAT IP address
 - IP address of sensor from router
 - Router's external IP address
8. At this point, the nrConfigure window displays the sensor under the correct folder. The folder name and the sensor's organization name should be the same. Exit the nrConfigure screen.

If you were to add a sensor that had been previously configured, you would change your selection in step 3 to **Connect to a previously configured Sensor**. You then finish the install by selecting **Finish**. The wizard uploads the configuration file from the sensor to the Director.

To delete a sensor from the nrConfigure screen, highlight the sensor to be deleted, right-click, and select **Delete Host**. Once the sensor is deleted, you remove the icon from nrConfigure by right-clicking the sensor icon to be deleted, and choose **Delete Symbol**.

Configuring & Implementing...

Changing a Sensor's IP Address

To change a sensor IP address, you have to update a sensor's IP address, or change its NetRanger communication infrastructure. The following steps show you how.

1. On the **Director** toolbar, choose **Configure | Security** to open nrConfigure.
2. Double-click the sensor icon you want to reconfigure.
3. Double-click the **System Files** folder. This shows you the current configuration version.
4. Double-click **Routes**.
5. Enter the new IP address and click **OK**.
6. Highlight the new transient version, save and apply the changes. Close.
7. At the sensor, log on as **root** and run **sysconfig-sensor**.
8. Select option **1**, IP address, and enter the new IP address. Exit and reboot the sensor.

Event Processing

Events are forwarded to the Director and translated into alarms. Similar to the other event viewers, they are color-coded red, yellow, and green, for high, medium, and low alarms, respectively.

To view alarms, you have to drill down into the icons. Follow these steps.

1. Double-click the **netranger** icon. The network topology submap opens. The network topology submap contains icons for all the sensors and Directors.
2. Double-click a sensor or Director icon and another submap opens with all the daemons running on that particular device.
3. Select a daemon and double-click. This opens another submap that displays all the events that have been generated by that daemon.

There are several different types of alarms:

- Intrusion Alarms
- Context Buffer Alarms
- Error Alarms
- OkAlarms

When an alarm is sent to the Director, one of the daemons, **nrdirmap**, translates the alarm and presents it in the submap. If multiple alarms from the same signature are sent, they are grouped into alarm sets.

Alarms are labeled with the name of the signature that corresponds to the signature ID. If the signature name cannot be located, then the alarm is labeled with the signature ID itself. The Director utilizes the signatures file in the `/usr/nr/etc/` directory.

Using the IDS Device Manager

If you need to get up and running fast, Cisco's Web-based Intrusion Detection Device Manager, IDM, is the way to go. IDM is by far the easiest of the three IDS Managers to implement. The Web server process runs on the IDS sensor. This is a clue that each IDS sensor is managed independently from one another. You will need to open a web browser for each IDS sensor that you are managing. There is a tool, IDS Event Viewer, you can download from your IDS sensor that allows you to look at more than one sensor's logs from a single graphical interface.

IDM is compatible with the following browsers:

- Netscape (version 4.79 or later)
- Internet Explorer (version 5.5 Service Pack 2 or later)

The browsers can run on an array of operating systems, including:

- Windows NT 4.0 Service Pack 6
- Windows 2000 Professional and Server
- Solaris SPARC version 2.7
- Solaris SPARC version 2.8

NOTE

IDM is not supported on IDS Sensor software prior to version 3.0 and is supported through version 4.0. If you are running IDS Sensor software version 2.2.1 you need to download and install the upgrade image from Cisco.com.

How to Configure IDS Device Manager

When you are bootstrapping the IDS sensor using the *sysconfig-sensor* command, option 6 Communications Infrastructure allows a shortcut. Remember the settings in Figure 3.9? If you are using IDM, you have the option of bypassing all the *IDS Manager Host* information shown earlier. You'll get a message after you set the Sensor IP Address, as seen in Figure 4.36.

Figure 4.36 Configuring IDM in *sysconfig-sensor*



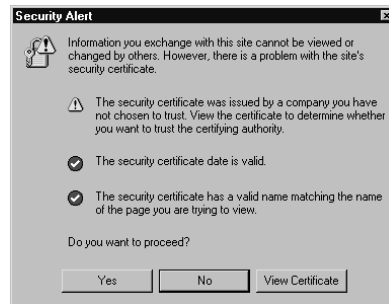
```
C:\WINNT\System32\telnet.exe
Communications Infrastructure
Sensor Host ID[8]:
Sensor Organization Id[100]:
Sensor Host Name[sensor1]:
Sensor Organization Name[lab]:
Sensor IP Address[10.0.0.8]:
Will IDM (WEB based Intrusion Detection Device Manager) be used
to configure the sensor (y/n)? y_
```

If you do not have a separate *Intrusion Detection Device Manager* such as the CSPM or Director solutions implemented, you can stop here and select **y** to let the sensor know you will be using IDM, the *Web-based Intrusion Detection Device Manager*. When the configuration is written, the *cidwebservice* is set to start up on boot.

Logging In

Once you have bootstrapped your sensor, you can log in to IDM. To do this, point your browser towards the sensor by simply typing the IP address in the Address bar in the browser using SSL *https:ip address*. SSL is activated by default. No configuration is required to utilize SSL. The first thing you see is a security alert for the security certificate, as shown in Figure 4.37.

Figure 4.37 Security Alert



It may sound trivial but best practices say you should always verify certificates. It is wise to view the certificate and make sure you are in fact getting the certificate from your sensor and not from somewhere/someone else.

Verifying the Certificate

IDS version 3.1 contains the Web server that runs the IDS Device Manager. Connecting to the IDS Device Manager is done via an encryption protocol called Transaction Layer Security (TLS). To access the IDS Device Manager, you have to enter the URL that starts with *https://ipaddress*. The Web browser serves the IDS Device Manager up by using TLS or SSL to negotiate a session with the host. The IDS Device Manager is enabled by default to use TLS/SSL. It can be disabled from IDS Device Manager by selecting **Device | Sensor Setup | Network**.

The server sends its certificate to the client. The client browser is shipped with a set of trusted Certificate Authority (CA) certificates. The certificate must be validated against the list of CAs, and its URL host name compared with the subject common name.

Follow these steps to verify the certificate:

1. With your browser, enter the sensor IP address and connect to IDM:
https://ip address.
2. You get the Security Alert for the certificate.
3. Select **View Certificate**.
4. The certificate information is shown.
5. Select the **Details** tab.
6. Locate **Thumbprint** and select it.
7. You will see the thumbprint in the corresponding field.
8. Leaving the screen open, connect to your sensor with a console port, SSH, or Telnet.
9. Log in as root.
10. Enter the following command: #
fingerprint[/usr/nr/idsRoot/etc/cert/mytestca.cer]
11. The MD5 fingerprint is displayed.
12. Compare the SHA-1 fingerprint with the value displayed in the open Certificate thumbprint text field. If the fingerprints match, you have validated your certificates' authenticity. If they do not match, you need to find out why.
13. Select the **General** tab.
14. Select **Install Certificate**. The Certificate Import Wizard dialog box appears.
15. Select **Next**. The Certificate Store dialog box appears.
16. Select the location for your certificates.
17. Select **OK** to close the Certificate Store dialog box.
18. Select **Yes** to open the IDS Device Manager.

Once you have validated and installed the certificate, the next dialog box prompts you to log in as shown in Figure 4.38. In order to properly configure and manage your IDS sensors, use netrangr.

Figure 4.38 Password Screen

Never save the password in the password list. You do not want an unauthorized user gaining access to your IDS sensor management console and modifying any of the settings. With access to the management console, an unauthorized user can make whatever changes to the configuration he wants, potentially disabling the sensors or reconfiguring the sensor so no alarms are issued during their attack. The IDS Device Manager console is shown in Figure 4.39.

Figure 4.39 IDS Device Manager Console

Configuring the IDS Device Manager

The IDS Device Manager is probably the easiest management tool to use. The installation is relatively painless and the price is right. It comes installed on the sensor and is free with the purchase of the sensor software. You will see that just about everything you did with the other management applications, you can do with IDM. The only exception to this is that you can only configure one sensor at a time—the one you are logged on to. For most of us though, the graphical interface is familiar territory and easy to maneuver through.

There are four tabs located at the top of the screen just underneath the application name. The tabs are named Device, Configuration, monitoring, and Administration. You'll use these tabs to make any configuration changes, tune signatures, view current and archived logs, and perform other tasks. Note at the very top right-hand side of the screen, buttons for Logout, Apply Changes, Help, NSDB, and About.

The Device Tab

The Device tab allows you to make some basic configuration changes to your sensors. Just under the tab is a menu bar with one option, Sensor Setup. Here, you can make basic sensor modifications, like those done in `sysconfig-sensor`. You can also configure SSH, set the time, and change passwords (see Figure 4.40).

Figure 4.40 The Device Tab



To make network changes, follow these steps:

1. Select **Network**. Figure 4.41 displays these settings, which are similar to those in `sysconfig-sensor`, with a few added fields.

Figure 4.41 Network Settings

Host Name	sensor1
Host ID	8
Organization Name	lab
Organization ID	100
Post Office Port	45000
Route Up Alarm Level	Information
Route Down Alarm Level	High
Heartbeat Interval Multiplier	3
IP Address	10.0.0.8
Netmask	255.255.255.0
Default Route	10.0.0.10
Enable TLS/SSL	Yes <input type="checkbox"/> Default Port
Web Server Port	443

Information
Complete the fields to specify the network and IDS communication parameters for this host. Click the Reset button to reset the form to the values that were present when the form was opened.

2. Make changes to your sensor configuration in this screen. Ensure your Host ID is unique and your organization name and ID match the rest of your IDS infrastructure.
3. The PostOffice Port defaults to port 45000. In the Route Up and Route Down Alarm Level boxes, select how you want the two alarms to be displayed in your event viewer. The route going down defaults to high. This is fairly important and should catch your attention. The route coming back up may be less important so it is marked as informational, as shown in Figure 4.42, and may not even be displayed on your event viewer, depending on your configuration.

Figure 4.42 The Alarm Level

Host Name	sensor1
Host ID	8
Organization Name	lab
Organization ID	100
Post Office Port	45000
Route Up Alarm Level	Information
Route Down Alarm Level	Information
Heartbeat Interval Multiplier	3
IP Address	10.0.0.8
Netmask	255.255.255.0
Default Route	10.0.0.10
Enable TLS/SSL	Yes <input type="checkbox"/> Default Port
Web Server Port	443

Information
Complete the fields to specify the network and IDS communication parameters for this host. Click the Reset button to reset the form to the values that were present when the form was opened.

4. Select the **Heartbeat Interval Multiplier**. The Heartbeat is the number of seconds between queries for PostOffice services. Enable or disable TLS/SSL. TLS/SSL is on by default and the port is 443.
5. Click **Allowed Hosts**.
6. On the screen, you can add specific IP addresses or entire networks that can access the sensor (see Figure 4.43). Try to be as specific as possible. Least privilege is a good practice when giving access.

Figure 4.43 Allowed Hosts



NOTE

The idea of least privilege is quite simple in definition but rather difficult when put into practice. It requires that a user be given only the necessary privileges to perform a job. First, the user's job is identified and a minimum set of privileges is associated with the job function, thus allowing the user to perform the job with those privileges and nothing more.

7. Select **Remote Access**. In this window, you can specify whether to allow or disallow FTP or Telnet. Make your selection and click **OK**.
8. Select **SSH | Host Key** to generate a new host key.

9. Click **Generate Host Key** and the system generates a new key, replacing the old one. The changes take effect once applied. Do not forget to update the fingerprint on remote systems (see Figure 4.44).

Figure 4.44 Generating a Host Key



10. Select **Time** to modify the system time.
 - Version 3.1 allows you to modify the time, date, and time zone.
 - Version 4.0 provides more granularity allowing changes to time, date, time zone, UTC settings, NTP settings, daylight savings, and the duration of daylight savings.
11. Select **Password** if you need to change the passwords for the accounts root or netnangr.
12. Once you have completed any sensor configurations, select **Finished**.

The Configuration Tab

In the Configuration tab, you can configure the sensing engines: Communications, Logging, and Blocking. You also have the option to restore default settings in this screen. Take your time and hopefully you won't have to restore defaults. Keep in mind, signature tuning is time-consuming.

The sensing engine configuration is for tuning and enabling/disabling signatures. You can tune all of the signatures, specify the level of traffic, what port is used, even filter certain signatures that you do not want to see. You can also configure IP Fragmentation Reassembly Options (see Figure 4.45).

Figure 4.45 Sensing Engine: Configuration



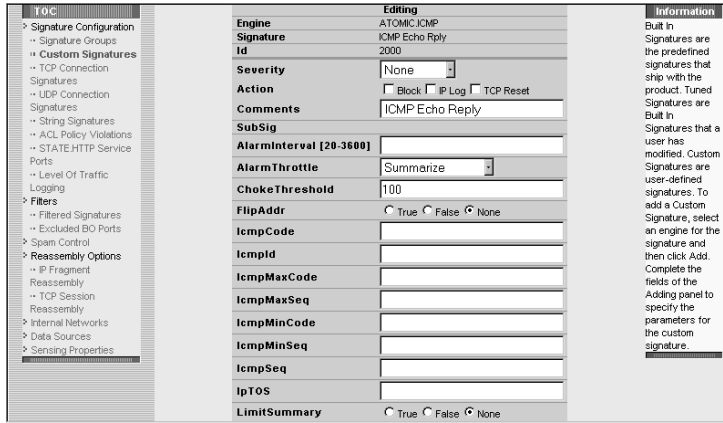
Notice that the different types of signatures are represented in groups on the screen. They have circles next to them that are either clear, half-filled, or filled. The clear circle means that none of the signatures in that specific group are enabled. The half circle means at least one signature is enabled, while a full circle means that all of the signatures are enabled.

If you want to enable all of the signatures in a certain group, put a check in the box next to the group, and click **Enable** at the bottom of the screen. To disable all of the signatures in a group, put a check in the box next to the group, and click **Disable**.

To configure or tune a signature, follow these steps:

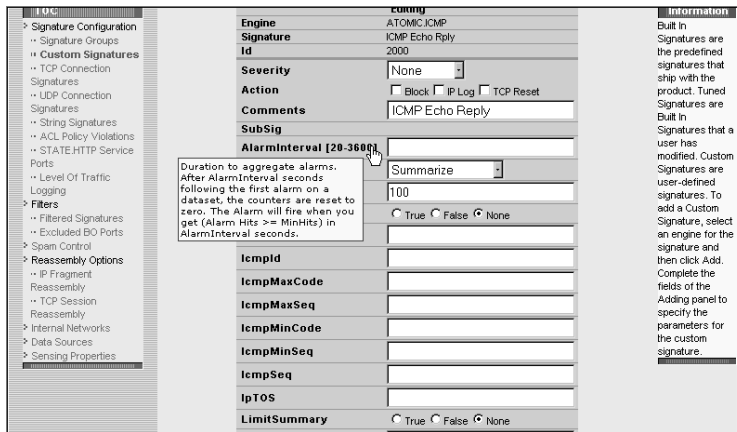
1. Select a signature group. The screen should display all of the signatures in that group. If there is more than a single screen of signatures, scroll to the bottom of the screen and select the signature IDs to move to.
2. Once you have selected the signature to tune, click the little notepad icon next to the signature name. You should get a screen similar to that shown in Figure 4.46.

Figure 4.46 Tuning a Signature



3. Make the changes necessary to meet the requirements in your security policy. If you move your cursor over the field name, it will tell you what needs to be entered in the field next to the name (see Figure 4.47).

Figure 4.47 Signature Fields



4. Once you have tuned all of your signatures, use the Apply Changes button to have them implemented.

The Remote Hosts screen in the Configuration tab is used to specify hosts that receive events from the sensor. (Refer to Figure 4.48.)

Figure 4.48 Remote Hosts



The Event Logging screen in the Configuration tab is to define at what level an event gets logged, as well as what type of alarms are logged (see Figure 4.49).

Figure 4.49 Event Logging



The Blocking screen is used to configure blocking and shunning. Be extremely cautious when configuring blocking. You do not want to deny access to a customer, client, or business partner (see Figure 4.50).

Figure 4.50 Blocking Configuration

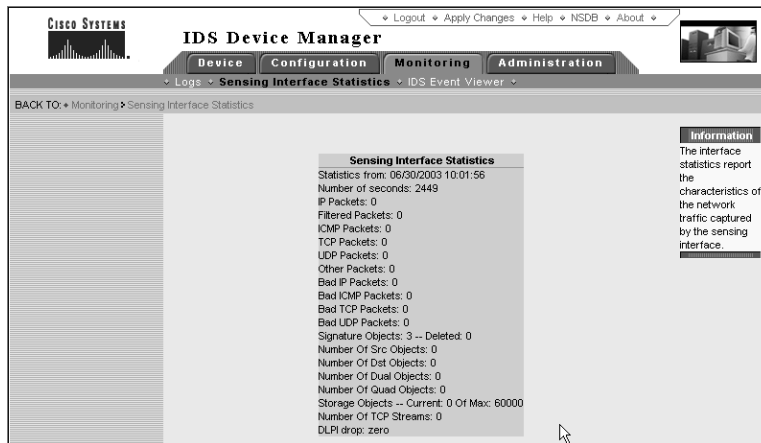


The Restore Defaults screen does exactly what it says. It sets all of your configurations back to factory defaults.

The Monitoring Tab

In the Monitoring tab you have the ability to view logs, interface statistics, and download the event viewer. To view interface statistics, simply click **Sensing Interface Statistics**. It may take a few moments for the statistics to be displayed. The display resembles Figure 4.51.

Figure 4.51 Sensing Interface Statistics



To view event logs, follow these steps:

1. Select **Logs**. Here you can view Error and Command Logs, IP Session Logs, Current and Archived Event Logs, and System Messages.
2. Choose **Current Events**. The resulting output should resemble that shown in Figure 4.52.

Figure 4.52 Log Output

```

3,9,2003/06/24,20:16:08,2003/06/24,14:16:08,10000,8,100,20012,8,100,GET
3,2,2003/06/24,20:16:08,2003/06/24,14:16:08,10005,8,100,20013,8,100,GET
3,2,2003/06/24,20:16:08,2003/06/24,14:16:08,10007,8,100,20014,8,100,GET
3,2,2003/06/24,20:16:08,2003/06/24,14:16:08,10010,8,100,20015,8,100,GET
3,29,2003/06/24,20:16:08,2003/06/24,14:16:08,10008,8,100,20016,8,100,GE
3,10,2003/06/24,20:16:18,2003/06/24,14:16:18,10000,8,100,20017,8,100,GE
3,11,2003/06/24,20:16:18,2003/06/24,14:16:18,10000,8,100,20018,8,100,GE
3,3,2003/06/24,20:16:18,2003/06/24,14:16:18,10005,8,100,20019,8,100,GET
3,3,2003/06/24,20:16:18,2003/06/24,14:16:18,10007,8,100,20020,8,100,GET
3,3,2003/06/24,20:16:18,2003/06/24,14:16:18,10010,8,100,20021,8,100,GET
3,30,2003/06/24,20:16:18,2003/06/24,14:16:18,10008,8,100,20022,8,100,GE
3,12,2003/06/24,20:17:04,2003/06/24,14:17:04,10000,8,100,20023,8,100,GE
3,13,2003/06/24,20:19:23,2003/06/24,14:19:23,10000,8,100,20024,8,100,GE
4,1000028,2003/06/24,20:21:45,2003/06/24,14:21:45,10008,8,100,OUT,OUT,5
4,1000029,2003/06/24,20:23:15,2003/06/24,14:23:15,10008,8,100,OUT,OUT,5
3,14,2003/06/24,20:25:46,2003/06/24,14:25:46,10000,8,100,20025,8,100,GE
4,1000030,2003/06/24,20:30:05,2003/06/24,14:30:05,10008,8,100,OUT,OUT,5
4,1000031,2003/06/24,20:31:36,2003/06/24,14:31:36,10008,8,100,OUT,OUT,5

```

The full output would look something like this:

```

3,10000030,2003/06/16,20:30:36,2003/06/16,14:30:36,10008,8,100,OUT,OUT,
5,2001,0,TCP/IP,192.168.2.5,10.0.0.32,0,0,0.0.0.0,

```

It's not very easy to read, but that's what the Event Viewer is for. It translates it all into easy-to-read records. We'll discuss the Event Viewer shortly. Table 4.2 describes each field in the .csv log file.

Table 4.2 Log File Field Values

Field Value	Field Type
3	Record Type
10000030	Record ID
2003/06/16	GMT Date Stamp
20:30:36	GMT Time Stamp
2003/06/16	Local Date Stamp
14:30:36	Local Time Stamp
10008	Application ID

Continued

Table 4.2 Log File Field Values

Field Value	Field Type
8	Host ID
100	Organization ID
OUT	Source Direction
OUT	Destination Direction
5	Alarm Level
2001	Signature ID
0	SubSignature ID
TCP	Protocol
192.168.2.5	Source Address
10.0.0.32	Destination Address
0	Source Port
0	Destination Port
0.0.0.0	Router Address

With this in mind, let's run through downloading the Event Viewer from IDM so we can look at events in a format that's a little easier on the eyes. To download the Event Viewer, follow these steps:

1. From the Monitoring tab, select **IDS Event Viewer** from the menu bar. The screen should have a couple of links to choose from (see Figure 4.53).

Figure 4.53 The Event Viewer Download Screen

2. Click the **Event Viewer Readme** link and review the signature updates and features.
3. Click the **Download the Windows NT/2000 IDS Event Viewer** link. This will initiate the download process to your workstation.
4. If there are signature updates, the link will be highlighted for you to download. Download if necessary.
5. Close this screen. (We discuss installing and configuring it later.)

The Administration Tab

In the Administration tab, you can configure automatic updates, view system information, run diagnostics on your sensor, set up severity levels for events, and start and stop processes. The two most useful options here are viewing system information and setting up automatic updates. To view the system information, click **System Information** in the menu bar (see Figure 4.54). This gives you the basic information necessary for troubleshooting. The following information should appear on the screen:

- Sensor Version
- Host Name
- Host ID
- Organization Name
- Organization ID
- PostOffice Port
- Web Server Port
- IP Address
- Netmask
- Default Route
- CSIDS Daemon Status—Displays running daemons
- CSIDS Connection Status—Displays the PostOffice connection status
- CSIDS Version—Displays daemon versions
- Administrative Tasks
- MAC Address

- Hardware
- Operating System
- CPU usage
- Memory usage (in MB)
- CSID Logging Disk Space Usage (in MB)
- TAC

Figure 4.54 System Information

Sensor Version	3.1(2)S23
Host Name	sensor1
Host Id	9
Organization Name	lab
Organization Id	100
Post Office Port	45000
Web Server Port	443
CIDS Daemon Status	netrangr 411 1 0 10:01:45 ? 0:00 ./usr/hr/bin/hr.fileXferd netrangr 410 1 0 10:01:42 ? 0:00 ./usr/hr/bin/hr.loggerd netrangr 352 1 0 10:01:35 ? 0:01 ./usr/hr/bin/hr.postofficec netrangr 412 1 0 10:01:49 ? 0:00 ./usr/hr/bin/hr.sapd netrangr 415 1 0 10:01:53 ? 0:08 ./usr/hr/bin/hr.packetd
CIDS Connection Status	Connection Status for sensor1.lab IDS-Mgr.lab Connection 1: 10.0.0.4 45000 1 [Established] sto:0002 with Version 1
CIDS Version	Application Versions for sensor1.lab The Version of the Sensor is: 3.1(2)S23 postoffice v220 (Release) 01/12/14-20:01 logger v220 (Release) 01/12/14-19:59 sap v220 (Release) 01/12/14-20:01 fileXfer v175 (Release) 01/07/11-21:48 sensor v262 (Release) 02/05/08-17:28
IP Address	10.0.0.8
Netmask	255.255.255.0
Default Route	10.0.0.10
MAC Address	0:d0:b7:88:8f:e2
Hardware	i86pc
Operating System	SunOS Generic_108529-14.5.8
CPU Usage	0%
Memory Usage (Mb)	Used: 104 Remaining: 152 (Total: 256)
CIDS Logging Disk Usage (Mb)	Used: 5 Remaining: 4260 (Total: 4265)
TAC Link	http://www.cisco.com/public/support/fac/home.shtml Phone: 1(800)553-2447

To configure automatic updates, follow these steps (see Figure 4.55):

1. In the **Administration** tab, select **Update** from the menu bar.
2. Enter the IP address of the FTP server in the FTP Server Field.
3. Enter the user account that will be used to connect to the FTP server in the Username field.
4. Enter the password of the user account in the Password field.
5. Enter the path (location) of the update files. Use a “/” at the beginning of the path.
6. Select the **Disabled** check box.
7. Select the **Performed at** check box and enter the times to check for updates.
8. Click **OK**.

Figure 4.55 Updates

There is also a Diagnostics option in the Administration tab. This is mainly used by the Cisco TAC personnel for troubleshooting. To run the diagnostics from the Administration tab, click **Diagnostics**, then click **Run Diagnostics** (see Figure 4.56). A diagnostics report will be displayed on the screen. Once you have run at least one diagnostics report, you will have the option of viewing the last diagnostics report by clicking **View Last Report**.

Figure 4.56 Diagnostics

Lastly, remember that after you make any changes in IDM, you must always apply the changes. To apply the changes you have made to the sensor, click the

Apply Changes button in the upper right-hand corner of the IDM screen. It may take some time, but when the changes are complete you will get a success message. Once you have made all of your configuration changes to IDM and your sensors, click **Logout** located next to the Apply Changes button.

Using the Cisco Network Security Database

The Cisco Network Security Database, or NSDB as it is commonly referred to, is Cisco's version of a security vulnerability database. The entries in the NSDB correspond with an event or a signature in the IDS. When researching and investigating alarms, the NSDB is used to make sense of what is going on within your enterprise.

Each IDS Management Console accesses the NSDB in the same manner. In order for you to access the NSDB entry for a signature, perform the following steps:

1. Access the events in the Event Viewer for IDM or CSPM or drill down to the event in the Director. You can either view the live database or a log file.
2. Select the record you want information about.
3. Right-click the record and select **NSDB**.
4. The NSDB will open in a Web browser with information about the signature in question (see Figure 4.57).

Figure 4.57 The NSDB Screen

The screenshot shows the Cisco NSDB interface for an "Exploit Signature". The main content area displays the following information:

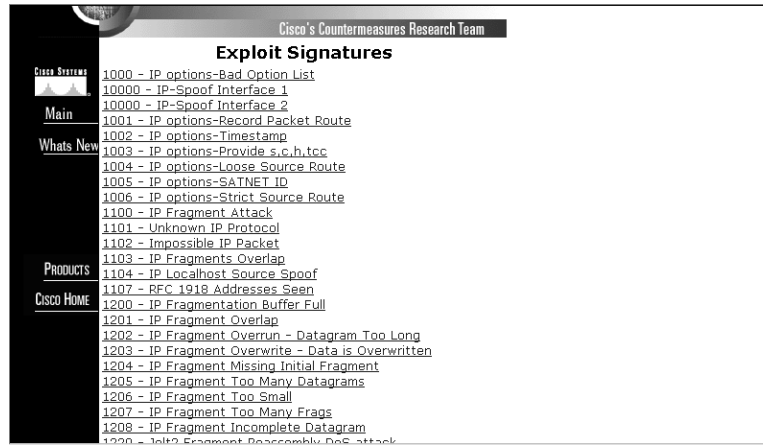
- Signature Title:** IP Fragment Attack
- ID:** 1100
- Sub ID:** 0
- Recommended Alarm Level:** 3
- Signature Type:** NETWORK
- Signature Structure:** ATOMIC
- Implementation:** CONTEXT
- Release Version:** 1.0
- Description:** Triggers when any IP datagram is received with an offset value less than 5 but greater than 0 indicated in the offset field. This indicates that the preceding fragment(s) was/ (were) unusually small, and is most likely an attempt to defeat packet filter security policies.
- Benign Trigger(s):** IP datagrams may be fragmented normally as they are transported across the network, but they will normally not be fragmented into sizes smaller than 256 bytes. Investigation of this traffic is especially important if the network is protected by a packet filtering firewall.
- Data Field Information Tag:** None
- Related Vulnerabilities:** [704](#)
- User Notes:** [User Notes Page](#)

The interface includes a left-hand navigation menu with options: Main, Whats New, PRODUCTS, and CISCO HOME. The top of the window has a "Cisco Systems" logo.

If there are related vulnerabilities for a particular signature, there will be links to those vulnerabilities.

You can view the entire database by clicking the **Main** link in the left pane. This offers a numerical list of all the signatures currently in the database (see Figure 4.58).

Figure 4.58 NSDB Main Menu



NOTE

If you are using the Director, you have to specify a browser preference to access NSDB. Open **nrConfigure**, select **Preferences** from the **File** menu and enter the path to the browser, then click **OK**.

Summary

As you can see there is a ton of information to absorb regarding management of sensors. Instead of a single method, Cisco presents three different ways to get the job done, CSPM, Unix Director, and IDM. Of the three, IDM is the easiest and quickest to get up and running. The Director is the hardest, while CSPM fits somewhere in the middle as the most commonly used solution.

We have gone through the installation of CSPM, the Director, and IDM. CSPM is quite finicky when it comes to software requirements, so make sure you have everything installed and on hand before you get started. It will save you some headaches. The Director is a monster of a system. If you do not have thorough knowledge of Unix and HP OpenView, I'd recommend looking into one of the other products. IDM is, of course, the easiest and cheapest way to manage the sensors, but keep in mind that some of the functionality is limited. You only have the option to configure one sensor at a time, whereas CSPM lets you make changes to a single signature file template and push those changes to multiple sensors.

Shunning requires coordination between both the security and networking teams. Access must be granted from the sensors to the devices doing the blocking. If you are going to configure your sensors to shun or do TCP resets, make sure you brief management on what it is and what it does. You may inadvertently deny access to customers and business partners to your resources. This can be a costly mistake. Check with Cisco to make sure your devices can be managed by the sensors before attempting to implement.

Solutions Fast Track

Managing the IDS Overview

- ☑ There is three different methods for managing Cisco IDSs: CSPM, Unix Director, and IDM.
- ☑ The goal of these solutions is to provide a central location for managing and monitoring IDS Sensors.
- ☑ Unix Director runs on a Solaris or HPUX Platform.
- ☑ IDM is a Web-based solution that comes with the sensor software.
- ☑ CSPM is the most commonly used solution for managing Cisco IDS sensors.

Using the Cisco Secure Policy Manager

- ☑ CSPM has specific software requirements when installing. These include the following:
 - NT 4.0
 - Service Pack 6a
 - IE 5.5
 - HTML Help 1.32 Update
 - MSXML3
- ☑ The PostOffice parameters must be correctly configured in order to properly install CSPM.
- ☑ A network must be defined first before you can add any hosts to the topology.
- ☑ The network parameters do not have to be exact. The communication parameters were previously configured on the sensor.
- ☑ When adding previously configured sensors, you will want to capture the configuration. In the Add Sensor Wizard, check the box on the first screen to capture the configuration.
- ☑ In order to push configuration changes to the sensor, you have to first save and update CSPM and then select the sensor you are updating. Choose the **Command** tab and click **Approve Now**.

Using the CSID Director for Unix

- ☑ The Director needs HP OpenView Network Node Manager (NNM) to run.
- ☑ The NetRanger Configuration File Management Utility (nrConfigure) is used to configure the sensors and the Director.
- ☑ To view the alarms, you have to drill down to them by double-clicking the Netranger icon, and then the daemon. The alarms will be displayed for the daemon that generated the event.
- ☑ You can only add one sensor or host at a time.
- ☑ To verify daemons are running on the Director, type **nrstatus**.

- ☑ The command to start HP OpenView is *ovw &*. The “&” forces OpenView to run in the background.

Using the IDS Device Manager

- ☑ IDM is the easiest management solution to install. It is installed when the sensor software is loaded on the sensor.
- ☑ The drawback to IDM is that you can only configure/manage one sensor at a time.
- ☑ Event Viewer software can be downloaded from IDM to better view the log files.
- ☑ Changes do not take place on the sensor until you have clicked the **Apply Changes** button in the upper right-hand corner of the IDM screen.

Using the Cisco Network Security Database (NSDB)

- ☑ The Network Security Database (NSDB) contains a description of each signature loaded on to a sensor.
- ☑ To view the description, right-click the record or icon of the alarm, then select **NSDB**.
- ☑ If there are related vulnerabilities, the page will provide links to them.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: What is the only version of the Windows Operating System that CSPM can be loaded on?

A: Windows NT 4.0

Q: What are the names of the eight tabs used to configure parameters on your sensors?

A: Properties, Sensing, Blocking, Filtering, Logging, Advanced, Command, Control

Q: What do you have to do in order to push changes from CSPM to the sensor?

A: You have to first save and update CSPM, then select the sensor you want to update. Access the **Command** tab and click **Approve Now**.

Q: Where are advanced PostOffice settings configured?

A: Highlight the sensor you want to configure. Choose the **Advanced** tab, then select the **PostOffice** subtab.

Q: What is the purpose of the PostOffice Heartbeat Interval?

A: The PostOffice Heartbeat Interval is the amount of time in seconds that a query is sent by PostOffice to a remote PostOffice to ensure they are communicating. The default is five seconds.

Q: What are the six parameters that can be set in the Watchdog Properties?

A: Watchdog Interval, Watchdog Timeout, PostOffice Heartbeat Interval, Number of Restarts, Daemon Down Alarm Level, and Daemon Unstartable Alarm Level

Q: What type of platform must CSID Director be loaded on?

A: Solaris or HP-UX

Q: What are the three host types that can be added in the Director?

A: A newly installed sensor, a previously configured sensor, or a secondary Director for alarm forwarding.

Q: What is the first account created during the Director installation?

A: netrangr

Q: After you have set the netrangr password during the CSID Director installation, what is the command you execute to initially configure communications parameters?

A: *sysconfig-director*. This command allows you to configure the Director Host ID, Director Organization ID, Director Host Name, Director Organization Name, Director IP Address, and HTML Browser Location.

Configuring the Appliance Sensor

Solutions in this Chapter:

- Configuring SSH
 - Configuring Remote Access
 - Applying the Sensor Configuration
 - Configuring Logging
 - Upgrading the Sensor
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Once the Cisco Network IDS appliance sensor has been installed, the next step before deployment of the sensor is configuration. The installation of the sensor software (whether by Cisco before shipping to the customer or through the upgrade process) leaves the appliance with specific default settings that are unsuitable for production deployment. This chapter covers the configuration and use of Secure Shell (SSH) for remote access and management, the application of new configurations to the sensor, and how to configure logging on the sensor. Secure shell has been the method of choice for accessing the command line interface (CLI) of the appliance since early versions of the IDS software. This is because Secure Shell provides the administrator the capability of establishing a secure communication channel with the sensor.

This chapter covers the initial configuration of the sensor appliance through the console interface as well as how to configure the appliance sensor using the command line interface through Secure Shell, configuring for remote access to the sensor, applying the modified sensor configuration to the device, logging, and how to upgrade the IDS sensor software and signature pack. Up-to-date signature packs are critical to the value of the IDS within the overall framework of security in the network. Without up-to-date signature packs, the sensor will not be able to detect newer exploits and attacks.

Logging allows the development of a baseline for alarms that may be detected on the network. These alarms may well represent benign traffic that the IDS sensor misinterprets as possible attacks—termed “false alarms.” Signature tuning can reduce the number of false alarms generated by the sensor, leaving only valid alarms that require investigation.

Configuring SSH

Secure Shell (SSH) is a protocol that provides a secure and encrypted connection between a client and a host. It uses TCP port 22 for all communication. SSH provides a method of providing secure and encrypted communications for such diverse protocols as X-Windows, Telnet, rlogin, and others. For the purposes of configuring the Cisco IDS sensors in this discussion, it will be used as a replacement for Telnet.

There are two different versions of SSH at this time, version 1 (SSH-1) and version 2 (SSH-2) and they are not compatible. The differences in the protocol are significant. The SSH-1 protocol is monolithic and encompasses a variety of

functions within this single protocol. SSH-2 consists of three protocols that work together in a modular form. These protocols are:

- SSH Transport Layer Protocol (SSH-TRANS)
- SSH Connection Protocol (SSH-CONN)
- SSH Authentication Protocol (SSH-AUTH)

Each of these protocols is specified in separate Internet drafts and are available from the Secure Shell (secsh) working group's section of the IETF Web site (www.ietf.org). A fourth Internet draft discusses the overall architecture of the SSH-2 protocol (SSH Protocol Architecture). Most Cisco products only support SSH-1. While there are known vulnerabilities in the SSH-1 protocol, it still provides a significantly more secure communication channel than using plaintext Telnet. Furthermore, even with these known vulnerabilities, the SSH-1 protocol provides a substantial hurdle for an attacker to overcome in order to gain access to the communication data stream.

Whether the IDS sensor was a new purchase or an upgrade to a currently deployed and supported IDS appliance, the first step that must be completed is an initial configuration of the device. This is achieved either by connecting a keyboard, mouse, or monitor to the device or by connecting to the device through a serial console. The initial configuration of the IDS was covered in a previous chapter. For the purposes of this discussion, it is assumed that the IDS sensor has been configured with a hostname of sensor as well as an IP address of 192.168.50.51 and a subnet mask of 255.255.255.0 or /24.

This section focuses on connecting into the IDS sensor and performing the initial configuration through the serial console. The back panel configurations for the IDS-4215 and the IDS-4235/4250 appliances are shown in Figures 5.1 and 5.2, respectively. Both the 4215 and the 4235/4250 models have serial console ports located on the back panel. The command and control interface for every IDS sensor appliance is the int1 interface.

Figure 5.1 IDS-4215 Back Panel

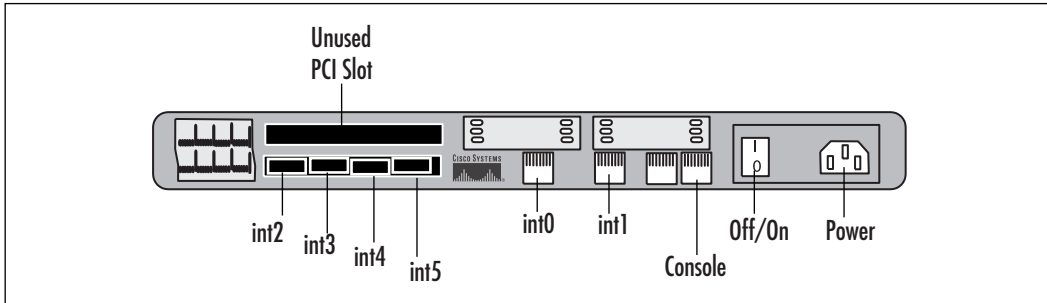
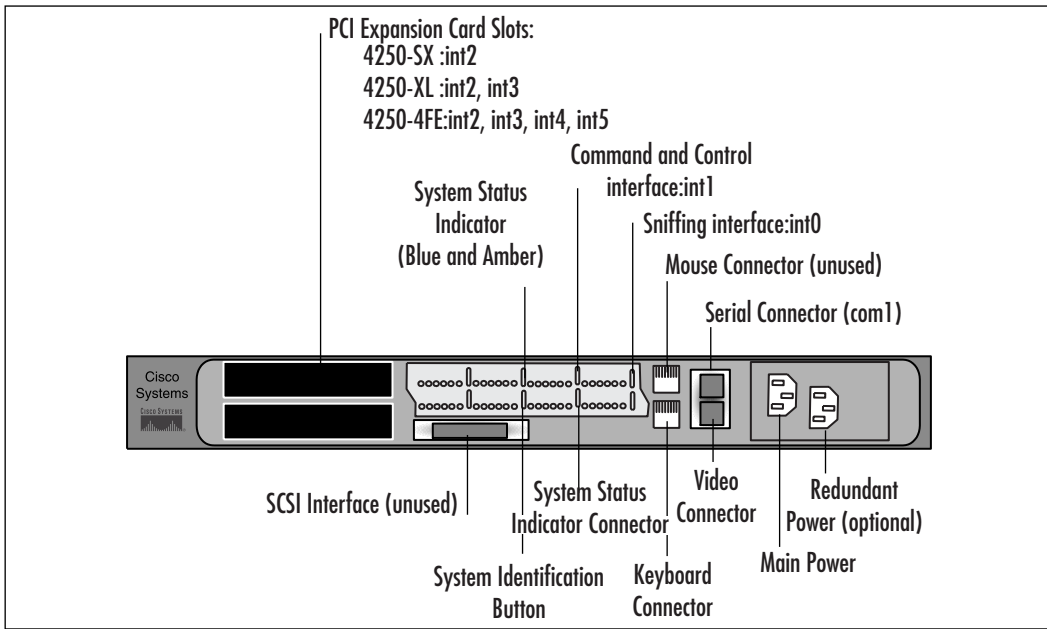


Figure 5.2 IDS 4235/4250 Back Panel



The procedure to connect to the serial connector on the back of the IDS sensor appliance is as follows:

For the IDS-4215:

1. Connect a nine-pin serial RJ-45 adapter (also known as the M.A.S.H.) to the back of a computer.
2. Using the rolled cable supplied with the IDS sensor, connect one end of the cable to the RJ-45 console port on the IDS and the other end into the M.A.S.H adapter. If a terminal server is being used for serial port

access, connect the other end of the rolled cable to one of the ports on the terminal server.

The serial port on the computer should be configured as shown in Table 5.1.

Table 5.1 Serial Port Settings for an IDS Console

Parameter	Setting
Baud Rate	9600
Data	8 bit
Parity	None
Stop	1 bit
Flow Control	Hardware or RTS/CTS

For the IDS-4210/4235/4250:

1. Connect the M.A.S.H. to the COM1 port on the back of the IDS sensor.
2. Connect one end of the 180/rolled cable supplied with the IDS sensor to the RJ-45 port of the M.A.S.H. Connect the other end either to a port on a terminal server (as discussed earlier) or to the RJ-45 port of a M.A.S.H. attached to a computer. If a computer is being used to provide a serial connection to the IDS sensor, the serial port settings should be set to the values shown in Table 5.1.

Once the serial connection to the IDS has been established, access to the IDS “console” is now possible. For the purposes of this discussion, it will be assumed that the IDS serial port is connected to a terminal server.

To connect to the serial port of the IDS sensor, simply Telnet to the proper port on the terminal server, as shown in Figure 5.3.

Figure 5.3 Telnet Server Access to IDS Sensor Serial Console

```
#####
This system is for authorized users only
All users will have their activities monitored and recorded
by the security personnel.
#####
User Access Verification
Username: user-1
```

Continued

Figure 5.3 Telnet Server Access to IDS Sensor Serial Console

```
Password: *****
```

```
Ciscoids-1
```

```
Ciscoids-1: login:
```

Cisco IDS Software v3

To configure Secure Shell under IDS software version 3.0 and 3.1, log in to the sensor appliance as **root**. Once logged into the sensor, the *sysconfig-sensor* utility can be used to configure and start up Secure Shell.

1. Log in to the sensor as **root**.
2. Start the *sysconfig-sensor* utility. A text-based menu will be displayed providing various options as shown next:

```
Cisco IDS Sensor Initial Configuration Utility
Select options 1 through 10 to initially configure the sensor.
1 - IP Address
2 - IP Netmask
3 - IP Host Name
4 - Default Route
5 - Network Access Control
6 - Communications Infrastructure
7 - Date/Time and Time Zone
8 - Passwords
9 - Secure Communications
10 - Display
x - Exit
Selection:
```

3. Select option **9** on the menu. This opens the Secure Communications sub-menu, shown next.

```
Secure Communications
1 - IPSec Communications
2 - Secure Shell Communications
x - Exit
Selection:
```

4. Select option **2** in the Secure Communications submenu to configure Secure Shell.

```
Secure Shell Communications
1 - Security Level (currently LOW)
2 - Manage Secure Shell Known Hosts
3 - Host Key Operations
x - Exit
Selection:
```

5. Select option **1** to change the security level of the sensor. By default, the security level is set to 3 (Low), which allows Secure Shell, Telnet, and FTP access to the sensor.

```
Security Level
## The Sensor always provides Secure Shell services (including
## scp). Increase the security of the Sensor by disabling two
## services that allow clear text password authentication:
## Telnet and FTP. For maximum security disable both.
The current setting is LOW.
```

```
Select the new security level:
```

```
1 - High (Telnet and FTP disabled)
2 - Medium (Telnet disabled)
3 - Low (insecure services available)
x - Exit
Selection:
```

6. Select options **1, 2, or 3**. It is *highly* recommended that the sensor's security level be set to **1** because of the role of the IDS sensor in the overall network security architecture. Once the security level has been set, select **x** to exit the Security Level sub-menu.
7. Select option **3** in the Secure Shell Communications menu. This displays the Host Key Operations sub-menu.

```
Host Key Operations
The system has a host key with fingerprint: 1024
6c:00:fa:53:5b:16:83:24:6e:f0:f4:68:21:22:bd:7c root@CISCO_IDS
```

Select an option:

- 1 - Delete host key and generate a new one
- 2 - Delete host key
- 3 - Exit

Selection:

8. Select either **1** to delete the current host key and generate a new one, or **2** to simply delete the current host key. Changing the host key may result in difficulty in connecting to the SSH server on the IDS sensor. SSH clients cache the host key of the servers that they connect to. When the client connects to an SSH server, it compares the host key of the server to the one stored in the cache. A change in a server's host key may indicate a problem. Either the host key was changed by an administrator or the client is connecting to a host that may be impersonating the server (a man-in-the-middle attack). In the case of a server host key that was re-created by an administrator, the old host key should be cleared out of the client's cache so that the new key will be written in its place.
9. Once the host key has been generated, exit out of the Secure Communications submenus by selecting **x** until the main menu of the configuration utility has been reached.

Cisco IDS Software v4.0

IDS software v4.0 and later changed the way the administrator managed the IDS sensor. With their release, Cisco switched the underlying operating system from Solaris 8 to Red Hat Linux 8. Additionally, IDS 4.0 provides an “IOS-like” command line interface to configure the IDS sensor appliance. Like IOS, the command line interface for the IDS 4.0 software is broken down into submenus that the administrator must use to configure various features in the IDS sensor. The default administrative account username/password combination for Cisco's IDS software 4.0 and later is: *Cisco /Cisco*. Cisco Systems developers realized the weakness of this username/password combination and required that the default password for the *Cisco* account be changed upon first login. Once the default password for the *Cisco* account has been changed, the user is logged in and the command line shell is started.

In order to have the proper time and date stamp placed on your log files, and for various security certifications to work properly if they are time-based, we

need to configure the sensor to have the correct time and maintain that time. The following steps, shown in Figure 5.4, easily accomplish this:

Figure 5.4 Configuring the Sensor's Time

```
sensor# clock set 20:32:00 September 27 2003
sensor# config t
sensor(config)# service host
// This is where we enter the time parameters mode
sensor(config-Host)# timeParams
// We need to adjust the offset from UTC in minutes
sensor(config-Host-tim)# offset -480
// Now we specify the standard time zone
sensor(config-Host-tim)# standardtimezone PST
// We enter the summer time parameter configuration mode
sensor(config-Host-tim)# summertimeparams
// Now we specify the summer time parameters that recur each year
sensor(config-Host-tim-sum)# active-selection recurringparams
// Enter the summertime recurring parameter mode
sensor(config-Host-tim-sum)# recurringParams
// Now specify the summertime timezone name
sensor(config-Host-tim-sum-rec)# summerTimeZoneName PST
sensor(config-Host-tim-sum-rec)# exit
sensor(config-Host-tim-sum)# exit
sensor(config-Host-tim)# exit
sensor(config-Host)# exit
Apply Changes?[yes]: yes
Warning: The node must be rebooted for the changes to go into effect.
Continue with reboot? [yes]:
```

The next step is to configure the Secure Shell server on the IDS sensor. Figure 5.5 shows how this is done. We will use the *ssh generate-key* command from the top-level prompt. Once the key has been generated, the sensor must be rebooted. After the sensor reboots, it can be accessed directly through SSH.

Figure 5.5 SSH Key Generation and Reboot

```

Ciscoids-1 login: Cisco
password:
last login: Thu Sept 25 15:58:25 on ttyS0
****NOTICE****

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer, and use.
Delivery of Cisco cryptographic products does not imply third-party
authority to import, export, distribute or use encryption. Importers,
exporters, distributors, and users are responsible for their compliance
with U.S. laws and regulations. If you are unable to comply with U.S. and
local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found
at: http://www.Cisco.com/ww1/export/crypto

Ciscoids-1# ssh generate-key
MD5: 05:2D:b1:E1:06:AE:40:C5:3D:DD:01:EE:34:92:CC:20
Bubble Babble: xires-rifs-vonuz-pubue-sapet-sauron-rings-lords-fatyn-gelin-
opera
Warning: The node must be rebooted for the changes to go into effect.
Continue with reboot? [yes]:

```

Once the sensor has finished rebooting, the next step is to configure the allowed hosts which can connect to the SSH server on the sensor. This can be accomplished as follows:

1. Log in to the sensor using the *cisco* account.
2. Enter configuration mode using the *configure terminal* command at the CLI prompt.
3. Enter the host service sub-menu using the *service host* command.
4. Select the network parameters sub-menu using the *networkParams* command.
5. Using the *accessList* command, enter the IP address and netmask of the hosts or subnets that will be allowed access to the IDS sensor through the network interface. The format of this command is: **accessList ipAddress <A.B.C.D> [netmask <A.B.C.D>]**.

6. Once all of the IP addresses or IP address ranges have been entered into the access-list, use the *show settings* command to verify them. This is shown in Figure 5.6.
7. Exit the networkParams sub-menu and return to the host service menu. Upon exiting the host service sub-menu, the IDS will request confirmation that the changes be applied to the sensor. Press **Enter** to select the default response of Yes. Otherwise, type **No** and press **Enter** .
8. Exit the host service sub-menu and the configuration menu.

Figure 5.6 Access-List Configuration on IDS Sensor

```
sensor(config)# service host
sensor(config-Host)# networkParam
sensor(config-Host-net)# accesslist ipaddress 10.16.17.0 netmask
255.255.255.0
sensor(config-Host-net)# show settings
networkParams
-----
  ipAddress: 10.1.9.201
  netmask: 255.255.255.0 default: 255.255.255.0
  defaultGateway: 10.1.9.1
  hostname: sensor
  TelnetOption: disabled default: disabled
  accessList (min: 0, max: 512, current: 2)
-----
  ipAddress: 10.0.0.0
  netmask: 255.0.0.0 default: 255.255.255.255
-----
  ipAddress: 10.16.17.0
  netmask: 255.255.255.0 default: 255.255.255.255
-----
-----
sensor(config-Host-net)#
```

Once the access-lists have been configured, the IDS sensor can be accessed using Secure Shell over the network.

The sensor needs to connect to hosts, which are SSH servers for software upgrades, signature updates, and file copying as well as other hosts, such as Cisco routers, PIX Firewalls, and Catalyst switches. In order to facilitate that communication, the SSH host keys of the hosts that the sensor can communicate with must be added to the `known_hosts` list. The following steps can be used to add hosts to this list:

1. Log in to the sensor using the *cisco* account.
2. Enter configuration mode using the *configure terminal* command from the CLI prompt.
3. Use the *ssh host-key* command to enter the IP address of the host whose SSH host key will be added to the `known_hosts` list. This is shown in Figure 5.7.
4. When asked if the key of the host should be added to the known hosts table, press **Enter** to select the default response of Yes. Otherwise, type **No** and press **Enter**.
5. To verify the SSH keys in the known hosts list on the sensor, use the *service sshKnownHosts* command at the top-level configure prompt.
6. Use the *show settings* command to list the hosts in the known hosts list, as shown in Figure 5.8.
7. Exit the *service sshKnownHosts* sub-menu and return to the top-level configure menu.
8. Exit configure mode.

Figure 5.7 Adding the SSH Host Key to the Known Hosts List

```
Ciscoids-1(config)# ssh host-key 192.168.50.14
MD5: 05:2D:b1:E1:06:AE:40:C5:3D:DD:01:EE:34:92:CC:20
Bubble Babble: xires-rifs-vonuz-pubue-sapet-sauron-rings-lords-fatyn-gelin-
opera would you like to add this to the known hosts table for this
host?[yes]
Ciscoids-1(config)#
```

Figure 5.8 Displaying the SSH Known Hosts List

```

sensor# config t
sensor(config)# service ssh
sensor(config-SshKnownHosts)# show settings
    rsa1Keys (min: 0, max: 500, current: 1)
    -----
        id: 192.168.50.3
        exponent: 35
        length: 1024
        modulus:
16508318659201744987257493934049916934023534822357915597860524173
8075615412030757209625612325747411882803771482511468683235829969888641604222
4132981902416287493190437220610204921172702794243732481684970354838327952077
2060730597444996382750101204023809139442273626501927211475878502549484330223
6884372899127817
    -----
    -----
sensor(config-SshKnownHosts)#

```

When we need to remove an entry, we use the following command:

```

sensor(config-SshKnownHosts)# no rsa1keys <id ip_address>

```

The *<ip_address>* parameter is the known host that we want removed from the rsa key ring. We see in the following sample how this command works:

```

(config-SshKnownHosts)# no rsa1keys id 192.168.0.20

```

The host 192.168.0.20 is removed from the SSH known hosts list. To verify the removal, we can use the command:

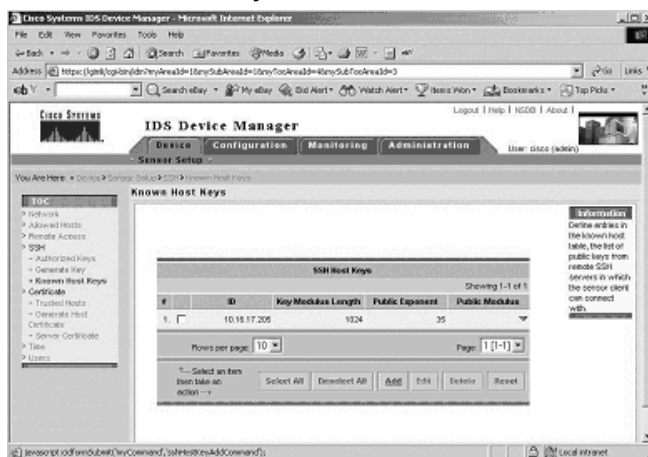
```

sensor(config-SshKnownHosts)# show settings
    rsa1Keys (min: 0, max: 500, current: 0)
    -----
    -----
    -----
sensor(config-SshKnownHosts)#

```


To add host keys to the sensor for use in updating the IDS software or signature packs, select the **Known Host Keys** link in the TOC menu at the left of the browser window. If a host key is already in the known hosts list, it will be displayed in the table in the middle of the window, as shown in Figure 5.11. To add a host key to the table, select the **Add** link at the bottom right of the table.

Figure 5.11 The Known Host Keys Table



Selecting this link brings up the next page, which asks you to add the host key of the host that the IDS will communicate with. Fill in the IP address as well as the key modulus length, public exponent, and public modulus of the host key. The values for the key modulus length, public exponent, and public modulus can be obtained from the `ssh_host_key.pub` file. An example of such a host key is shown in Figure 5.12. Here the public exponent is 35, the key modulus length is 1024, and the public modulus is the long number between the public exponent value and the name identifier at the end of the host key.

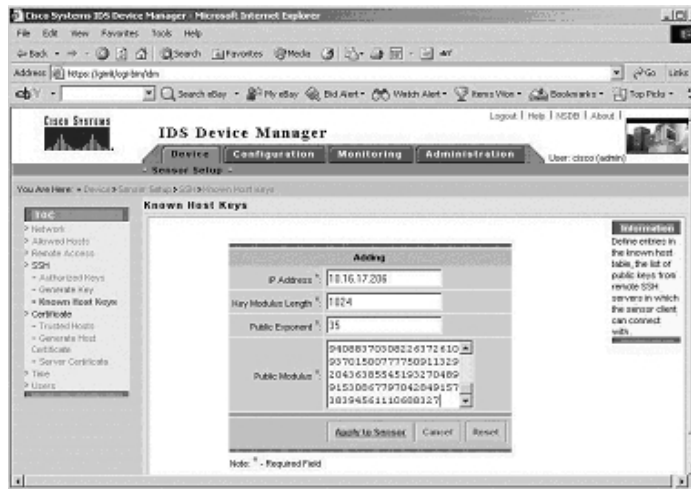
Figure 5.12 The SSH Host Key Structure

```
1024 35 165083186592017449872574939340499169340235348223579
155978605241738075615412030757209625612325747411882803771482
511468683235829969888641604222413298190241628749319043722061
0204921172702794243732481684970354838327952077206073059744499
63827501012040238091394422736265019272114758785025494843
302236884372899127817
```

The first number, 1024, is the *Public Exponent*. The second number, 35, is the *Key Modulus Length*. The final set of numbers is the *Public Modulus number*. All of this can be found in the `/etc/ssh/ssh_host_key.pub` file. This example was from Red Hat 7.2, but most flavors of Unix/Linux will follow the same format. For a Windows ssh client like Tera Term, you will find this information in the `C:\program files\teraterm\ssh_known_hosts` file.

Using the values in the SSH host key, fill in the required fields in the **Adding Known Host Keys** page, as shown in Figure 5.13. Select **Apply to Sensor**. The host key is added to the known_hosts list.

Figure 5.13 Adding an SSH Host Key to an IDS Sensor



The final option in configuring SSH through IDM is entering the individual user SSH keys. This allows for public key authentication rather than using passwords as a means of accessing the IDS sensors. To enter the necessary information, use a key generation tool such as `ssh-keygen` on Unix/Linux systems to generate a public/private key pair for the user on the client where the private key is going to reside. Then, display the generated public key as a set of three numbers (Key Modulus Length, Public Exponent, Public Modulus) and enter those numbers in the proper fields.

Compatible Secure Shell Protocol Clients

There are many SSH clients that can be used to access the IDS sensors. An SSH client that supports the SSH-1 protocol should be used in order to access the

IDS sensor CLI. The following SSH clients have been tested by Cisco and verified to work with the SSH server in the IDS sensor software.

For Windows clients:

- SecureCRT 3.1 is available at www.vandyke.com/products/securecrt.
- PuTTY 0.53b is available at www.chiark.greenend.org.uk/~sgtatham/putty.
- The SSH Secure Shell for Workstations 3.2 is available at www.ssh.com/support/downloads/secureshellwks.
- Tera Term Pro 2.3 with TTSH 1.5.4 is available at www.packetattack.com/downloads.html.

For Unix/Linux clients:

- OpenSSH 3.4p1 is available at www.openssh.com/pub/OpenBSD/OpenSSH/portable.
- The SSH Secure Shell for Servers 3.2 is available at www.ssh.com/support/downloads/secureshellserver.

NOTE

While officially the preceding list represents SSH clients that are guaranteed to be compatible with the SSH server in Cisco's IDS sensor software, the fact is there is a much wider range of SSH clients that are compatible. These clients include

- OpenSSH 3.5–3.7 clients (both the portable version and the OpenBSD version)
- NiftyTelnet 1.1 SSH r3 (a Macintosh SSH client)
- SSH 1.2.3

Configuring Remote Access

All IDS sensors can have their serial consoles available through a terminal server. With IDS software v4.0 and later, this connection is easy (it's described earlier in this chapter). IDS sensors running IDS software 3.0 or 3.1 require a slight modification to the serial port setup on the terminal server in order for remote access to the serial port to operate properly. The following list identifies the necessary configuration in order to access version 3.0 and 3.1 sensors remotely.

- Terminal Server Setup
- BIOS setup for the IDS-4210 Sensor
- BIOS setup for the IDS-4220 and DIS-4230 Sensors

Terminal Server Setup

The terminal server port configuration that the IDS sensor console will connect to must be modified slightly from the default values. For the purposes of the rest of this section, the terminal server is assumed to be a Cisco 2511-RJ router used as a terminal server. For other terminal server hardware, consult the proper documentation. To change the configuration of the terminal server, Telnet to the terminal server (or, more preferably, if the terminal server software supports SSH, use Secure Shell) and enter configuration mode, as shown in Figure 5.13. To configure the terminal port for proper operation with a version 3.0 or 3.1 sensor use the commands displayed in Figure 5.14:

Figure 5.14 The Terminal Server Line Configuration

```

termsrv#config t
termsrv(config)# line <line number>
termsrv(config-line)# no exec
termsrv(config-line)# login
termsrv(config-line)# transport input all
termsrv(config-line)# stopbits 1
termsrv(config-line)# flowcontrol hardware
termsrv(config-line)# exit
termsrv(config)# exit
termsrv# wr mem

```

If a terminal session does not receive a proper exit signal, the terminal session may remain open. This leaves the terminal session open and accessible without any authentication. Typically, this occurs when the physical connection to the sensor is disrupted (such as a line drop or disconnect). Another possible source for this problem may be when the application connected to the terminal server is terminated prematurely and the connection is dropped. In these cases, the next connection to the terminal server port will be provided access directly to the IDS sensor console without requiring authentication. It is imperative that any session with the

terminal server be properly terminated (exit the session and return to a login prompt before terminating the terminal server session) in order to ensure the security of the IDS sensor. If a connection is broken or dropped by accident, the user should reestablish the connection and exit normally back to the login prompt and then exit the application used to connect to the terminal server session.

BIOS Modifications for IDS 4210/4220/4230 Sensors

In addition to the configuration of the terminal server, some older sensor models require modifications to their system BIOS in order to redirect their consoles over to the serial port. This section covers the modifications necessary in order for the older IDS 4210, 4220, and 4230 sensors to redirect their consoles. Newer sensors do not require this modification as they direct their consoles to the serial ports by default.

The IDS-4210 Sensor

The IDS 4210 sensor is a 1U rack mount appliance that can be connected to with a keyboard, mouse, and monitor or through the serial port located at the back of the device. The 4210 BIOS can redirect the entire console of the device to the serial through the following modifications. In order to make the following changes, a keyboard and monitor must be connected to the 4210 sensor, as the console redirection has not been configured yet. To redirect the console, use the following steps:

1. Boot or reboot the sensor.
2. During POST, press **F2** when prompted to enter BIOS setup.
3. Click **Serial Features** on the **System Management** menu.
4. Enable **Serial Console Redirection** and change settings to match the following:

```
Serial Port: COM1 3F8 IRQ4
```

```
Baud Rate: 9600
```

5. Press **Esc** to return to the System Management menu.
6. Click **Exit Saving Changes**.
7. When asked to confirm the changes, press **Y** and then **Enter**.

The Sensor will automatically reboot and redirect the console to the serial port.

The BIOS Setup for the IDS-4220 and IDS-4230 Sensors

Connecting to the serial console of an IDS sensor is useful should a problem arise in the IDS sensor software that prevents access to the sensor either through the IDM or Secure Shell. A serial connection through either a terminal server or directly through a serial cable connection provides direct access to the IDS sensor console without the requirement of a keyboard or monitor. To redirect the consoles of the IDS-4220 and 4230 sensors to the serial port, the following BIOS changes are required. As with the 4210, these changes need to be performed locally on the sensor using a keyboard and monitor since redirection has not yet been configured.

1. Boot or reboot the sensor.
2. During POST, press **F2** when prompted to enter BIOS setup.
3. Select **Console Redirection** on the **Server** menu.
4. Change the **COM Port Address** from Disabled to **3F8**.
5. Make sure all other settings match the following:
 - IRQ# 4
 - Baud Rate: 9600
 - Console Type: PC ANSI
 - Flow Control: CTS/RTS + CD
6. Press **Esc** to return to the **Server** menu.
7. Click **Exit Saving Changes** on the **Exit** menu.
8. When asked to confirm the changes, press **Y** and then **Enter**.

Applying the Sensor Configuration

You are ready to assign interfaces, configure signatures, set up blocking, set up automatic signature updates, and restore defaults after you have completed configuring system information.

The following sections describe how to use the Configuration tab to configure the following options:

- Configuring Interfaces
- Configuring Blocking
- Configuring Automatic Updates
- Restoring Default Settings

Cisco Enabling and Disabling Sensing Interfaces

For every sensor, there is only one command and control interface. Depending on the model of sensor you have, you can set up to five sniffing or monitoring interfaces. In Table 5.2, we can see the matrix showing the monitoring interfaces of every IDS sensor, and the name of each interface.

Table 5.2 Sensor Models and Monitoring Interface Names

Sensor	Sniffing Interface
IDS-4210	int0
IDS-4215	int0
IDS-4215-4FE	int0, int2, int4, int5
IDS-4220 and IDS-4230	int0
IDS-4235	int0
IDS-4235-FE	int0, int2, int3, int4, int5
IDS-4250	int0
IDS-4250-SX	int0, int2
IDS-4250-XL	int0, int2, int3
IDS-4250-FE	int0, int2, int3, int4, int5
IDSM-2	int7 and int8
NM-CIDS	int1

Make sure the monitoring interfaces are part of Group 0 and are enabled for the sensor to monitor the network traffic.

NOTE

Sensors that have factory-installed Cisco IDS version 4.1 are shipped with all sniffing interfaces added to Interface Group 0 and disabled. On the sensor that you want to monitor, you must enable the sniffing interfaces. If you do not enable the sniffing interfaces, the sensor will not be able to monitor your networks. Only enable those interfaces that you want to monitor; you do not need to enable all interfaces.

WARNING

When upgrading from version 4.0 to 4.1, some interfaces may be left enabled that are not assigned to a group. You must choose to disable these interfaces or add them to Group 0 to prevent inconsistencies in reporting to the sensor.

To show the current interfaces and what they are assigned as, use the *show interface* command, as displayed in Figure 5.15.

Figure 5.15 Showing the Interface Configuration

```

sensor# show interface
command-control is up
    Internet address is 192.168.50.51, subnet mask is 255.255.255.0, Telnet
is disabled.
    Hardware is eth1, tx

Network Statistics
    eth1      Link encap:Ethernet  HWaddr 00:E0:29:75:46:75
              inet addr:192.168.50.51  Bcast:192.168.50.255
Mask:255.255.255.0
              UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
              RX packets:2819 errors:0 dropped:0 overruns:0 frame:0
              TX packets:2293 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:100
              RX bytes:340909 (332.9 Kb)  TX bytes:1070419 (1.0 Mb)
              Interrupt:17 Base address:0x1400

Group 0 is up
    Sensing ports int0
    Logical virtual sensor configuration: virtualSensor
    Logical alarm channel configuration:  virtualAlarm
..VirtualSensor0
    General Statistics for this Virtual Sensor
        Number of seconds since a reset of the statistics = 12887
:::output trimmed for brevity:::

```

As you can see from Figure 5.15, our management interface is eth1 and the monitoring interface (or sniffing interface) is int0. The monitoring port is part of Group 0.

Adding Interfaces to an Interface Group

To group monitoring interfaces into one logical virtual sensor, you will use an interface group. At this time, only interface Group 0 is supported. More than one monitoring interface can be assigned to the interface group. The monitoring interfaces must be added to Group 0 and be enabled for the sensor to monitor the sniffing interfaces.

NOTE

You will not be able to assign the command and control interface to the interface group.

To add or remove interfaces from Group 0 is very straightforward. In Figure 5.16, we outline the addition and removal of an interface to Group 0.

Figure 5.16 Adding and Removing Interfaces from Group 0

```
sensor# config t
sensor(config)# interface group 0
sensor(config-ifg)# no sensing-interface int0
sensor(config-ifg)# exit
// This removes int0 from the Group 0.
sensor(config-ifg)# sensing-interface int0
sensor(config-ifg)# exit
// This adds int0 back into Group 0.
```

WARNING

The IDS-4250-XL, interface 0 (int0) is used for sending TCP resets and cannot be a sensing interface.

Configuring Logging

Logging provides a way to record the events that the IDS sensor sees for later analysis either by security personnel, network operations, or event correlation software. This section covers how to configure event logging as well as IP logging, how to export event logs, and how to configure automatic IP logging. Logging changes between IDS software version 3.1 and 4.0 include the discontinuation of event logging to files in 4.0. All events are logged to the internal database running on the IDS sensor. IP logging does not change between the two software versions.

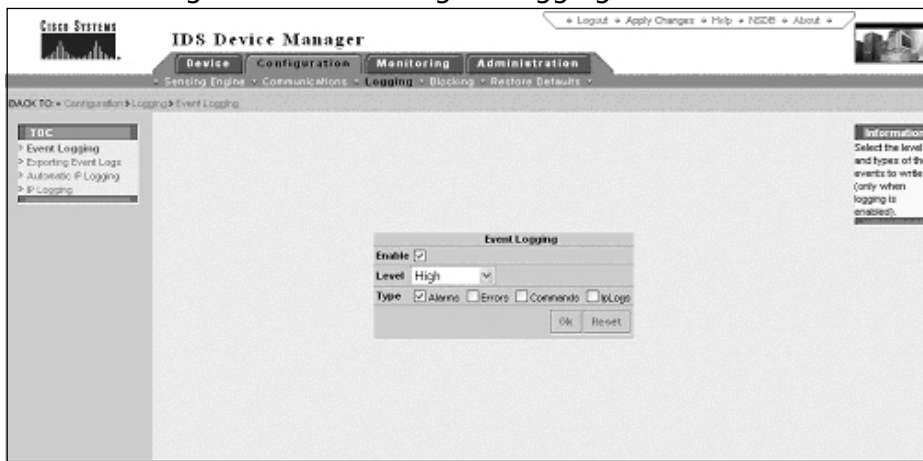
Configuring Event Logging (IDS version 3.1)

Depending on what the sensor had been configured to watch, it can generate audit event logs locally on the sensor based on syslog data streams, network data streams, or both. Follow these steps and examine Figure 5.17 to see how events will be logged:

1. In the IDS Device Manager main window, select **Configuration | Logging | Event Logging**.
2. The **Event Logging** panel appears. Select the **Enable** check box. Once event logging has been enabled, the only two options that can be set are the **Level** and **Type** options.
3. Select the severity level of the signature from the **Level** list box:
 - **Information** Attacks not relevant to security are categorized. These attacks are shown in the IDS Event Viewer with a blue icon.
 - **Low** Mildly severe attack. These attacks are shown in the IDS Event Viewer with a yellow icon.
 - **Medium** Moderately severe attack. These attacks are shown in the IDS Event Viewer with an orange icon.
 - **High** Highly severe attack. These attacks are shown in the IDS Event Viewer with a red icon.
4. To specify types of events you want to log, select one or more of the **Type** check boxes.
 - Alarms
 - Errors

- Cmd Logs
 - IP Logs
5. Click **OK**.

Figure 5.17 Using 3.1 IDM to Configure Logging



If alarm events are selected to be logged, then all alarms for enabled signatures which have severity levels that are greater than, or equal to, the selected level chosen in the Event Logging Panel are logged to the file `/usr/nr/var/log/log.timestamp`. If IPLogs are desired as well, then the severity level must be set to **Information**. IPLogs are stored in a binary format in the `/usr/ne/nr/iplog/iplog.address.timestamp` files.

NOTE

ComdLogs, Errors, and Alarms are also written to the event logs.

To view the event log files, select **Monitoring** | **Logs** in the IDM browser window.

Exporting Event Logs

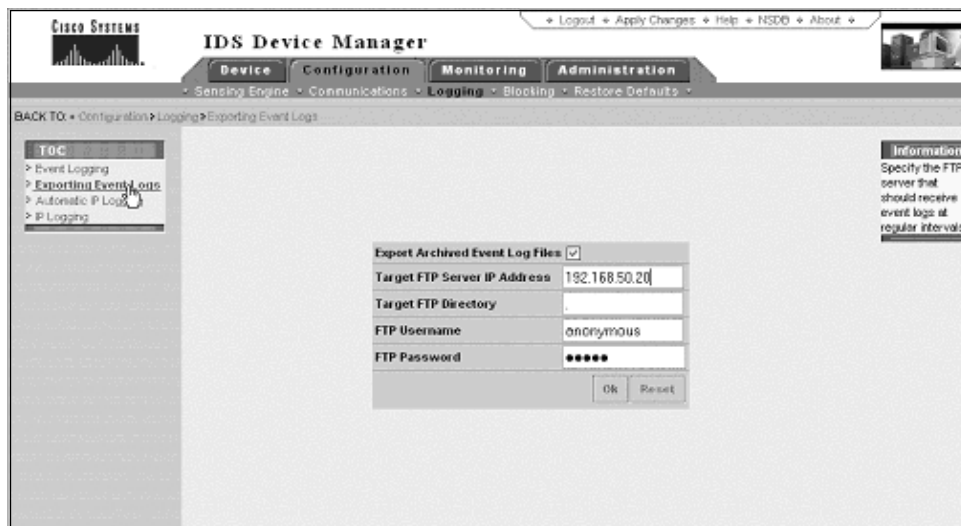
By default, the IDS sensor logs all events locally on the sensor by both severity and type. A feature of the IDS sensors is that you can export the event logs to an

FTP server. This allows you to run detailed analysis using other tools such as Sawmill. Once the logs are exported, you can maintain an archive of events over time that can be of help if you need to pull up the logs of several months ago because of legal issues such as hacking attempts. You can configure the export function to use an FTP server that event logs will be sent to at regular intervals.

The following steps illustrate how to configure the export of event logs (also see Figure 5.18):

1. Select **Configuration | Logging | Exporting Event Logs**.
2. The **Exporting Event Logs** panel appears. Check the box for **Export Archived Event Log Files**
3. Enter the IP address of the FTP server you want to connect to and send the logs to in the Target FTP Server IP Address field.

Figure 5.18 Configuring Exporting Log Files



NOTE

The following FTP servers support FTP log export functions:

- Windows NT 4.0 (Microsoft ftp server ver 3.0)
- Sambar FTP Server Ver 5.0 (win32)
- Windows 2000 (Microsoft ftp server ver 5.0)
- Web-mail Microsoft FTP Service Version 5.0 (win32)
- HP-UP (HP-US qdir-5 B.10.20 A 9000/715)

- Serv-U FTP-Server v2.5 for WinSock (win32)
 - Solaris 2.8
-

4. Enter the target directory on the remote FTP server in the **Target FTP Directory** field. This can be 1 to 128 characters.
5. Enter the FTP server login name in the **FTP Username** field. This can be 1 to 16 characters.
6. Enter the FTP server password associated with the login name in the **FTP Password** field. This can be from 1 to 8 characters. Click **OK**.
7. View the messages.sapd file to verify the event logs are being exported by selecting **Monitoring | Logs | Messages | Sapd**. If there is an error, this is where you will see it.

NOTE

Every time the event log is closed and archived, logs are FTPed. This occurs once a day by default or when the logs fill up the 104,876 bytes allocated to them, whichever comes first.

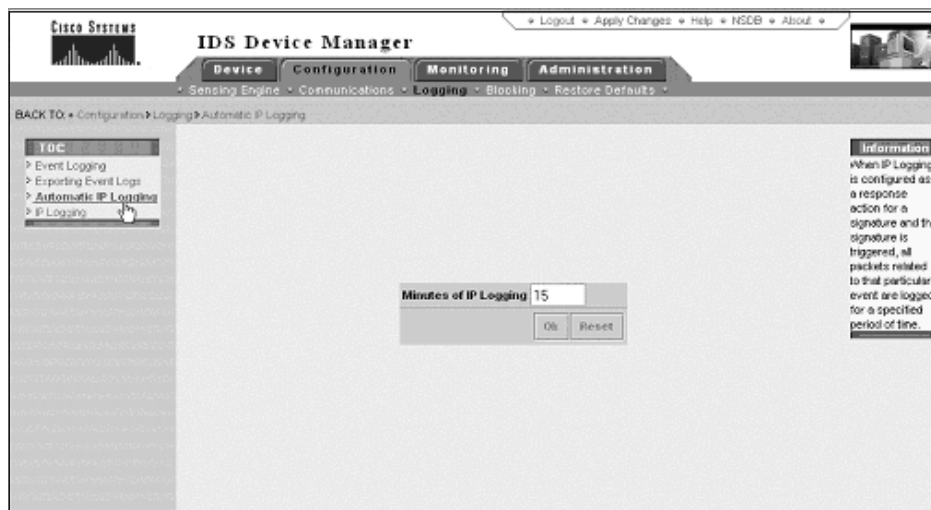
Configuring Automatic IP Logging

You can configure a sensor to generate an IP session log when the sensor detects an attack. All packets to and from the source address of the alarm are logged for a specific period of time when IP logging is configured as a response action for a signature and the signature is triggered. Additionally, you can set the number of minutes events are logged. The IP log file is in the tcpdump format for ease of exporting into other tools if required. Follow these steps to set the amount of minutes of automatic IP logging and see Figure 5.19 for the screen shot of the IDSM interface:

1. Select **Configuration | Logging | Automatic IP Logging** in the **IDS Device Manager** main window.
2. Enter the number of minutes you want IP logging to be done (from 1 to 60) in the **Minutes of IP Logging** field. Note that the default is 15. Click **OK**.

3. Select **Monitoring | Logs | IP Logs | Archived** to download the logs.

Figure 5.19 Configuring Automatic IP Logging Using the Cisco IDM



Configuring IP Logging

One option you have is to configure the sensor to capture all traffic related to the specified hosts. We can use the IP logging option to log all traffic or a list of IP addresses.

NOTE

You must enable event logging with Information as the severity level and at least IPLogs for the type since this is an IP logging requirement.

Follow these steps to generate logs for specific IP addresses:

1. Select **Configuration | Logging | IP Logging** in the **IDS Device Manager** window. The IP Logging panel will appear.
2. To enter IP addresses, click **Add**.
3. Enter the source IP address to log in the IP address field.

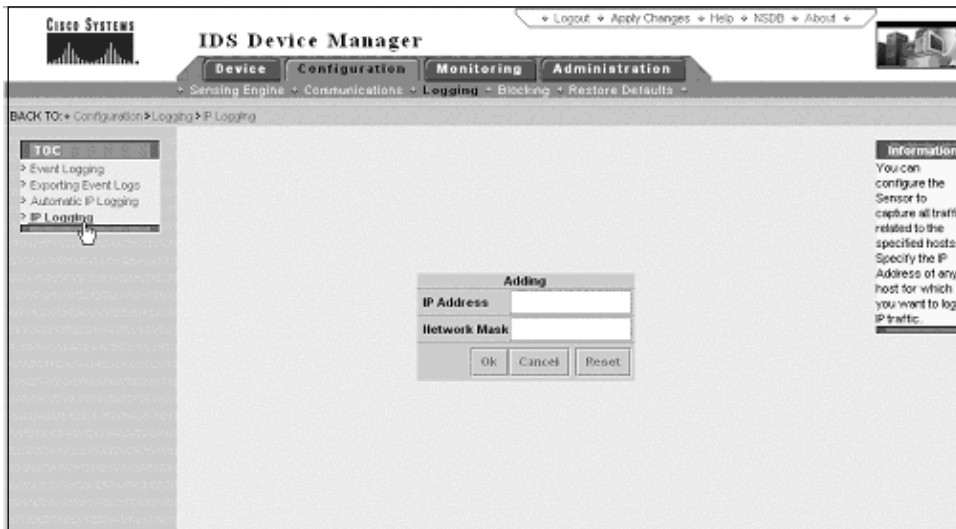
4. Enter **255.255.255.255** if it is a single IP address, or enter the netmask if it is a network in the Network Mask field.

NOTE

By selecting **Monitoring | Logs | IP Logs | Archived**, the sensor begins logging and thus creates a log file that can be viewed. Logging will continue until the address is removed from the IP Logging list. Be aware that logging slows down the performance of the sensor.

Figure 5.20 shows the panel for configuring IP logging using the IDM for version 3.1.

Figure 5.20 Configuring IP Logging Using the Cisco 3.1 IDM



When we use version 4.x software, the process is a little bit different, as shown in the following steps and in Figure 5.21:

Follow these steps to generate logs for specific IP addresses:

1. Select **Administration | IP Logging** in the **IDS Device Manager** window. The IP Logging panel will appear.
2. To enter IP addresses, click **Add**.

3. Enter the source IP address to log in the **IP Address** field.
4. Enter values in the optional **Duration**, **Number of Packets**, and **Number of Bytes** field.

Figure 5.21 Configuring IP Logging Using the Cisco 4.x IDM



Generating IP Logs

The sensor can be configured to catch all IP traffic associated with the hosts you specify using the IP address. To generate log files for a specific IP address, first log into the CLI using an account with administrator or operator privileges. For each address, you can either specify that the sensor log IP traffic until a specific threshold is reached (using number of minutes, packets, or bytes), or you can configure the sensor to continue logging IP traffic until you later disable IP logging for that address.

Type in the following command to configure the sensor so it continues logging indefinitely for a specific IP address:

```
Sensor# iplog interface group number (0) <IP address>
```

The components of this command include

- **interface group number** Group ID to begin or end logging on. There is only one interface group supported. Use 0 as the value.
- **ip address** Only log packets that contain the specified IP address.

Type the following command to configure the sensor to log IP traffic until a specified threshold is reached.

```
Sensor# iplog interface group number (0) <ip_address> <duration minutes>  
packets <numPackets> bytes <numBytes>
```

The components of this command include:

- **minutes** The duration the logging should be active in minutes from 0 to 60 (the default is 10 minutes).
- **numPackets** The maximum number of packets to log from 0 to 4294967295 (the default is 1000 packets).
- **numBytes** The maximum number of bytes to log from 0 to 429496295.

NOTE

You do not have to specify all three parameters; these are optional. If you choose to include more than one parameter, the sensor will continue logging only until the first threshold is reached. For example, if you set the duration to five minutes with the number of packets to 1000, the sensor will stop logging after the 1000th packet is captured, even if only two minutes have elapsed.

Based on the parameters you specified, the sensor begins logging. A log ID number will appear. If you later want to stop the logging session, you will need this log ID. When you type in the command *iplog-status*, you will get a short version of the status of the logging, as shown in Figure 5.22:

Figure 5.22 IP Logging ID Status

```
sensor# iplog-status  
IPAddress: 10.1.1.2  
Status: In-Progress  
Start Time: 10:02:34 8/24/2001  
Minutes Remaining: 5 minutes  
Packets Captured: 1039438  
Packets Remaining: 48 Packets
```

Continued

Figure 5.22 IP Logging ID Status

```
// To stop a specific IP logging session you will type in:
sensor# no iplog Log ID
// To stop all IP logging you will type in:
sensor# no iplog
```

Upgrading the Sensor

Cisco Systems periodically releases updates of sensor software and signature versions. It is highly recommended that you regularly install the updates of signature versions as well as sensor software in order to ensure the value of the IDS sensor in the overall security architecture. Without regular updates, the IDS sensor will become no more than a pretty decoration in the rack since it will not be able to sense current threats to your network.

Updating Sensor Software (IDS 3.1) Upgrading the IDS sensor software version 3.0 or 3.1 can be done by downloading the service pack from the Cisco Web site and applying it to the IDS sensor. The following procedure can be used to update IDS sensor software versions 3.0 or 3.1.

1. From the Cisco.com Web site (www.Cisco.com/cgi-bin/tablebuild.pl/ids3-app) download the self-extracting binary file.
2. On the target sensor, copy the binary file to the **/tmp** directory.
3. Log into the sensor as **root**.
4. Change the directory to the **/tmp** directory.
5. Change the binary file's attributes so it is an executable:

```
sensor# chmod +x IDSk9-sp-3.1-4-S50.bin
```

6. Type the following to execute the binary file with the **-I** option.

```
sensor# ./IDSk9-sp-3.0-1-S4.bin -I
```

7. Then the installation is complete, review the file */usr/nr/sp-update/output.log* for the status of this service pack.

Upgrading from 3.1 to 4.x

To upgrade the IDS sensor appliance from IDS software versions 3.0 or 3.1 to version 4.0 or later requires the installation of the 4.0 software from the install CD.

Before upgrading from 3.1 to 4.0, the configuration information of the IDS sensor

must be saved. The easiest way to do so is through the IDS Device Manager (IDM). Use the following procedure to save the configuration information of the IDS sensor before upgrading.

1. From the IDM browser, select **Administration | Diagnostics**. The diagnostics panel will display.
2. Click **Run Diagnostics**.
3. Click **View Results**. The diagnostics results are displayed in a report.

To save the results of the diagnostics, select **Menu | Save As** in the browser. With the configuration information saved, the sensor can be upgraded. The following procedure can be used to upgrade a sensor from IDS sensor software version 3.1 to version 4.0 or later.

1. Power on the sensor.
2. Insert the IDS 4.0(1) or 4.0(2) Upgrade/Recovery CD into the CD-ROM drive.

On the IDS console, the following message will be displayed:IDS-4220/4230 customers:

```
Sniffing and Command-and-Control interfaces have been swapped in
CIDS 4.0. Reference the 4.0 software documentation before
proceeding.
```

IDS-4235/4250 customers:

```
BIOS version "A04" or later is required to run CIDS 4.0 on your
appliance. Reference the 4.0 software documentation before
proceeding.
```

```
- To recover the Cisco IDS 4.0 Application using a local
keyboard/monitor, type: k <ENTER>. (WARNING: ALL DATA ON DISK 1
WILL BE LOST)
```

To recover the Cisco IDS 4.0 Application using a serial connection, type: **s**, or just press **ENTER**. If the upgrade is being done with the console redirected to a serial port, press **S**. Otherwise, press **K** if the upgrade is being done with a keyboard and monitor connected directly to the sensor. After selecting either **S** or **K**, the upgrade will continue without requiring user intervention. Once the upgrade is complete, the sensor will eject the CD from the CD-ROM drive and reboot automatically. When the sensor has completed rebooting, log into the sensor using the *Cisco* account with the default password of **Cisco** and continue with the initial configuration of the device as discussed earlier in the book

NOTE

When upgrading an IDS-4220-E or IDS-4230-FE appliance, the command and control interface cable must be swapped with the monitoring interface cable before the software upgrade. IDS software version 4.0 switches the interfaces by making the former command and control interface into the sniffing interface. If the cables are not switched on the IDS-4220-E or IDS-4230-FE, it will not be possible to connect to the appliance through the network.

The reason for the interface switch is because the PCI-based card that was used as the sniffing interface in the IDS-4220-E and the IDS-4230-FE does not support the monitoring of the 802.1q tagged VLAN trunk packets or the tracking of the 993 Dropped Packet Alarm. Also, the performance of the PCI-based card is lower compared to the onboard NIC. For more information, see: www.Cisco.com/en/US/products/sw/secursw/ps2113/products_installation_and_configuration_guide_chapter09186a008014a23a.html#533236

1. Updating Sensor Software (IDS 4.0) The IDS sensor software version 4.0 and later uses a different method for updating the sensor. Since the 4.0 series software is based on Red Hat Linux, all sensor service packs are released as RPM packages. In addition, the 4.0 series software supports one of two methods for uploading service packs: *ftp* or *Secure Copy Protocol (scp)*. To use *scp*, the host key of the system that the IDS will connect to upload the service pack must be installed in the sensor's `known_hosts` table. The following procedure can be used to update a sensor to the latest service pack. Go to the Cisco Connection Online (CCO) Web site URL (www.Cisco.com/kobayashi/sw-center/Ciscosecure/ids/crypto/). The choice of software service packs depends on which model IDS sensor is being updated. Select the link for the appropriate sensor model. As of this writing, there are two links: one for the 4210/4235/4250 model sensors and one for the IDS 4215 sensor.
2. Select the appropriate service pack (for the purposes of this example, the latest service pack is: `IDS-K9-min-4.1-1-S47.rpm.pkg`).
3. Download the service pack.

The update can be completed using either the command line interface or the IDM.

Updating Sensor Software (IDS 4.0) from the Command Line

The IDS sensor software upgrade command can be accessed from the configuration mode of the command line interface. To upgrade the IDS sensor using a service pack, do the following:

1. Log into the IDS sensor using the administrative account Cisco.
2. Enter configuration mode.
3. Use the *upgrade* command to upload and apply the service pack. The supported protocols for the upgrade command include ftp and scp. The location of the service pack is given in a URL format as follows: <protocol name>://username@IP Address/directory/service pack name. For example, to upgrade an IDS 4.1 sensor to the S47 service pack on host 10.16.17.205 using scp, and also upgrade the account name Cisco, use the following URL: scp://Cisco@10.16.17.205/IDS-K9-min-4.1-1-S47.rpm.pkg.

```
sensor# config t
sensor(config)# upgrade scp://Cisco@10.16.17.205/upgrades/IDS-K9-min-4.1-1-S47.rpm.pkg
Password:
Warning: Executing this command will apply a minor version upgrade to the application partition. The system may be rebooted to complete the upgrade.
Continue with upgrade? : yes
```

Updating Sensor Software (IDS 4.0) with IDM

The sensor software can be manually updated using IDM. Use the following procedure to update the sensor software through the IDM:

1. Select **Administration | Update** in the IDM window.
2. This displays the **Update Settings** panel in the IDM, as shown in Figure 5.23.
3. Enter the URL of the update service pack.
4. Enter the password of the account to access the host where the update service pack can be found.
5. Click **Apply to Sensor**.
6. The update will be downloaded to the sensor and applied.

Figure 5.23 The IDM Update Settings Panel



1. In addition to manual updates, IDS 4.0 software supports autoupdating of sensor software and signature packs. The configuration of the autoupdate feature can be done either through the command line or with the IDM. Updating Sensor Software (IDS 4.0) from the Command Line To configure autoupdate of the sensor software using the command line interface, use the following procedure: Log into the IDS sensor using the administrative account **Cisco** and enter configuration mode:

```
sensor# configure terminal
```

2. Enter the *Host* service mode using the *service host* command.

```
sensor (config) # service Host  
sensor (config-Host) optionalAutoUpgrade  
sensor (config-Host-opt) autoUpgradeParams
```

3. Enter the IP address of the update host using the *IP Address* command and then select the protocol to use for copying the update packs to the sensor (either *scp* or *ftp*).

```
sensor (config-Host-opt-aut) ipAddress 10.16.17.205  
sensor (config-Host-opt-aut) fileCopyProtocol scp
```

- Specify the account name to use to access the update host, as well as the account password necessary to access the update host.

```
sensor (config-Host-opt-aut) username netrangr
sensor (config-Host-opt-aut) password attack
```

- Specify the directory where the updates can be found. This directory must be a relative directory to either the ftp home directory (if the FTP protocol is used) or a directory relative to the home directory of the account specified.

```
sensor (config-Host-opt-aut) directory updates
sensor (config-Host-opt-aut) schedule
```

- Select whether the updates will be either based on a calendar schedule or a frequency schedule. A calendar schedule specifies the time and day you will download the updates. The frequency update stipulates that the sensor will check for updates every *X* number of hours regardless of what day it is.

Updating Sensor Software (IDS 4.0) Using the IDM

The Cisco IDM provides a clean and easy way to update the sensor software. In order to start filling out the parameters, choose **Configuration | Autoupdate**. The screen shown in Figure 5.24 should appear.

Figure 5.24 The IDM Autoupdate Screen Shot

NOTE

The sensor cannot automatically download updates from Cisco's Web site. They need to be downloaded and moved to a local server.

Upgrading Cisco IDS Software from Version 4.0 to 4.1

At the time of this writing, the latest major version of Cisco's IDS sensor software was 4.1. The only way to upgrade to this version of the IDS sensor software was with the Upgrade/Recovery CD for version 4.1, using the same procedure utilized in updating from version 3.0 or 3.1 to IDS sensor software version 4.0.

NOTE

The IDS-4210 sensor must be upgraded to 512MB of memory in order to upgrade the sensor to software version 4.1 requirements. This requires an additional 256MB of memory that can be purchased from Cisco. Customers with a current SmartNET contract can request the memory upgrade at no charge.

NOTE

The IDSM (WS-X6381) cannot be upgraded to Cisco IDS 4.1. The IDSM (WS-X6381) must be replaced with WS-SVC-IDSM2-K9, which supports version 4.x software.

Updating IDS Signatures

Every two weeks, signatures packs for the IDS sensors are updated and posted to the Cisco Web site. In order to obtain updated signature packs, a CCO login account is required. Without this account, access to software upgrades and signature pack upgrades is not possible. This section covers how to update signature

packs on the IDS sensor as well as how to configure automatic updates and active update notifications.

IDS version 3.0 uses the *idsupdate* command both for scheduled and manual updates of service packs and signature packs. The *idsupdate* command also can be used to set up scheduled updates. The IDS 4.0 software uses the *upgrade* command.

Updating Signatures (IDS 3.0)

To manually update signatures under IDS 3.0, the *idsupdate* command can be used to download and install the signature update. The *idsupdate* command only supports ftp as a communication protocol. The following command can be used to update a 3.0 sensor with a new signature pack.

```
/usr/nr/bin/idsupdate netrangr@192.168.45.100/updates <ftppasswd>
```

NOTE

Care must be used when using the *idsupdate* command. If there is already an automatic update schedule in effect, a new schedule will be created and the old schedule will be cancelled.

Automatic Updates

IDS 3.0 sensors can be configured to automatically download and apply signatures and service pack updates. This removes the administrative burden of updating sensors by the network operations staff or security personnel. To configure automatic updates on IDS 3.0 sensors:

1. Select a remote machine where sensor updates will be stored. Make sure that this host is running an FTP server since the sensors will download the updates using FTP.
2. Log in as **root** on the sensor via Telnet, SSH, or a local console.
3. Change the directory to the **/usr/nr/bin** directory:

```
sensor# cd /usr/nr/bin
```

- Use the following command to set up *idsupdate*. If the directory */usr/nr/bin* is not in *root*'s execution path, use the full pathname. The format for the *idsupdate* command is as follows:

```
idsupdate username@ftpserver/<directory> <FTP Password> <Day>
      <hh:mm>
```

The components of this command are

- **ftpserver** Must be an IP address
- **/** separates the FTP server and the FTP home directory
- **directory** The name of the directory that is relative to the ftp home directory. For example, if the FTP home directory is */usr/home/ftp* and the directory name is *updates*, then the FTP server will look in */usr/home/ftp* for a directory named “updates” where the service packs and signature updates can be found. The directory specified can include several levels of subdirectories.
- **Day** Consists of a comma-separated list of one to seven digits that have the values of 0–6. Each day of the week is specified by a single number according to the following convention: 0=Sunday, 1=Monday, 2=Tuesday, 3=Wednesday, 4=Thursday, 5=Friday, and 6=Saturday.
- **hh:mm** Represents the hour and minutes in 24-hour military convention.

For example, to update the IDS 3.1 sensor at 11:15 p.m. every night with updates from the *updates* directory on the FTP host 10.1.1.101 using the *netrangr* account with the password **attack**, the following command can be used:

```
sensor# /usr/nr/bin/idsupdate netrangr@10.1.1.101/updates attack
0,1,2,3,4,5,6 23:15
```

- To view the current update schedule use:

```
sensor# /usr/nr/bin/idsupdate show
```

- To cancel the current schedule use:

```
sensor# /usr/nr/bin/idsupdate stop
```

Supported FTP Servers

The IDS sensor cannot download signature updates and service packs directly from Cisco's FTP servers. Rather the signature updates and service packs must be downloaded to a local FTP server for use by the IDS sensor. At the time of this writing, the following FTP servers were officially supported by Cisco as being compatible with the FTP client on the IDS sensor:

- Windows NT 4.0 (Microsoft ftp server version 3.0)
- Sambar FTP Server Version 5.0 (win32)
- Windows 2000 (Microsoft ftp server version 5.0)
- Web-mail Microsoft FTP Service Version 5.0 (win32)
- HP-UX (HP-UX qdir-5 B.10.20.A 9000/715)
- Serv-U FTP-Server v2.5 for WinSock (win32)
- Solaris 2.8

Active Update Notification

The Active Update Notification feature on Cisco's Web site can be used to receive updates on changes to IDS signatures. To receive notification of signature updates:

1. Go to www.Cisco.com/warp/public/779/largeent/it/ids_news/subscribe.html.
2. Fill in the **E-mail Address** box.
3. Enter the CCO password associated with the e-mail address in the **Password** box.
4. Click **Submit**.

When a signature update occurs, an e-mail will be sent to the e-mail address entered, with instructions on how to obtain it.

Updating Signatures (IDS 4.0)

IDS signature updates under IDS sensor software version 4.0 use the *upgrade* command in the same fashion as the software upgrades. The manual update process for signatures uses the *upgrade* command as well. The primary difference in updating signatures between IDS software version 3.0 and 4.0 is in the configuration of

automatic updates. Automatic updates can be configured in IDS 4.0 using either the command line or through IDM.

How to Restore the Default Configuration

If necessary, the default sensor configuration can be restored. This may be due to a configuration mistake that cannot be located, or a retasking of the IDS sensor in the network. The following procedure can be used to restore the default configuration to the sensor:

1. Select **Configuration | Restore Defaults**. The Restore Default page appears.
2. To restore the default configuration, click **Apply to Sensor**.

The netmask, default gateway, IP address, allowed hosts, password, and time will not be reset. Automatic and manual blocks will also remain in effect.

If you have found yourself in a situation that requires more than a simple restoration of the defaults, version 4.x has a recovery partition available that is a very nice safety net to have. Simply go to the command line and use the recover command to rebuild the sensor.

Summary

You can see that there is a lot that goes into the configuration of the Cisco IDS sensor appliance. There is the choice of Telnet or SSH. There is remote access to the IDS sensor via a terminal server, a dozen different models of IDS sensors, three major software releases, as well as a couple of minor ones just to add to your fun. We have learned that it is pretty easy to get started with SSH on the Cisco IDS sensor. Using either the IDM interface or the command line, you can use the `ssh generate-key` command. Once that is done, you can add the desired subnet to the access-list and use the ssh client to log in remotely. In the new 4.x code, Telnet is disabled by default, but by using the command `telnet-server` we can enable or disable the service.

When we want to configure remote access of IDS sensors like the 4210, 4220, and the 4230, we need to perform a BIOS update in order for the serial port to work correctly with something like a terminal server. This fix will allow the IDS sensor to redirect the output to the serial port much like how the console port on a Cisco router works. Cisco's newer IDS sensors do not need this modification.

Each sensor has only one command and control interface but the sensors can have multiple monitoring interfaces. These monitoring interfaces need to be part of Group 0 in order to work correctly. We can enable or disable the monitoring interfaces one at a time or all at once. Of course, best practices from Cisco say we should only enable those interfaces we need at the time. To enable the interfaces in version 3.1 software, use the IDM and choose **Configuration | Sensing Engine | Interfaces**. To enable the monitoring interface in the 4.x command line, use the `interface group 0` command. This gives us the `config-ifg` prompt. Now we can use the `sensing-interface <interface name>` command to add the interface to Group 0.

One of the most important functions of the Cisco IDS sensor aside from sensing potential threats is the capability to report back about the threats, and to log those threats. In the version 3.1 software, we could enable event logging locally, but this was dropped in version 4.x. We can export the event logs to a workstation/server by using FTP simple by using the IDM and choosing **Configuration | Logging | Exporting Event Logs**. IP logging is available for both versions of software. For the 3.1 software, using the IDM we select **Configuration | Logging | IP Logging** to enable it. For the 4.x code, we select **Administration | IP Logging | Add** to enable IP logging.

Without updating the Cisco IDS sensor software and signatures regularly, you run the risk of a threat getting past the IDS sensor due to old signatures and software. To update the Cisco IDS sensor with version 3.1, you need to download the correct service pack from Cisco's Web site and place it on the sensor in the */tmp* directory. Then you need to change the attributes so it can be executed by using *chmod +x*. Once that is completed, execute the binary file with the *-I* option. Keep in mind that if you are using 3.1, you should seriously consider moving up to 4.x since Cisco is dropping support for the 3.1 signatures and patches. With version 4.x, we can now update the Cisco IDS sensor through either the command line or with the IDM. For the command line upgrade, we can use the *upgrade* command and choose either the scp or ftp protocol. To upgrade with the IDM, we choose **Administration | Update** and then enter the correct information in the IDM window. We then click **Apply to Sensor** and the file will be downloaded to the sensor and installed.

Sometimes in configuring a Cisco IDS sensor, a mistake might be made that requires you to reset the IDS sensor back to the factory defaults. This is easily done by choosing **Configuration | Restore Defaults** and **Apply to Sensor**. When you do this, the IP address, default gateway, allowed hosts, password, and time are not reset. If you need to actually rebuild the IDS sensor, you can use the recovery partition in version 4.x. In version 3.x, you have to reinstall the sensor from the CD. In order to use the recovery partition, go to the command line and use the *recover* command.

Solutions Fast Track

Configuring SSH

- ☑ Use the *ssh generate* command to make a key. This requires a reboot of the IDS sensor but once it is rebooted, it can be accessed through SSH.
- ☑ To create an access-list to limit who and what can have access to your IDS sensor by SSH, use the *ssh host-key* command to start the process, or use the IDM. The Public Exponent, the Key Modulus Length, and the Public Modulus will be needed to configure the ssh known hosts.
- ☑ From within the (*sensor config-SshKnownHosts*) prompt, use the *show settings* command to verify that the known host has been added.

- ☑ The Public Exponent, Key Modulus Length, and the Public Modulus can be found in the `/etc/ssh` directory on most Unix/Linux computers, or in the application directory on Windows computers.

Configuring Remote Access

- ☑ To access a version 3.0 or 3.1 on either a 4210, 4220, or 4230 IDS sensor remotely, changes in the BIOS need to be made to allow the sensor to redirect output to the serial or console port by default.
- ☑ In version 4.x, Telnet is disabled by default so you can use the `setup` command or the `telnet-server` command from the command line to enable Telnet access.
- ☑ In version 3.x, the default security level setting is Low, which allows Telnet and FTP. To adjust this, use either Medium (Telnet disabled) or High (Telnet and FTP disabled) from the `sysconfig-sensor` utility.

Applying the Sensor Configuration

- ☑ When you upgrade from version 3.x to 4.x sensor code, you need to switch the cables on the monitoring interface and the command and control interface.
- ☑ Make sure the monitoring interfaces are in Group 0 for the interfaces to be able to see the network traffic.

Configuring Logging

- ☑ Event logging is available only in 3.x software; this feature was dropped in version 4.x software. IP logging is the same in both versions of software.
- ☑ The sensor can be configured to automatically export event log files by FTP through the IDM. To do so, choose **Configuration | Logging | Export Event Logs**. The IP address, directory, FTP username, and FTP password, will need to be input.
- ☑ To configure IP logging, use the IDM and choose **Configuration | Logging | IP Logging**, then enter the IP address by clicking **Add**.

- ☑ To configure IP logging from the command line, use the following *iplog* command: *iplog 0 192.168.50.14*. The 0 is the group. The IP address to be logged must be supplied.

Upgrading the Sensor

- ☑ The installation CD of 4.x software from Cisco is needed to upgrade from 3.x to 4.x software.
- ☑ Existing information can be retained by using the IDM, choosing **Administration | Diagnostics**, and selecting **Run Diagnostics**. Select **Menu | Save As** to save the results.
- ☑ The upgrade CD will automatically start when the sensor boots and you will have the choice of either upgrading through the serial port or through the local keyboard and monitor. The upgrade does not require any more user intervention until it's complete, whereupon the CD must be removed before rebooting.
- ☑ The new username and password is *Cisco/Cisco*. The user will be prompted to change the password upon the first login to the newly upgraded sensor.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: I want to reset my sensor to the default state. How can I do this?

A: From the IDM application, select configuration and then restore defaults. You can also use the *recover* command from the command line to recover the complete installation from the recovery partition.

Q: I don't want to completely reset my sensor, I just want to change the IP address and the gateway IP address. What is the easy way to do this?

A: From the command line interface, use the command *setup* and step through the questions to reset the IP address for the sensor and the gateway.

Q: I installed an update but it is not working correctly. Is there any way I can remove the update without having to restore the entire sensor?

A: From the command line interface, use the command *downgrade* to remove the last installed update.

Q: What Cisco IDS Sensor Secure Shell operation enables a network security administrator to remove hosts from the list of those previously connected to the devices?

A: Manage the Sensor's known host file.

Q: Which two Cisco IDS Sensor services must be disabled in order to only allow secure device management?

A: Telnet and FTP.

Q: What is the default user ID and passwords for my IDS sensor?

A: For 3.1, the default user ID is *netrangr* or *root*, and the password is “attack.” For the newer 4.x software, the default user ID is *Cisco* with the password of “Cisco.” The IDS sensor will prompt you to change it the first time you log in.

- Q:** I upgraded my sensor from 3.1 to 4.x and now I cannot log in to the sensor through the network. What's the problem?
- A:** When upgrading to the 4.x code, you need to switch the command and control cable with the monitoring port cable.
- Q:** When I try to Telnet or ssh to my IDS sensor with 4.x code, I can't make the connection. I have checked the cables and they are plugged into the correct ports.
- A:** You need to configure the access-list to allow the subnet to have access to the IDS sensor. To do this, use the *service host* command and then the *networkParams* command.
- Q:** I want to export my IP log files. What format are they in?
- A:** The IP log file is in the tcpdump format.

Configuring the Cisco IDSM Sensor

Solutions in this Chapter:

- Understanding the Cisco IDSM Sensor
- Configuring the Cisco IDSM Sensor
- Updating the Cisco IDSM Sensor
- Troubleshooting the Cisco IDSM Sensor

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

The Cisco IDSM sensor blade is viewed with a mixture of awe, dread, and ignorance. This sensor is certainly one of the least understood and underutilized sensors in the Cisco IDS product line. In part, this stems from the cost of the hardware to support the IDSM sensor module and the difficulty in finding solid information on the sensor itself. In this chapter, we try to dispel the myths of the IDSM sensor and help you understand it and use it effectively.

This chapter provides an overview of the architecture of the IDSM sensor, how it fits into the network, how to configure the sensor, and how to troubleshoot it. You will see that the sensor, even though it is a module in the Catalyst switch, is not much different than any other IDS sensor from an operational perspective. There are differences in the command line (which we'll discuss), as well as other dissimilarities, like having direct access to the span ports and VLAN access-lists which more conventional IDS sensors do not have. There are also a few things the IDSM can't do that more conventional IDS sensors can. We will discuss some of the differences between the IDSM and conventional IDS sensors, which are now falling by the wayside with the advent of the new IDSM sensor version 2 released by Cisco.

We would be remiss if we did not explore one of the most critical skills in managing the Cisco IDSM sensor: how to apply service packs and updated signatures. As seen in earlier chapters, one of the best ways to stay ahead of threats is to keep current with both service packs and signature files, so this is a “must have” skill.

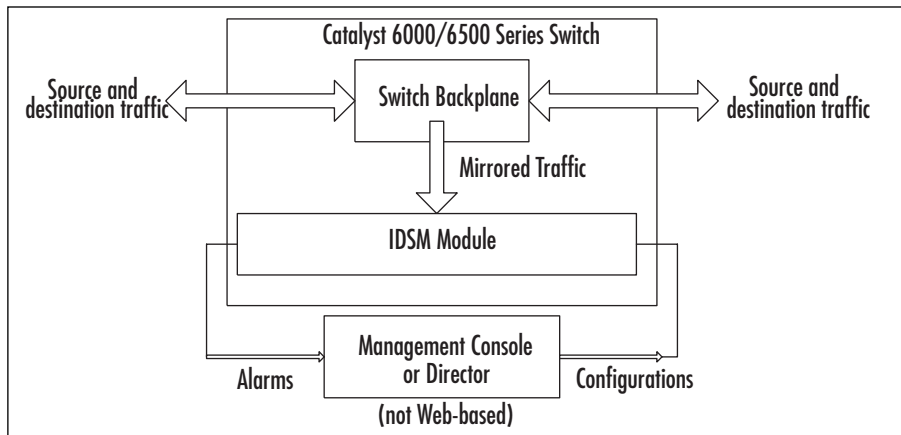
In a perfect world, everything would work correctly the very first time we configured it, but alas, we do not live in a perfect world. Therefore, we will show you how to troubleshoot the IDSM sensor should you have problems getting it to work correctly.

Understanding the Cisco IDSM Sensor

The IDSM sensor module differs from other, more conventional IDS sensors from Cisco by being a blade- or module-based solution. Unlike the 4230 sensor that uses a form of Solaris as the OS and runs on a dual Pentium PC in a box, the first generation of IDSM blades uses an embedded form of Windows, while the second generation uses Red Hat Linux as the OS. In both types of sensor, there is a complete PC on a card with hard drive, memory, ports, and the OS. The IDSM card occupies one slot on a Cisco Catalyst 6000/6500 series switch.

You have to stack the IDSM sensors in the switch to aggregate the throughput of the IDSM which until recently was 250 Mbps of traffic. Figure 6.1 shows an architectural view of the IDSM. Notice how traffic flows through the switch, and how the IDSM can see the traffic.

Figure 6.1 The Cisco IDSM Architecture



This architecture is one of the reasons the Cisco IDSM is so little understood. The fact that the IDSM sensor actually sits in the switch as a blade and directly connects to the switch backplane confuses many people. Contributing to the confusion is the difficulty in finding any information on installing and configuring the IDSM sensor. The Cisco IDSM sensor is not a cheap device and, as a result, there are not very many IDSM sensors installed. This cost is multiplied when you consider that you first need the Catalyst 6000/6500 chassis before installing and using the IDSM sensor. The cost of the Catalyst switch chassis alone precludes having one of the IDSM sensors in your home lab or even in most commercial Cisco labs.

As we mentioned, the IDSM sensor has a processor, RAM, BIOS, and a hard drive, which is typical of a PC-like architecture. Unlike a PC, however, there are two ports. The first port is the monitoring port, while the second is called a Control Port. The monitoring port is port 1 and is set up to automatically trunk all VLANs by default. Port 1 uses 802.1q as the trunking protocol. The control port is port 2, which will have an IP address assigned to it through which the IDSM sensor is managed.

The older IDSM v1 sensor was much more limited in what it could, and could not, do relative to the IDS appliance. For example, the older IDSM sensor

could not offer TCP resets, command-line management, Web-based management, or IP logging. The new IDSMv2 sensor offers all of these improvements and more, such as not having to use the Cisco Postoffice protocol to communicate between sensors or the director. Since the new IDSM-2 sensor can run 4.0 and newer code, the sensor can use a new protocol called Remote Data Exchange or RDEP. This new protocol is not available to the IDSM-1 since the IDSM-1 code development stopped at version 3.1. However, with version 3.0 and 3.1 code, the port number used for the Cisco Postoffice Protocol can now be specified to whatever the administrator would like the port number to be.

The IDSM-1 sensor can monitor up to 100 Mbps of traffic based on a minimum packet size of 64 bytes, while the IDSM-2 sensor can monitor up to 600 Mbps of traffic. In both IDSM sensor units, the traffic is captured directly off the switch backplane. It is possible to use more than one IDSM sensor to scale the monitoring ability of the IDSM sensor and switch combination. If the Cisco switch has a Policy Feature Card (PFC), the switch can be configured to use VACLs to capture the traffic. All supported Cisco switches can use the Switch Port Analyzer (SPAN) option. If the switch has an MSFC, it can use MLS IP to capture traffic. The IDSM differs from the conventional IDS in that it can monitor multiple VLANs simultaneously by having the monitoring port configured as a trunk. Since the IDSM sensor sits on the backplane of the switch and has direct access to the data flow, it does not impact the switch performance. The signatures and attacks that the IDSM can detect mirror those used by the 4200 series of sensors.

Configuring the Cisco IDSM Sensor

Before we attempt to configure the IDSM v1 sensor, we need to verify that we have the correct hardware and software for the IDSM to function. The following are required:

- Catalyst OS 6.1(1) or later
- PFC for the VACL feature
- Supervisor 1A or 2
- MSFC or MFSC2 (this is optional)

Designing & Planning...

IDSM Sensor Catalyst Requirements

This chapter spans two IDSM sensors. There is the first version that is still on the Cisco tests, and a second—the new IDSM-2 (version 2)—meant to replace IDSM version 1. The requirements for the two units are very different. The following are needed for the IDSM-2 sensor:

- Catalyst OS 7.5(1) min
- Native Cisco IOS software 12.1(19)E min
- Supervisor 1A
- Supervisor 1A/PFC2
- Supervisor 1A/MSFC1
- Supervisor 1A/MSFC2
- Supervisor 2
- Supervisor 2/MSFC2

This new version of the IDSM sensor uses Red Hat Linux as the OS and has a Pentium 1.13GHz processor on board. The new code has a Web-based IDM native, unlike version 1 which must have a director.

In order to configure IDSM-1, you will treat the IDSM sensor much like any other blade on the switch. You can access the blade through a Telnet session or through the management port. Once the IDSM is configured, you can configure the IDSM sensor to be accessible directly by Telnet instead of Telneting to the switch and then using the *session* command. To begin the configuration, we will first Telnet to the switch and start by using the *show module* command on the switch to see where the IDSM sensor is located in the chassis. We also want to verify that the module is powered up and enabled. In Figure 6.2, we see that the module has just been powered up and is coming online.

Figure 6.2 The IDSM Sensor Initializing and Coming Online from Bootup

```

switch>(enable) 2003 Jul 13 03:30:25 PDT -07:00 %SYS-3-SUP_
    OSBOOTSTATUS:Star
ting IDSM Diagnostics
switch>(enable) 2003 Jul 13 03:31:05 PDT -07:00 %SYS 3SUP_
    OSBOOTSTATUS:IDSM diagnostics completed successfully.
2003 Jul 13 03:31:14 PDT -07:00 %SYS-5-MOD_OK:Module 4 is online
2003 Jul 13 03:31:14 PDT -07:00 %SYS-3-MOD_PORTINTFINSYNC:Port Interface
    in sync for Module 4
2003 Jul 13 03:31:14 PDT -07:00 %PAGP-5-PORTFROMSTP:Port 4/1 left bridge
    port 4/1
2003 Jul 13 03:31:15 PDT -07:00 %DTP-5-TRUNKPORTON:Port 4/1 has become
    dot1q trunk
2003 Jul 13 03:31:15 PDT -07:00 %PAGP-5-PORTTOSTP:Port 4/2 joined bridge
    port 4/2
2003 Jul 13 03:31:15 PDT -07:00 %PAGP-5-PORTTOSTP:Port 4/1 joined bridge
    port 4/1

```

If you had needed to power up the module, you would first use the *show module* command to get the module number and then use the *set power up <module>* command to turn the module power on. In Figure 6.3, we see the results of *show module*.

Figure 6.3 The *show module* Command Results

```

switch>(enable) show module

```

Mod	Slot	Ports	Module-Type	Model	Sub	Status
1	1	2	1000BaseX Supervisor	WS-X6K-SUP2-2GE	yes	ok
15	1	1	Multilayer Switch Feature	WS-F6K-MSFC2	no	ok
2	2	48	10/100BaseTX Ethernet	WS-X6348-RJ-45	no	ok
3	3	16	1000BaseX Ethernet	WS-X6516-GBIC	no	ok
4	4	2	Intrusion Detection System	WS-X6381-IDS	no	ok

```

:::output truncated for brevity:::

```

NOTE

If the status of the module reads “other,” then the IDSM sensor is not online yet.

In this example, we see that module 4 is our IDSM sensor and that the status is listed as “ok.” It has power and is enabled at this point. Now we can connect to the module by using the set-based command of *session <module>*.

```
switch>(enable) session 4
```

When you use the *session* command, it starts a process that will act like a Telnet session and give us access to the IDSM sensor module. We have to log into the IDSM sensor and this requires a user ID and password. Since this is a new configuration, the default user ID of *ciscoids* and the password of *attack* will be used.

The Cisco IDSM sensor has three command modes to work with. The following list shows us the three modes and what we can do in each mode:

- **Exec** This allows the administrator to perform commands such as reboot, setup, show, and shutdown the IDSM sensor.
- **Configuration** This mode allows the administrator to change the password, assign Telnet access, upgrade IDSM signatures, and add or remove service pack.
- **Diagnostic** This mode is how the administrator can upgrade the sensor, use network commands such as PING, show communication with the *nrconn* command, and show various reports.

To start configuring the IDSM sensor, we will use the command *setup*. This will walk us through all the parameters that the IDSM sensor needs in order to have its initial configuration completed. In Figure 6.4, we see the various parameters we will be prompted to configure for the IDSM sensor.

Figure 6.4 Using the IDSM *setup* Command for Initial Configuration

```

Tera Term - 10.1.1.11 VT
File Edit Setup Control Window Help
Continue with configuration dialog? [yes]:
Enter virtual terminal password[Use Current>]:
Enter sensor IP address[10.1.1.101]:
Enter sensor netmask [255.255.0.0]:
Enter sensor default gateway [10.1.1.1]:
Enter sensor host name [pacsensor2]:
Enter sensor host id [1]:
Enter sensor host post office port [45000]:
Enter sensor organization name [test]:
Enter sensor organization id [100]:
Enter director IP address [10.1.1.64]:
Enter director host name [idsmc]:
Enter director host id [10]:
Enter director host post office port [45000]:
Enter director heart beat interval [5]:
Enter director organization name [pactest]: test
Enter director organization id [100]: 100

```

This configuration is the same that we have read about and used in earlier chapters of this book for the more traditional Cisco IDS sensor appliances. We will be configuring an IP address, subnet mask and default gateway for the sensor. Then we will assign a host name for the sensor, a host ID, the post office protocol port, and the organization name/ID. We then need to configure the director information since we cannot manage the IDSM sensor from the command line. It requires the director.

NOTE

In the IDSM software version 4, there is now a CLI to manage the IDSM sensor. This version of software cannot run on the old style IDSM version 1 sensor. Thus, for the current Cisco certification test, there's no way to manage the IDSM sensor from the command line.

In Table 6.1, we see the list of parameters and the values each needs in order to be configured.

Table 6.1 Listing of Cisco IDSM Sensor Setup Parameters

Cisco IDSM Setup Parameters	Value	Description of Parameter
IDSM Virtual Terminal UserID	<userid> default is ciscoids	IDSM session user ID
IDSM Virtual Terminal Password	<password> default is attack	IDSM session password
Sensor IP address	<aaa.bbb.ccc> IP address	IP address of the sensor

Continued

Table 6.1 Listing of Cisco IDSM Sensor Setup Parameters

Cisco IDSM Setup Parameters	Value	Description of Parameter
Sensor Subnet Mask	<aaa.bbb.ccc> Subnet Mask	Subnet Mask of the sensor
Sensor Default Gateway	<aaa.bbb.ccc> Default Gateway	Default Gateway for the sensor
Sensor Host Name	<hostname>	Name of the Sensor
Sensor Host ID	<1–65535>	Numeric ID of the sensor
Sensor Host Postoffice Port	<1–65535> default is 45000	Postoffice protocol port to use
Sensor Organization Name	<name>	ID of a group of Cisco IDS devices
Director IP Address	<aaa.bbb.ccc>	IP address of the Director device
Director Host Name	<name>	Name of Director Device
Director Host Postoffice Port	<1–65535> default is 45000	Postoffice protocol port to use
Director Heart Beat Interval	<1–65535> default is 5	System heartbeat to monitor routes
Director Organization Name	<organization_name> case sensitive	Name of Director Organization
Director Organization ID	<1–65535>	ID of Director Organization

NOTE

For the Director and sensor name, you can have up to 255 characters. They are case-sensitive and spaces are invalid. The “_” and “-” are acceptable to use.

Once all the information is entered, we need to save it to the IDSM sensor. In Figure 6.5, we see the final screen before the information is saved and applied along with the warning of the required reboot. This reboot betrays the Windows pedigree of the Cisco IDSM sensor.

Figure 6.5 Final Configuration of IDSM Sensor before Application and Save

```

Tera Term - 10.1.1.11 VT
File Edit Setup Control Window Help
Configuration last modified Never
Sensor:
IP Address:          10.1.1.101
Netmask:             255.255.0.0
Default Gateway:    10.1.1.1
Host Name:          pacsensor2
Host ID:            1
Host Port:          45000
Organization Name:  test
Organization ID:    100

Director:
IP Address:          10.1.1.64
Host Name:           idsmc
Host ID:             10
Host Port:           45000
Heart Beat Interval (secs): 5
Organization Name:  test
Organization ID:    100
WARNING: Applying this configuration will cause all configuration files
to be initialized and the card to be rebooted.
Apply this configuration?: 2003 Jun 13 09:02:43 PDT -07:00 %CDP-4-NULANMISMATCH:
Native vlan mismatch detected on port 3/5

```

NOTE

The 4.0 IDSM sensor code for Version 2 sensor does not use the Postoffice protocol. Instead, it uses Remote Data Exchange Protocol (RDEP).

NOTE

Once the initial configuration of the IDSM is completed, there are some concepts that we need to know about when working with the IDSM. One of these is the Virtual Lan Access List (VACL). This is an ACL applied to a VLAN. To configure the VACL, we take the following steps:

1. Create a VACL that can capture interesting traffic
2. Commit the VACL to memory
3. Map a VACL to the VLANs
4. Assign the sensor monitoring port as a VACL capture port.

VACLs are one of two methods used to capture traffic for analysis on the switch. The second way is to use SPAN to mirror VLANs to capture the traffic. The VACL offers a much more granular approach to the capture than SPAN. A critical point to remember is that VACLs have the same implicit deny at the end that other ACLs have. All traffic that does not match the VACL will be dropped.

In order to configure the IDSM to do something useful like examine traffic, we need to perform a series of tasks. One of the first tasks is to initialize the IDSM, which includes the configuration of the post office parameters by using the *setup* command. We just completed that task, so we move on to the second step of assigning the Command port to the VLAN that will allow communication to the Director. This is accomplished by the *set vlan* command on the switch. This is where it is helpful to have two Telnet sessions open on the switch. There would be one Telnet session for configuring the IDSM module and one Telnet session for configuring the switch itself. We will configure the 6500 switch to capture traffic for the IDS by using either SPAN sessions, VACLs, or MLS IP capturing, depending on the configuration of the switch. The *set vlan* command is used as the following to assign the command and control port to a VLAN:

```
set vlan <vlan_number> <src_module/src_ports>
```

The parameter *<vlan_number>* is the number identifying the VLAN we want to place the port into. The parameter *<src_module/src_port >* is the slot number of the module that has the ports to be included in the VLAN. The *src_port* is the actual port to be placed in the VLAN. The command will look like this:

```
switch>(enable) set vlan 1 4/2
```

This will assign VLAN 1 to module 4 and port 2, which is the command and control port. This assumes the director is on VLAN1. If the director is not on VLAN 1, then we need to have routing enabled between VLAN 1 and whatever the VLAN is that the director is located on.

Configuring & Implementing...

To Clear an Old Configuration

The command *clear config* will erase existing configuration and disable the IDSM module. To bring the module back online, you can use the *set module enable <module number>* command. If you need to power off the module or to cycle the power, use the *set module power <up/down>* command. Remember that it takes a few minutes for the IDSM to shut down and recycle itself to where you can session back into it.

Setting Up the SPAN

SPAN is a method of mirroring or duplicating traffic from a single VLAN, a group of VLANs, a single port, or group of ports to a single monitoring port. This provides a way for an IDS or a sniffer to “see” a flow of traffic without actually having to be in the flow of traffic. There is a limit of four Tx (transmit sessions) or two Rx (receive sessions). If the command *both* is used, the limit is still two SPAN sessions. The SPAN port can be 10, 100, or 1000 Mbps. The command to set the span is slightly different between various switches, but since we are working with the IDSM, we will be working with the Cisco 6000, 6500 series chassis, prompting us to use the *set* command. To configure SPAN, use the following command:

```
Set span <src_mod/src_port> <dest_mod/dest_port> [rx | tx | both]
      [create]
```

What we are saying here is that the `<src_mod/src_port>` is the source module and source port, while the `<dest_mod/dest_port>` is the destination module and destination port. The `[rx | tx | both]` tells the switch if we are to have Rx only, Tx only, or both. The `[create]` tells the switch that we are creating the SPAN. If you do not use the argument *create* at the end of the *set span* command and you have only configured a single session, that session is overwritten.

```
switch> (enable) set span 2/5 4/1 both create
Created Port 4/1 to monitor transmit/receive traffic of Port 2/5
Incoming Packets disabled. Learning enabled. Multicast enabled.
switch> (enable)
```

We can disable the SPAN session either in pieces or all at once, as shown in the following example:

```
switch> (enable) set span disable all
This command will disable your span session(s).
Do you want to continue (y/n) [n]?y
Disabled all sessions
switch> (enable)
```

Setting Up the VACLs

A VACL can be configured to capture traffic for the IDSM from either a single VLAN or multiple VLANs. This differs from the conventional ACL that we use

on routers where the rules apply to a given interface. This also differs significantly from SPAN in that with VACLs you can filter down to a specific type of packet or flow you want to look at. The VACL applies to all packets and the processing is done in hardware. If there is a field specified for the VACL that does not apply, then it is ignored. As we read earlier, port 1 (or the monitoring port) is, by default, a trunked port and will monitor all VLANs that have an ACL applied to capture traffic. If you want to capture specific VLAN traffic, you need to clear the VLANs other than the ones you want to capture. There can be only one VACL per protocol applied to a single VLAN.

To configure a VACL to capture any traffic from a SPAN port, use the following command:

```
set security acl ip <acl name> permit < > capture
set security acl ip
SPANCOPY permit any any capture
```

Then we commit the VACL using this command:

```
commit security acl
```

Next, we map the VACL to the VLANs or VLAN of interest to us:

```
set security acl map <acl name> [vlans]
```

Finally, we add the IDSM port 1 to the VACL capture list

```
set security acl capture <idsm_mod/1>
```

```
switch1> (enable) set security acl capture-ports 4/1
Successfully set 4/1 to capture ACL traffic.
```

NOTE

By default, port 1 on the IDSM is set as the security ACL capture port.

For example, if we wanted to catch all Web traffic for the IDSM, we would use a VACL configured like the following example:

```
switch>(enable) set security acl ip WEBTRAF permit tcp any host 10.10.
10.50 eq 80 capture
switch>(enable) set security acl ip WEBTRAF permit ip any any
switch>(enable) commit security acl WEBTRAF
```

```
switch>(enable) set security acl map WEBTRAF 10
switch>(enable) set security acl capture-ports 4/1
```

This sets up the capture for only Web traffic, permitting everything else to pass the IDSM. The permit any any is the magic key to let the rest of the traffic go past the IDSM. We then commit the VACL called WEBTRAF. The security ACL map is set to WEBTRAF, and VLAN 10 is mapped to the ACL. Lastly, we set the ACL to use module 4, and employ port 1 as the capture port for the IDSM.

Configuring Trunks to Manage Traffic Flow

A method of managing the amount of traffic seen by the IDSM sensor is to manage the trunks and VLANs on the trunks. An example of this would be to have a single IDSM sensor and the need to monitor a single VLAN. This can be accomplished by clearing VLANs from the IDSM sensor monitoring port and then assigning the VLAN that we are interested in back to the monitoring port. In the following example, we step through the process. We have three VLANs, VLAN 501, VLAN 502, and VLAN 503 on module 4, port 1. So we will first clear the VLANs from the port by using this command:

```
switch>(enable) clear trunk 4/1 2-1005, 1025-4094
```

Now we will reassign VLAN 502 back to the monitoring port

```
switch>(enable) set trunk 4/1 502
switch>(enable) set vlan 502 4/1
```

We now assign module 4 and port 1 as the capture port using the following command:

```
switch>(enable) set security acl capture-ports 4/1
```

Verifying the Configuration

To verify that the IDSM is configured correctly, we have several commands at our disposal. The most common command as you might guess is just like a router, the *show config* command at the switch. This will give us the entire configuration of the switch. The next command of great use is called *show span* and tells us to span the configuration on the switch. We can use the *show security acl*, which shows us the VACL settings.

On the IDSM itself, we can use the same *show configuration* command to get the config of the IDSM. The *show eventfile current* command allows us to look at the logfiles of the IDSM.

Updating the Cisco IDSM Sensor

Updating the IDSM sensor might result from a need to move to newer code, or because the current image has been corrupted. A different reason for updating (or more appropriately: to recover the IDSM sensor) is that the password has been forgotten. In any case, the image of the IDSM sensor OS needs to be replaced. The IDSM sensor has two partitions on the internal hard drive. The first is the application partition or *hdd:1*. The second is the maintenance partition or *hdd:2*. Both of these partitions contain a complete operating system and therefore the IDSM sensor can be booted from either partition. The partition that the IDSM sensor booted from is called the *active partition*. Any updates to the IDSM sensor operating system must be done to an offline partition so the production partition would need to be offline by booting to the maintenance partition.

Be aware that when updating the IDSM sensor, the process must be done at the command line. To update the IDSM requires administrative privileges to the maintenance partition. This is why we reboot to the maintenance partition and log in as *ciscoids*, using the password *attack*. If no upgrade has been done before, we need to set the network settings for the IDSM sensor to communicate with the network—in particular, to communicate with the FTP server that holds the new CAB files for the update. This setting of the network parameters in the maintenance mode is accomplished by using the *ids-installer* command. The update file that the *ids-installer* will use must reside on an FTP server or the IDS Director. In the following examples, we used an FTP server called “Cerberus FTP Server,” which is free for personal and non-profit use and can be found at www.cerberusftp.com.

Booting the IDSM Sensor from Partition 2

In order to boot from a particular partition, we can set the default partition by using the command *set boot device*, as shown in the following example:

```
switch> (enable) set boot device hdd:2 4
Device BOOT variable = hdd:2
Warning: Device list is not verified but still set in the boot string.
switch> (enable)
```

Alternatively, we can have the IDSM boot from a given partition temporally, as shown in the following example.

```
Switch> (enable) reset 4 hdd:2
```

This command will reset module 4 and have it boot off the boot device: hdd number 2, which is the maintenance partition. We can see this in Figure 6.6.

Figure 6.6 Booting IDSM Module 4 off Partition 2

```
switch> (enable) reset 4 hdd:2
This command will reset module 4.
Unsaved configuration on module 4 will be lost
Do you want to continue (y/n) [n]? y
Module 4 shut down in progress, please don't remove module until shutdown
  completed.
2003 Jun 15 07:29:44 PDT -07:00 %PAGP-5-PORTFROMSTP:Port 4/1 left bridge
  port 4/1
2003 Jun 15 07:29:44 PDT -07:00 %DTP-5-NONTRUNKPORTON:Port 4/1 has become
  non-trunk
2003 Jun 15 07:32:01 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:Starting IDSM
  Diagnostics
2003 Jun 15 07:32:41 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:IDSM diagnostics
  completed successfully.
2003 Jun 15 07:32:50 PDT -07:00 %SYS-5-MOD_OK:Module 4 is online
2003 Jun 15 07:32:51 PDT -07:00 %SYS-3-MOD_PORTINTFINSYNC:Port Interface
  in sync for Module 4
2003 Jun 15 07:32:51 PDT -07:00 %DTP-5-TRUNKPORTON:Port 4/1 has become
  dot1q trunk
2003 Jun 15 07:32:51 PDT -07:00 %PAGP-5-PORTTOSTP:Port 4/1 joined bridge
  port 4/1
2003 Jun 15 07:32:51 PDT -07:00 %PAGP-5-PORTTOSTP:Port 4/2 joined bridge
  port 4/2
2003 Jun 15 07:33:21 PDT -07:00 %CDP-4-NVLANMISMATCH:Native vlan mismatch
  detected on port 3/5
switch2> (enable)
```

As we saw in Figure 6.6, there are several messages that tell us module 4 is being reset and that diagnostics are being run. We can see the bridge port messages of ports 1 and 2 leaving the switch and coming back into the switch.

In Figure 6.7, we are logging into the IDSM after the reset to partition 2. We can see that the hostname of the IDSM is now shown as maintenance.

Figure 6.7 Logging in to the Maintenance Partition of the IDSM

```
switch> (enable) session 4
Trying IDS-4...
Connected to IDS-4.
Escape character is '^]'
login: ciscoids
Password: attack
maintenance# show
configure      Enter configuration mode
diagnostics    Enter diagnostic command menu
exit           Exit from Telnet session
show           Show system parameters
shutdown       Shutdown the system
maintenance#
```

We can also see that there are very limited commands from this version of the IDSM sensor operating system to work with. No IDS commands are available from the maintenance partition. To get back to our production IDSM operating system, all we need to do is log out of the IDSM sensor and use the *reset module* command but leave the boot device off.

Now that we have learned about how to boot the IDSM sensor into the maintenance mode using the second partition, we are ready to upgrade the OS of the IDSM. In the following example, we will upgrade the IDSM V1 sensor from version 2.5 to 3.0 of the OS. The first step is to boot to the second partition just as we did before using the *reset* command, as shown in Figure 6.8.

Figure 6.8 Using the *reset* Command to Boot to the Maintenance Partition

```
Switch>(enable) #reset 4 hdd:2
This command will reset module 4.
Unsaved configuration on module 4 will be lost
Do you want to continue (y/n) [n]? y
```

Continued

Figure 6.8 Using the `reset` Command to Boot to the Maintenance Partition

```

Module 4 shut down in progress, please don't remove module until shutdown
  completed.
Switch> (enable) 2003 Jun 15 07:29:44 PDT -07:00 %PAGP-5-PORTFROMSTP:Port
  4/1 left bridge port 4/1
2003 Jun 15 07:29:44 PDT -07:00 %DTP-5-NONTRUNKPORTON:Port 4/1 has become
  non-trunk
2003 Jun 15 07:32:01 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:Starting IDSM
  Diagnostics
2003 Jun 15 07:32:41 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:IDSM diagnostics
  completed successfully.
2003 Jun 15 07:32:50 PDT -07:00 %SYS-5-MOD_OK:Module 4 is online
::text truncated for clarity::

```

Upgrading the IDSM Sensor

Remember that the `hdd:2` will boot the IDSM off the OS on the second partition. Once the IDSM has completely rebooted and run through its diagnostics, we are ready to configure the maintenance IDSM OS for a network connection. First, we will *session* into the IDSM and log in as we have done before. Then we will use the `ids-installer` command to verify any network configuration, or to add the network information, as shown in the following example:

```

switch-2> (enable) session 4
Trying IDS-4...
Connected to IDS-4.
Escape character is '^]'.

```

```

login: ciscoids
Password: attack

```

We change to the diagnostic mode by typing in **diag**, and then we verify the existing network configuration, if there is one:

```

maintenance#(diag) ids-installer netconfig /view
IP Configuration for Control Port:
IP Address           : 0.0.0.0
Subnet Mask          : 0.0.0.0
Default Gateway      : 0.0.0.0
Domain Name Server   : 77.1.1.1

```

```
Domain Name      : cisco
Host Name       : CISCO_IDS
```

```
maintenance(diag)#
```

To either change the network settings or to configure the network settings, we use the *ids-installer* command and the following command-line parameters:

```
ids-installer netconfig /configure /ip=ip_address /subnet=subnet_mask
/gw=default_gateway /dns=dns_server /domain=nw_domain
/hostname=host_name
```

In the following example of the *ids-installer* command, we see how to change the network configuration in the diag mode of the maintenance partition:

```
maintenance(diag)# ids-installer netconfig /configure /ip=10.10.10.101
/subnet=255.255.0.0 /gw=10.10.10.1 /hostname=testids
```

In Table 6.2, we show the *ids-installer netconfig* parameters and what they mean:

Table 6.2 *ids-installer netconfig* Parameters

Parameters	Notes
netconfig	This keyword specifies that a network configuration action will take place.
/configure	This keyword specifies the configuration of port parameters.
/ip	This keyword specifies an IP address as a parameter.
ip_address	This is the IP address of the IDSM command and control port (port 2).
/subnet	This keyword specifies the subnet mask address parameter.
Subnet	This is the subnet mask for the IDSM command and control port.
/gw	This keyword specifies the Default Gateway parameter.
default_gateway	This is the IP address of the default gateway for the IDSM.
/dns	This is an OPTIONAL keyword that specifies the DNS server.

Continued

Table 6.2 *ids-installer netconfig* Parameters

Parameters	Notes
ip_address	This is the IP address of the optional DNS server parameter.
/domain	This is an OPTIONAL keyword that specifies a network domain name.
nw_domain	This is the network domain name assigned to the command and control port.
/hostname	This OPTIONAL keyword specifies the hostname assigned to the IDSM.
host_name	This is the hostname assigned to the IDSM.

To install the image to the partition, we use the *ids-installer* command mentioned earlier. This command has several parameters that can be used to install the image. The command line is structured as shown in this example:

```
ids-installer system /nw /install /server=ip_address /user=username
    /dir=directory /prefix=update_file /save=yes
```

In Table 6.3, we see a listing of the command-line arguments that can be used:

Table 6.3 *ids-installer* Command-Line Parameters to Install an Image

Parameters	Notes
system	This keyword specifies that a system action will be performed.
/nw	This keyword specifies that the installation of the image will be done from the network.
/install	This keyword specifies the system action will be to install.
/server	This keyword specifies that the image file will be on an FTP server.
ip_address	This is the IP address of the FTP server.
/user	This specifies that a username is required to log in to the FTP server.
username	This is the username required.
/dir	This specifies that the files are stored in a specific directory.
directory	This is the directory name of where the files are stored.

Continued

Table 6.3 *ids-installer* Command-Line Parameters to Install an Image

Parameters	Notes
/prefix	This specifies that the update filename prefix is required.
update_file	This is the update filename that will be installed but without the extension.
/save	This keyword specifies that the image will be saved as a cached copy.
yes no	If yes, then the image will be cached. If no, the image is installed but not cached.

In the following example, we will have the IDSM do a network install of the new code from an FTP server and a certain user account:

```
maintenance(diag)# ids-installer system /nw /install /server=10.1.2.11 /
user=ciscoids /save=yes /dir='ftpupload' /prefix=IDSMk9-a-3.0-1-S4
```

The FTP server is 10.1.2.11 using a user ID of ciscoids. We are saving the image to cache, and the directory name on the FTP server is ftpupload. The filename is IDSMk9-a-3.0-1-S4 but without the .bin extension on it.

In Figure 6.9, we see the complete upgrade of an IDSM V1 in progress. Note that it has been shortened in some places for brevity.

Figure 6.9 Complete Upgrade of IDSM V1

```
maintenance(diag)# ids-installer system /nw /install /server=10.1.2.11
/user=ciscoids /save=no /dir='ftpupload' /prefix=IDSMk9-a-3.0-1-S4
Please enter login password: *****
Downloading the image.. File 01 of 05
Downloading the image.. File 02 of 05
Downloading the image.. File 03 of 05
Downloading the image.. File 04 of 05
Downloading the image.. File 05 of 05

FTP STATUS: Installation files have been downloaded successfully!
Validating integrity of the image... PASSED!
Formatting drive C:\....
Verifying 4016M
0 percent completed.1 percent completed.2 percent completed.3 percent
completed.4 percent completed.5 ::shortened for brevity::
```

Continued

Figure 6.9 Complete Upgrade of IDSM V1

```

100 percent completed.Format completed successfully.
4211310592 bytes total disk space.
4206780416 bytes available on disk.

Volume Serial Number is C49D-CFDA
Extracting the image...

::shortened for brevity::

STATUS: Image has been successfully installed on drive C:\!
maintenance(diag)# exit
maintenance# exit
switch>(enable) reset 4 hdd:2
This command will reset module 4.
Unsaved configuration on module 4 will be lost
Do you want to continue (y/n) [n]? y
Module 4 shut down in progress, please don't remove module until shutdown
  completed.
switch>(enable) 2003 Jun 17 13:15:06 PDT -07:00 %SYS-3-
  SUP_OSBOOTSTATUS:Starting IDSM Diagnostics
2003 Jun 17 13:15:49 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:IDSM diagnostics
  completed successfully.
2003 Jun 17 13:15:49 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:IDSM has not been
  configured. Network is unguarded!
2003 Jun 17 13:15:49 PDT -07:00 %SYS-3-SUP_OSBOOTSTATUS:Use session to
  login to IDSM and run setup.
2003 Jun 17 13:15:58 PDT -07:00 %SYS-5-MOD_OK:Module 4 is online

```

Verifying the IDSM Sensor Upgrade

Once the IDSM sensor has rebooted and completed its self-diagnostics, we need to log back into the IDSM sensor and run the *setup* command since the original configuration has been overwritten. We can see in Figure 6.10 that the new configuration is void of data except for the default IP address and mask. We also see that the version of the software is 3.0(1)S4.

Figure 6.10 Verifying the Successful Upgrade of the IDSM Sensor

```
switch>(enable) session 4
Trying IDS-4...
Connected to IDS-4.
Escape character is '^]'.
login: ciscoids
Password:
# show config
Using 38240256 out of 267702272 bytes of available memory
!
Using 439668736 out of 4211310592 bytes of available disk space
!
Sensor version is : 3.0(1)S4 ;
```

Note that the preceding line shows our new version number of the OS.

```
!
Sensor application status:
nr.postofficed      not running
nr.fileXferd       not running
nr.loggerd         not running
nr.packetd         not running
nr.sapd            not running

Configuration last modified Never
Sensor:
IP Address:         10.0.0.1
Netmask:           255.0.0.0
Default Gateway:
Host Name:         Not Set
Host ID:          Not Set
Host Port:        45000
Organization Name: Not Set
Organization ID:  Not Set
Director:
IP Address:       Not Set
Host Name:       Not Set
```

```

Host ID:                               Not Set
Host Port:                             45000
Heart Beat Interval (secs): 5
Organization Name:                     Not Set
Organization ID:                       Not Set
Direct Telnet access to IDSM: disabled
#

```

Shutting Down the IDSM Sensor

In order to disable or to remove the IDSM sensor from a live switch, we need to shut down the IDSM sensor. If we do not, given Windows tendency to corrupt on a dirty shutdown, we could easily find ourselves reinstalling the OS without the clean shutdown. The good news is that this is very easy to accomplish. As shown in Figure 6.11, just log in to the IDSM and issue a *shutdown* command.

Figure 6.11 Sample of the Module in Shutdown Mode

```

switch> (enable) session 4
Trying IDS-4...
Connected to IDS-4.
Escape character is '^]'.
login: ciscoids
Password:
# shutdown
WARNING: Shutting down the line card will disable IDS.
Continue with shutdown?: y
Shutting down the module...
# exit
switch> (enable)

```

If we use the command *show module*, we will see that the current state of the module is in the shutdown mode, as seen in Figure 6.12.

Figure 6.12 Sample of Module in Shutdown Mode

```

switch> (enable)
switch> (enable) show module 4

```

Mod Slot	Ports	Module-Type	Model	Sub Status

Continued

Figure 6.12 Sample of Module in Shutdown Mode

```

-----
4   4   2   Intrusion Detection System WS-X6381-IDS   no   shutdown
Mod Module-Name           Serial-Num
-----
4                               SAD052800JV
Mod MAC-Address(es)           Hw       Fw       Sw
-----
4   00-03-32-bd-41-3a to 00-03-32-bd-41-3b 1.1     4B4LZ0XA  3.0(1)S4
switch> (enable)

```

Now for the final command, we issue a *set power* command to actually shut off the power to the IDSM. Once this is completed, we can safely remove the IDSM from the switch even with the switch live. In Figure 6.13, we see the command and resulting output:

Figure 6.13 Sample of the *set module power* Command

```

switch> (enable)
switch> (enable) set module power down 4
Module 4 powered down.
switch> (enable) 2003 Jun 17 12:31:40 PDT -07:00 %SYS-5-MOD_PWRDN:Module
 4 powered down
switch> (enable) show module 4
Mod Slot Ports Module-Type           Model           Sub Status
-----
4   4   0   Intrusion Detection System WS-X6381-IDS   no   power-down
Mod Module-Name           Serial-Num
-----
4
Mod MAC-Address(es)           Hw       Fw       Sw
-----
4   unknown
switch> (enable)

```

To bring the IDSM sensor back online, all we do is reverse the commands. We apply power to the IDSM sensor and wait for about two minutes for the

IDSM sensor to boot up and then we enable the IDSM sensor to bring it back online. In Figure 6.14, we see the steps and results:

Figure 6.14 Bringing the IDSM Sensor Back Online from a Power-Off Condition

```
switch> (enable) set module power up 4
Module 4 powered up.
switch> (enable) 2003 Jun 17 12:32:28 PDT -07:00 %SYS-5-MOD_PWRON:Module
    4 powered up
switch> (enable) set module enable 4
Enabling module 4. Please wait until module on line.
switch> (enable)
```

Updating the IDSM Sensor Signatures and Service Packs

To update the signatures on the IDSM sensor, we use a command called *apply*. This command is used from the primary partition when the IDSM sensor is in the configuration mode. In the following sample, we apply a typical signature.

Apply `ftp://username@server/path/filename`

This installs the signature or update in the active partition from the path set in the *apply command* argument. In this case, the entire filename is needed, not just the prefix, as seen in Figure 6.9. In Figure 6.15, we see the results of the command when used to install a service pack on an IDSM v1.

Figure 6.15 Service Pack Installation on an IDSM v1 Sensor

```
IDSM(config)# apply ftp://ciscoids@10.4.2.11/ftpupload/IDSMk9-sp-3.0-6-
    S42.exe
WARNING: Installing Service Pack will temporarily disable IDS.
Continue with IDS Service Pack install?: y

Enter the FTP user password: *****
Connecting to site...

Receiving file.
```

Continued

Figure 6.15 Service Pack Installation on an IDSM v1 Sensor

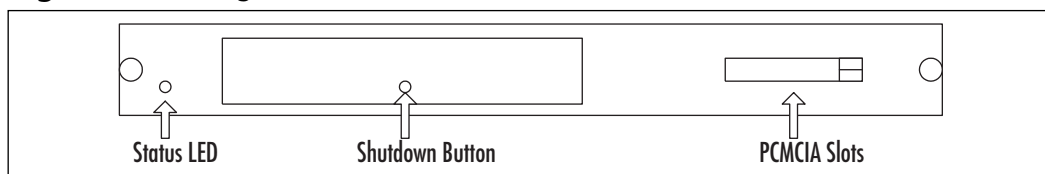
Installing files from 3.0(6)S23

```
Starting NetRanger Signatures Merging Utility...
Checking file: C:\Program Files\Cisco Systems\Netranger/etc/packetd.
    conf...
Adding signature: SigOfGeneral 993 to C:\Program Files\Cisco
    Systems\Netranger/etc/packetd.conf.
Adding signature: SigOfGeneral 1107 to C:\Program Files\Cisco
    Systems\Netranger/etc/packetd.conf.
::trimed for brevity::
The Install for IDSM Service Pack file IDSMk9-sp-3.0-6-S42.exe was
    successful
System needs to be restarted. Rebooting...
```

At the end of the update, the IDSM will be rebooted and you will have to log back into the IDSM to verify the service pack was applied. To verify the update, we will use the *show config* command, as detailed in Figure 6.10. If, during the updates or service pack installation, you can not get the IDSM sensor to talk to the FTP server, from the diag prompt of the maintenance partition, execute the *PING* command. This is a quick and simple way to make sure the IDSM sensor can, in fact, see the FTP server. More often than not there is a configuration issue with the network configuration of the IDSM sensor such as the incorrect default gateway or an incorrect subnet mask.

Troubleshooting the Cisco IDSM Sensor

Troubleshooting the IDSM might feel somewhat overwhelming at first, but in reality you know a lot of the procedure already. There are commands and even LEDs that we can look at to get an idea of what the problem of our broken IDSM could be. We will start with the simplest of items, the physical diagram of the IDSM. In Figure 6.16, we have a basic diagram of the IDSM.

Figure 6.16 Diagram of the Front Panel of the IDSM Sensor

The two most critical parts to know about are the Status LED and the shutdown button. The status LED will show three different colors, or be off completely if the power is off.

- Green means all diagnostics have passed and the IDSM is operational.
- Red means a diagnostic test other than an individual port test.
- Amber means the IDSM is running through the bootup OR the IDSM is disabled.
- Off means the IDSM power is off.

To keep from corrupting the Windows-based operating system, you need to properly shut down the IDSM before hitting the power switch. The proper way to shut down the IDSM is to use the *shutdown* command from the Catalyst switch console. If the *shutdown* command fails to work, you can use the Shutdown button to force the IDSM to shut down.

NOTE

The default for the IDSM configuration is to have the direct Telnet feature of the IDSM disabled. Do not mistake this default as an error of the IDSM.

One of the first commands to use to check a difficult IDSM sensor is the *show module* command. This command will let you quickly verify that the module is in the slot you think it is and what its current state is. If the module is in an “other” state, use the *reset* command to try and jumpstart the IDSM sensor back to life. Remember, you are dealing with Windows in version 1 and some of our favorite “features” are alive and well in the IDSM sensor, thus it does not handle errors in the configuration very well. In one system we used, an error occurred

while configuring Telnet permissions, and when the IDSM sensor was rebooted, it went into a fault mode and refused to let anyone connect. The only fix was to reinstall the OS using the upgrade process discussed earlier in this chapter. In extreme cases, you might need to power off the module or, if necessary, remove the module from a live switch. To do this, use the *set module power* command as discussed earlier in the chapter. It's shown next:

```
switch> (enable) set module power down <module>
```

When the module is powered down and ready to be powered back up, just reverse the command to say:

```
switch> (enable) set module power on <module>
```

If you can not Telnet to the module or get it to reset from the switch, the last resort is to use the Shutdown button on the front of the IDSM sensor unit. This forces the system to shut down regardless of its current state.

A common problem is that the IDSM can't see the expected traffic when it is enabled. This occurs most often when the monitoring port or port 1 is not in the correct VLAN, or the access-lists are incorrect. This also holds true when you are trying to upgrade the IDSM and you can't get to the FTP server from the IDSM. Check the VLAN that the command and control port is in and verify that it is the correct VLAN. In Figure 6.17, we can see that port 4/2 is in the backbone VLAN.

Figure 6.17 Sample of the *show vlan* Command

```
switch> enable
Password:
switch> (enable) show vlan
```

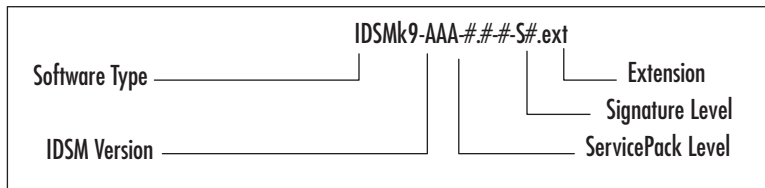
VLAN	Name	Status	IfIndex	Mod/Ports, Vlans
1	default	active	5	3/15-16
3	Finance	active	83	
4	IDF-1	active	77	2/27-29, 2/37-48
5	IDF-2	active	84	
6	IDF-3	active	79	
7	IDF-4	active	80	
10	HR	active	86	
20	backup	active	92	1/1-2

Continued

Figure 6.17 Sample of the *show vlan* Command

				2/1-6, 2/9-26, 2/30-36
				3/6-14
				4/2
100	delete	active	89	
101	FAILOVER	active	91	
1002	fddi-default	active	6	
1003	token-ring-default	active	9	
1004	fddinet-default	active	7	
1005	trnet-default	active	8	

To verify that the correct IDSM software has been uploaded to the IDSM sensor, or to prepare for an upgrade, we need to look at how the IDSM software filename is structured. In Figure 6.18, we see the basic structure of the filename.

Figure 6.18 The IDSM Filename Structure

The filename is composed of five parts, as outlined in the following list:

- **Software type** This will be one of the following:
 - **Application (a)** Cisco IDS engine image
 - **Maintenance (m)** Cisco IDS maintenance image
 - **Service Packs (sp)** Cisco IDS engine fixes
 - **Signatures (sig)** Cisco IDS signature updates
- **Cisco IDSM version** The version number is a numeric value and is separated by the use of a decimal point. The preceding number is the major version and the later number is the minor version.

- **Service pack level** This is the level to which the code has been patched to.
- **Signature level** The signature version is the Cisco IDS major and minor release level.
- **Extension** This can be one of the following filename extensions:
 - **Exe** Self-extracting executables such as signature or service packs
 - **Cab** A Microsoft format used for the IDSM software images
 - **Lst** List of cab files required for an IDSM software image
 - **Dat** A binary file containing information required for the installation of an IDSM image

For example, in previous examples we used the file IDSMk9-a-3.0-1-S4.DAT. This file is application 1 for the IDSM major version 3 and the minor version of 0. The signature is version 4 and composes the DAT file for the update.

Other useful commands to aid in troubleshooting the IDSM sensor are used from the switch prompt (switch>). These include:

- **(enable) show config** This prints out the entire configuration of the IDSM
- **show span** This shows us the span configuration and which ports are used
- **show security ACL** This displays the current security access-list in use

From the IDSM sensor prompt, we have the following commands to aid us with troubleshooting the IDSM sensor:

- idsm# **show configuration**
- idsm(diag)# **show eventfile current**

The *show configuration* command will display the current memory statistics, the disk space used, the sensor version, and the current IDS processes running (a key item). In a properly configured IDSM, the following processes should be running:

- **nr.postofficed**
- **nr.filexferd**

- **nr.loggerd**
- **nr.packetd**
- **nr.sapd**

If any one of these processes is not running, we move onto the next command, which is *show eventfile current*. The *show eventfile current* command displays the Windows event log, which may provide clues as to what might be the issue with the IDSM sensor. In Figure 6.19, we show a sample from the eventfile log:

Figure 6.19 Sample from the Eventfile Log

```
idsm(diag)# show eventfile current
4,47,2003/06/18,22:40:23,2003/06/18,14:40:23,10008,57,100,OUT,OUT,2,
    3030,0,TCP/I
P,10.4.2.75,0.0.0.0,0,139,0.0.0.0,
4,48,2003/06/18,23:21:50,2003/06/18,15:21:50,10008,57,100,OUT,OUT,2,
    3030,0,TCP/I
P,10.8.3.24,0.0.0.0,0,139,0.0.0.0,7
```

To start with clear counters and to clear out the statistics, we use the *diag resetcount* command, as shown next:

```
idsm(diag)# diag resetcount
```

To clear out a configuration, we can use the *clear config* command and remove the IDS configuration. Be warned, however: this also disables the IDSM as mentioned earlier in the chapter.

```
idsm# clear config
```

We saw earlier how to apply a service pack to the IDSM, but what happens if something goes wrong with the service pack installation? In Windows, we can uninstall files and the IDSM offers something along the same lines of functionality. The *remove* command removes the most recently applied service pack or signature from the IDSM.

```
Idsm(config)# remove
```

Summary

We can see from this chapter that the IDSM sensor, although intimidating on the surface, is no more difficult to configure and manage than the more-conventional Cisco IDS appliances. It consists of two versions: the original version of the IDSM sensor (based on an embedded version of Windows) and version 2 (based on Red Hat Linux).

The Cisco IDSM sensor has three command modes: exec mode, configuration mode, and diagnostic mode. Through them, we manage and configure the IDSM sensor at the command line.

In order to start using the IDSM sensor, you need to configure the monitoring port to capture the appropriate VLAN traffic. To do this on a Catalyst 6000/6500 switch, we use the `set vlan <vlan_number> <src_module/src_ports>` command. Once we have the monitor port in the correct VLAN, we can either configure SPAN or use a VACL depending on the need. SPAN is easier to configure but does not have as much flexibility as the VACL. The VACL, meanwhile, can capture very specific traffic—for instance, a single given protocol such as HTTP only. Or it can filter on a given MAC address. To configure the SPAN, we use the `set span <src_mod/src_port> <dest_mod/dest_port> [rx | tx | both] [create]` command.

Configuring the VACL is a bit more involved. We first start with the command `set security acl ip <acl name> permit < > capture` which sets up the ACL name, permits IP, and instructs the VACL to capture traffic. Next, we commit the ACL by using the `commit security acl` command and apply it to the VLAN of interest using the command `set security acl map <acl name> [vlans]`.

The IDSM sensor has two interfaces that sit on the backplane of the switch. The first, or port 1, is the monitoring interface. The second, or port 2 interface, is the command and control interface that we use to control and manage the IDSM sensor. Since the IDSM sensor is a line card for the Catalyst 6000/6500 series switch, there is no impact on the switching performance.

The IDSM sensor can have the operating system upgraded or patched by using an FTP server, the `ids-installer` command and the `apply` command. To update or upgrade the IDSM sensor software, you need to boot to a different partition than the one that will be upgraded. In most cases, you will be booting to partition 2 or the maintenance partition using the `reset <module/port> hdd:2` command. Before we can upload the image to the partition, we need to configure the maintenance partition with a network configuration using the `ids-installer netconfig` command. Using FTP and the `ids-installer system` command on the IDSM sensor uploads the update/patch image to the IDSM sensor.

Solutions Fast Track

Understanding the Cisco IDSM Sensor

- ☑ The IDSM sensor is a module or blade in the Catalyst 6000/6500 series switch.
- ☑ The IDSM uses SPAN, RSPAN, or VACLs to capture traffic for analysis.
- ☑ The IDSM sensor can capture all VLANs or a selection of VLANs.
- ☑ The IDSM sensor does not impact the performance of the switch during its operation.
- ☑ If the IDSM sensor fails or is disabled, it does not block the flow of traffic since it is a passive device.
- ☑ There are two ports on the IDSM sensor. The first, port 1, is for monitoring the traffic. The second, port 2, is used to command and control the IDSM sensor.
- ☑ The IDSMv1 needs to have a director to manage the sensor while IDSMv2 can be managed by web, Telnet, or a director.

Configuring the Cisco IDSM Sensor

- ☑ The initial configuration is accomplished by using the *setup* command.
- ☑ There are two partitions on a Cisco IDSM: one for the operation and one for maintenance.
- ☑ In order for the IDSM sensor to analyze traffic, we need to assign it to the correct VLAN(s) that we want to analyze by using the *set vlan* command.
- ☑ If we want to just filter traffic at the IP level, we can use the *SPAN* command.
- ☑ If we want to filter traffic at a port level or a MAC level, we use VACLs

Updating the Cisco IDSM Sensor

- ☑ Updating the operating system of the sensor requires you to boot the sensor from the maintenance partition either by setting the boot device or by using the *reset* command.

- ☑ Before any upgrades to the sensor can be completed, the IDSM sensor must have the network settings configured on the maintenance partition.
- ☑ To upgrade the operating system, use the *ids-installer system* command from the diag mode on the maintenance partition.
- ☑ To install a service pack to the operating system of the IDSM sensor, use the *apply* command from the config mode on the primary partition of the IDSM sensor.
- ☑ The signature updates, operating system updates, or patches are downloaded to the IDSM sensor by FTP.

Troubleshooting the Cisco IDSM Sensor

- ☑ The status LED can tell you if the system has completed all diagnostics, failed, or if the IDSM is disabled
- ☑ If you can't Telnet to the IDSM sensor directly, verify you have at least version 3.0 code and that Telnet has been enabled (by default, it's disabled).
- ☑ If the IDSM sensor cannot see any traffic, check that the monitor port is in the correct VLAN by using the *show vlan* command from the enabled mode of the switch.
- ☑ To verify the IDSM processes are running, use the *show configuration* command, which gives the status of the nr.postoffice, nr.filexd, nr.loggerd, and nr.packetd processes.
- ☑ To remove a configuration from the IDSM sensor, use the *clear config* command. Remember though, this command will leave the IDSM in a disabled state.
- ☑ If a newly installed service pack is problematic, we can remove it by using the *remove* command from the config mode on the primary partition.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: How do I get into the IDSM to configure it?

A: With a default configuration, there is only one way in and that is to use the session command from the switch console. This can be changed to allow Telnet directly to the IDSM.

Q: How do I upgrade my IDSM?

A: To upgrade the IDSM sensor, boot to the maintenance partition using the *reset* command and go into the diagnostic mode. Then use the *ids-installer* command to install the image from an FTP server. Reboot the IDSM sensor back to the primary partition and reconfigure the sensor.

Q: How do I start over with an IDSM sensor that has already been configured?

A: The easiest way is to clear the configuration of the IDSM sensor. This is accomplished by using the *clear config* command and remembering that the IDSM sensor will be disabled after the operation is complete.

Q: Can I have more than one IDSM sensor in the switch chassis?

A: Yes, you can use more than one IDSM sensor in the chassis provided you follow the basic rule that slot 1 is reserved for the supervisor module.

Q: Can I manage the IDSM sensor from a Web or command-line interface?

A: Yes and no. The older IDSM sensor (version 1) only goes to version 3.0 of the code. This version of code does not have any command-line or Web interface. The new IDSM sensor uses version 4.0 code and has both Web and command-line interfaces.

Q: If my IDSM sensor fails or I place it into disabled mode, will that stop traffic from passing through the switch?

- A:** No, the IDSM sensor is a passive device and traffic will flow without regard to the state of the IDSM sensor.
- Q:** Do I have to set up the SPAN session to use both Tx and Rx, or can I just use Tx?
- A:** If you configure the switch to SPAN with Tx only, the IDSM sensor will only see part of the traffic flow. In order to see all the traffic, you need to use both the Rx and Tx.
- Q:** I can't upgrade my IDSM sensor from the maintenance partition. What might be the problem?
- A:** The most common error is that the network configuration was not set up or that it is incorrect. Use the *ids-installer netconfig /view* command to verify the current network configuration of the IDSM maintenance partition.
- Q:** Can I have more than one IDSM sensor in a given switch chassis?
- A:** Yes, you can. In the Catalyst chassis, slot 1 is reserved for the supervisor blade while slot 2 is usually reserved for the redundant supervisor. However, you can install the IDSM sensor in slot 2 if there is no redundant supervisor, or install it into any other slot in the chassis.

Cisco IDS Alarms and Signatures

Solutions in this Chapter:

- Understanding Cisco IDS Signatures
 - Understanding the Cisco IDS Signature Series
 - Configuring the Sensing Parameters
 - Excluding or Including Specific Signatures
 - Creating a Custom Signature
 - Working with SigWizMenu
 - Understanding Cisco IDS Alarms
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Once the Cisco IDS sensor is racked and operational, and the IDS management device or director is configured and communicating properly, it is time to tune the IDS signatures to the traffic patterns that occur on your network. We need to run the sensor for a period of time, normally a week or so to build a baseline of activity to look at. Without the baseline it is impossible to know for sure if the alarm is real or if it has resulted from an incorrect setting for your network traffic. Without optimized signatures, the IDS sensor is relatively useless to us. To start the baselining of the network, the sensor is placed in a strategic location on your network where it can see and analyze all of the targeted traffic that passes by the sensor. To put it simply, you are data-mining from a security perspective. With data-mining, there needs to be a query; in this case, the tuned signature is the query. Anything that meets the parameters of the signature triggers an alarm and sends an event to the IDS management device. We are studying the traffic behavior of the network and teaching the IDS sensor to make decisions on data and patterns that are considered out of the norm for the network and which provide some type of notification or action such as shunning.

As you can see in our discussion of IDS signatures, the IDS signature is the heart and soul of successful IDS deployment and operation. Without the correct signatures, the IDS sensor is useless for maintaining your network security. However, an IDS sensor that constantly generates false positives or false alarms is useless as well, since you will learn to ignore the sensor's alarms even when they might be valid. And when time comes that a real attack does take place, you will miss it because you thought it was just another false alarm. This is not an effective way to use the Cisco IDS system. We will show you in this chapter how to avoid this pitfall. We will also discuss exactly what the Cisco IDS signature is, what makes up the signature, how to tune the signatures, and how to make your very own custom IDS signature. The Cisco IDS sensor can also provide various responses to signature triggers such as logging, TCP resets, or blocking. We will cover the various alarms and why alarms are useful for the IDS and your sanity.

Understanding Cisco IDS Signatures

It is important to understand what a signature is, and what exactly a signature does. A signature is a known type of activity. It has already been detected in the wild and someone has captured the personality or traffic pattern of the attack or intrusive activity and documented it. In many ways, the signature is something

akin to a fingerprint. The fingerprint is unique to a person just like the signature is unique to a certain attack or type of activity. A Cisco IDS sensor then compares traffic against the signatures it has configured and will match up this activity when it appears on your network. The parameters you set for the signature will tell the sensor how to respond to the threat. The sensor can send an alarm to your IDS management device, log the event, send e-mail alerts, or even block the suspect traffic at the router, switch, or firewall.

When you load signature updates up to the IDS sensor, the signatures are loaded onto the sensor with their recommended settings already preconfigured. To view those signature settings with CSPM, scroll down the network topology in the left pane and select **Tools and Services | Sensor Signatures**. The name of the signature files is listed there. By default, CSPM creates a *Default* signature file when the sensor is added, as we see in Figure 7.1. You can have a different signature file for each sensor on your network or use one for all of them. To get to the signatures from inside Cisco's Intrusion Detection Manager (IDM), choose **Configuration | Sensing Engine | Signature Configuration | Signature Groups**, shown in Figure 7.2. The most critical signatures are usually configured and set to generate high- or, at the least, medium-level alarms. When the sensor detects traffic that meets the enabled signatures, it fires off an alarm. The sensor stores all alarms in the sensor logs that are informational and above. If you have a Cisco IDS Management device, and it is configured as a destination for alarms, the alarms are also sent to that device for viewing.

Figure 7.1 The CSPM Signature File

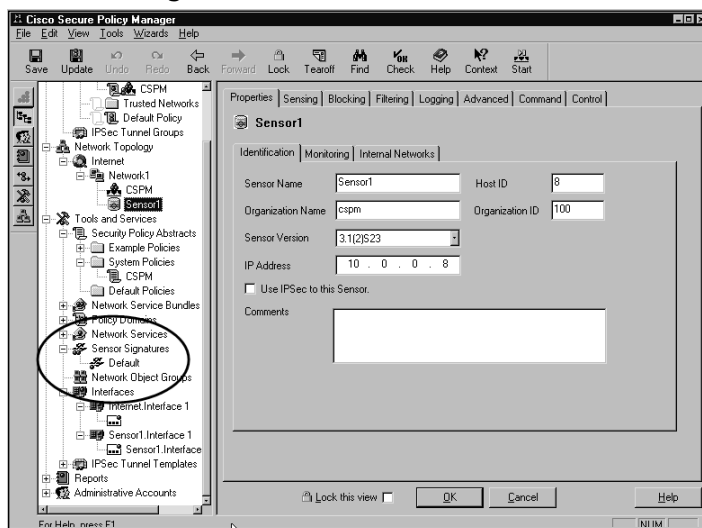


Figure 7.2 IDM Signatures



Signature Implementation

The complexity of signatures can be explained fairly easily. There are several components that make up the signatures and as long as you understand the role each component plays, you will not have a problem with understanding them. It is not a black art or magic, just a bit of common sense. As we mentioned earlier, the signature is created from an already known activity. Once intrusive or malicious activity is discovered in the wild, a signature is created that looks for that specific behavior and nothing else. The sensor has a database of all the signatures and their specific configurations, and compares the traffic against that database. Signatures are implemented as either content-based or context-based.

NOTE

Content-based signatures are triggered by information contained in the payload of the packet such as a URL string that could possibly compromise a web server application.

Context-based signatures are triggered by the data in the packet headers. This is an enhancement to Packet Signature Detection, which does not consider any context. The most common implementations of Context-Based Signature Detection are designed to look for attack signatures in particular fields or use a particular offset within a packet stream (based on the protocol).

You need to keep this straight in your head when taking the Cisco IDS exam.

Signature Classes

The class of the signatures is important to understand. The attack and the intentions of the attack will drive the classification of the signatures. Reconnaissance, Informational, Access, and Denial of Service are the four main categories.

Reconnaissance is what the attackers do that enable them to map out a network such as DNS queries, ports scans, and even pings. This type of activity will trigger the reconnaissance class signatures. Once the active IP addresses and open ports have been identified, information is gathered about the hosts by attempting to connect or communicate with the host. The attacker may try to connect to the host on a specific port. If the connection is successful, the attacker can deduce what type of system it is by what ports are open. The activity is not necessarily malicious but can be intrusive. Informational class signatures are configured to detect this type of activity. Access signatures fire alarms when known unauthorized access or attempts to access are detected. Denial-of-Service or DoS class signatures trigger when the level of activity on the network is detected as having the ability to disrupt services.

Signature Structure

The structure of the signature depends on the number of packets that have to be inspected. They can be either atomic or composite. Atomic signatures can be detected by inspecting a single packet. No state information is required. Some examples of an atomic signature are

- 1004-IP options-Loose Source Route
- 3050-Half-open SYN Attack
- 3455-Java Web Server Cmd Exec
- 3652-SSH Gobblers

A composite signature is detected by inspecting multiple packets. If the sensor detects the first packet that is a potential attack, it stores that information and the information of the following packets. State information is required in order to perform this function. Examples of a composite signature are:

- 3225-WWW websendmail File Access
- 3250-TCP Hijack
- 3314-Windows Locator Service Overflow
- 3990-BackOrifice BO2K TCP Non Stealth

For example, in the SYN Attack, a single packet with the SYN bit set is sent without the rest of the normal TCP three-way handshake. All the IDS sensor needs to see is the single SYN IP packet out of order. With the Windows Locator attack, it requires more than a single packet of information and the IDS sensor will match on the first one in the sequence, tag it as interesting and look for more matches of the known attack sequence. Once the IDS sensor sees more of the attack, it will trigger whatever alarms or actions it was programmed to carry out.

Signature Types

Cisco also categorizes the signatures into different traffic types. The different types are

- General Connection
- String
- Access Control List (ACL)

General signatures cover the 1000, 2000, 5000, and 6000 signature series. Depending on the type of attack, the General signatures look for abnormalities in a known type of traffic such as making sure a certain protocol is behaving correctly or the payload in packets is or looks correct. An example of a general signature is *3037-TCP FRAG SYN FIN Host Sweep*. This signature triggers when a series of packets (TCP) with both the SYN and FIN flags set have been sent to multiple hosts with the same destination port. Having the SYN and FIN flags set is abnormal, as is fragmentation.

Connection signatures are covered in the 3000 and 4000 signature series. They observe traffic to UDP ports and TCP connections. An example of connection signature is *3001-TCP Port Sweep*. TCP Port sweep is the perfect example of a connection signature. It fires when a series of TCP connections are initiated on a host to multiple ports. The port range is less than 1024. Be vary aware of these types of detects. It can be a prelude to a major attack.

String signatures are highly flexible. They monitor strings (text) within packets that you deem important. An example of a string signature is *8000:2303-Telnet-+ +*. When a Telnet session is initiated and the command “++” is entered, this signature will fire. All string detects will generate an 8000 series alarm. It is the subID, 2303, that differentiates the string signatures.

Access-Control-List signatures apply to traffic or activity that is attempting to circumvent access control lists on the routers. These are signatures in the 10000 series. Like the string signatures, the subID is what differentiates the different signatures. An example of an Access-Control-List signature is *10000:1001-IP-Spoof Interface 2*. This particular signature triggers when there is notification from a NetSentry device that an IP datagram has been received from a source in front of the router with an IP address that belongs behind the router.

Cisco IDS Signature Micro-Engines

The Cisco Secure IDS software divides signature processing into different categories or engines. We can see the types of engines in Table 7.1.

Table 7.1 Cisco IDS Signature Micro-Engine Overview

Engine Type	Description
Atomic	This is used for single packets.
Flood	This is used to detect attempted DoS attacks.
Service	This is used when services at layers 5,6, and 7 require protocol analysis.
State	This is used when stateful inspection is required. At this time, only http is supported.
String	This is used for string pattern matching.
Sweep	This is used to detect network reconnaissance sweeps or probes.

Each engine contains a parser and inspector and multiple signatures are supported within specific categories. When the IDS is sniffing the network, it reads from a signature file that contains all of the signature definitions. Each of the definitions contains configurable parameters that can be tweaked to define activity on your network that you would consider intrusive and possibly malicious. Signature parameters have three attributes to them. They can be Protected, Required, or Hidden. The Protected attribute affects the fundamental behavior of the parameter and applies only to the Cisco set of default signatures. The

Required attribute is a parameter value that must be declared. The Hidden attribute is that the parameter is not viewable because modifications to the parameter are not allowed. The parameters are themselves broken down into two categories:

- Master or Global engine parameters
- Engine-specific parameters

The Master engine parameters apply to each of the signatures in the subengines. Master engine parameters are the basis for parsing the input (traffic) and producing output (alarms). Table 7.2 lists the Master engine parameters. It is up to the subengines to provide the specific protocol needed for the sensor to decode and inspect the traffic.

Table 7.2 Master or Global Engine Parameters

Parameter	Description
AlarmDelayTimer	This is the number of seconds (1–3600) to delay further signature inspection after an alarm.
AlarmInterval	Special handling for time events (2–1000). Uses AlarmInterval Y with MinHits X for X alarms in a Y-second interval.
AlarmSeverity	The severity of the alert (high, medium, low, or informational) reported in the alarm.
AlarmThrottle	Limits the number of alarms sent to the IDS management device. The following options can be selected: FireAll: Send all alarms when the signature conditions are met. FireOnce: Send the first alarm when signature conditions are met. Then, do not send any more alarms from the same source and destination address combination. Summarize: Send only one alarm per ThrottleInterval per address combination. Usually, the first alarm that starts a summary is sent. The ThrottleInterval is a configurable number in seconds that the sensor counts until that number (ThrottleInterval) is reached. It then fires another alarm and starts the count all over again.

Continued

Table 7.2 Master or Global Engine Parameters

Parameter	Description
	GlobalSummarize: Similar to the Summarize parameter but expands to all address combinations instead of one. For example, once an alarm is sent the sensor counts the subsequent alarms per the ThrottleInterval for all address combinations being monitored. This reduces the number of alarms triggered during flood attacks.
ChokeThreshold	Switches between Summarize and Global Summarize. During the ThrottleInterval, the sensor autoswitches the AlarmThrottle mode to Summarize if the frequency of alarms from a single signature is greater than the ChokeThreshold. The sensor will autoswitch the AlarmThrottle mode to GlobalSummarize if the frequency of alarms from single signature is double or twice the ChokeThreshold. The ChokeThreshold may not be set to ANY to autoswitch the AlarmThrottle.
FlipAddr	Swaps the addresses and ports if they are detected as being reversed in the alarm message.
MaxInspectLength	The Maximum length in bytes to inspect.
MinHits	Throttle for firing the alarm when the minimum number of signature hits has been detected by the sensor.
ResetAfterIdle	When a signature stops firing alarms, this is the number of seconds the sensor waits before it resets the counters (ThrottleInterval, MinHits, etc...).
SigComment	Comment section to input your own notes about the signature.
SIGID	Unique number identifier for each signature. Cisco designates 1000–19,999 as the range for default signatures and 20,000–50,000 as the range for user signatures.
SigName	Official signature name.
SigStringInfo	Any extra information included in the alarm message.
SubSig	ID of Subsignatures, if any. Usually a variation of the original signature.

Continued

Table 7.2 Master or Global Engine Parameters

Parameter	Description
ThrottleInterval	A counter in seconds defining the interval that alarms are triggered. Used in conjunction with the AlarmThrottle parameter when configuring Summarize or Global Summarize settings.
WantFrag	Has the sensor inspect fragmented packets against the signature. Can be set to TRUE if you want to inspect reassembled fragmented packets or fragments, FALSE if you do not want to inspect reassembled fragmented packets or fragments, or ANY to ignore all reassembled packets and/or fragments.

Figure 7.3 shows all of the micro-engines available on the 4200 series sensors.

Figure 7.3 SigWizMenu Showing the Micro-Engines

Signatures Display Menu : CSIDS Signature Wizard

Engine	Sigs:	Default	Custom
1 - ATOMIC.ICMP		14	0
2 - ATOMIC.IPOPTIONS		6	0
3 - ATOMIC.L3.IP		5	0
4 - ATOMIC.TCP		21	0
5 - ATOMIC.UDP		7	0
6 - FLOOD.HOST.ICMP		2	0
7 - FLOOD.HOST.UDP		1	0
8 - FLOOD.NET		5	0
9 - FLOOD.TCPSYN		4	0
10 - SERVICE.DNS.TCP		18	0
11 - SERVICE.DNS.UDP		16	0
12 - SERVICE.PORTMAP		7	0
13 - SERVICE.RPC		11	0
14 - STATE.HTTP		287	0
15 - STRING.HTTP		7	0
16 - STRING.ICMP		0	0

Continued

Figure 7.3 SigWizMenu Showing the Micro-Engines

17 -	STRING.TCP	81	0
18 -	STRING.UDP	8	0
19 -	SWEEP.HOST.ICMP	3	0
20 -	SWEEP.HOST.TCP	8	0
21 -	SWEEP.PORT.TCP	12	0
22 -	SWEEP.PORT.UDP	1	0
23 -	SWEEP.RPC	9	0
ENTER - Back to Main			

Selection>

The ATOMIC Micro-Engines

The ATOMIC engine is used to create or tune existing signatures for simple, single packet conditions that cause alarms to be triggered. Every packet's conditions have specialized parameters that deal with each of the protocol-specific inspections within the scope of the engine. Table 7.3 shows the different ATOMIC micro-engines. These engines do not store any persistent data whatsoever. The ATOMIC micro-engines have parameters that are set for their specific protocol.

Table 7.3 ATOMIC Micro-Engines

Engine	Description
ATOMIC.ARP	ARP simple and cross-packet signatures.
ATOMIC.ICMP	Simple ICMP alarms based on the following parameters: Type, Code, Sequence, and ID. See Figure 7.1.
ATOMIC.IPOPTIONS	Simple alarms based on the decoding of layer-3 options. See Figure 7.2.
ATOMIC.L3.IP	Simple layer-3 IP alarms. See Figure 7.3.
ATOMIC.TCP	Simple TCP packet alarms based on the following parameters: Port, Destination, Flags, and single-packet Regex. Use SummaryKey to define the address view for MinHits and Summarize counting. For best performance, use a StorageKey. See Figure 7.4.
ATOMIC.UDP	Simple UDP packet alarms based on the following parameters: Port, Direction, and DataLength. See Figure 7.5.

Continued

Table 7.3 ATOMIC Micro-Engines

Engine	Description
OTHER	This engine is used to group generic signatures so common parameters can be changed. It defines an interface into common signature parameters.

SigWizMenu option 1 ATOMIC.ICMP (as seen in Figure 7.3) and SigWizMenu option 5 ATOMIC.UDP (shown in Figure 7.4) work specifically on layer 4. None of the parameters are required even though there are several parameters that can be manually configured. You can use all the single parameters together in a signature or configure specific ones.

Figure 7.4 SigWizMenu Option 1 ATOMIC.ICMP

```

Selection> 1

      2000  (SubSig 0) ICMP Echo Rply :
      2001  (SubSig 0) ICMP Unreachable :
      2002  (SubSig 0) ICMP Src Quench :
      2003  (SubSig 0) ICMP Redirect :
      2004  (SubSig 0) ICMP Echo Req :
      2005  (SubSig 0) ICMP Time Exceed :
      2006  (SubSig 0) ICMP Param Prob :
      2007  (SubSig 0) ICMP Time Req :
      2008  (SubSig 0) ICMP Time Rply :
      2009  (SubSig 0) ICMP Info Req :
      2010  (SubSig 0) ICMP Info Rply :
      2011  (SubSig 0) ICMP Addr Msk Req :
      2012  (SubSig 0) ICMP Addr Msk Rply :
      2150  (SubSig 0) Fragmented ICMP :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

The SigWizMenu option 2 ATOMIC.IPOPTIONS decodes layer-3 options as shown in Figure 7.5.

Figure 7.5 SigWizMenu Option 2 ATOMIC.IPOPTIONS

```
Selection> 2
```

```
1001 (SubSig 0) Record Packet Rte :
1002 (SubSig 0) Timestamp :
1003 (SubSig 0) Provide s,c,h,tcc :
1004 (SubSig 0) Loose Src Rte :
1005 (SubSig 0) SATNET ID :
1006 (SubSig 0) Strict Src Rte :
```

```
(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

The SigWizMenu option 3 ATOMIC.L3.IP inspects the traffic at layer 3 (as we can see in Figure 7.6). It handles fragment, partial ICMP packets, DataLength, and Protocol Number comparisons. Again, these parameters are optional.

Figure 7.6 SigWizMenu Option 3 ATOMIC.L3.IP

```
Selection> 3
```

```
1101 (SubSig 0) Unknown IP Proto :
1107 (SubSig 0) RFC 1918 Addresses Seen : RFC 1918 Address
2151 (SubSig 0) Large ICMP :
2154 (SubSig 0) Ping Of Death :
2154 (SubSig 1) Ping Of Death :
```

```
(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

ATOMIC.TCP looks at layer-4 TCP packets. This menu option does comparisons on TcpFlags/Mask in conjunction with port filters and the SinglePacketRegex. TcpFlags/Mask compares packets against the configured parameters to determine packets of interest. The SinglePacketRegex provides a simple Regex match capability to combine ports, flags, and Regex matches in single signatures. Refer to Figure 7.7. Figure 7.8 shows the SigWizMenu option 5 ATOMIC.UDP.

Figure 7.7 SigWizMenu Option 4 ATOMIC.TCP

Selection> 4

```
3038 (SubSig 0) TCP FRAG NULL Packet :
3039 (SubSig 0) TCP FRAG FIN Packet :
3040 (SubSig 0) TCP NULL Packet :
3041 (SubSig 0) TCP SYN/FIN Packet :
3042 (SubSig 0) TCP FIN Packet :
3043 (SubSig 0) TCP FRAG SYN/FIN Packet :
9000 (SubSig 0) Back Door SYN-port 12345 : back door SYN-port 12345
9001 (SubSig 0) Back Door SYN-port 31337 : back door SYN-port 31337
9002 (SubSig 0) Back Door SYN-port 1524 : back door SYN-port 1524
9003 (SubSig 0) Back Door SYN-port 2773 : back door SYN-port 2773
9004 (SubSig 0) Back Door SYN-port 2774 : back door SYN-port 2774
9005 (SubSig 0) Back Door SYN-port 20034 : back door SYN-port 20034
9006 (SubSig 0) Back Door SYN-port 27374 : back door SYN-port 27374
9007 (SubSig 0) Back Door SYN-port 1234 : back door SYN-port 1234
9008 (SubSig 0) Back Door SYN-port 1999 : back door SYN-port 1999
9009 (SubSig 0) Back Door SYN-port 6711 : back door SYN-port 6711
9010 (SubSig 0) Back Door SYN-port 6712 : back door SYN-port 6712
9011 (SubSig 0) Back Door SYN-port 6713 : back door SYN-port 6713
9012 (SubSig 0) Back Door SYN-port 6776 : back door SYN-port 6776
9013 (SubSig 0) Back Door SYN-port 16959 : back door SYN-port 16959
9014 (SubSig 0) Back Door SYN-port 27573 : back door SYN-port 27573
```

(Sig Number to EDIT) or (ENTER to CONTINUE) >

NOTE

Figure 7.7 only shows a portion of the signatures within the ATOMIC.TCP micro-engine. There are approximately 60 total signatures in this engine.

Figure 7.8 SigWizMenu Option 5 ATOMIC.UDP

```

Selection> 5

      4050   (SubSig 0) UDP Bomb :
      4051   (SubSig 1) Snork  :
      4051   (SubSig 2) Snork  :
      4051   (SubSig 3) Snork  :
      4052   (SubSig 1) Chargen DoS :
      4052   (SubSig 2) Chargen DoS :
      4600   (SubSig 0) IOS Udp Bomb :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

ATOMIC.ARP is for basic layer-2 ARP signatures and also for more advanced detection of the ARP spoof tools *dsniff* and *ettercap*. Refer to Table 7.4 for the ATOMIC.ARP parameters.

NOTE

ettercap supports active and passive dissection of several protocols. It features network and host analysis tools. In essence, it acts as a sniffer, interceptor, and logger for switched LANs. *dsniff* is a collection of tools used for penetration testing and auditing networks.

Table 7.4 ATOMIC.ARP Parameters

Name	Data Type	Protected	Required	Description
ArpOperation	Number 0–255	No	No	The ARP operation code the signature is interested in.
MacFlip	Number 0–65535	No	No	If the MAC address changes this many times for the same IP address, an alarm will fire

Continued

Table 7.4 ATOMIC.ARP Parameters

Name	Data Type	Protected	Required	Description
RequestInBalance	Number 0–65535	No	No	If there is this many more requests than there are replies on a particular IP address, an alarm will fire.
WantDstBroadcast	Boolean True/False	No	No	If the sensor detects an ARP destination address of 255.255.255.255, an alarm will fire.
WantBroadcast	Boolean True/False	No	No	If the sensor detects an ARP source address of 255.255.255.255, an alarm will fire.

The SERVICE Micro-Engine

Of all the different service micro-engines (see Table 7.5), SERVICE.DNS and SERVICE.RPC are two of the more important engines. SERVICE works at layer 5 and above to analyze traffic between two hosts. Service engine signatures are one-to-one signatures that interpret the payloads similar to the way the live services would interpret them. The result of the interpretation is the decoded fields of the protocol used in comparison against the signatures. These engines only decode enough of the data to make comparisons. Once a comparison can be made, the alarm is triggered and keeps resource utilization to a minimum.

Table 7.5 Service Micro-Engines

SERVICE.DNS	Analyzes the DNS service.
SERVICE.FTP	FTP service special decode alarms.
SERVICE.GENERIC	Custom service/payload decode. For expert use only.

Continued

Table 7.5 Service Micro-Engines

SERVICE.HTTP	HTTP protocol decode-based string engine. Includes anti-evasive URL deobfuscation.
SERVICE.IDENT	IDENT service (client and server) alarms.
SERVICE.MSSQL	Microsoft SQL service inspection engine.
SERVICE.NTP	Network Time Protocol–based signature engine.
SERVICE.RPC	Analyzes the RPC service.
SERVICE.SMB	SMB SuperInspector signatures.
SERVICE.SMTP	Inspects SMTP protocol.
SERVICE.SNMP	Inspects SNMP traffic.
SERVICE.SSH	SSH header decode signatures.
SERVICE.SYSLOG	Processes SYSLOGS.

The SERVICE.DNS micro-engines specialize in traffic on both TCP (see Figure 7.9) and UDP (see Figure 7.10) port 53. Port 53 is the standard port for DNS traffic. The SERVICE.DNS does not have any required parameters, but for full coverage on DNS, you must specify TCP or UDP. Other than that necessity, the engine is open for full customization of the signatures.

Figure 7.9 SigWizMenu Option 10 SERVICE.DNS.TCP

```
Selection> 10

6050 (SubSig 1) DNS HINFO-TCP :
6051 (SubSig 1) DNS Zone Xfer-TCP :
6052 (SubSig 1) DNS High Zone Xfer-TCP :
6053 (SubSig 1) DNS Request All-TCP :
6054 (SubSig 1) DNS Version Request-TCP :
6055 (SubSig 1) DNS IQUERY Overflow-TCP :
6055 (SubSig 2) DNS IQUERY Overflow-TCP :
6056 (SubSig 1) DNS NXT Overflow-TCP :
6056 (SubSig 2) DNS NXT Overflow-TCP :
6057 (SubSig 1) DNS SIG Overflow-TCP :
6057 (SubSig 2) DNS SIG Overflow-TCP :
6058 (SubSig 1) DNS SRV DoS-TCP :
6059 (SubSig 2) DNS TSIG Overflow-TCP :
6060 (SubSig 2) DNS Complain Overflow-TCP :
```

Continued

Figure 7.9 SigWizMenu Option 10 SERVICE.DNS.TCP

```

6060 (SubSig 3) DNS Complain Overflow-TCP :
6061 (SubSig 1) DNS Infoleak-TCP :
6062 (SubSig 1) DNS Authors Request-TCP :
6063 (SubSig 1) DNS Incremental Zone Transfer-TCP :

```

(Sig Number to EDIT) or (ENTER to CONTINUE) >

Figure 7.10 SigWizMenu Option 11 SERVICE.DNS.UDP

Selection> 11

```

6050 (SubSig 0) DNS HINFO-UDP :
6051 (SubSig 0) DNS Zone Xfer-UDP :
6052 (SubSig 0) DNS High Zone Xfer-UDP :
6053 (SubSig 0) DNS Request All-UDP :
6054 (SubSig 0) DNS IQUERY Overflow-UDP :
6055 (SubSig 0) DNS NXT Overflow-UDP :
6056 (SubSig 0) DNS SIG Overflow-UDP :
6057 (SubSig 0) DNS SRV DoS-UDP :
6058 (SubSig 0) DNS TSIG Overflow-UDP :
6059 (SubSig 1) DNS TSIG Overflow-UDP :
6060 (SubSig 0) DNS Complain Overflow-UDP :
6060 (SubSig 1) DNS Complain Overflow-UDP :
6061 (SubSig 0) DNS Infoleak-UDP :
6062 (SubSig 0) DNS Authors Request-UDP :
6063 (SubSig 0) DNS Incremental Zone Transfer-UDP :
6064 (SubSig 0) BIND Large OPT Record DoS : Large OPT

```

(Sig Number to EDIT) or (ENTER to CONTINUE) >

NOTE

You need to add UDP and TCP signatures to have full coverage.

The SERVICE.RPC engine decoder has full decode as an anti-evasive strategy. It handles fragmented messages or batch messages. The RPC port mapper operates on port 111. Regular RPC messages are on any port greater than 550. RPC sweeps are very similar to TCP port sweeps with one exception: they only count unique ports when valid RPC messages are sent. One other unique characteristic of the SERVICE.RPC engine is they segregate on each RPC program type for sweep unique counting. In other words, counting occurs on an individual program basis. Figure 7.11 shows the signatures that fall into this category.

Figure 7.11 SigWizMenu Option 13 SERVICE.RPC

Selection> 13

```
6180 (SubSig 0) rexd Attempt :
6190 (SubSig 0) statd Buffer Overflow :
6191 (SubSig 0) ttldbserverd Buffer Overflow :
6192 (SubSig 0) mountd Buffer Overflow :
6193 (SubSig 0) cmsd Buffer Overflow :
6194 (SubSig 0) sadmind Buffer Overflow :
6195 (SubSig 0) amd Buffer Overflow :
6196 (SubSig 0) snmpXdmid Buffer Overflow :
6197 (SubSig 0) rpc yppaswdd overflow : yppaswdd overflow
6198 (SubSig 0) Long rwalld Message : rwalld String Format
6199 (SubSig 0) cachefsd overflow : cachfsd overflow
6275 (SubSig 0) SGI fam Attempt : Fam Attempt
6276 (SubSig 0) TooltalkDB overflow : TooltalkDB overflow
6277 (SubSig 0) Show Mount Recon : Show Mount Recon
6277 (SubSig 0) Show Mount Recon : Show Mount All Recon
```

(Sig Number to EDIT) or (ENTER to CONTINUE) >

The FLOOD Micro-Engine

Simply stated, FLOOD engines analyze flood type traffic, that is traffic from many sources to a single host (n to 1), specified in FLOOD.HOST or floods to the network, traffic from many sources to many destinations (n to n), specified in FLOOD.NET. Host floods use a counter that counts the packets-per-second

(PPS) to the destination. Net floods, however, do not use the address for counting, but instead utilize the count rate on a virtual sensor basis. Analysis is done on a per-second basis for both host and net floods.

FLOOD engines have one configuration restriction. You have to specify the Rate parameter in both the host and net flood engine groups. FLOOD engines also ignore the WantFrag, MaxInspectLength, and ResetAfterIdle parameters from the Master engine parameters.

NOTE

The concept of a virtual sensor is that if the physical sensor is monitoring more than one interface, all the interfaces are configured into interface groups. There can be more than one interface group. But virtual sensors are attached to only one interface group.

There are three FLOOD micro-engines. We will look at each in detail in the following sections.

FLOOD.HOST.ICMP

FLOOD.HOST.ICMP analyzes ICMP floods directed at a single host. Figure 7.12 shows the two signatures 2152 – *ICMP Flood*, and 2153 – *ICMP Smurf attack* that are host flood signatures based on ICMP traffic.

Figure 7.12 SigWizMenu Option 6 FLOOD.HOST.ICMP

```
Selection> 6

2152 (SubSig 0) ICMP Flood :
2153 (SubSig 0) ICMP Smurf attack :
(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

Table 7.6 shows the configurable parameters for FLOOD.HOST.ICMP signatures.

Table 7.6 FLOOD.HOST.ICMP Parameters

Name	Data Type	Protected	Required	Description
IcmpType	Number 0–256	No	No	ICMP header TYPE
Rate	Some number	No	Yes	The maximum allowed packets-per-second (PPS)

FLOOD.HOST.UDP

FLOOD.HOST.UDP analyzes UDP floods directed at a single host. Figure 7.13 shows the single signature, *4002 – UDP Flood*, that is a host flood signature based on UDP traffic.

Figure 7.13 SigWizMenu Option 7 FLOOD.HOST.UDP

```
Selection> 7

4002 (SubSig 0) UDP Flood :

(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

Table 7.7 shows the configurable parameters for FLOOD.HOST.UDP signatures.

Table 7.7 FLOOD.HOST.UDP Parameters

Name	Data Type	Protected	Required	Description
ExcludeDst1	Number 0–65536	No	No	Destination port to exclude from flood counting.
ExcludeDst2	Number 0–65536	No	No	Destination port to exclude from flood counting.
ExcludeDst3	Number 0–65536	No	No	Destination port to exclude from flood counting.
Exclude1	Number 0–65536	No	No	Source port to exclude from flood counting.

Continued

Table 7.7 FLOOD.HOST.UDP Parameters

Name	Data Type	Protected	Required	Description
Exclude2	Number 0–65536	No	No	Source port to exclude from flood counting.
Exclude3	Number 0–65536	No	No	Source port to exclude from flood counting.
Rate	Some number	No	Yes	Threshold number of PPS. When the PPS is greater than the specified Rate, an alarm fires.

FLOOD.NET

FLOOD.NET analyzes network floods directed at a single network segment. Figure 7.13 displays the current signatures in the FLOOD.NET micro-engine. Of special interest in the FLOOD.NET micro-engine is FLOOD.Net Learning Mode. This configuration option is feedback mode. Feedback mode replaces the normal inspection of packets with a diagnostic alarm. Simply stated, the alarm will have the maximum count of PPS in the alertDetails values seen during the interval. This is good for baselining network traffic in order to tune the signatures. The configuration is set to feedback mode when the Rate parameter is set to 0. Figure 7.14 shows the five signatures that are part of the FLOOD.NET micro-engine.

Figure 7.14 SigWizMenu Option 8 FLOOD.NET

```
Selection> 8

6901 (SubSig 0) NET FLOOD Icmp Reply :
6902 (SubSig 0) NET FLOOD Icmp Request :
6903 (SubSig 0) NET FLOOD Icmp Any :
6910 (SubSig 0) NET FLOOD UDP :
6920 (SubSig 0) NET FLOOD TCP :

(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

Table 7.8 shows the configurable parameters for FLOOD.NET signatures.

Table 7.8 FLOOD.NET Parameters

Name	Data Type	Protected	Required	Description
Gap	Number	No	No	The number of seconds allowed within the ThrottleInterval where PPS < Rate. Alarms will not be triggered if you get greater than Gap seconds that are not suspects and counting is reset.
IcmpType	Number 0–256	No	No	This is the ICMP type value found in the header. Only valid when Protocol is set to ICMP.
Peaks	Number	No	No	The threshold of suspect seconds. Alarm is triggered when the Peaks suspect seconds is reached in a ThrottleInterval.
Rate	Number	No	No	The threshold for PPS. Suspect second occurs when PPS > Rate. Remember for diagnostics/feedback mode to set the Rate value to 0.

The STATE.HTTP Micro-Engine

The STATE micro-engine encompasses the 3000 and 5000 series signatures. There are approximately 415 signatures covered in this micro-engine. The STAT.HTTP micro-engine is especially helpful if you are running a web server on nonstandard HTTP ports. Use the **Configuration | Sensing Engine | Signature Configuration | STATE.HTTP Service Ports** in IDM to add those ports (see Figure 7.15). Choose **option 14** for configuring the parameters

in SigWizMenu. For all the configuration parameters for this engine, refer to Table 7.9. Examples of some of these signatures are

- **3221-WWW cgi-viewsource Attack** Fires when someone attempts to use the `cgi-viewsource` script to view files above the `http` root directory.
- **3222-WWW PHP Log Scripts Read Attack** Fires when someone attempts to use the PHP scripts `mlog` or `mylog` to view files on a machine.
- **3223-WWW IRIX cgi-handler Attack** Fires when someone attempts to use the `cgi-handler` script to execute commands.
- **3224-HTTP WebGais** Fires when someone attempts to use the `web-gais` script to run arbitrary commands.
- **3225-WWW websendmail File Access** Fires when unauthorized attempts are made to read a file using the `websendmail` CGI program.
- **3226-WWW Webdist Bug** Fires when attempts are made to use the `webdist` program. False positive alarms will fire from legitimate use of the `webdist` program.
- **3227-WWW Htmlscript Bug** Fires when attempts are made to view files above the `html` root directory.
- **3228-WWW Performer Bug** Fires when attempts are made to view files above the `html` root directory.
- **3229-Website Win-C-Sample Buffer Overflow** Fires when attempts are made to access the `win-c-sample` program in the WebSite server distribution. Testing new Web site servers or upgrades using the `win-c-sample` program can cause false positives. This script is for testing purposes and should be removed on production servers.
- **3230-Website Uploader** Fires when attempts are made to access the uploader program in the Web site server distribution.

For a full list of all of these signatures, refer to Appendix A.

Figure 7.15 IDM STATE.HTTP Service Ports



Table 7.9 STATE.HTTP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
ArgNameRegex	Number	Yes	No	Regular expression searches the HTTP Arguments field.
ArgValueRegex	Number	Yes	No	Regular expression searches the HTTP Arguments field after ArgNameRegex is matched. You have to define ArgNameRegex for this match to work. It is an ordered match.
Deobfuscate	Boolean True/False	No	No	Use anti-evasive deobfuscation prior to searching for the RegexString.

Continued

Table 7.9 STATE.HTTP Parameters

Parameter	Data Type	Protected	Required	Description
Direction	Boolean from Service to Service	Yes	No	Indicates the direction in which the sensor is watching traffic at the service port.
HeaderRegex	String	Yes	No	Regular expression used to search within the HTTP Header field.
MaxArgFieldLength	Number	No	No	Maximum length of the Arguments field.
MaxHeaderField Length	Number	No	No	Maximum length of the Header field.
MaxRequestField Length	Number	No	No	Maximum length of the Request field.
MaxUriFieldLength	Number	No	No	Maximum length of the URI field.
ServicePorts	Set	No	No	Comma-separated list of ports or port ranges where the service resides.
UriRegex	String	Yes	No	Regular expression to use to search within the HTTP URI field.

The STRING Micro-Engine

The STRING micro-engine provides pattern inspection and alarm generation against regular expressions. It works against TCP, UDP, and ICMP. There are currently four STRING micro-engines.

STRING HTTP has eight signatures (shown in Figure 7.16). These are specifically tailored to look for certain command strings in HTTP traffic.

Figure 7.16 SigWizMenu Option 15 STRING.HTTP

```

Selection> 15

5123 (SubSig 0) WWW IIS Internet Printing Overflow : Host:<250+ Chars>
5168 (SubSig 0) Snapstream PVS Directory Traversal Vulnerability : ../
5169 (SubSig 0) Snapstream PVS Plaintext Password Vulnerability :
../ssd.ini
5172 (SubSig 0) WinWrapper Admin Server Directory Traversal : ../
5188 (SubSig 0) HTTP Tunnelling : GET /erc/Poll?machineKey
5188 (SubSig 0) HTTP Tunnelling : POST /index.html?crap
5191 (SubSig 0) Active Perl PerlIS.dll Buffer Overflow : *.pl
5289 (SubSig 0) SQLXML ISAPI Buffer Overflow : contentType=text/AAA...
<240+>...

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

Table 7.10 shows the configurable parameters for STRING.HTTP signatures.

Table 7.10 STRING.HTTP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
Deobfuscate	Boolean True/False	No	No	Use anti-evasive deobfuscation prior to searching for the RegexpString.
Direction	Boolean From Service To Service	Yes	No	Indicates the direction in which the sensor is watching traffic at the service port.
MinMatch Length	Number	No	No	Minimum number of bytes the RegexpString must match.
MultipleHits	Boolean True/False	No	No	Search for multiple RegexpStrings in a single packet.

Continued

Table 7.10 STRING.HTTP Parameters

Parameter	Data Type	Protected	Required	Description
PreFilterDepth	Number	No	No	This is a list of strings to filter on or match before Regex starts its search. At least one of the strings in this list must be found in the first PreFilterDepth bytes of the stream to be considered a valid web stream.
RegexString	String	Yes	Yes	Regular expression to search on.
ServicePorts	Set	No	No	Comma-separated list of ports or port ranges where the service resides.
StripTelnet Options	Boolean True/False	No	No	Strips Telnet option characters from data before searching.

STRING ICMP signatures will fire upon detecting a series of three pluses (+) in an ICMP packet, as shown here:

```
Selection> 16
```

```
2155 (SubSig 0) Modem DoS : +++ (ICMP)
```

```
(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

Table 7.11 shows the configurable parameters for STRING.ICMP signatures.

Table 7.11 STRING.ICMP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.

Continued

Table 7.11 STRING.ICMP Parameters

Parameter	Data Type	Protected	Required	Description
Direction	Boolean from Service to Service	No	No	Indicates the direction in which the sensor is watching traffic at the service port.
MinMatchLength	Number	No	No	Minimum number of bytes the RegexpString must match.
MultipleHits	Boolean True/False	No	No	Search for multiple RegexpStrings in a single packet.
RegexpString	String	Yes	Yes	Regular expression to search on.
ServicePorts	Set	No	No	Comma-separated list of ports or port ranges where the service resides.
StripTelnetOptions	Boolean True/False	No	No	Strips Telnet option characters from data before searching.

STRING.TCP looks for strings in commands or text in TCP sessions. There are approximately 165 different signatures in this micro-engine. Refer to Appendix A for a complete list.

Examples of some of the signatures are

- **3117-KLEZ worm** The alarm triggers when a filename Gn.Exe is found as an audio/x-wav attachment to an e-mail.
- **3118-rwhoisd format string** This sig fires upon detecting a *soa* command sent to a rwhois server with a large argument.

- **3119-WS_FTP STAT overflow** Fires upon detecting a *stat* command with an argument that is greater than 450 characters.
- **3120-ANTS virus** The alarm triggers when an e-mail is found with the attachment ANTS3SET.EXE.
- **3121-Vintra MailServer EXPN DoS** Fires when **@* is detected as the argument to the SMTP command *EXPN*.
- **3122-SMTP EXPN root Recon** Fires when an attempt to expand the e-mail alias of the *root* user with the SMTP command *EXPN* is detected.
- **3123-NetBus Pro Traffic** Alarm fires upon detecting a Netbus Pro communications channel setup.
- **3124-Sendmail prescan Memory Corruption** This signature looks for an abnormally long (1000+ characters) *MAIL FROM* (SubSig 0) or *RCPT TO* (SubSig 1) SMTP command.

Table 7.12 shows the configurable parameters for STRING.TCP signatures.

Table 7.12 STRING.TCP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
Direction	Boolean from Service to Service	Yes	No	Indicates the direction in which the sensor is watching traffic at the service port.
MinMatch Length	Number	No	No	Minimum number of bytes the RegexString must match.
MultipleHits	Boolean True/False	No	No	Search for multiple RegexStrings in a single packet.
RegexString	String	Yes	Yes	Regular expression to search on.

Continued

Table 7.12 STRING.TCP Parameters

Parameter	Data Type	Protected	Required	Description
ServicePorts	Set	No	No	Comma-separated list of ports or port ranges where the service resides.
StripTelnetOptions	Boolean True/False	No	No	Strips Telnet option characters from data before searching.

STRING.UDP looks for strings in UDP traffic. Without beating this to a pulp, remember we are looking at strings in payloads. A lot of the tools used to exploit systems use UDP. Refer to Appendix A for a complete list. Some examples of UDP string signatures are

- **4607-Deep Throat Response** This signature triggers when the string *My Mouth is Open* is detected in a UDP packet sent on well-known Deep Throat UDP ports. Alarm level 5.
- **4608-Trinoo (UDP)** This signature triggers when the string *trinoo* is detected on any UDP port known to have Trinoo traffic. Alarm level 5.
- **4609-Orinoco SNMP Info Leak** This signature triggers when a specially crafted packet is detected with a destination of UDP port 192. This is a good indicator that attempts are being made to retrieve the SNMP community names from the target. Alarm level 4.
- **4610-Kerberos 4 User Recon** This signature triggers when a null character sent to UDP port 750 is detected. This is a good indicator that a Kerberos user recon attack may be occurring. Alarm level 0.

Table 7.13 shows the configurable parameters for STRING.UDP signatures.

Table 7.13 STRING.UDP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.

Continued

Table 7.13 STRING.UDP Parameters

Parameter	Data Type	Protected	Required	Description
Direction	Boolean from Service to Service	No	No	Indicates the direction in which the sensor is watching traffic at the service port.
MinMatchLength	Number	No	No	Minimum number of bytes the RegexpString must match.
ServicePorts	Set	No	No	Comma-separated list of ports or port ranges where the service resides.

The SWEEP Micro-Engine

All of the SWEEP signatures alarm conditions depend on the count of the *Unique* parameter. Unique is the threshold parameter that causes the signature to fire the alarm when more than the configured “Unique” number of ports and hosts is seen on the address set within the time period. This process, tracking unique port/host traffic, is referred to as counting. In order for traffic to be put into the counting section, other parameters such as Mask/TcpFlags, IcmpType, WantFrag Boolean, and/or the UDP ports. If the packet conditions are not met and the sweep occurs, review the settings for these parameters and tune as necessary.

The SWEEP micro-engines include the following types.

*SWEEP.HOST.**

The SWEEP.HOST.* micro-engines analyze traffic from a single host to many hosts, particularly ICMP and TCP. The two micro-engines are SWEEP.HOST.ICMP and SWEEP.HOST.TCP (see Figures 7.17 and 7.18). The signatures fire when the Unique count of host exceeds the configured setting. Examples of these signature are

- 2100-ICMP Network Sweep w/Echo** Fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8 (Echo Request). Alarm level 3.

- **2101-ICMP Network Sweep w/Timestamp** Fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13 (Timestamp Request). Alarm level 5.
- **2102-ICMP Network Sweep w/Address Mask** Fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17 (Address Mask Request). Alarm level 5.
- **3030-TCP SYN Host Sweep** Fires when a series of TCP SYN packets have been sent to the same destination port on a number of different hosts. Alarm level 2.
- **3031-TCP FRAG SYN Host Sweep** Fires when a series of fragmented TCP SYN packets have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3032-TCP FIN Host Sweep** Fires when a series of TCP FIN packets have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3033-TCP FRAG FIN Host Sweep** Fires when a series of TCP FIN packets have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3034-TCP NULL Host Sweep** Fires when a series of TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3035-TCP FRAG NULL Host Sweep** Fires when a series of fragmented TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3036-TCP SYN FIN Host Sweep** Fires when a series of TCP packets with both the SYN and FIN flag sets have been sent to the same destination port on a number of different hosts. Alarm level 5.
- **3037-TCP FRAG SYN FIN Host Sweep** Fires when a series of TCP packets with both the SYN and FIN flag sets have been sent to the same destination port on a number of different hosts. Alarm level 5.

Figure 7.17 SigWizMenu Option 19 SWEEP.HOST.ICMP

```

Selection> 19

2100 (SubSig 0) Net Sweep-Echo :
2101 (SubSig 0) Net Sweep-Time :
2102 (SubSig 0) Net Sweep-Mask :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

Table 7.14 shows the configurable parameters for SWEEP.HOST.ICMP signatures.

Table 7.14 SWEEP.HOST.ICMP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
IcmpType	Number	No	Yes	ICMP header type of interest.
Unique	Number 2–40	No	Yes	Maximum Unique connections to the target.

Figure 7.18 SWEEP.HOST.TCP

```

Selection> 20

3030 (SubSig 0) TCP SYN Host Sweep :
3031 (SubSig 0) TCP FRAG SYN Host Sweep :
3032 (SubSig 0) TCP FIN Host Sweep :
3033 (SubSig 0) TCP FRAG FIN Host Sweep :
3034 (SubSig 0) TCP NULL Host Sweep :
3035 (SubSig 0) TCP FRAG NULL Host Sweep :
3036 (SubSig 0) TCP SYN/FIN Host Sweep :
3037 (SubSig 0) TCP FRAG SYN/FIN Host Sweep :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

Table 7.15 shows the configurable parameters for SWEEP.HOST.TCP signatures.

Table 7.15 SWEEP.HOST.TCP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
Mask	BITSET: FIN/SIN/RST/ PSH/ACK/URG	No	Yes	Mask used for TcpFlags comparison.
TcpFlags	BITSET: FIN/SIN/RST/ PSH/ACK/URG	Yes	Yes	TCP used to match when masked by the Mask parameter.
Unique	Number 2–40 connections to the target.	No	Yes	Maximum Unique

*SWEEP.PORT.**

The SWEEP.PORT.* micro-engines analyze the traffic between two specific hosts and ports. Like the SWEEP.HOST.* engines, SWEEP.PORT.* engines count unique port connections between the hosts. The two micro-engines that fall into this category are SWEEP.PORT.TCP and SWEEP.PORT.UDP (see Figures 7.19 and 7.20). The signatures fire when the Unique count of port connections exceeds the configured setting. At this time, there are only 14 signatures total in these two micro-engines. They are

- **3001-TCP Port Sweep** Fires when a series of TCP connections to a number of different privileged ports (port number < 1024) on a specific host have been initiated. Alarm level 4.
- **3002-TCP SYN Port Sweep** Fires when a series of TCP SYN packets have been sent to a number of different destination ports on a specific host. Alarm level 3.
- **3003-TCP Frag SYN Port Sweep** Fires when a series of fragmented TCP SYN packets are sent to several different destination ports on a specific host. Alarm level 5.

- **3005-TCP FIN Port Sweep** Fires when a series of TCP FIN packets have been sent to a number of different privileged ports (port number < 1024) on a specific host. Alarm level 5.
- **3006-TCP Frag FIN Port Sweep** Fires when a series of fragmented TCP FIN packets have been sent to several different privileged ports (having port number less than 1024) destination ports on a specific host. Alarm level 5.
- **3010-TCP High Port Sweep** Fires when a series of TCP connections to several different high-numbered ports (port number > 1023) on a specific host have been initiated. Alarm level 0.
- **3011-TCP FIN High Port Sweep** Fires when a series of TCP FIN packets have been sent to several different destination high-numbered ports (having port number greater than 1023) on a specific host. Alarm level 5.
- **3012-TCP Frag FIN High Port Sweep** Fires when a series of fragmented TCP FIN packets have been sent to several different destination high-numbered ports (port number > 1023) on a specific host. Alarm level 5.
- **3015-TCP Null Port Sweep** Fires when a series of TCP packets with none of the SYN, FIN, ACK, or RST flag sets have been sent to several different destination ports on a specific host. Alarm level 5.
- **3016-TCP Frag Null Port Sweep** Fires when a series of fragmented TCP packets with none of the SYN, FIN, ACK, or RST flag sets have been sent to several different destination ports on a specific host. Alarm level 5.
- **3020-TCP SYN FIN Port Sweep** Fires when a series of TCP packets with both the SYN and FIN flag sets have been sent to several different destination ports on a specific host. Alarm level 5.
- **3021-TCP Frag SYN FIN Port Sweep** Fires when a series of fragmented TCP packets with both the SYN and FIN flags set have been sent to several different destination ports on a specific host. Alarm level 5.
- **4001-UDP Port Sweep** Fires when a series of UDP connections to several different destination ports on a specific host have been initiated. This is an indicator of a reconnaissance sweep of your network. Be wary of potentially more serious attacks. Alarm level 0.

- 4003-Nmap UDP Port Sweep** Fires when a series of UDP connections to several different privileged ports (port number < 1024) on a specific host have been initiated. This is an indicator of a reconnaissance sweep of your network. Be wary of potentially more serious attacks. Alarm level 5

Figure 7.19 SigWizMenu Option 21 SWEEP.PORT.TCP

```

Selection> 21

3001 (SubSig 0) TCP Port Sweep :
3002 (SubSig 0) TCP SYN Port Sweep :
3003 (SubSig 0) TCP FRAG SYN Port Sweep :
3005 (SubSig 0) TCP FIN Port Sweep :
3006 (SubSig 0) TCP FRAG FIN Port Sweep :
3010 (SubSig 0) TCP High Port Sweep :
3011 (SubSig 0) TCP FIN High Port Sweep :
3012 (SubSig 0) TCP FRAG FIN High Port Sweep :
3015 (SubSig 0) TCP Null Port Sweep :
3016 (SubSig 0) TCP FRAG Null Port Sweep :
3020 (SubSig 0) TCP SYN FIN Port Sweep :
3021 (SubSig 0) TCP FRAG SYN FIN Port Sweep :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

Table 7.16 shows the configurable parameters for SWEEP.PORT.TCP signatures.

Table 7.16 SWEEP.PORT.TCP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.

Continued

Table 7.16 SWEEP.PORT.TCP Parameters

Parameter	Data Type	Protected	Required	Description
InvertedSweep	Boolean: True/False	No	NO	Parameter to force the sensor to compare the signature against traffic to the source port instead of the destination port for unique counting.
Mask	BITSET: FIN/SIN/RST/ PSH/ACK/URG	Yes	Yes	Mask used for TcpFlags comparison.
PortRange	Number	No	Yes	Three port range options: (1) for low ports, (2) for high ports, (0) for all ports.
SupressReserve	Boolean: True/False	No	No	Suppresses the alarm when a sweep is going in the opposite direction.
TcpFlags	BITSET: FIN/SIN/RST/ PSH/ACK/URG	Yes	Yes	TCP used to match when masked by the Mask parameter.
Unique	Number 2–40	No	Yes	Maximum Unique connections to the target.

Figure 7.20 SigWizMenu Option 22 SWEEP.PORT.UDP

```
Selection> 22
```

```
4001 (SubSig 0) UDP Port Sweep :
```

```
4003 (SubSig 0) Nmap Udp Port Sweep : NMAP UDP port Sweep :
```

```
(Sig Number to EDIT) or (ENTER to CONTINUE) >
```

Table 7.17 shows the configurable parameters for SWEEP.PORT.UDP signatures.

Table 7.17 SWEEP.PORT.UDP Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
PortsInclude	String	Yes	Yes	List of ports and/or ranges for the engine to inspect for sweeps.
Unique	Number 2–40	No	Yes	Maximum Unique connections between two hosts.

SWEEP.RPC

SWEEP.RPC is the final SWEEP micro-engine (Figure 7.21). It analyzes Remote Procedure Call (RPC) traffic between hosts. The signatures that fall under the SWEEP.RPC micro-engine are

- **6110-RPC RSTATD Sweep** Fires when RPC requests are made to many ports for the RSTATD program. Alarm level 5.
- **6111-RPC RUSERSD Sweep** Fires when RPC requests are made to many ports for the RUSERSD program. Alarm level 5.
- **6112-RPC NFS Sweep** Fires when RPC requests are made to many ports for the NFS program. Alarm level 5.
- **6113-RPC MOUNTD Sweep** Fires when RPC requests are made to many ports for the MOUNTD program. Alarm level 5.
- **6114-RPC YPPASSWDD Sweep** Fires when RPC requests are made to many ports for the YPPASSWDD program. Alarm level 5.
- **6115-RPC SELECTION_SVC Sweep** Fires when RPC requests are made to many ports for the SELECTION_SVC program. Alarm level 5.
- **6116-RPC REXD Sweep** Fires when RPC requests are made to many ports for the REXD program. Alarm level 5.

- **6117-RPC STATUS Sweep** Fires when RPC requests are made to many ports for the STATUS program. Alarm level 5.
- **6118-RPC ttldb Sweep** Fires on an attempt to access the tooltalk database daemon on multiple ports on a single host. Alarm level 5.

Figure 7.21 SigWizMenu Option 23 SWEEP.RPC

```

Selection> 23

6110 (SubSig 0) RPC RSTATD Sweep :
6111 (SubSig 0) RPC RUSESRD Sweep :
6112 (SubSig 0) RPC NFS Sweep :
6113 (SubSig 0) RPC MOUNTD Sweep :
6114 (SubSig 0) RPC YPASSWDD Sweep :
6115 (SubSig 0) RPC SELECTION SVC Sweep :
6116 (SubSig 0) RPC REXD Sweep :
6117 (SubSig 0) RPC STATUS Sweep :
6118 (SubSig 0) RPC TTDB Sweep :

(Sig Number to EDIT) or (ENTER to CONTINUE) >

```

Table 7.18 shows the configurable parameters for SWEEP.RPC signatures.

Table 7.18 SWEEP.RPC Parameters

Parameter	Data Type	Protected	Required	Description
Master parameters				Refer to Table 7.1 for the master parameters.
RpcProgram	Number	Yes	Yes	RPC program number request.
Unique	Number 2–40	No	Yes	Maximum allowed destination ports receiving RPCs with program number request RpcProgram.

If you would like more information regarding any of the preceding signatures refer to Appendix A or go to Cisco's web site: <http://www.cisco.com>.

The OTHER Engine

After going through the ten or so different signature series and becoming familiar with the different micro-engines, you may have wondered: what if there is a signature that does not fit the other engines? What happens? Does Cisco just forget about it? Not a chance. What Cisco has done is create an engine for all the signatures that do not fit any other engine protocol decode. It's called the OTHER engine. The OTHER engine does not allow you to define any custom signatures or add any signatures. The signatures that fall into the OTHER engine are

- **993-Missed Packet Count** This signature is triggered when the sensor is dropping packets and the percentage dropped can be used to help you tune the traffic level you are sending to the sensor. For example, if the alarms show that there is a low count of dropped packets or even zero, the sensor is monitoring the traffic without being overutilized. On the other hand, if 993 alarms show a high count of dropped packets, the sensor may be oversubscribed. Alarm level 1.
- **994-Traffic Flow Started** This signature triggers when traffic to the sensing interface is detected for the first time or resumes after an outage. SubSig 1 fires when initial network activity is detected. SubSig 2 fires when the link (physical) layer becomes active. Alarm level 1.
- **995-Traffic Flow Stopped** Subsignature 1 is triggered when no traffic is detected on the sensing interface. You can tune the timeout for this via the TrafficFlowTimeout parameter. SubSignature 2 is triggered when a physical link is not detected. Alarm level 1.
- **996-Route Up** This signifies that traffic between the sensor and director has started. When the services on the director and/or sensor are started, this alarm will appear in the event viewer. Alarm level 1.
- **997-Route Down** This signifies that traffic between the sensor and director has stopped. When the services on the director and/or sensor are started, this alarm will appear in the event viewer. Alarm level 1.
- **998-Daemon Down** One or more of the IDS sensor services has stopped.

- **999-Daemon Unstartable** One or more of the IDS sensor services is unable to be started.
- **1200-IP Fragmentation Buffer Full** This signature is triggered when there is an extraordinary amount of incomplete fragmented traffic detected on the protected network. Alarm level 1.
- **1201-IP Fragment Overlap** This signature is triggered when two fragments contained within the same IP datagram have offsets that indicate that they share positioning within the datagram. Alarm level 5.
- **1202-IP Fragment Overrun - Datagram Too Long** Fires when a reassembled fragmented datagram would exceed the declared IP data length or the maximum datagram length. Alarm level 5.
- **1203-IP Fragment Overwrite - Data is Overwritten** Fires upon detecting an IP fragment that overlaps a previous fragment. This behavior is consistent with the *Ping of Death*. Alarm level 5.
- **1204-IP Fragment Missing Initial Fragment** Fires when a datagram can not be reassembled due to missing initial data. Alarm level 1.
- **1205-IP Fragment Too Many Datagrams** This signature is triggered when there is an excessive number of incomplete fragmented datagrams detected on the network. Alarm level 2.
- **1206-IP Fragment Too Small** Fires when any fragment other than the final fragment is less than 400 bytes, indicating that the fragment is likely intentionally crafted. Alarm level 2
- **1207-IP Fragment Too Many Frags** This signature is triggered when there is an excessive number of fragments for a given datagram. This is most likely either a Denial-of-Service attack or an attempt to bypass security measures. Alarm level 2
- **1208-IP Fragment Incomplete Datagram** Fires when a datagram can not be fully reassembled due to missing data. Alarm level 2
- **1220-Jolt2 Fragment Reassembly DoS attack** This alarm will fire when multiple fragments are received, all claiming to be the last fragment of an IP datagram. Alarm level 5.
- **3050-Half-open SYN Attack** Fires when multiple TCP sessions have been improperly initiated on any of several well-known service ports. Alarm level 5.

- **3250-TCP Hijack** Fires when both data streams of a TCP connection indicate that TCP hijacking has occurred. TCP Hijacking is used to gain illegal access to system resources. False positives are possible. Alarm level 5
- **3251-TCP Hijacking Simplex Mode** Fires when both data streams of a TCP connection indicate that TCP hijacking has occurred. TCP hijacking is used to gain illegal access to system resources. Simplex mode means that only one command is sent, followed by a connection RESET packet, which makes recognition of this signature different from regular TCP hijacking (sigID 3250). False positives are possible. The most common network event that may trigger this signature is an idle Telnet session. The TCP Hijack attack is a low-probability, high level-of-effort event. If it is successfully launched, it could lead to serious consequences, including system compromise. The source of these alarms should be investigated thoroughly before any actions are taken. Recommend security professional consultation to assist in the investigation. Alarm level 5.
- **5249-IDS Evasive Encoding** This signature looks for special characters such as Null %00, New Line %0a, Carriage Return %0d, Period “.” %2e, Forward Slash “/” %2f, and Back Slash “\” %5c in the URL of an HTTP request that have been encoded in hexadecimal vice the actual character. This is a technique used to evade detection of an attack. This signature is triggered if any of the aforementioned characters are detected as being encoded in part of the URL. Alarm level 4.
- **5250-IDS Evasive Double Encoding** This signature looks for special characters such as Null %00, New Line %0a, Carriage Return %0d, Period “.” %2e, Forward Slash “/” %2f, and Back Slash “\” %5c in the URL of an HTTP request that have been encoded in hexadecimal vice the actual character in the URL of an HTTP request that have been “doubly” encoded. This is a technique used to evade detection of an attack. This signature is triggered if any of the before mentioned characters are detected as being doubly encoded as part of a URL. Alarm level 4.

Table 7.19 shows the configurable parameters for the OTHER micro-engine signatures.

Table 7.19 OTHER Micro-Engine Parameters

Parameter	Data Type	Protected	Required	Description
HijackMax OldAck	Number	No	No	Maximum number of old dataless client-to-server ACKs allowed before a Hijack alarm is triggered.
HijackReset	BOOLEAN; True/False	No	No	Hijack signature requires a reset.
ServicePorts	Port Range	No	No	List of ports and/or port ranges the target service may be listening to.
SynFloodMax Embryonic	Number	No	No	The maximum number of simultaneous embryonic connections allowed to any service. Embryonic connections are half-open connections.
TrafficFlow Timeout	NUMBER	No	No	This is the number of seconds that no traffic is detected on the segment.

Understanding Cisco IDS Signature Series

Now we are going to discuss each of the signatures. I have taken the time to separate them into the numbered series. The signatures range from 1000 all the way into the 11000s. Besides numerically grouping signatures, the series number represents another type of grouping. They help the administrator narrow down what type of attack is generating the alarms. Are they atomic? Is the attack a string, sweep, or web site exploit? Although the numbers do cover multiple signature types, they help the administrator narrow down his search.

The following list gives a brief description of each signature series.

- The 1000 series covers the signatures that analyze the content of IP headers.
- The 2000 series focuses on ICMP signatures.
- The 3000 series is all about TCP-based signatures.
- The 4000 series is all about UDP connections and ports on the network.
- The 5000 series is probably the largest. It covers web (HTTP) traffic.
- The 6000 series focuses on multiprotocol signatures.
- The 7000 series has the ARP signatures.
- The 8000 series is string-matching signatures.
- The 9000 series covers Back Doors.
- The 10000 series has signatures that focus on policy enforcement.

Configuring the Sensing Parameters

Configuring the sensing parameters is very important on the network. You have to tell the sensor how to do TCP Session reassembly, IP fragment reassembly, how to define internal networks, and specify data sources. These are critical steps. I'll explain what the benefits are as we go along.

TCP Session Reassembly

TCP reassembly causes the sensor to reassemble a TCP session's packets before they are compared against the signatures. This helps keep resources from being tied up. There are three TCP session reassembly options you can choose from: No Reassembly, Loose Reassembly, and Strict Reassembly.

NOTE

This only applies to version 2.5(X) software and later for the IDSM. If you do not have an IDSM, this section will not apply.

No Reassembly

Simply stated, the sensor does not reassemble TCP sessions. All packets are processed on arrival. No reassembly can generate false positives and negatives because of the potential for packets being processed out-of-order. It is not recommended unless your network is subject to a higher-than-normal rate of packet loss.

Loose Reassembly

A step up from not reassembling at all, loose reassembly does process all packets in order. The problem loose reassembly causes is the same though. False positive alarms are generated because the sensor allows gaps in the sequence when reassembling the session record.

Strict Reassembly

If you are going to do TCP session reassembly, strict reassembly is the way to go. I'd like to say there is no chance of any false positives or negatives, but you might try and hold me to it. The odds are in my favor though. Unless all of the packets are received and the session is completely reassembled, the sensor will not analyze the session.

WARNING

Remember, when we talk about reassembly (whenever you have a network device do any type of reassembly of fragments, sessions, and so on...), we're talking about the overhead involved. It will consume memory and be CPU-intensive.

Configuring TCP Session Reassembly

In order to configure TCP Session Reassembly, follow these steps:

1. In CSPM, select the **Sensing** configuration tab of the sensor you want to configure.
2. Select **TCP Three-Way Handshake** in the configuration screen. This tracks only three-way handshakes that are complete.
3. Choose what method you will use for reassembly.

4. Define values for **TCP Open Establish Timeout** and **TCP Embryonic Timeout**.
5. Once you have finished configuring the Sensing parameters, click **OK**, then save and update your configuration.
6. Finally, from the **Command** tab, click **Approve Now** to push the new configuration to your sensor.

NOTE

TCP Open Establish Timeout gives the number of seconds before the sensor frees the resources allocated for established TCP sessions. Ninety seconds is the default. *TCP Embryonic Timeout* gives the number of seconds before the sensor frees the resources allocated for half-open TCP sessions. Fifteen seconds is the default.

IP Fragment Reassembly

IP fragment reassembly is very similar to the TCP session reassembly. IP reassembly causes the sensor to reassemble IP packets before they are compared against the signatures. This helps to keep resources from being tied up, since reconstruction does consume some resources. IP fragment reassembly has three parameters:

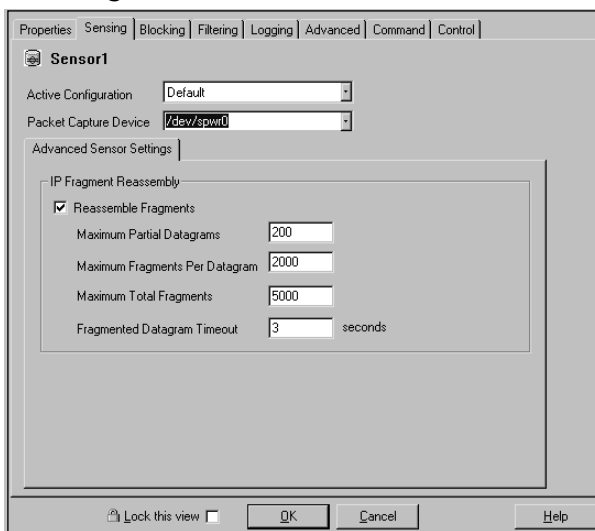
- **Maximum Partial Datagrams** The maximum number of partial datagrams the sensor will attempt to reconstruct at any time.
- **Maximum Fragments Per Datagram** The maximum number of fragments that are accepted for a single datagram.
- **Fragmented Datagram Timeout** The maximum number of seconds before the sensor stops trying to reassemble a datagram.

Configuring IP Fragment Reassembly

To configure IP fragment reassembly, follow these steps:

1. Select the **Sensing** tab on the sensor you want to configure.
2. Check the **Reassemble Fragments** check box (refer to Figure 7.22).

2. Enter the settings for **Maximum Partial Datagrams**, **Maximum Fragments Per Datagram**, and **Fragmented Datagram Timeout**.
3. Once you have finished configuring the Sensing parameters, click **OK**, then save and update your configuration.
4. From the **Command** tab, click **Approve Now** to push the new configuration to your sensor.

Figure 7.22 The Sensing Tab**NOTE**

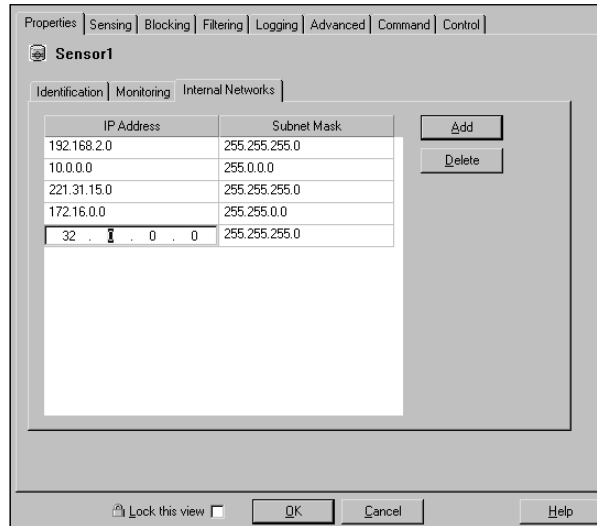
Cisco's recommended guidelines for determining the maximum partial datagrams and maximum fragments per datagram is as follows (it takes a little math here):

- The partial datagrams multiplied by the fragments per datagram should be less than 2,000,000. This applies to all 4200 series sensors running versions 2.2.1.5 or 2.5(X).
- The partial datagrams multiplied by the fragments per datagram should be less than 5000. This applies to the IDSMs running versions 2.5(X).

Internal Networks

What is the purpose of identifying internal networks, you ask? Well, you want to log all the alarms, right? You want the events to make sense to you, right? How much use would your logs be if everything was considered an external address marked with “OUT”? So, to be able to differentiate from internal and external networks and hosts, Cisco has given you the ability to configure internal networks into the mix so the events are easier to understand. In this section, you will define your Internal Protected networks that the sensor is protecting. CSPM uses this to parse the events in Event Viewer. Any address space that is not identified in this section is considered an external address designated as “OUT”. The internal addresses are designated as “IN” (see Figure 7.23).

Figure 7.23 Internal Networks



Adding Internal Networks

To add networks that are labeled as internal networks (IN), follow these steps:

1. Select the sensor you want to configure. The first tab showing should be the **Properties** tab. If it is not, select the **Properties** tab.
2. Select the **Internal Networks** subtab and click **Add**.
3. Enter all of the networks and subnet masks you want to be identified as internal (IN) addresses for logging purposes.

4. Once you have finished adding networks, click **OK**, then save and update your configuration.
5. From the **Command** tab, click **Approve Now** to push the new configuration to your sensor.

Sensing Properties

As you have read in Chapter 4, the Sensing tab allows you to configure what signature configuration file the sensor is using, what Packet Capture Device (Interface) the sensor is using, and how to handle IP fragment reassembly. You can specify the active configuration, which is the signature file the sensor is using for comparison. You also set the Packet Capture Device. This is the sniffing interface. This is also the tab that you configure for IP fragment reassembly (discussed earlier in this chapter).

Configuring Sensing Properties

To configure the sensing properties, follow these steps:

1. Select the **Sensing** tab on the sensor you are going to configure (see Figure 7.22 earlier).
2. In the Active Configuration field, select the Sensor Signature file template that the sensor will be using to monitor the network. It is not uncommon to have a different Sensor Signature file template for each sensor. Some signatures may be disabled or tuned differently depending on the positioning on the network.
3. Select the appropriate Packet Capture device for your device and network. The Packet Capture device is the interface that is doing the sniffing. (Refer to Chapter 3 for help with the different interfaces on a sensor.)
4. If you are configuring IP fragment reassembly, make your configuration changes here. IP fragment reassembly causes your sensor to reassemble a fragmented IP packet first, and then compare that packet with a signature. This can be a resource hog depending on your network traffic patterns. Unless you are very familiar with the traffic patterns on your network, do not modify the default settings.
5. Once you have finished configuring the Sensing parameters, click **OK**, then save and update your configuration.

6. From the **Command** tab, click **Approve Now** to push the new configuration to your sensor.

Excluding or Including Specific Signatures

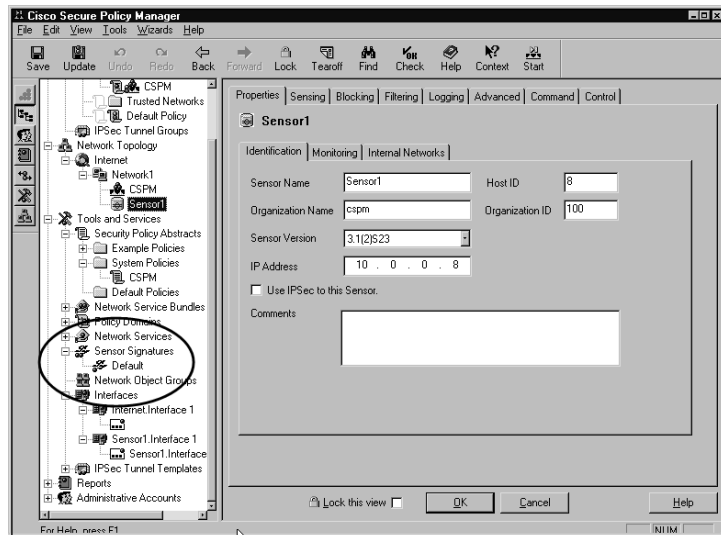
After viewing events for several days and analyzing the traffic along with the source and destination addresses, you may want to turn certain signatures off and others on. There could be several reasons why you would want to exclude signatures. They range from too many alarms to false positives being generated by legitimate traffic patterns such as networking monitoring tools using ICMP to check that a node is alive. The ICMP would trigger most ICMP alarms even though the traffic is perfectly legitimate. This tuning process of the sensor by excluding signatures that are not pertinent to your network, or perhaps turning some on that were previously off, will add quite a bit of value to your security effort.

Excluding or Including Signatures in CSPM

To exclude or include a signature in CSPM, perform these steps:

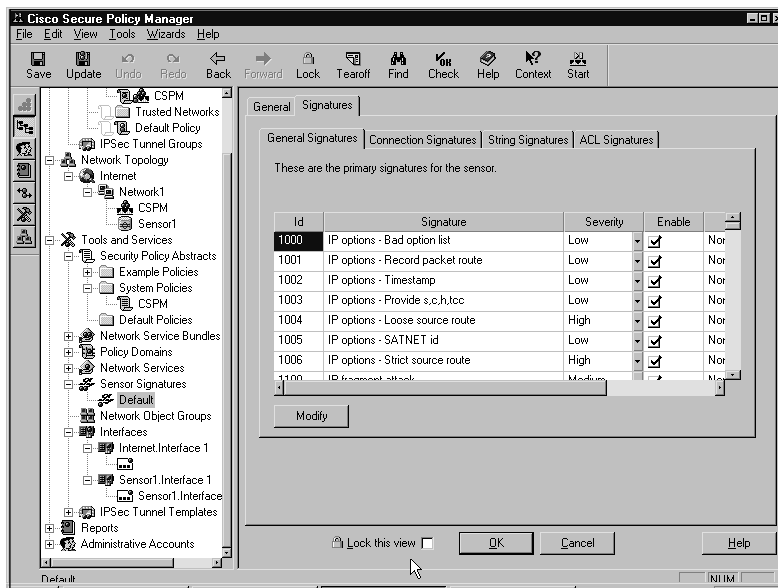
1. Select the signature file you want to edit from the topology map (as seen in Figure 7.24).

Figure 7.24 Signature Files



2. Click the **Signatures** tab and select the appropriate subtab, **General Signatures**, **Connection Signatures**, **String Signatures**, or **ACL Signatures**. Refer to Figure 7.25.

Figure 7.25 The Signatures Tab



3. You will see the **Enable** column to the right of the signature screen. To disable the signature, uncheck the boxes, or, if you want to enable a signature, put a check in the box to enable it. Continue this process until you have finished making changes.
4. Once you have finished enabling and disabling the signatures, click **OK**, then save and update your configuration.
5. From the **Command** tab, click **Approve Now** to push the new configuration to your sensor.

Excluding or Including Signatures in IDM

To exclude or include signatures using the Cisco IDM, follow these steps:

1. Once you have logged in to IDM, go to **Configuration | Signature Groups**. Click the group name that your signature is associated with (see Figure 7.26). Drill down until you get to the signature you want to configure. Select the signature you want to enable or disable.

Figure 7.26 IDM Signature Groups



- Simply check the box of the signature to enable and uncheck the boxes of the signatures you want to disable or have excluded.
- Once you have tuned all of your signatures, use the **Apply Changes** button to implement the changes.

Creating a Custom Signature

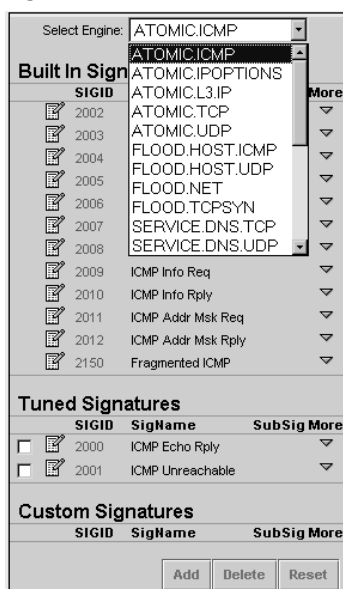
The task of creating custom signatures can be difficult and, at first glance, seem overwhelming, but the following steps will hopefully have you off and running in no time. Even though Cisco supplies us with several hundred signatures, you may have to still create a custom signature because of odd traffic on your network or because of a new security threat. Also, string signatures may come in handy when new vulnerabilities are published on the network without patches and/or tuned signatures to combat them. A good source of signature files to work with as a starting point is the Snort signature file archive. While you can not use the Snort file directly, you can use the offsets and strings contained within the Snort signature file to help build your own Cisco signatures in less time than waiting for the next update from Cisco. In view of how quickly some recent Internet attacks have taken place, this is a good way to provide additional security for your network in a hurry.

Creating Custom Signatures Using IDM

Custom signatures using IDM has the same feel as if you were doing it with the Signature Wizard, discussed later in the chapter. Once you get logged into IDM for the sensor you want to create a custom signature for, follow these steps:

1. From the main screen, go to **Configuration | Custom Signatures**. Select the engine that your custom signature will apply to, as shown in Figure 7.27.

Figure 7.27 Custom Signatures



NOTE

Notice the *Tuned Signatures* section in Figure 7.27. Once you have changed any of the preconfigured signatures in a micro-engine, that signature will appear in this section.

2. At the bottom of the screen, click **Add**. On the **Adding** screen, start filling in the information and setting the parameters on the page that will be the signature. Refer to Figure 7.28. If you have questions about

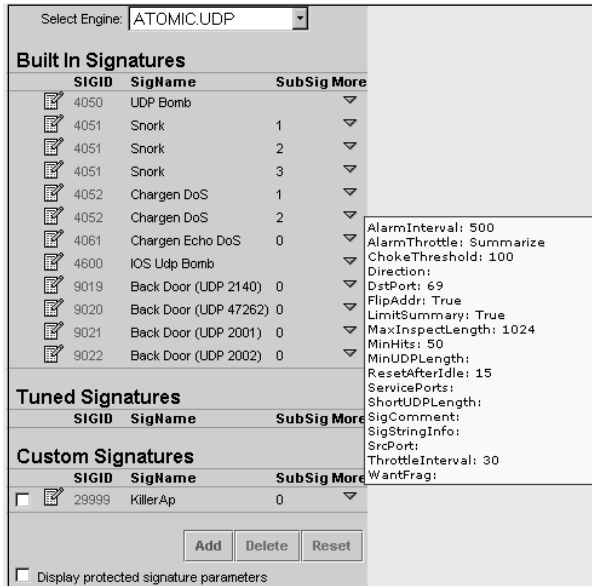
the type of information to add, move your cursor across the field title to get more information.

Figure 7.28 Adding Screen

Adding	
Engine	ATOMIC.UDP
Id (required)	20000
Severity	None
Action	<input type="checkbox"/> Block <input type="checkbox"/> IP Log <input type="checkbox"/> TCP Reset
Comments	
SigName	ATOMIC.UDP
SubSig	
AlarmInterval [20-3600]	
AlarmThrottle	Summarize
ChokeThreshold	100
Direction	<input type="radio"/> FromService <input type="radio"/> ToService <input checked="" type="radio"/> None
DstPort	
FlipAddr	<input type="radio"/> True <input type="radio"/> False <input checked="" type="radio"/> None
LimitSummary	<input type="radio"/> True <input type="radio"/> False <input checked="" type="radio"/> None
MaxInspectLength	
MinHits [0-1000]	
MinUDPLength	
ResetAfterIdle [15-1000]	15
ServicePorts	
ShortUDPLength	<input type="radio"/> True <input type="radio"/> False <input checked="" type="radio"/> None
SigComment	
SigStringInfo	
SrcPort	
ThrottleInterval [0-1000]	30
WantFrag	<input type="radio"/> True <input type="radio"/> False <input checked="" type="radio"/> None
<input type="button" value="Ok"/> <input type="button" value="Cancel"/> <input type="button" value="Reset"/>	

- After you have added all of the required information, click **OK**. The result is having your signature added to the sensor configuration and listed in the **Custom Signatures** section of the micro-engine (see Figure 7.29). When you scroll your mouse across the down-arrow icon to the right, you will see what the configuration is without actually having to open the signature for editing.
- Once you have added all of your custom signatures, you have to apply the changes to the sensor before they will take effect. Click **Apply Changes** in the upper right-hand corner of the IDM screen. Once the changes have been applied, you can then check your event view to see if the custom signatures are firing alarms.

Figure 7.29 Custom Signature in IDM



Creating Custom Signatures Using CSPM

When using CSPM, it can be something of a surprise to you that CSPM can only set a signature's actions and severities. It cannot tune signatures for the IDS sensor appliance. In other words, CSPM can set the severity and the action to associate to the signature but cannot set what triggers that signature. This is where *SigWizMenu* on the Sensor has to be used to tune the Sensors. SigWizMenu and CSPM can both be used to configure the same Sensor since they affect different parts of the configuration. The parameters that will cause the signature to trigger are set by tuning with the SigWizMenu. The tuning involves changing what it takes for a signature to trigger (such as the number of hosts in a sweep) and does not mean setting actions and severity levels.

Working with SigWizMenu

SigWizMenu is the signature wizard that allows you to make changes to IDS signatures directly on the Sensor. CSPM does not allow you to tune thresholds and other parameters. These same changes can also be made via the version 2.2.3 Unix Director. The Signature Wizard is an interim tool for version 2.2.2 Unix Director users until they upgrade to version 2.2.3, as well as Cisco Secure PM users until these options are included in Cisco Secure PM. If you use Cisco Secure PM, you need the Signature Wizard to configure the version 3.0 features.

Starting SigWizMenu

To start SigWizMenu, follow these steps:

1. From the console or Telnet session, login as **netrangr** to the sensor you want to start SigWizMenu on. You should verify you are in the `/usr/nr/bin` directory by using the `pwd` command. If you are not in that directory, use the `cd` command to change to the `/usr/nr/bin` directory. The file is hidden by default so a plain `ls` command will not show the executable.
2. Type **.SigWizMenu** at the command prompt. Don't forget to put the period in front and remember that Unix environments are case-sensitive. Press Enter when prompted. You should get a screen that looks like Figure 7.30.

Figure 7.30 The SigWizMenu Menu

```
-----  
Current Sig Data File '/usr/nr/etc/SigData.conf'  
Current Sig User File '/usr/nr/etc/SigUser.conf'  
Current Settings File '/usr/nr/etc/SigSettings.conf'  
-----  
1 - Tune Signature Parameters  
2 - Add NEW Custom Signature  
3 - Set Custom Signature Severity/Action  
4 - Edit Signature Address Mapping  
5 - Delete Signature Tunings and Custom Signatures  
6 - Other 3.x Tokens  
7 - Display Signatures  
8 - Global Settings  
x - EXIT  
-----  
Selection>
```

3. Enter the option number you want to work with. From this menu, you can perform tasks that are specific to signature behavior.

Notice the three files referenced at the top of the preceding menu printout:

- Current Sig Data File ‘/usr/nr/etc/SigData.conf’
- Current Sig User File ‘/usr/nr/etc/SigUser.conf’
- Current Settings File ‘/usr/nr/etc/SigSettings.conf’ SigData.conf

These files are what the signature wizard uses to operate and maintain a current configuration of all the signatures. The *SigData.conf* file contains the default signatures. When signature update files are applied to a sensor, this file is also updated with current data and is encrypted. The *SigUser.conf* configuration file is where signature modifications and additions are stored. This file is updated when changes are made in the signature wizard, SigWizMenu. The *SigSettings.conf* file is updated and managed through the signature wizard also. It has the global Device Management (packetd) tokens.

Tune Signature Parameters

To tune a signature to your specific needs, you would use option 1 from the SigWizMenu. This allows you to change signature parameters directly on the sensor. There may be a chance that you do not want to see every little ICMP Echo Request generate an alarm. By tuning the signature, you can customize it to summarize the amount of alarms, or raise thresholds before the signature fires. Tuning improves the sensor’s performance and adds credibility to reports by tuning out false positives and false negatives. Cisco provides a list of configurable signature parameters for all versions of the IDS software online at www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/.

Follow these steps to tune your signatures:

1. Select option 1 from the SigWizMenu menu to tune an existing signature.
2. Enter the signature ID of the signature you would like to tune. The list of available configurable parameters will be displayed (see Figure 7.31). Select the number next to the parameter you want to modify. Notice that the bottom-left corner of the screen displays the current value if there are any. Just above the cursor and the current value, a brief description of the parameter is displayed.

- Once you have made all of your modifications, type **X** to save it and continue. This will take you back to the main menu. If you make a mistake, type **U** to undo any changes and continue. This will also take you back to the main menu. To delete a value, type **D** to delete settings for a specified parameter.

Figure 7.31 SigWizMenu Signature Parameters

```

0 - Edit ALL Parameters
1 - AlarmInterval           =
2 - AlarmThrottle          = FireOnce
3 - ChokeThreshold         = 100
4 - FlipAddr               = 8
5 - IcmpCode               =
6 - IcmpId                 =
7 - IcmpMaxCode           =
8 - IcmpMaxSeq            =
9 - IcmpMinCode           =
10 - IcmpMinSeq           =
11 - IcmpSeq              =
12 - IpTOS                 =
13 - LimitSummary         =
14 - MaxInspectLength     =
15 - MinHits              =
16 - ResetAfterIdle       = 15
17 - SigComment           =
18 - SigStringInfo        =
19 - ThrottleInterval     = 30
d - Delete a value
u - UNDO and continue
x - SAVE and continue
-----
Selection> 10
Minimum allowed IcmpSeq. Packets with Seq les than this value will alarm.
      (NUMBER)
- IcmpMinSeq -
[current value]
[new value] >

```

Adding a New Custom Signature

Here is where your specific network traffic patterns can be monitored by using custom signatures. Follow these steps to add a custom signature:

1. Select option 2 from the main menu to add a new custom signature. Several things must take place (see Figure 7.32). You have to select the engine the signature will be used with. A Signature ID must be assigned. If you don't assign it, Cisco will do it for you. Give your signature a name. Configure all of the parameters available to meet your needs. Step 1: Determine what you want the signature to detect.

Figure 7.32 SigWizMenu Adding a New Custom Signature

```
Add NEW Custom Signature : CSIDS Signature Wizard
```

```
-----
1 - Engine Name      'Not Set'
2 - Generate SIGID
3 - Signature ID     'Not Set'
4 - Signature Name   'Not Set'
5 - INSERT NOW
ENTER - BACK TO MAIN
```

```
-----
Selection> 10
```

2. Select option 1 to choose the engine name. All of the micro-engines will appear. Select the one that applies to you by entering the corresponding number at the prompt.
3. Two things can happen on this step. You can either select option 2 and have the signature wizard create a signature ID or you can select option 3 and create your own. Make your choice.
4. Select option 4 to give the signature a name.
5. By selecting option 5, you will insert the new signature into the database. The result is the *Adjust Severity and Action* menu (see Figure 7.33).

Figure 7.33 The Adjust Severity and Action Menu

```
Adjust Severity and Action : CSIDS Signature Wizard
-----
Signature:      21435
Alarm Level:   0  (OFF)
Alarm Action:  0  None
-----
    0 - Turn Signature OFF
    1 - Engine Name      'Not Set'
    2 - Generate SIGID
    3 - Signature ID    'Not Set'
    4 - Signature Name  'Not Set'
    5 - INSERT NOW
ENTER - BACK TO MAIN
-----x--- DONE
-----
Selection>
```

6. Select the Alarm Severity level 1–5 and press Enter. The *Adjust Severity and Action* menu appears (see Figure 7.34).

Figure 7.34 Adjust Severity and Action

```
Adjust Severity and Action : CSIDS Signature Wizard
-----
Signature:      21436
Alarm Level:   4
Alarm Action:  0  None
-----
    0 - Set Action NONE
    1 - Set Action Shun
    2 - Set Action Log
    3 - Set Action Shun & Log
    4 - Set Action Reset
    5 - Set Action Shun & Reset
    6 - Set Action Log & Reset
    7 - Set Action Shun & Log & Reset
```

Continued

Figure 7.34 Adjust Severity and Action

```

ENTER - adjust SEverity
-----x--- DONE
-----
Selection>

```

7. Choose the action you want the signature to perform, then type **x** to complete the task.
8. Type **x** when you are finished. The signature screen with all of the configurable parameters appears. Modify any or all of the parameters you wish. (Refer to Figure 7.35.) Any parameter number that has an asterisk (*) is required and must be set in order to save the settings. Once all of the information is entered, select **x** to SAVE and continue. The signature is now in the database.

Figure 7.35 The Signature Wizard

```

SigName: test sweep
-----
0 - Edit ALL Parameters
1 - AlarmInterval           =
2 - AlarmThrottle          = FireOnce
3 - ChokeThreshold         = 100
4 - FlipAddr               =
5 - LimitSummary           =
6 - MaxInspectLength       =
7 - MinHits                =
8 - ResetAfterIdle         = 15
9 * RpcProgram             =
   10 - SigComment          =
   11 - SigName              = test sweep
12 - SigStringInfo         =
13 - ThrottleInterval      = 30
14 * Unique                =
15 = WantFrag              =
d - Delete a value
u - UNDO and continue
x - SAVE and continue

```

Continued

Figure 7.35 The Signature Wizard

Selection>

9. When you have finished making additions and modifications to your signature database, you must activate the signature. To do this, type **x** to exit the Signature Wizard. Type **y** to save and activate the changes (see Figure 7.36). The packetd activates the new configuration.

Figure 7.36 Activating the Signature

Current Sig User File '/usr/nr/etc/SigUser.conf'

Current Settings File '/usr/nr/etc/SigSettings.conf'

-
- 1 - Tune Signature Parameters
 - 2 - Add NEW Custom Signature
 - 3 - Set Custom Signature Severity/Action
 - 4 - Edit Signature Address Mapping
 - 5 - Delete Signature Tunings and Custom Signatures
 - 6 - Other 3.x Tokens
 - 7 - Display Signatures
 - 8 - Global Settings
 - x - EXIT
-

Selection> x

Save changes and Exit?

Activate Changes on Sensor?

y - Exit, Save, ACTIVATE CHANGES

s - Exit, Save, Do Not Activate

n - Exit. Do Not Save

Enter - Back to Menu

Selection >

NOTE

If you are using Unix Director version 2.2.3 or later, the nrConfigure utility will be able to configure everything that SigWizMenu configures. After upgrading to 2.2.3, you should use nrConfigure instead of SigWizMenu to tune the signatures.

Understanding Cisco IDS Alarms

It is important to understand the relationship between signatures and alarms. Not all signatures are labeled as a high or low signature. Some signatures are not even enabled and are therefore useless until enabled. Depending on what you want to see, you may end up tuning a signature that once was disabled or considered informational or a low-level event, and tune it to high because you have been seeing strange activity, or have been tasked with researching an event. While Cisco has taken the time and assigned a severity level to all of the alarms, it is up to you to make the final call regarding how the alarms need to be configured. This will change over time, so note that just because you spent the time once to configure the IDS sensor alarms, you are not done. The signature tuning and alarm tuning is an ongoing task. Within the Cisco IDS sensor alarms, there are three levels of severity, *Low*(3), *Medium*(4), and *High*(5). Cisco also provides a *None*(1) and an *Informational*(2) level.

Alarm Level 5 – High Severity

It only makes sense to cover the highest severity level first. They are the most important and you should be more concerned with them than most of the others. Most of the signatures that trigger on unauthorized access, circumvent Access Control Lists, and Denial-of-Service attacks are by default set to a high severity level. Only high-level signatures are mapped to this severity level. Some examples of signatures with high severity levels are

- 3525-IMAP Authenticate Buffer Overflow
- 3250-TCP Hijacking
- 3251-TCP Hijacking Simplex Mode
- 5036-WWW Windows Password File Access Attempt

Alarm Level 4 – Medium Severity

Medium severity level signatures fire based on unusual or abnormal activity on the network. If you have legacy systems on your network, they may generate some false positives or it could be legitimate. The problem with these legacy systems is the fact that they may have gone unpatched for some time. *Low* and *Medium* signatures are mapped to this severity level. Some examples of signatures with medium severity levels are

- 3327-Windows RPC DCOM Overflow
- 4052-Chargen DoS
- 5068-WWW formmail.pl Access
- 5101-WWW CGI Center Auction Weaver Attack

Alarm Level 3 – Low Severity

These are, of course, a low threat to the environment. They pose very little threat. In most cases, the traffic they look at is benign, meaning they are of very little threat by themselves. Cisco provides them as more of an FYI of the different types of traffic that is traversing your network. This severity level is mapped to the *None* and *Informational* signatures. Some examples of these signatures are

- 3602-Cisco IOS Identity
- 5082-WWW WEBactive Logfile Access
- 6053-DNS Request for All Records

Sensor Status Alarms

Sensor status alarms are used to monitor the health of the sensor daemons. Events like *998 - Daemon Down* and *999 - Daemon Unstartable!* appear when sensor services fail or cannot be started or restarted. Communication between the sensor and director is also monitored. *993 - Missed Packet Count* fires when a threshold for dropped packets is met. Signature 993 is very useful in tuning the sensor. Signatures *994 - Have Traffic* and *995 - NO Traffic* detect traffic at the interface. If traffic is detected, signature 994 will fire. If traffic is not detected for a certain period of time signature 995 will fire. The last two, *996 - Route Up* and *997 - Route Down* provide communication information between the sensor and director. The following is a complete list of the status alarms.

- **993-Missed Packet Count** This signature is triggered when the sensor is dropping packets. The percentage dropped can be used to help you tune the traffic level you are sending to the sensor. For example, if the alarms show there is a low count of dropped packets or even zero, the sensor is monitoring the traffic without being overutilized. On the other hand, if 993 alarms show a high count of dropped packets, the sensor may be oversubscribed. Alarm level 1.
- **994-Traffic Flow Started** This signature triggers when traffic to the sensing interface is detected for the first time or resumes after an outage. SubSig 1 fires when initial network activity is detected. SubSig 2 fires when the link (physical) layer becomes active. Alarm level 1.
- **995-Traffic Flow Stopped** Subsignature 1 is triggered when no traffic is detected on the sensing interface. You can tune the timeout for this via the TrafficFlowTimeout parameter. SubSignature 2 is triggered when a physical link is not detected. Alarm level 1.
- **993-Missed Packet Count** This signature is triggered when the sensor is dropping packets and the percentage dropped can be used to help you tune the traffic level you are sending to the sensor. For example, if the alarms show that there is a low count of dropped packets or even zero, the sensor is monitoring the traffic without being overutilized. On the other hand, if 993 alarms show a high count of dropped packets, the sensor may be oversubscribed. Alarm level 1.
- **994-Traffic Flow Started** This signature triggers when traffic to the sensing interface is detected for the first time or resumes after an outage. SubSig 1 fires when initial network activity is detected. SubSig 2 fires when the link (physical) layer becomes active. Alarm level 1.
- **995-Traffic Flow Stopped** Subsignature 1 is triggered when no traffic is detected on the sensing interface. You can tune the timeout for this via the TrafficFlowTimeout parameter. SubSignature 2 is triggered when a physical link is not detected. Alarm level 1.
- **996-Route Up** This signifies that traffic between the sensor and director has started. When the services on the director and/or sensor are started, this alarm will appear in Event Viewer. Alarm level 1.
- **997-Route Down** This signifies that traffic between the sensor and director has stopped. When the services on the director and/or sensor are started, this alarm will appear in Event Viewer. Alarm level 1.

- **998-Daemon Down** This is issued when one or more of the IDS sensor services has stopped. Alarm level 1.
- **999-Daemon Unstartable** Issued when one or more of the IDS sensor services is unable to be started. Alarm level 1.

NOTE

Study these Sensor Status Alarms. They are covered on the test.

Identifying Traffic Oversubscription

Traffic oversubscription is caused by too much traffic being inspected. This can be caused by not tuning signatures to the proper level for traffic on the network. The sensors resource utilization becomes too high to inspect all the packets on the network and begins to drop.. *Signature 993-Missed Packet Count* alarms are used to detect if the sensor is dropping packets or not. The percentage of dropped packets can then be used to tune the traffic level being sent to the sensor. If the percentage rate is very small, it may be normal and the percentage of dropped packets could be within an acceptable level for your network. If the percentage rate is extremely high or higher than you normally expect, the signatures may need to be tuned down to accommodate for the amount of alarms being generated. Some things to help besides tuning signatures is to disable *TCP3WayHandshake* and enabling *TCPReassemblyMode* to loose, discussed earlier in the chapter. This helps to ensure a good level of security.

NOTE

Signature 993 should never show a 100-percent packet loss. This is a good sign that your sensor is having problems.

Summary

Understanding Cisco IDS signatures is understanding what a sensor is comparing traffic against and knowing why a signature triggers an alarm and when it will do it. This understanding is what provides the value of an IDS sensor to the network security arena as well as for your network security. Cisco IDS sensor signatures represent a known type of activity in the wild and the sensor uses this signature, like a fingerprint, to compare traffic for a possible match. If the IDS sensor finds a match to a given signature, the sensor will send an alarm or other means of notification, such as sending an alert to the management console.

The act of simply loading signature updates on to your sensor is not enough to provide good security. You have to take an active role by tuning the signatures for them to be of any value. This tuning takes time and a thorough understanding of your network traffic patterns. We have discussed all of the different components that make up a signature. Content-based and Context-based signatures are the two ways a signature can be implemented. Content-based signatures are triggered by information contained in the payload of the packet. While context-based signatures are triggered by the data in the packet headers.

The structure of the signature depends on the number of packets that have to be inspected. They can be either atomic or composite. Remember, atomic signatures can be detected by inspecting a single packet. A composite signature is detected by inspecting multiple packets. Once the sensor detects a potential signature match, it stores all the information for that stream until it determines a match. State information is required in order to perform this function.

Signature classes, describing the type of attack you are seeing, are another component you need to understand. Reconnaissance, Informational, Access, and Denial of Service are the four main signature classes. Depending on the attack patterns in your environment, you may see some of these, all of these, or none of these.

The different types of signatures are also grouped by traffic patterns. Groups include: General, Connection, String, and Access Control List (ACL).

Configuring signatures does take time and effort. Adding new ones is beneficial only if a similar signature isn't already looking at a particular pattern. *Signature 993-Missed Packet Count* alarms are very useful in determining if you are dropping too many packets because of oversubscribing your sensor. Make sure you remember to tune according to your traffic and that you do not leave yourself open to attack.

Solutions Fast Track

Understanding Cisco IDS Signatures

- ☑ A signature is a pattern or personality of the attack or intrusive activity that has already been discovered.
- ☑ In many ways, the signature is something akin to a fingerprint.
- ☑ You can have a different signature file for each sensor on your network or use one for all of them.
- ☑ The sensor stores all alarms in the sensor logs that are informational and above.
- ☑ The sensor has a database of all the signatures and their specific loaded configurations, and compares the traffic against that database.
- ☑ Content-based signatures are triggered by information contained in the payload of the packet such as a URL string that could possibly compromise a web-server application.
- ☑ Context-based signatures are triggered by the data in the packet headers. This is an enhancement to Packet Signature Detection, which does not consider any context. The most common implementations of Context-Based Signature Detection are to look for attack signatures in particular fields or use a particular offset within a packet stream (based on the protocol).
- ☑ Reconnaissance, Informational, Access, and Denial of Service are the four main categories of signature classes.
- ☑ Reconnaissance is what the attackers do that enable them to map out a network such as DNS queries, ports scans, and even pings.
- ☑ The structure of the signature depends on the number of packets that have to be inspected.
- ☑ Atomic signatures can be detected by inspecting a single packet. No state information is required.
- ☑ A composite signature is detected by inspecting multiple packets. If the sensor detects the first packet that is a potential attack, it stores that

information and the information of the following packets. State information is required in order to perform this function.

Understanding Cisco IDS Signature Series

- ☑ Cisco categorizes the signatures into different traffic types: General, Connection, String, and Access Control List (ACL).
- ☑ General signatures cover the 1000, 2000, 5000, and 6000 signature series. Depending on the type of attack, the General signatures look for abnormalities in a known type of traffic such as making sure a certain protocol is behaving correctly or the payload in the packets is, or looks, correct.
- ☑ Connection signatures are covered in the 3000 and 4000 signature series. They observe traffic to UDP ports and TCP connections.
- ☑ String signatures are highly flexible. They monitor strings (text) within packets that you deem important.
- ☑ Access Control List signatures apply to traffic or activity that is attempting to circumvent Access Control Lists on the routers.
- ☑ The micro-engine types are broken down into the type of activity they detect. They are Atomic, Flood, Service, State, String, and Sweep.
- ☑ When the IDS is sniffing the network, it reads from a signature file that contains all of the signature definitions. Each of the definitions contains configurable parameters that can be tweaked to define activity on your network that you would consider intrusive and possibly malicious.
- ☑ Signature parameters have three attributes to them. They can be Protected, Required, or Hidden. The Protected attribute affects the fundamental behavior of the parameter and applies only to the Cisco set of default signatures. The Required attribute is a parameter value that must be declared. The Hidden attribute denotes that the parameter is not viewable because modifications to the parameter are not allowed.
- ☑ The parameters for the signatures are broken down into two categories, Master or Global engine parameters, and engine-specific parameters.
- ☑ The Master engine parameters apply to each of the signatures in the subengines. Master engine parameters are the basis for parsing the input (traffic) and producing output (alarms).

- ☑ The ATOMIC engine is used to create or tune existing signatures for simple, single-packet conditions that cause alarms to be triggered. Every packet's conditions have specialized parameters that deal with each of the protocol-specific inspections within the scope of the engine.
- ☑ Service engine signatures are one-to-one signatures that interpret the payloads similar to how live services would interpret them. The result of the interpretation is that the decoded fields of the protocol used in comparison against the signatures. These engines only decode enough of the data to make comparisons.
- ☑ FLOOD engines analyze flood type traffic—that is, traffic from many sources to a single host (n to 1), specified in FLOOD.HOST or floods to the network, traffic from many sources to many destinations (n to n), specified in FLOOD.NET.
- ☑ The STAT.HTTP micro-engine is helpful if you are running a web server on nonstandard HTTP ports. Go to Configuration | Sensing Engine | Signature Configuration | STATE.HTTP Service Ports in IDM to add those ports.
- ☑ The STRING micro-engine provides pattern inspection and alarm generation against regular expressions. It works against TCP, UDP, and ICMP. All of the SWEEP signatures alarm conditions depend on the count of the “Unique” parameter.
- ☑ Unique is the threshold parameter that causes the signature to fire the alarm when more than the configured “Unique” number of ports and hosts is seen on the address set within the time period.
- ☑ The OTHER engine does not allow you to define any custom signatures or add any signatures.

Configuring the Sensing Parameters

- ☑ TCP reassembly causes the sensor to reassemble a TCP session's packets before they are compared against the signatures.
- ☑ There are three TCP session reassembly options you can choose from: No Reassembly, Loose Reassembly, and Strict Reassembly.
- ☑ No Reassembly means the sensor does not reassemble TCP sessions. All packets are processed on arrival. This option can generate false positives

and negatives because of the potential for packets being processed out-of-order.

- ☑ Loose Reassembly processes all packets in order. False positive alarms are generated because the sensor allows gaps in the sequence when reassembling the session record.
- ☑ Strict Reassembly does not analyze the session unless all of the packets are received and the session is completely reassembled.
- ☑ IP fragment reassembly is very similar to the TCP session reassembly. IP reassembly causes the sensor to reassemble IP packets before they are compared against the signatures. This helps keep resources from being tied up.
- ☑ IP fragment reassembly has three parameters: Maximum Partial Datagrams, Maximum Fragments Per Datagram, and Fragmented Datagram Timeout.

Excluding or Including Specific Signatures

- ☑ To exclude or include a signature in CSPM, perform the following steps:
 1. Select the signature file you want to edit from the topology map.
 2. Choose the Signatures tab and select the appropriate subtab: General Signatures, Connection Signatures, String Signatures, or ACL Signatures.
 3. To disable the signature, uncheck the boxes, or, if you want to enable a signature, put a check in the box to enable it.
 4. Click **OK**, then save and update your configuration.
 5. From the Command tab, click **Approve Now** to push the new configuration to your sensor.
- ☑ To exclude or include a signature in IDM use these steps:
 1. Go to Configuration | Signature Groups. Choose the group name that your signature is associated with. Drill down until you get to the signature you want to configure. Select the signature you want to enable or disable.

2. Check the box of the signature to enable and uncheck the boxes of the signatures you want to disable or have excluded.
3. Click the Apply Changes button for changes to take affect.

Creating a Custom Signature

- ☑ To create a custom signature in IDM for the sensor you want to create a custom signature for, follow these steps:
 1. From the main screen, go to Configuration | Custom Signatures. Select the engine that your custom signature will apply to.
 2. At the bottom of the screen, click Add. On the Adding screen, start filling in the information and setting the parameters on the page that will be the signature.
 3. Click OK. The result is having your signature added to the sensor configuration and listed in the Custom Signatures section of the micro-engine.
 4. Click Apply Changes in the upper right-hand corner of the IDM screen.
- ☑ CSPM can only set a signature's actions and severities. CSPM cannot tune signatures for the IDS sensor appliance. CSPM can set the severity and the action to associate to the signature but cannot set what triggers that signature.

Working with SigWizMenu

- ☑ SigWizMenu is the signature wizard that allows you to make changes to IDS signatures directly on the Sensor.
- ☑ The Signature Wizard is an interim tool for version 2.2.2 Unix Director users until they upgrade to version 2.2.3, and Cisco Secure PM users until these options are included in Cisco Secure PM.
- ☑ To start SigWizMenu, follow these steps:
 1. From the console or Telnet session, log in as **netrangr** to the sensor you want to start SigWizMenu on.

2. From the `/usr/nr/bin` directory, type **.SigWizMenu** at the command prompt. Don't forget to put the period in front and remember that Unix environments are case-sensitive.
- ☑ The files `/usr/nr/etc/SigData.conf`, `/usr/nr/etc/SigUser.conf`, and `/usr/nr/etc/SigSettings.conf` are what the Signature Wizard uses to operate and maintain a current configuration of all the signatures.
 - ☑ `nrConfigure` in Unix Director version 2.2.3 or later can do everything that `SigWizMenu` configures.

Understanding Cisco IDS Alarms

- ☑ Cisco assigns a severity level to all of the alarms. This is completely customizable.
- ☑ Within the Cisco IDS sensor alarms, there are three levels of severity: Low(3), Medium(4), and High(5). Cisco also provides a None(1) and an Informational(2) level.
- ☑ Only High level signatures are mapped to alarm level 5.
- ☑ Low and Medium signatures are mapped to alarm level 4.
- ☑ None and Informational severity level signatures are mapped to alarm level 3.
- ☑ Sensor status alarms are used to monitor the health of the sensor daemons.
- ☑ Traffic oversubscription is caused by too much traffic being inspected. This can be caused by not tuning signatures to the proper level for traffic on the network.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: What is the difference between content-based signatures and context-based signatures?

A: Content-based signatures are triggered by information contained in the payload of the packet, such as a URL string. Context-based signatures are triggered by the data in the packet headers.

Q: What are the four categories of signature classes?

A: Reconnaissance, Informational, Access, and Denial of Service are the four main categories. Reconnaissance class signatures identify traffic that is representative of your network and systems being mapped. Informational signatures are triggered from activity attempting to connect or communicate with the host(s). Access signatures fire alarms when known unauthorized access or attempts to access them are detected. Denial of Service or DoS class signatures trigger when the level of activity on the network is detected as having the capability to disrupt services.

Q: What is the difference between atomic signatures and composite signatures?

A: Atomic signatures can be detected by inspecting a single packet. No state information is required. A composite signature is detected by inspecting multiple packets. State information is required for composite signatures.

Q: Signatures are also categorized into traffic types. What are they?

A: General Connection, String, and Access Control List (ACL).

Q: What is meant by virtual sensor?

A: The concept of a virtual sensor is that if the physical sensor is monitoring more than one interface, all the interfaces are configured into interface

groups. There can be more than one interface group. But virtual sensors are attached to only one interface group.

Q: Explain the different types of micro-engines.

A: ATOMIC micro-engines are used for single packets. Flood micro-engines are used to detect attempts to cause DoS attacks. Service micro-engines are used when services at layers 5, 6, and 7 require protocol analysis. State micro-engines are used when stateful inspection is required. String micro-engines are used for string pattern matching. Sweep micro-engines are used to detect network reconnaissance sweeps or probes.

Q: What are the three different configuration settings for TCP Session Reassembly?

A: No Reassembly, Loose Reassembly, and Strict Reassembly. No Reassembly does not reassemble TCP sessions. All packets are processed on arrival. It is not recommended unless your network is subject to a higher-than-normal rate of packet loss. Loose Reassembly does process all packets in order. In Strict Reassembly, unless all of the packets are received and the session is completely reassembled, the sensor will not analyze the session.

Q: What command do you use to tune signatures in CSPM?

A: *.SigWizMenu* is used to tune signatures when using CSPM. There is not a method for tuning in the CSPM console itself.

Q: What are the different severity alarms for signatures?

A: High, Medium, Low, but also None and Informational.

Q: What are Sensor Status Alarms?

A: Sensor Status Alarms are used to verify the health and status of the sensor daemons, interfaces, and the communication between the sensor and director.

Configuring Cisco IDS Blocking

Solutions in this Chapter:

- Understanding the Blocking Process
 - Understanding Master Blocking
 - Using ACLs to Perform Blocking
 - Configuring the Sensor to Block
 - Determining the Status of the Managed Device and Blocked Addresses
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Blocking... This is a word that just sounds like security, doesn't it? We will block you from our network. In the world of Cisco, blocking is another name for "shunning," which is the art of actively interacting with a device such as a router and reconfiguring the Cisco device to stop or "block" the attack. The managed device could be a router or a firewall. The IDS sensor uses the control port to establish the connection with the device and applies an ACL to the managed interface. We can have the Cisco IDS sensor talk with the Cisco PIX firewall and dynamically change the configuration to shun an attack. The Cisco IDS sensor can also manage other Cisco IOS devices such as the following:

- 1600
- 2500
- 2600
- 3600
- 4500
- 4700
- 7200
- 7500
- PIX firewalls such as the 501, 506E, 515E, 525, and 535

IP blocking eliminates the need for the engineer to log in to the device and make the blocking changes manually. However, you need to be careful with blocking so as not to inadvertently block someone or something that is not attacking your network, such as a particular server or an extranet connection.

NOTE

The PIX firewall uses the *shun* command to block. Unlike the routers, the PIX ACLs are not modified.

Other devices that can be managed are the Cisco Catalyst 6000 series switches with CatOS, 6000 switches with MSFC (Multilayer Switching Feature Card) and the Catalyst 5000 switch with an RSM (Route Switch Module). In order for the

blocking to work, the IDS sensor must be able to communicate with the Cisco device and must have VTY (Telnet) access enabled, a line password, and the privileges to make configuration changes. The Cisco PIX can use either VTY (Telnet) or SSH. A subtle but critical item to remember is that the IDS sensor either needs to be on the same subnet or routed to the subnet of the managed device. You might laugh at this basic concept, but people forget it all the time.

NOTE

SSH is optional, but if SSH is configured, then the IDS sensor and the PIX must exchange keys manually.

Cisco blocking is a very powerful and dangerous feature that should only be used after detailed planning. For example, there are always critical resources such as certain hosts that should never be blocked. These need to be identified and prevented from ever being blocked. You need to make sure you have some type of antispoofing in place to ensure a spoofed address will not enable blocking on a legitimate address by mistake. By default, the blocking process will last 30 minutes. Is this too long for your network? This needs to be thought about before a block takes place and you are scrambling around trying to fix it in real time. We will cover all this information in more detail throughout this chapter.

The importance of network entry points, or ingress points, will be discussed as well. If the attacker is blocked off on one entry point to our network, can he find another way in? We will investigate Master blocking, a feature that can help us manage this type of situation. Let's now delve into the basics of IP blocking and its processes.

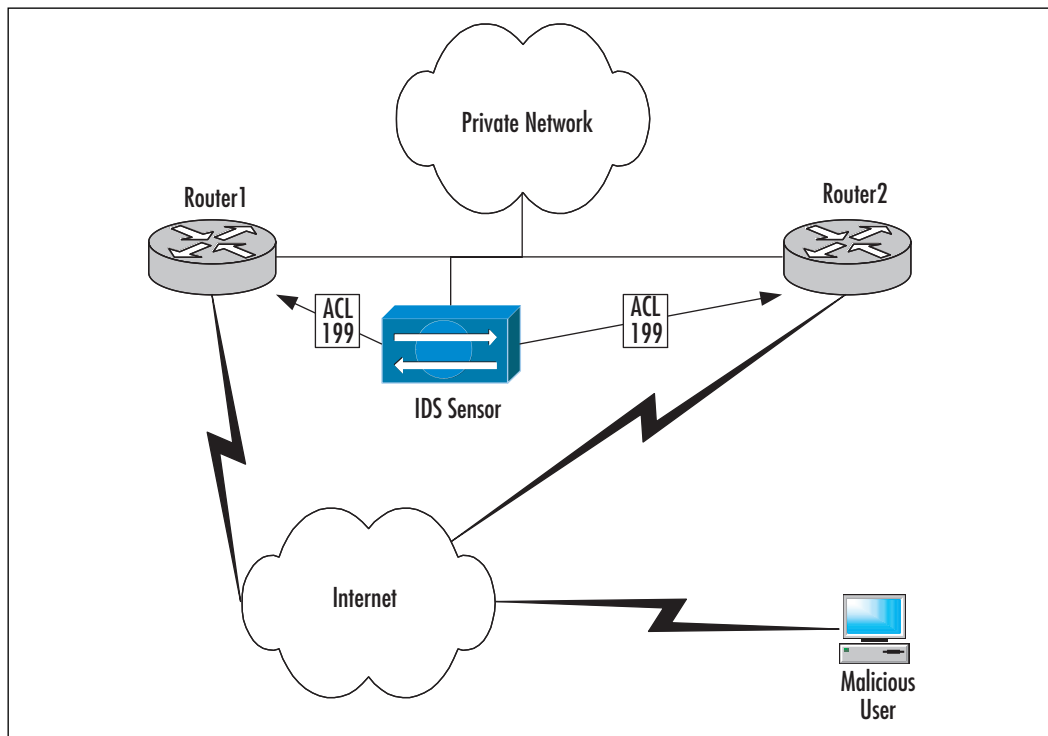
Understanding the Blocking Process

Threats to our networks never sleep or wait until Monday to become a burden. While there are many mitigating tools available to the security administrator such as firewalls, password security, encryption standards, and even complex networks with private IP addresses, malicious traffic still seems to find its way into the network. Hence, we have the need for network intrusion detection systems, or NIDSs, to find these intruders and make the administrator's aware of the threats to their network and data.

However, Security Administrators and their teams may not always be available and nobody likes to be awoken in the middle of sleep, or called while vacationing in Bermuda, to find out their network has just been a victim of a Denial-of-Service attack, web defacement, or worse. This is where IP blocking can come in handy. Instead of just watching, logging, and sending alerts as malicious activity happens on the network, blocking can be put in place to dynamically stop intruders and allow ample time for someone to take the needed corrective actions.

If blocking is configured correctly, it can also stop this traffic from using any other possible entrances to the same network, say from a redundant connection with a different ISP, and keep the unwanted traffic out however long the Security Administrator has configured it to do so. We can see in Figure 8.1 that the IDS sensor can send the Access Control List “199” to both Internet bordering routers, thus effectively keeping the attacker from entering the network at either point. Now the Security Administrator can rest easier and handle the problem when he is ready to handle the problem, not work according to the attacker’s schedule.

Figure 8.1 Blocking Multiple Network Entry Points



What Is Blocking?

So what actually is IP blocking? IP blocking is very much as it sounds, when a Cisco Secure IDS sensor detects malicious behavior, IP blocking is implemented to deny further network traffic entering our network from the identified source host IP address. What's more, we can configure what specific traffic patterns, or *signatures*, we want the intrusion detection system (IDS) to watch for and manage according to an administrator-assigned severity level.

IP blocking makes use of Access Control Lists (ACLs) to stop the suspect traffic from entering the network and will block the network traffic until the block is either manually removed or it exceeds its predetermined *blocking duration*.

Designing & Planning...

IP Blocking or Shunning?

IP blocking was known as “shunning” in previous versions of Cisco Secure IDS and its predecessor, *NetRanger*. Be aware that this term may occasionally come up from some of the security veterans out in the field or may be seen in older configuration files. Later in this chapter, we will see an example of this term being used in a couple of pop-up windows.

Access Control Lists

At this point, it is important to have a thorough understanding of Cisco's Access Control Lists (ACLs), sometimes simply referred to as access-lists. There are two main types of ACLs we will be concerned with in this chapter. These are the Standard Access Control List and the Extended Access Control List. A number that is assigned to them can quickly identify these two types of ACLs. Two steps must be followed when creating an access-list. The first step is to create the list itself and the second is to apply the list to an interface.

Step 1: Standard Access-Lists

These ACLs are used in situations when we only want to block a source IP host address or an entire network IP address from accessing a certain network. This is

specifically working with the IP addresses and not giving any mind to other information, such as port number or protocol. The number assigned to a standard access-list will be in the range of 1–99, and an expanded range of 1300–1999. Here is the basic format of a standard access-list:

```
access-list access-list-number [permit|deny] [source-ip-address] [source-wildcard]
```

Here is an example:

```
access-list 19 permit 172.16.2.0 0.0.0.255
```

The access list components are as follows:

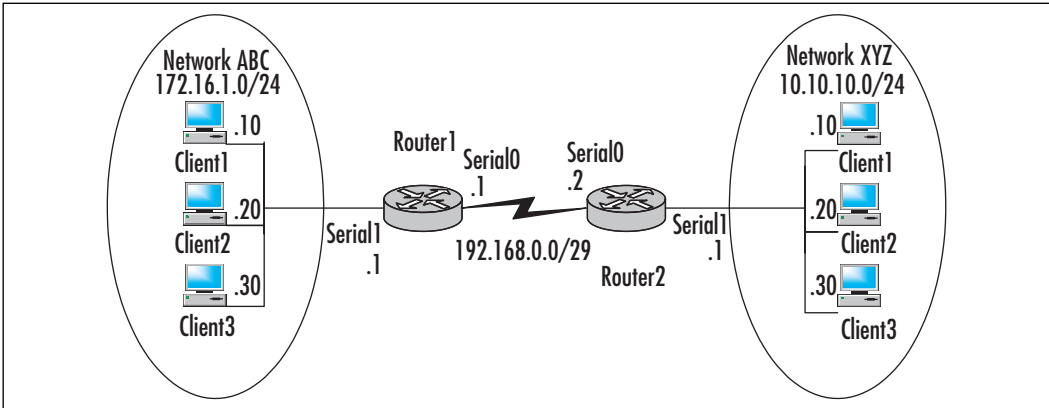
- **access-list** The command used to initiate the creation of an access-list
- **access-list-number** A number within the range previously specified for the type of access-list being created.
- **[permit|deny]** This tells the network device to either “permit” the traffic to pass the interface or to “deny” the traffic from passing and drop the traffic.
- **source-ip-address** Indicates the particular host IP or network IP address to be checked.
- **source-wildcard** Used to indicate an actual host, subnet, or network. This is broken down into comparing subnet masks and the ip address. A “0” indicates the number in its respective octet matches perfectly. Any “1s” indicate any number can be used. For instance, in the example, for any ip to be matched by the access-list, 172.16.2 must be the first part of the IP address. The last octet can be anything from 0–255. The example “permits” any traffic from the IP range of 172.16.2.1–172.16.2.254 to pass the network device’s interface.

NOTE

More on wildcard masks and their usage can be found in various articles at www.cisco.com.

Let's look at an example to better understand how this configuration works on a network referring to Figure 8.2.

Figure 8.2 Implementing a Standard Access-List



Let's say Client1 on the Network XYZ needs access to resources located on the Network ABC. By using an access-list we can provide this admission and still protect the network from others trying to enter. Here is the configuration shown as an excerpt from a *show run* command:

```
Router1
!
interface Serial0
ip address 192.168.0.1 255.255.255.252
ip access-group 10 in
!
access-list 10 permit 10.0.0.10
!
```

Notice the access-list has been applied to the Serial0 interface on Router1 with the *ip access-group 10 in* statement. This configuration will permit only incoming traffic from the address 10.0.0.10, and allow it access to the entire ABC network.

If it were desired to block Client1 on XYZ and allow the rest of the network, we would use the configuration:

```
Router1
!
```



```

interface Serial0
ip address 192.168.0.1 255.255.255.252
ip access-group 10 in
!
access-list 10 deny 10.0.0.10
access-list 10 permit any
!

```

This configuration will allow all users to access every area of the network except Client1. Keep in mind that access-lists are read from top to bottom. When a match is found, that match will be used and the router stops comparing the packets to the access-list. Therefore, once the router reads the “permit any” statement, it will stop reading and simply allow “any” address through. Also, the most heavily matched items should be placed at the beginning of the access-list to help ease the CPU load.

NOTE

Not shown in the configuration is an implicit “deny all” statement. This does not need to be manually edited because it is automatically applied. This is a wonderful security feature, but if you are not aware of its existence, it can lock out all critical systems attempting to access resources on the network. As in the preceding configuration, only the host 10.0.0.1 will be allowed to traverse Router1. All other traffic attempting to enter this interface will be dropped.

Step 2: Extended Access-Lists

These ACLs give us much more depth in how to control network traffic. Extended access-lists can be configured to check port number, protocol, and the destination address as well as the source address. The number assigned to an extended access-list is in the range of 100–199, and an expanded range of 2000–2699. Here is the basic format of an extended access-list:

```

access-list access-list-number [permit|deny] protocol source ip address
source-wildcard destination destination-wildcard [operator]

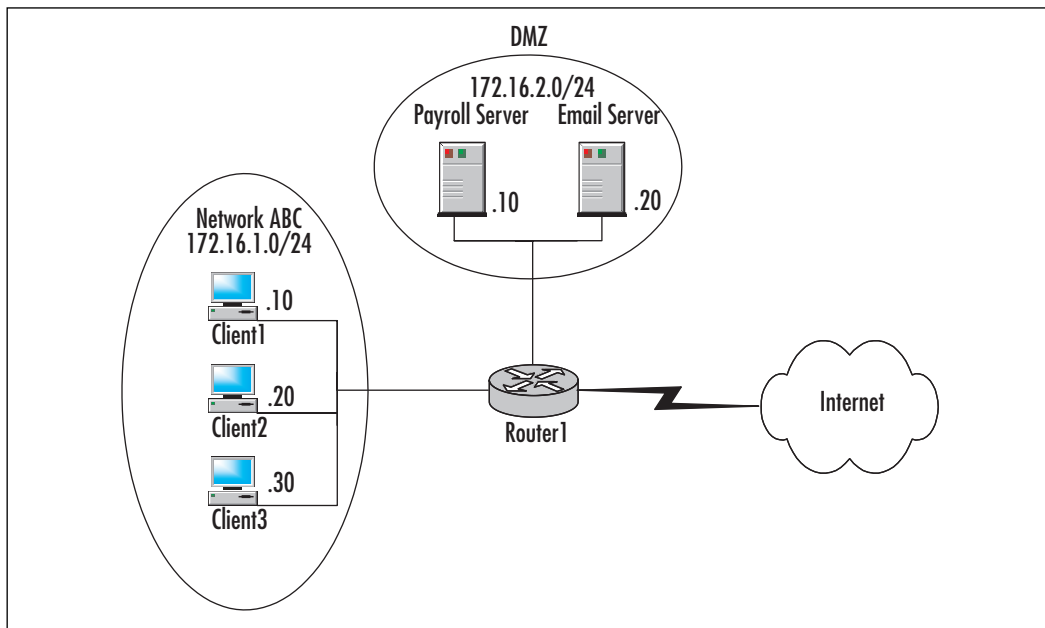
```

Here is an example:

```
access-list 190 deny TCP 172.16.1.0 0.0.0.255 172.16.2.0 0.0.0.255 neq 23
```

The extended access-list is different than the standard ACL in the following ways:

- **access-list-number** This is a number within the range previously specified for the type of access-list being created. In this case, it is an extended access-list, as indicated by the 190.
 - **protocol** This allows us to filter based on a particular protocol—typically, IP for filtering by IP address or TCP for filtering by protocol.
 - **destination** This is the destination IP address a particular packet is trying to reach. In the example, it is the network address of 172.16.2.0.
 - **destination-wildcard** This works as the source-wildcard to indicate the host IP address or range of IP addresses trying to be reached. This saves the time of having to type lines for each IP address within a particular subnet.
 - **Operator** This can be used to indicate a port number when filtering by protocol, such as TCP or UDP. There are four options that can be used with this feature.
 - **eq** Equals—when we know exactly what port needs to be monitored
 - **gt** Greater than—allows us to specify a particular range over a particular port number
 - **lt** Less than—allows us to specify a particular range lower than a particular port number
 - **neq** Not equal—allows us to assert the access-list to all but on port
- Let's now see how this works on a network. Refer to Figure 8.3.

Figure 8.3 Implementing an Extended Access-List

Looking at this network example, imagine that all the clients need access to the e-mail server and only Client3 needs Telnet access to the payroll server. Our configuration would look similar to this:

```
Router1
!
interface Ethernet1
ip address 172.16.2.1 255.255.255.0
ip access-group 110 out
!
access-list 110 permit tcp 172.16.1.30 any eq telnet
access-list 110 deny tcp any 172.16.2.10
access-list 110 permit tcp 172.16.1.0 0.0.0.255 any
!
```

This access-list configuration first allows Client3 to access to the Demilitarized Zone (DMZ) and specifically access the payroll server using Telnet (indicated by eq). The next line in the access-list is to deny anyone to access the payroll server. Notice this comes after the line that allows Client3 access to the same server. Remember, when the router finds the first match, it will stop

reading and take action. The third line allows anyone from the internal network 172.16.1.0 to access any resource and any protocol, and so on, not already denied, on the DMZ. If a match is still not found, the implicit deny will drop all remaining network traffic trying to access the DMZ.

One condition we used in the preceding configuration had to do with whether we applied the access-list to an internal or external interface and the particular direction of the traffic flow. In the last example, we can see the ACL was applied to an external interface, in relation to the internal network. We also have the command “out” used. This means that traffic trying to leave the router on the Ethernet1 interface will be checked. The first example, Figure 8.1, used an ACL on the Serial0 interface, applying the *in* command. This prevents the network traffic specified in the ACL from entering the router. The traffic is then dropped right at the front door.

NOTE

Only one access-list can be applied to an interface/direction at a time. If another ACL is applied to the same interface/direction, the original ACL will be nullified.

IP blocking relies on a system that will compare inbound and/or outbound IP datagrams on a particular interface to a list of signatures and trigger an alarm if a match is made. The alarm will indicate to the governing sensor that there is a threat. At this point, the sensor will create a new ACL and distribute that ACL to the network device providing access to the network in jeopardy. When the new ACL reaches the network device, it will replace the resident ACL with the newly arrived one, thus dynamically updating the network device. This process is referred to as *Device Management*.

Device Management

The process of device management includes the construction of a new ACL and applying that ACL to the appropriate network device interface thus stopping, or blocking, the network breach immediately. As mentioned earlier, the block will remain in effect until either it has been removed manually or the blocking duration has expired. Device management relies on a Telnet connection, via its com-

mand and control interface, to the network device in question. For the sensor to successfully Telnetted to the network device, it requires the following:

1. There must be a route to the network device within the device management list on the sensor or the sensor must reside within the same subnet as the network device.
2. The network device must be configured to accept Telnet connections by configuring the vty lines with a login password and, if used, a Telnet username, and allow secure configuration changes by setting an enable password.

It is important to keep in mind that the IDS sensor will be contacting the router with an enable-privileged password. This could foster an insecure environment if this password were to get out. When possible, it is recommended to keep the Telnet sessions isolated to a “Telnet only” network. Another option is to apply ACLs to lock down all Telnet access to the network device except from approved systems and/or users.

Device management uses extended ACLs to provide its blocking functionality. The ACLs are 198 and 199. The first ACL to be used by Device Management is 199. When a change is demanded via an alarm from the sensor, Device Management will create a new ACL—that’s right, 198—and send it to the rescue.

Understanding Master Blocking

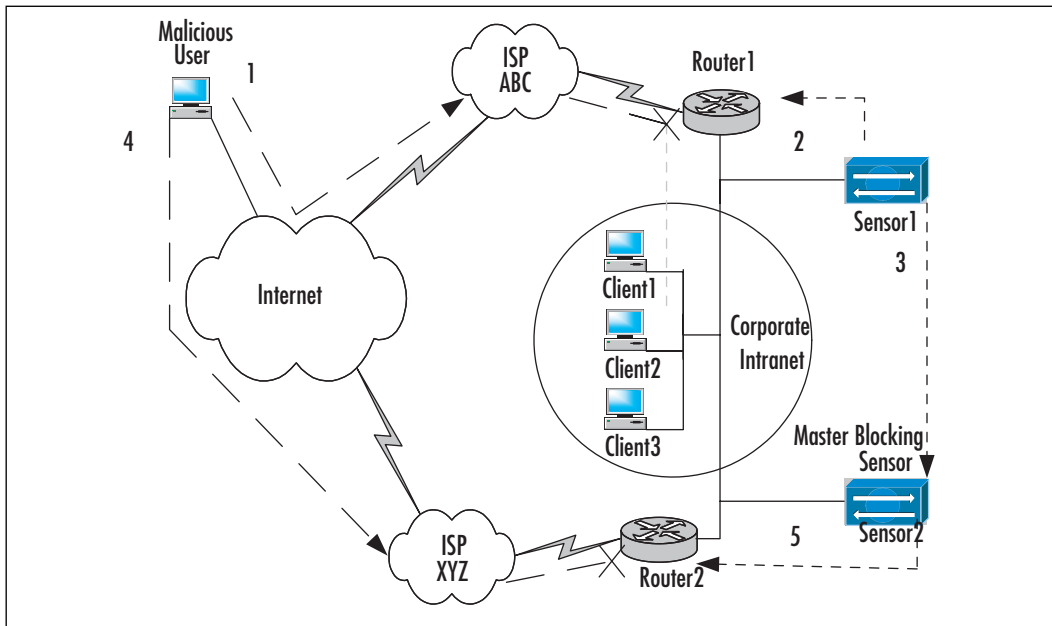
In some network architectures, for reasons such as redundancy or perhaps cost, another ISP may be a feasible solution. An Extranet connection or two may also be present. These connections create multiple entryways to our network and thus generate more risk areas that will need to be monitored.

This is where a feature called master blocking comes in. Master blocking allows one sensor to perform the blocking for another. In a nutshell, one sensor learns of a triggered alarm and updates the triggering router with a new ACL. After the ACL has been updated, the sensor will communicate with any other sensors on the network that are configured for *master blocking*. The communication will take the form of a Telnet session request. At this point, the initializing sensor becomes the *blocking forwarding* sensor.

The contacted *master blocking* sensor(s) will accept the Telnet connection and update any of their respective network devices with the same ACL to keep the intruding data from entering the network via another path.

In Figure 8.4, we see how this process works.

Figure 8.4 A Master Blocking Sensor



Let's follow the steps taken when a malicious user attempts to access resources on a private network.

1. The malicious user connects through the Internet to ISP ABC. From this point, he has somehow (perhaps by brute force attack) accessed the internal network.
2. The Cisco Secure IDS Sensor1 has noticed the strange traffic on the network and just so happens to match one of the signatures it has been configured to monitor. This could possibly be a brute force attack on an internal system.
3. Sensor1 creates and sends a new ACL to the perimeter router, Router1. This action stops the attack in its place.
4. Now, with master blocking configured, Sensor1 requests all sensors listed within its Master Blocking Sensors panel, in this case Sensor2, to block for this same attack. Meanwhile, the attacker now tries to reroute his traffic to any other available interface to the network. If the attacker is prepared, the entry point via ISP XYZ will already be known.

Therefore, the attack is attempted to continue through this other interface.

5. Sensor2 sends the ACL it received from Sensor1 to Router2 and blocks the traffic at this entry point as well.

In a nutshell, Sensor2 was completely unaware of the attack on Router1 until Sensor1 contacted it. This saves our sensor's resources from having to detect the same traffic over and over again and, most importantly, stops the traffic from entering again.

Using ACLs to Perform Blocking

As previously discussed, Cisco's ACLs are a list of rules that will either permit or deny traffic entering or leaving the network. We can use either standard ACLs, for controlling network access for a particular source IP address, or extended ACLs for a more specific control such as source and destination IP address, protocol, and so on. There are other types of ACLs used by Cisco with varying functions, such as Lock and Key (Dynamic ACLs) which force an authentication before a router allows traffic to pass, and IP-named ACLs which allow a user to name an access-list as opposed to using a number. These other ACL types are beyond the scope of our subject but more information can be found at www.cisco.com. Give an actual link, not just the generic Cisco site.

Device management takes advantage of ACLs by creating and applying ACLs to network devices being monitored by a particular sensor. We have already discussed how blocking works, now let's take a look at what actually happens with the device management process.

ACL 198 and 199 are the two extended IP access-list numbers that device management utilizes. These are the only two and a sensor-created ACL will take priority over any previously established ACL by the same number. A sensor begins with the ACL 199. When a security threat triggers an alarm, the monitoring sensor will create a new ACL number—you guessed it, 198. This new ACL is equipped with the appropriate settings to block out the newly realized threat. The sensor will then Telnet to the network device, authenticate, and then apply the ACL 198 to the interface in question. Since we can only have one ACL per interface and direction combination, ACL 198 will replace the current ACL 199. The next alarm that is triggered on this interface will receive the ACL 199 again but with new configurations. These two ACLs are switched back and forth as needed and the method used to apply them ensures consistent protection.

NOTE

To use an ACL as a preshun or a postshun ACL, it *must* be an extended IP ACL and can be either named or numbered. Preshun and postshun ACLs can be used to indicate a standard group of settings to be included with all ACLs being implemented. Therefore, when the IDS sensor deems an ACL creation necessary, the preshun and postshun settings will be included with the deployed list. Preshun indicates the standard information will be included before the IDS created portion of the ACL. Postshun would be in the ACL after the IDS created portion of the ACL.

This feature could be very important in helping to keep particular, or critical, connections open all the time or blocked all the time, whatever the case may be.

General Considerations for Implementation

Before implementing any new type of technology to our networks, it is, of course, critical to have a plan that has been thought out and preferably tested. A few things that should be considered before we move forward with our implementation have to do with a sensor accidentally creating a Denial of Service (DOS) on our network. The following are topics that can have a positive and negative effect on our networks and must be considered carefully before implementation.

1. **Knowing your network's entry points** As mentioned earlier in the chapter about master blocking, there are times when we will have multiple entries to our networks. Be sure to locate each one of these entries and document their purpose and what type of services and traffic traverse each one. This is helpful in establishing our sensors in the correct places and utilizing master blocking to lock-down our networks. Hackers love a good challenge and finding the one spot that was overlooked during this lock-down is an excellent one.

NOTE

Nowadays, use of the terms *hacker* and *cracker* reflect a strong revulsion against the theft and vandalism perpetrated by cracking rings in the everyday understanding (or misunderstanding) of the terms. While it is expected that any real hacker will have done some playful cracking and knows many of the basic techniques, any true hacker past larval stage is

expected to have outgrown the desire to do so except for immediate, benign, practical reasons (for example, if it's necessary to get around some security in order to get some work done).

Thus, there is far less overlap between hackers and crackers than the mundane reader misled by sensationalistic journalism might expect. Crackers tend to gather in small, tightly knit, very secretive groups that have little overlap with the huge, open polyculture of hackers. Though crackers often like to describe *themselves* as hackers, most true hackers consider them a separate and lower form of life.

2. **Configuring the blocking duration** When an alarm is triggered, the ACL is applied to the appropriate interface and the traffic violation is halted. The default duration for a Cisco Secure IDS to block the suspect traffic is 30 minutes. During this time, the network security team can research the problem and create a fix to prevent the issue from reoccurring or determine the necessary steps to pursue legal action. The blocking duration may be extended for times when network security staffing is unable to respond immediately, such as weekends or holidays. If the block has been set manually, the default blocking duration will be 1440 minutes, or 24 hours.
3. **Clever anti-spoofing attacks** Anti-Spoofing is accomplished by only allowing valid internal IP addresses to exit the network and disallow any incoming traffic with a source IP address that is used on the internal network. If a crafty cracker has a go at our network, the hacker may opt to create suspect traffic while using a valid internal network IP address, thus causing the monitoring sensor to create and apply an ACL that restricts a valid IP address or block of addresses. If this happens, we can see how legitimate internal clients may be cut off from needed network resources.

NOTE

Spoofing is when an external attacker infiltrates an internal network by using an IP address that is common to that internal network. The idea behind it is that the firewall will automatically assume the traffic is internal and allow it to pass to areas restricted to intranet clients.

Another popular choice for configuring firewalls is to disallow any external source IP addresses to establish a connection with any system within the internal network. If a valid remote connection is needed, a connection should be made directly to a firewall, remote access server, and so on located within the Demilitarized Zone (DMZ). From that point, the DMZ system can establish a secure (most likely authenticated) connection with the internal resources requested.

4. **Choosing signatures** Carefully performing this step will allow the monitoring sensor to perform its duties efficiently. If we chose to include every signature available to us while monitoring, a massive strain will be realized on the sensor and the potential for errors and missed packets increases. Thus, we will want to choose signatures that relate to the type of traffic and services being passed through our network. It is very easy for someone to run a utility that scans for these services and types of traffic and either explains how to take advantage of them or provides a list that the attacker can use to find their failings on one of hundreds of web sites out there.

Configuring & Implementing...

Finding Vulnerabilities to Monitor

When we choose which signatures to configure, we should consider the way most attacks are constructed. Of course, we should first clarify that network attacks are becoming more complex and intricate everyday, yet most of them seem to follow a particular format. This process is important for any Security Administrator to understand and it will help us to decide which signatures to select.

The ease of performing a vulnerability assessment to determine weaknesses is rather frightening. Even more frightening are the so-called "Script Kiddies," who have no prior knowledge of the scanning tools they are using against your network and can produce volatile results.

The basis of most attacks is for the attacker to launch a scan of a network using particular tools to find out as much about a network as

Continued

possible. This is called the reconnaissance phase of an attack, and can include DNS lookups and zone transfers, which could potentially provide an enormous amount of information about a company's web servers, such as server names, IP addresses, and operating systems. Once this is completed, a list of targets is now ready. Using any IP addresses the attacker found, a *ping sweep* could be launched to discover if any other vulnerable systems have been missed. While the ping sweep is in action, the attacker can then use the newfound information to scour the Internet and find potential vulnerabilities to try.

This is when it becomes crucial to know what services we are running. The attacker can connect to one of many web sites to find what vulnerabilities could be tried on the particular operating system. For instance, imagine an attacker found out we were running Microsoft IIS 5.0 on one of our web servers. The attacker could go to a hacker web site, run a search for "Microsoft IIS 5.0" and come up with a list of vulnerabilities to try and probably get the list in order of severity. These listed exploits, more than likely, will provide step-by-step instructions on how to perform the attack, (usually followed with the step-by-step instructions on how to fix them). Perhaps the vulnerability of a "built-in" administrator account or FTP domain authentication is discovered, we can only guess what would happen from that point on.

It should now be easy to see the importance of knowing what services we are running when picking our signatures. With more and more vulnerabilities being discovered daily, we can also see how important it would be to keep our signatures up-to-date with Cisco-provided updates.

If properly prepared, it may be reasonable to run a vulnerability scan on our own networks. However, vulnerability scanning can put a major strain on our networks and perhaps create the unwanted Denial-of-Service conditions we are trying to avoid. The best option for this is to know exactly what the pros and cons are of particular scanning tools and note which tool would be the best choice for our situation. Owners of critical systems should be made aware of, and be present during, a vulnerability scan in the event something should go awry.

5. **Defining your critical hosts** This is a step that will help in the preceding step as well, Choosing Signatures. It's important to find critical systems that may need to perform their functions free of blocking.

Examples include: DNS Servers, Windows Domain Controllers, and WINS and DHCP servers. The Secure IDS Sensors and Director will also need to be free from the potential of blocking. These systems may be detrimental to a company's productivity if blocked. Thus, you should have local security controls in place so Cisco Secure IDS can perform tasks such as logging and trigger alarms without the use of IP blocking.

Where Should I Put My Access Control Lists?

As we know from earlier in the chapter, we can only have one ACL applied to an interface in a specified direction—for instance, Serial 0 out. If a specific interface and direction has been previously configured with an ACL manually, when a sensor is appointed to monitor that interface, it will replace that ACL with the sensor-generated one, thus replacing the manually configured ACL. If, for some reason, the manually configured ACL is using the same ACL number the sensor does, 198 or 199, the ACL will simply be replaced with the new sensor-generated one. The sensor's ACL will take priority over whatever ACL is currently in place and will not merge any of the two rules sets together.

The ACL may be configured for either the internal interface of a router, facing the Internet, or the internal interface, facing the internal network. There are both good and bad results from either method chosen. Before we decide on an interface, we will want to consider the direction the traffic will be heading. For instance, if traffic from an internal system enters a router on Serial 0 and exits to the Internet on Serial 1, the traffic direction will be “in” for Serial 0 (into the router), and “out” for Serial 1 (out of the router).

When applying an ACL to an *external interface*, we are essentially keeping any external traffic from even entering the router. This traffic could include different types of network-scanning tools, Internet broadcast advertisements and so forth. This is a highly protective state and if all the services and network traffic needed by the internal network are well defined, it can be a good choice. With the external “inbound” ACL in place, denied data packets not processed by the router will allow the router to perform its duties free of wasteful excess processing power. The external packets will be dropped at the door.

When configuring an ACL to an internal interface “outbound,” the data packet enters the router and is processed and sent to the correct interface to traverse the router. When the packet is switched to the correct interface, the ACL will be checked to see if the data packet will be allowed to proceed into the Internal network. As you can imagine it is less than desirable to have our routers

processing packets only to have them dropped afterwards at the interface. The packets are also reaching the network device, which is not quite network-security friendly.

It seems obvious that performing blocking on the external interface “in” would be the best solution. However, it may be suitable to perform IP blocking on the internal “out” interface. This may be because there already exists an external “in” ACL and the only option for the sake of stability is to configure the internal “in” interface. If a DMZ is in place, we may need to allow some traffic to enter the router and proceed to that DMZ and at the same time deny that very traffic from entering our local network. This situation could include a general ACL on the external “in,” with regards to ICMP echo requests and perhaps some unneeded ports, and two internal “out” interfaces (one for each internal interface). Now its time to turn our attention to how we can configure our sensors to react to triggered alarms and how our routers can accept requested modifications.

Configuring the Sensor to Block

In this section, let’s delve into how to actually configure IP blocking step by step. As we mentioned earlier in the chapter, there are many different possibilities for network set-ups. Thus, different options may work poorly for one configuration, and well for another configuration. Since we have already decided on which signatures we will want to incorporate in our configuration, and we have specified our blocking devices and reviewed the option of utilizing master blocking on our network, our next steps will be configuring our sensors and routers. This will include configuring the *blocking device*, or router, for Telnet communications as well as preparing the sensor for which interface will need to be monitored.

Configuring a Router for a Sensor Telnet Session

First, we will configure the router for Telnet access and assign a login password. The login password is essential for allowing us to Telnet to a router and should be something complex and easy for us to remember. Password security is very important and we will need to use this password when configuring our sensors.

```
Router>enable
Router#configure terminal <enter>
Router#line vty 0 4 <enter>
Router<config-line>#login password syngress <enter>
```

```
Router<config-line>#exit <enter>
!
```

Now we will set an enable password for the security of remote configuration changes:

```
Router<config>#enable password Syngress <enter>
Router<config>#^Z      #This is actually ctrl+z
Router#write memory <enter>
Building Configuration...
[OK]
Router#
```

At this point, we can exit out of the router or type **show running-config** and view our configuration. Our interest in a show run would be an enable password at the start of the configuration and a vty login at the bottom. It should look somewhat similar to this:

```
Router# show running-config
Building configuration...

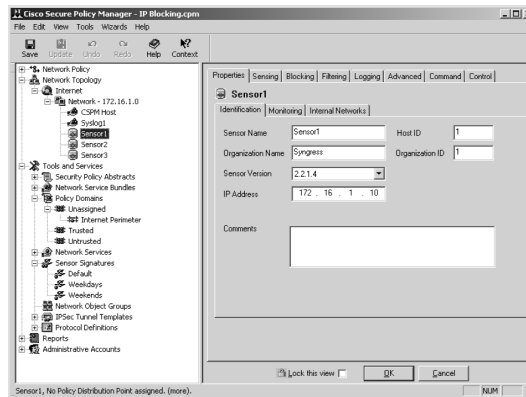
Current configuration : 2350 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Router
!
enable password 7 00071A150754
#
# Router specific data...
#
line vty 0 4
  password syngress
  login
!
end
```

Configuring the Sensor

Now we need to set up the sensor for the blocking devices it will monitor by using the Cisco Secure Policy Manager (CSPM). These settings indicate to the sensor which routers, by Telnet IP address, will be governed and updated as well as indicate the correct settings for dynamic Telnet sessions, including login password and possible usernames to use.

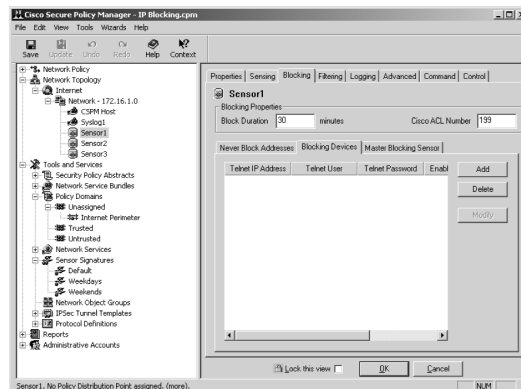
First, we will need to start our Cisco Secure Policy Manager. Once the CSPM is open, we will select our target sensor from the Network Topology Tree in the left pane, as shown in Figure 8.5.

Figure 8.5 The Network Topology Tree



Second, we will select the **Blocking** tab from the sensor view panel on the right side of the CSPM and then select the **Blocking Devices** tab. This will give us a list of the configured network devices currently monitored by the sensor, if any. This can be seen in Figure 8.6.

Figure 8.6 The Blocking Devices Tab



At this point, we can add the blocking device we want to configure to this sensor. By selecting **Add**, we will be given the options we need to configure the sensor to both recognize and manage this blocking device. This can be seen in Figure 8.7.

Figure 8.7 The Blocking Device Properties Dialog

Interface Name	Interface Direction
Ethernet0/0	Inbound

The following fields appear in the Blocking Device Properties dialog:

- **Telnet IP Address** This is needed by the sensor to establish a connection to the blocking device if any changes are to be made to the interface's ACL usage.
- **Telnet Username** This is not always necessary. If usernames are used on the network, then this option will need to be filled in to provide the sensor with the ability to log in. If it is not used, then it is fine to leave this option blank.
- **Telnet Password** This is the login password configured on the blocking device to allow Telnet connections from the sensor.
- **Enable Password** This is necessary for the implementation of any new ACLs. If this is not configured, any sensor-configured ACL updates will not be accepted by the blocking device.
- **Blocking Interfaces** This area specifies the interface and traffic direction of the blocking device the sensor will be managing. To configure this, we will select **Add** and configure the following:
 - **Interface Name** The interface on the blocking device we want to be monitored. This would include the name of the interface and its

respective number. Examples would include, Serial0, FastEthernet2/8. Notice there is no space between the name and the number. This lack of a space is imperative for the sensor to distinguish the interface.

- **Interface Direction** This is where we configure which direction of traffic we want the sensor to monitor. Here we can choose from either Inbound or Outbound. The implications of the direction were covered earlier in the chapter.

To configure more than one interface on a router, select **Add** and configure the appropriate settings for each one individually.

Once we have finished entering our configuration settings, select **OK** twice to accept our changes and then click the **Save** button to save the new configuration in the CSPM database.

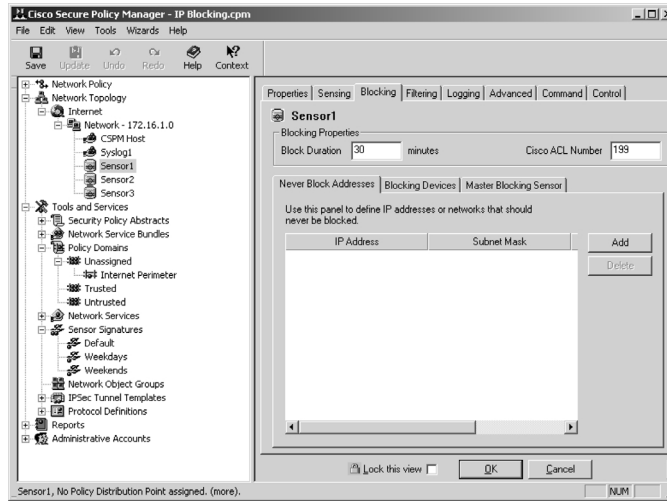
To complete the blocking device configuration, we will now need to push the configuration to the blocking device's respective sensor. After we have saved our new configuration, select the **Update** button in the toolbar to generate the new configuration files used by the sensor. Select the sensor we wish to push the files to; it should already be selected since we chose this for our initial configuration changes in the first step. We then select the **Command** tab. If the preceding configurations have been saved and updated, the **Approve Now** button on the Command tab will be enabled. Click the **Approve Now** button and the configuration files will be transferred. When the **Refresh** button becomes enabled, select it to view the configuration update status.

The Never Block IP Addresses Setup

The **Never Block Addresses** tab is an answer to the critical host issue mentioned earlier in this chapter. As we mentioned, some systems on our networks should never be blocked like a DNS server or a Cisco Secure IDS Director and sensors. This option allows us a safe network-monitoring tool and allows these systems to function normally. The following lists how we can configure these systems as *Never Block Addresses*.

From the Network Topology Tree in the left pane of the CSPM, select the sensor that is monitoring the network that a particular critical host resides upon. Now select the **Blocking** tab as in the previous exercise. We should now be looking at the **Never Block Addresses** tab. If not, select the appropriate tab. This tab can be seen in Figure 8.8.

Figure 8.8 The Never Block Addresses Tab

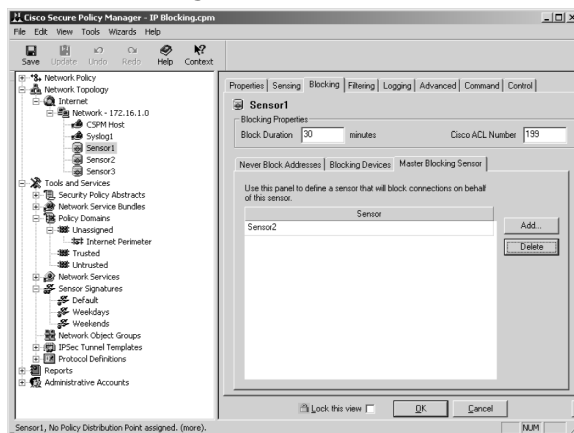


Click the **Add** button to add the critical host(s), or critical subnets of what we will never want to be blocked. These hosts, or networks, will be identified by IP address and subnet mask. We will need to select, add, and configure each host, or network, individually. Once this list is complete, we can choose **OK** and then save our settings. We then need to update our sensors as mentioned in the last exercise. This is done by using the **Update** and **Approve Now** buttons under the **Command** tab of our sensors. This process will need to be repeated for each sensor on the network utilizing IP blocking.

Using the Master Blocking Sensor

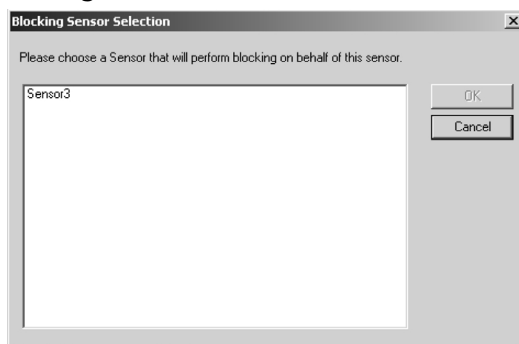
We previously discussed master blocking and its methods for securing various entrances to our networks. If we have a large network with master blocking in place, our sensors will dynamically update each other to protect all entries before an attack can reroute and attempt to regain access. Let's take a look at how this option can be configured.

Select a sensor that will use master blocking from the Network Topology Tree in the left pane of the Cisco Secure Policy Manager. Select the **Blocking** tab and the *Master Blocking Sensor* subtab. The Master Blocking Sensor subtab can be seen in Figure 8.9. In this area, we can see the sensors, if any, that are currently serving as this sensor's master blocking sensors.

Figure 8.9 The Master Blocking Sensor

Select the **Add** button which will open the Blocking Sensor Selection window, this can be seen in figure 8.10. From this window, select the name of the sensor that has been chosen to be a master blocking sensor and select **OK**. In this example, we see that Sensor3 is our only option.

Now select **OK** and click **Save** to save the new settings. From here, we need

Figure 8.10 The Blocking Sensor Selection Window

to update and distribute, or push, our new configuration files as mentioned earlier. Again, this is performed by using the **Update** and **Approve Now** buttons under the **Command** tab of our sensors.

Manually Blocking and Removing a Block

Another option given to use with Cisco Secure IDS is to manually block, or remove a block from, an IP address. Some administrators may like this option, as it will give much more freedom to choose when and where IP Blocking takes place. This may also be an option for a Cisco Secure IDS implementation that

was done quickly and has not yet been fully configured. Another reason could be Mr. Smith in payroll forgot to add your bonus to your last paycheck, (of course we don't condone this type of behavior). Whatever the reason, this process is a simple and effective method for IP Blocking.

Let's first look at manually blocking a specific IP address of a host or a network. Using the Cisco Secure Policy Manager, we need to perform the following steps:

1. Select **Tools | View Sensor Events | Database** to open the Event Viewer – Database Events.
2. Choose **View | Connection Status Pane** for an easier window format to view.
3. Pick an alarm with the source IP address of the target to be blocked.
4. From the menu bar, select **Actions | Block | [Host... or Network...]**.

Shortly, a Shunning Hosts window will appear with the current status of this operation and if the block was successfully executed, a "Success" message will appear. This manually configured IP Block will have a default *Blocking Duration* of 1440 minutes, or 24 hours.

Now that we have covered how to invoke blocking manually on a host or network, let's take a look at how to remove a block from a host or network. This may be a desirable option if a critical host was not identified during the planning process of implementation, a false positive wasn't really an attack, or if a vulnerability was mitigated and the block is not needed anymore.

To remove a block, open the CSPM Event Viewer—do this the same way as when adding a block. Select the sensor which will allow us to view the block. Choose the block with the source IP address of the system or network we want to free up and select **Actions | Block | [Host... or Network...]**. As when implementing a manual block, a window will pop up with the current status information and a "Success" message will appear if the operation succeeded.

Determining the Status of the Managed Device and Blocked Addresses

We have determined our specific needs for Signature selections, picked our blocking devices (less our critical hosts), and established our master blocking sen-

sors. We now need to see what is happening on our network in regards to IP blocking. This is, in fact one, of the most important elements of IP blocking. It probably wouldn't be very beneficial to utilize IP blocking and monitor the usage and threats our network has been, or is being, protected from. We will use the Cisco Secure Policy Manager Event Viewer for monitoring our managed devices and blocked addresses. The CSPM Event Viewer is covered in more depth later in the book.

Using the CSPM, we need to perform the following steps:

1. Select **Tools | View Sensor Events | Database** to open the Event Viewer – Database Events. The CSPM Event Viewer can be seen in Figure 8.11.

Figure 8.11 The Cisco Secure Policy Manager Event Viewer

The screenshot shows the 'Event Viewer - Database Events - CSIDS Alarms' window. The main area contains a table with the following columns: Count, Name, Source Address, Dest Address, Details, and Source Location. The table lists various network events such as DNS requests, ICMP echo requests, and Net Flood attacks.

Count	Name	Source Address	Dest Address	Details	Source Location
5	DNS request for all records	192.168.1.2	66.75.160.41	<none>	OUT
38	ICMP echo reply	+			
38	ICMP echo request	192.168.1.2	+		
8	ICMP unreachable	+			
3	IOS Cisco identification	192.168.1.2	192.168.1.10	<none>	OUT
4	Net Flood ICMP ANY	+			
246	Net Flood TCP SYN	+			
159	Net Flood UDP SYN	+			
5	PostOffice Initial Notification	<none>	<none>	postofficed initial notification msg	OUT
260	Route Down	<none>	<none>	+	
269	Route Up	<none>	<none>	+	
9	TCP Conn Req	192.168.1.2	192.168.1.10		
1	TCP SYN host sweep	192.168.1.2	207.50.4.92	<none>	OUT
1	TCP SYN port sweep	192.168.1.2	192.168.1.10	<none>	OUT
1	Traffic Status Down	<none>	<none>	No traffic flowing for past 90 seconds on fastethernet interface.	OUT
1	Traffic Status Up	<none>	<none>	Traffic flow has started for fastethernet interface.	OUT
449	UDP Packet	192.168.1.10	255.255.255.255	<none>	OUT

2. Select **View | Connection Status Pane** that will give us a cleaner look for information we want by listing the reporting sensors in the left pane of the window.
3. Select a sensor to view its current blocking information in the right pane. An example of this can be seen in Figure 8.12.

Figure 8.12 Event Viewer – Connection Status Pane

Count	Name	Source Address	Dest Address	Details	Source Loc	Dest Loc	SubSig ID
5	DNS request for all records	192.168.1.2	66.75.160.41	<none>	OUT	OUT	0
38	ICMP echo reply	+					
38	ICMP echo request	192.168.1.2	+				
8	ICMP unreachable	+					
3	IDS Cisco identification	192.168.1.2	192.168.1.10	<none>	OUT	OUT	0
4	Net Flood ICMP ANY	+					
246	Net Flood TCP SYN	+					
159	Net Flood UDP ANY	+					
6	PostOffice Initial Notification	<none>	<none>	postofficed initial notification msg	OUT	OUT	0
780	Route Down	<none>	<none>	+			
710	Route Up	<none>	<none>	+			
9	TCP Conn Req	192.168.1.2	192.168.1.10	+			
1	TCP SYN host sweep	192.168.1.2	207.90.4.92	<none>	OUT	OUT	0
1	TCP SYN port sweep	192.168.1.2	192.168.1.10	<none>	OUT	OUT	0
3	Traffic Status Down	<none>	<none>	No traffic flowing for past 90 seco	OUT	OUT	1
1	Traffic Status Up	<none>	<none>	Traffic flow has started for fastelth	OUT	OUT	1
449	UDP Packet	192.168.1.10	255.255.255.255	<none>	OUT	OUT	69

- Choose **View | Block List...** to view a list of currently blocked IPs and their corresponding block duration time left. The title of the window is actually called the Shun List. This list has all the currently blocked IP addresses for the sensor currently selected. Next to the IP addresses is the time, in seconds, left for the IP address to be blocked.

One method that can also be used to monitor blocked addresses is to log on to a specific blocking device and check the ACL manually with a *show access-list* command. This may be a good choice if no CSPM access is available or working on a specific issue.

Summary

When suspect traffic is found either entering or trying to enter our networks, Cisco IDS sensors can implement IP blocking to stop this traffic in its tracks. By using a Cisco Access Control List, the suspect traffic can be stopped by filtering data packets by IP address, port, or protocol. This process is applicable for Cisco IOS network devices, particularly Cisco routers and PIX firewalls.

Device Management is the process a sensor takes after accepting new alarms from monitored devices. The sensor receives an alarm and produces an ACL suitable to stop the offending traffic at the interface we have configured it to. This could be on an outgoing interface, such as leaving the router into our network, or on an incoming interface, such that the router will no longer accept the network traffic and saving valuable router processing resources. Device Management can use Telnet or SSH, to connect to any devices being monitored by the sensor. Of course, this means the router must support and be able to be configured to accept certain Telnet or SSH connection requests.

After the network device accepts the Telnet/SSH request from the sensor, the sensor pushes a newly configured ACL to the appropriate interface regardless of any currently existing ACL. Those ACLs will simply be “unapplied” and replaced with the new ones. No ACLs will be merged. This process only uses two ACLs: 198 and 199. This provides consistent security on the network while an update takes place. The process creates an ACL and pushes it to the appropriate device, or devices. The preconfigured ACL is applied automatically, thus removing the old ACL at the same time. The old ACL will be used in the event another violation occurs and the same process happens again.

When there is more than one entry to a network, which is often the case, the use of multiple IDS sensors may be desirable. One should be able to monitor each of the border routers for incoming and outgoing traffic. When an attack occurs on a router’s interface at an intranet entry point, the sensor monitoring the traffic on that router will use device management to stop the attack. This, however, does not protect the other entry points from the same attacker who will more than likely find another route into the intranet. In this case, Cisco Secure IDS provides us with *master blocking*. Master blocking, configured on one sensor, will push the same ACL to any of the sensors monitoring other entry point network devices with a request to implement this ACL to their respective network devices. This will keep the malicious traffic from entering any other way. It is advisable to make master blocking sensors perform as master blocking sensors for

each other. If it works one way, it will work the other way as well and thus protect the network if this situation were to be reversed.

Access Control Lists, or access-lists, are the primary tool used with IP blocking. These lists are used to filter particular traffic from an interface, either trying to enter a router or leave a router, depending upon the interface it has been applied to. Access-lists can be standard or extended (as well as various other types which are beyond the scope of this book). Their access-list numbers, 1–99 and 1300–1999 for standard, and 100–199 and 2000–2699 for extended, can identify these two types of access-lists. As we know, the two being used by device management for IP blocking are 198 and 199 and therefore are extended access-lists. Extended access-lists allow us to configure network traffic to be denied at an interface by source or destination IP address, port numbers, and protocol.

Using the Cisco Secure Policy Manager (CSPM), we can add, and remove, sensors to be managed by one director and have signature updates pushed manually or automatically to other sensors. This utility is also used to assign network devices to particular sensors and allows us to easily configure for master blocking. We may choose our signatures from this feature as well which is a very important part in planning and should be thought out considerably. From a software level, the CSPM is a great tool for all around secure network device management.

The CSPM is also a valuable tool for manually blocking a host or network IP address range and, in turn, removing blocks from a host or network IP address range. The Cisco Event Viewer database events allow us to view currently blocked IP addresses and their current block duration. With the event viewer, we can also view the status of currently managed network devices.

Solutions Fast Track

Understanding the Blocking Process

- ☑ IP blocking is the process of blocking IP addresses from entering or leaving a particular interface based on a signature comparison previously created. When a traffic pattern is detected, the source address of that traffic will be blocked from passing any more traffic through that interface.

- ☑ When IP blocking is implemented, it will only be in place for the blocking duration configured, 30 minutes by default, or 24 hours for a manual block.
- ☑ Blocking can be applied on an interface for either traffic coming in (inbound) or traffic coming out (outbound). The difference is traffic coming in to a router is not processed and dropped at the front door so to speak. Traffic blocked at the outbound side of the interface has already been processed by the router and actually been switched to the correct interface only to be halted when it arrives.

Understanding Master Blocking

- ☑ Master blocking is the process of using one sensor, monitoring a perimeter router, to perform the same blocking function as another sensor, on the same network, that has already been implemented. This helps to protect all network entry points from the same damaging traffic, if that traffic tries to enter the network from another ingress.
- ☑ The master blocking device accepts the request of the blocking forwarding device.
- ☑ Large networks with more than one network entry point should have this feature in place. It is recommended to have all perimeter routers monitored by master blocking sensors. This will keep all entry points protected from the same attack without each sensor having to find out for itself and perhaps sustain network damage.

Using ACLs to Perform Blocking

- ☑ An ACL, or access-list, is a feature used by Cisco network devices. It is a packet filtering capability that can be specifically configured to block, or allow, certain traffic.
- ☑ IP blocking takes advantage of access-list 199 and 198. ACL 199 is the first to be implemented and when a violation occurs, ACL 198 will be created and updated to all associated network devices.
- ☑ Device management is the actual process of creating and updating ACLs to sensor monitored network devices.

Configuring the Sensor to Block

- ☑ Using the Cisco Secure Policy Manager (CSPM) allows us to configure sensors to monitor particular network devices, establish our signature selection, assign a blocking duration, configure master blocking, and much more.
- ☑ Simply choosing a sensor and adding the network devices to its blocking devices list will enable IP blocking to take place. However, the network device will need to be configured to allow Telnet connections. The Telnet and enable passwords will need to be known on the sensor as well.
- ☑ The Event Viewer is a method used for manually blocking or unblocking IP traffic on network devices. By selecting an alarm, the source IP address which caused the alarm, can be blocked or unblocked.

Determining the Status of the Managed Device and Blocked Addresses

- ☑ The CSPM Event Viewer can be used to monitor the status of managed network devices. The network device window will show information regarding the device's current time setting, status, device type, and the version of the device.
- ☑ Blocked IPs can be viewed through the Cisco Event Viewer database events "Shun" window. This window will list currently implemented blocks in place with their respective source and destination IP addresses and their block duration time remaining.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Frequently Asked Questions

Q: Why would I want my IP blocking ACL to expire?

A: You would want your IP blocking ACL to expire over time in an effort to keep your router’s interfaces clean from unnecessary configurations. Sometimes a blocked attack might be a false positive and simply be something the Cisco Secure IDS thought was an attack. In this case, the ACL would expire after the configured Blocking duration. The bottom line, however, is if you left all the ACLs in place and let them grow to block all alarms, over time it’s easy to imagine the list that could develop. The router would have to parse this list every time a datagram would try to pass out (or in) the interface. This simply wastes processing resources and time.

Q: Can I use IP blocking with network devices other than Cisco IOS products?

A: No. Cisco Secure IP blocking is used exclusively with the Cisco IOS access-list technologies. Therefore, it must be used with Cisco IOS products.

Q: Will IP blocking work with my 6500 switch?

A: Theoretically, this should work as long as it is configured to use a Cisco IOS interface. Be sure to check www.Cisco.com for the latest enhancements to IP blocking and be sure to thoroughly test the configuration before implementing.

Q: Can I use IP blocking on the internal network as well as the Internet interfaces?

A: This would depend on how sensitive your protected data is to both the Internet or to unauthorized associates on the internal network. For instance, there may be an administrative subnet on a network that only authorized sys-

tems should access. If the authorized systems have static IP addresses, then it may be a good idea to use IP blocking. If the systems obtain their IP addresses dynamically, via DHCP, then this would not be a good option. Either way, it will require a great deal of planning and testing to make sure authorized hosts are never blocked out.

- Q:** If applying an ACL to an “external – in” interface stops all unwanted traffic from entering my router and saving processing time, why would I ever put it on my “internal – out” interface?
- A:** On some networks there can be standardized or complex ACLs already in place on the external interface when implementing an IDS design. It may take some time to reconfigure these ACLs or there may be too much “red tape” to go through to make a change. Another possible reason could be a public web server on a DMZ may be reached through the networking device and should never have IP addresses blocked (an exception to this, of course, is an external IP address that matches an internal network IP address).

Capturing Network Traffic

Solutions in this Chapter:

- Switching Basics
- Configuring SPAN
- Configuring RSPAN
- Configuring VACLs
- Using Network Taps
- Using Advanced Capture Methods
- Dealing with Encrypted Traffic and IPv6

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Capturing traffic is one of the most basic configuration skills needed for a successful IDS deployment. Capturing traffic is also one of the most misunderstood processes of deploying an IDS sensor. The axiom “if the switch port can’t see the traffic, then neither can the IDS sensor” must be followed. A successful IDS sensor deployment requires that the sensor see all the traffic of interest wherever it has been placed on the network. To add to the fun of capturing traffic are virtual LANs (VLANs). And to kick up the anxiety level a notch, there are VPNs, SSL, and IP version 6. All of this must be accounted for when trying to roll out the IDS sensors. In the old days of networks, there were hubs or what is called “transparent bridges.” These were very simple devices and it was easy to sniff or capture traffic since the traffic went everywhere. With the advent of switching, however, life became more difficult. The switch is nothing more than single-port transparent bridges tied to together in a common chassis. So the collision domain has been broken up but not the broadcast domain. This is why on a switched network you can capture broadcast traffic till the cows come home but not much else. We will show you in this chapter how to get around this troublesome improvement in network design. Of course, there are VLANs which thankfully many IDS sensors can work with, but this is not true of encryption. It’s almost impossible to use an IDS sensor on encrypted traffic. And encryption comes in a lot of flavors nowadays. We have SSL, VPNs, IPSec, SSH, and many others. To effectively capture traffic, we must be aware of these limitations and how to get around them. One of the newest kinks in the world of IDS sensors capturing traffic is the deployment of IP version 6. While it’s still not a very mainstream issue, it will be in the coming years and we need to be aware of it now.

NOTE

To verify that the monitoring interface actually sees traffic, use the Solaris snoop command:

```
snoop -d [name of interface]
```

For a 4230 IDS sensor, the Ethernet interface name is `spwrX`, as shown in the following example:

```
snoop -d spwr0 ; where spwr0 is the monitor interface, and
```

```
snoop -d spwr1 ; where spwr1 is the control interface
```

For Token Ring, the interface name is `mtok36`, and for FDDI, the interface name is `ptpci`.

For a 4210 IDS appliance sensor, the Ethernet interface name is different, as shown next:

snoop -d iprb0 ; where iprb0 is the monitor interface, and

snoop -d iprb1; where iprb1 is the control interface

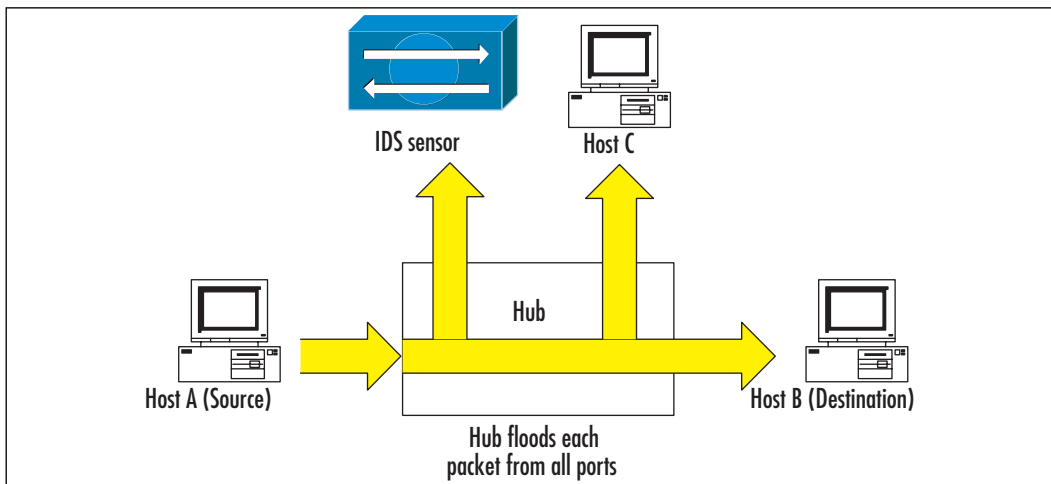
Use CTRL-C to break out of snoop.

Switching Basics

During the last five or so years, Ethernet networks have silently undergone a major change. Earlier, they were built using hubs, but now almost everywhere switches are used. This change becomes very apparent when we start to consider the effects on the traffic-capturing process and the implementation of intrusion detection systems. Let's see what the major difference between hubs and switches is and what problems a switched environment presents to IDS.

The primary difference between a switch and a hub is that the hub is considered shared media or a single collision domain. Anything that one port on a hub sees, all ports will see, such as that in Figure 9.1.

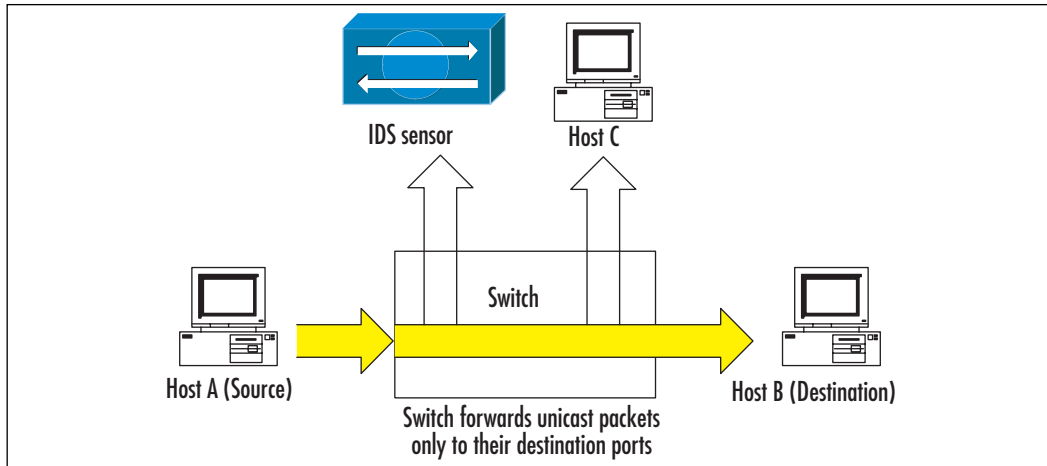
Figure 9.1 A Hub Broadcasts All Traffic



On the other hand, a switch is a more intelligent device than the average hub, it learns which MAC addresses are located on each of its ports and then stores that information in a lookup table. When the switch receives an Ethernet

packet destined for a specific MAC address, the switch forwards it only to the corresponding port, as shown in Figure 9.2.

Figure 9.2 Switch Operation



But there are exceptions to this rule on switches. The switch will send the frame out a single port unless it is a broadcast frame, in which case all ports except the one the frame arrived on will get a copy of the frame. There is a second modification to this rule if the frame's MAC address is not in the forwarding table of the switch. In this situation, the switch then "floods" the frame out of all of its ports except the one the frame arrived on.

So, to review switch theory in simple terms, a switch consists of a set of one-port hubs (each port) which breaks up the collision domain into multiple collision domains. Since the switch is a layer-2 device, the broadcast domain does not change until we get to the router. Neither hubs nor switches will change the header of the frame so we will see the term "transparent bridges," something which refers to the fact that the frame header is not changed in transit through the hub or switch. It is this "switching" of the frame between ports that makes our life with the IDS sensor much more difficult, but not impossible.

The problem posed by switches is that no matter how you connect a traffic-capturing device to a switch, it will not see any traffic, with the exclusion of broadcast packets. There are several options available to avoid this problem (besides using hubs instead of switches, which is usually not practical from the point of view of bandwidth consumption).

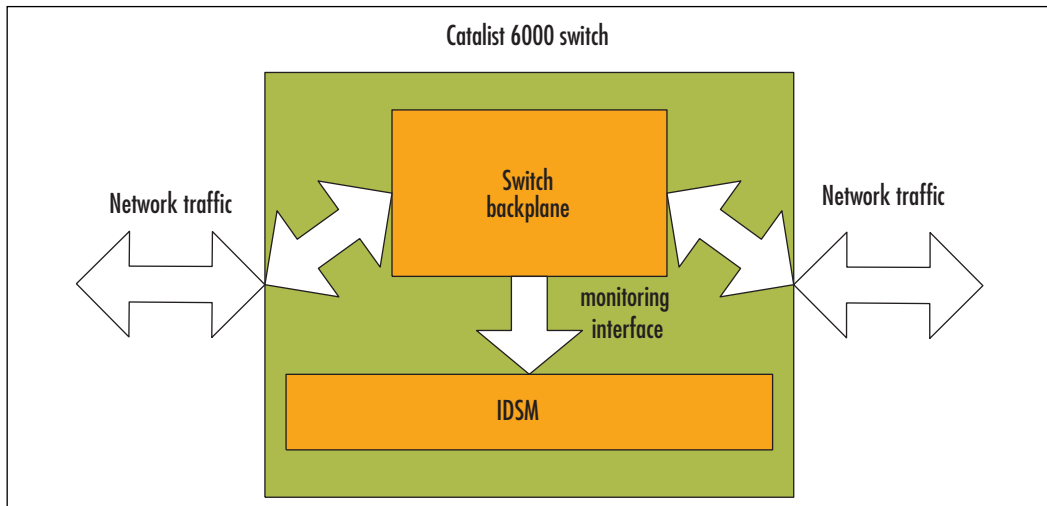
One approach is to use network taps that tend to be passive devices and which are inserted between a monitored network device and a switch. A network tap copies the information from the monitored link to a separate cable which is plugged into an IDS sensor. Taps are designed in a “fail-open” way so that if they break or lose power, the monitored link is not affected. Taps exist for almost any type of line or connection speed, including optical and Gigabit Ethernet lines. We will discuss the usage of taps in more detail at the end of this chapter.

Another way to address the capturing problems created by switches is to use a SPAN ports feature, provided by most switches currently on the market. SPAN stands for Switch Port Analyzer and is also sometimes called “port mirroring,” although technically port mirroring is a subset of port spanning features. A switch can be configured to have a dedicated port to which any packet that passes through the switch is copied. Depending on the switch model, this process can cause an overhead in packet processing, although there are switches where spanning ports do not affect switching capacity.

NOTE

When using spanning ports, only packets that get inside the switching backplane are copied to the spanning port. So, for example, frames with incorrect CRCs are dropped when they enter the switch and are consequently not copied to any of the SPAN ports.

The last option, which is available only with the Cisco Catalyst 6000 IDS Module, is to monitor network traffic directly on a switch backplane. Since IDSM has access to the switching fabric, there is no need to copy packets between ports to redirect them to IDS, thus the only configuration task remaining is to specify the “interesting” traffic that needs to be monitored (see Figure 9.3). This is done using VLAN access-lists or VACLs, which we look at in more detail next.

Figure 9.3 Monitoring Traffic by IDSM

All three options are discussed in this chapter, although the main means of using IDS in a switched environment is still the port spanning feature, which will be described in more detail than the other two.

Configuring SPAN

Different models of Cisco switches have different capabilities regarding the number of ports that can be dedicated simultaneously as SPAN ports, restrictions on how VLAN-separated traffic is monitored, and so on. They also differ in the way the SPAN feature is configured, mainly because there are two different command-line interfaces—one for IOS-based switches, and the other for CatOS switches (supervisor engines of high-end switches, to be more precise).

We will start from the simpler IOS-based interface, which is applicable to the 2900/3500 series and those 4000/6000 switches that run the integrated Cisco IOS feature set (the supervisor engine in native mode).

Configuring an IOS-Based Switch for SPAN

With IOS-based switches, there are two configuration types depending on which switch model you are working on. A simpler SPAN feature is used on series 2900/3500 switches, while a more powerful SPAN feature set can be applied to 4000 or 6000 series switches running an integrated Cisco IOS command set. We will discuss both, starting with a simpler SPAN configuration.

Configuring 2900/3500 Series Switches

The Catalyst 2900/3500 series have basic port spanning features, while the IOS-based SPAN configuration is initiated using just one main command:

```
port monitor <interface>
```

This command is used in the configuration of a port dedicated to the SPAN feature (also called a monitor port or SPAN destination port—essentially, the port where traffic is copied to), and the parameter *<interface>*, which lists interfaces that should be monitored by this SPAN port (SPAN source ports). Two main restrictions must be taken into consideration when configuring port spanning on these switches:

1. The SPAN destination port and all the ports it monitors must belong to the same VLAN.
2. If the parameter *<interface>* is not specified, all ports from this VLAN (to which a monitor port belongs) are monitored.

There are also some restrictions regarding which ports can act as SPAN destination ports (all restrictions are described in the corresponding model documentation):

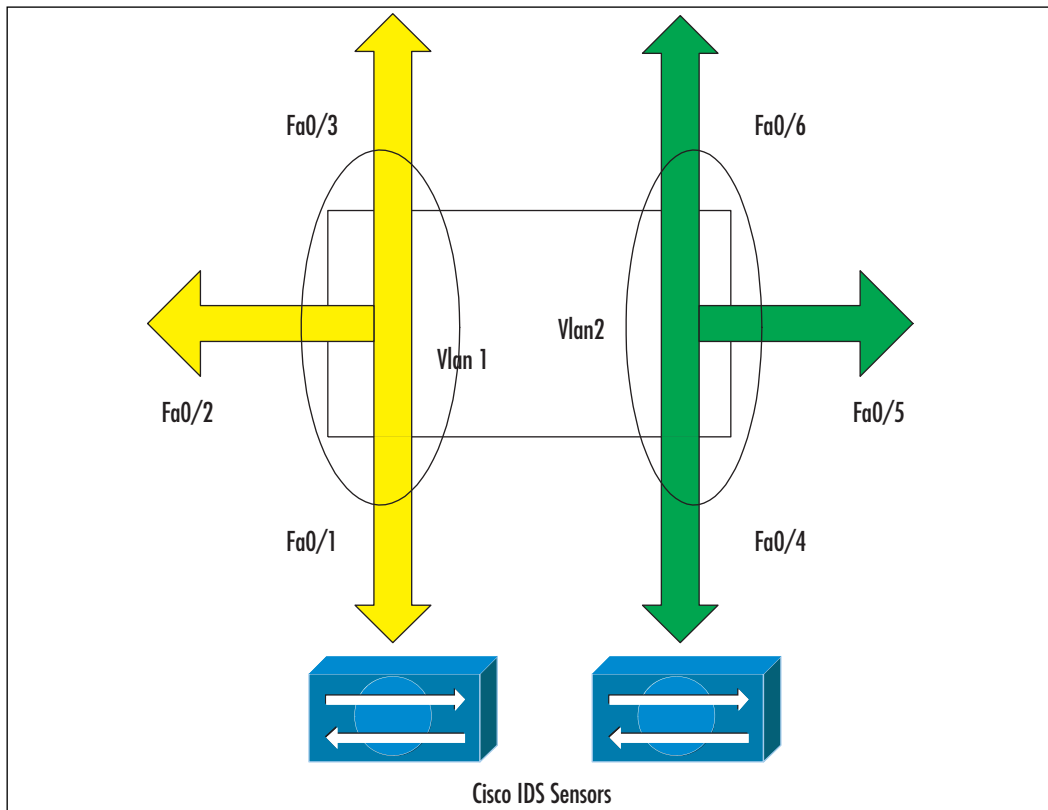
- The monitor port must belong to the same VLAN as the monitored ports. It is not possible to change VLAN membership on the monitor port or ports being monitored.
- The monitor port cannot be a trunk port or dynamic-access port. On the other hand, a static-access port can monitor a VLAN on a trunk, dynamic-access, or multi-VLAN port. The VLAN monitored will be the VLAN to which the monitor port belongs.
- An ATM port cannot be a monitor port.
- The monitor port cannot belong to a Fast EtherChannel or Gigabit EtherChannel port group.
- The monitor port cannot have more security enabled.
- The monitor port cannot be a multi-VLAN port.
- Port monitoring does not work if both the monitor and the monitored ports are protected ports.

NOTE

The monitor port does not run STP (Spanning Tree Protocol—the word “span” in this term is not related to SPAN ports), so it is advisable not to connect this port to anything but IDS systems. If, for example, it is connected to a hub or bridge so that it creates a loop in the network, it can affect packet forwarding heavily.

Let’s take a look at the following situation shown in Figure 9.4. We have a Catalyst 2900 switch with ports Fa0/1, Fa0/2, and Fa0/3 belonging to a VLAN 1, and ports Fa0/4, Fa0/5, and Fa0/6 belonging to a VLAN 2. Port Fa0/1 will be used to monitor VLAN 1 (source ports Fa0/2 and Fa0/3), and port Fa0/4 will monitor VLAN 2 (ports Fa0/5 and Fa0/6).

Figure 9.4 An Example Using the 2900 Series Switch



Before SPAN ports are configured, the corresponding part of switch configuration appears as the following:

```
!  
interface FastEthernet0/1  
!  
interface FastEthernet0/2  
!  
interface FastEthernet0/3  
!  
interface FastEthernet0/4  
switchport access vlan 2  
!  
interface FastEthernet0/5  
switchport access vlan 2  
!  
interface FastEthernet0/6  
switchport access vlan 2  
!
```

This simply states that ports Fa0/1 to Fa0/3 belong to the default VLAN 1, while ports Fa0/4 to Fa0/6 belong to VLAN 2.

In order to configure port Fa0/1 as a monitor port, we need to put it in the configuration mode and enter the list of ports to be monitored:

```
sw2900(config)# int Fa0/1  
sw2900(config-if)# port monitor fastethernet 0/2  
sw2900(config-if)# port monitor fastethernet 0/3  
sw2900(config-if)# ^Z
```

These commands state that each packet received or transmitted through ports Fa0/2 and Fa0/3 will be copied to port Fa0/1. If there are any other ports in VLAN 1, they will not be monitored. If we want to monitor the whole VLAN 2, we would simply use these commands:

```
sw2900(config)# int Fa0/1  
sw2900(config-if)# port monitor  
sw2900(config-if)# ^Z
```

When SPAN source ports are not specified in the *port monitor* command, traffic from the whole VLAN is monitored. If you try to specify as a source a port from another VLAN, you will get an error message saying it is impossible.

A similar configuration applies to VLAN 2 and resembles the following:

```
!
interface FastEthernet0/1
port monitor FastEthernet0/2
port monitor FastEthernet0/3
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
port monitor FastEthernet0/3
port monitor FastEthernet0/6
switchport access vlan 2
!
interface FastEthernet0/5
switchport access vlan 2
!
interface FastEthernet0/6
switchport access vlan 2
!
```

You can check which SPAN sessions are configured on a switch by using either the *show running* or *show port monitor* commands. The latter displays a list of monitor ports and corresponding SPAN sources for each SPAN port.

```
Switch#show port monitor
Monitor Port Port Being Monitored
-----
FastEthernet0/1 FastEthernet0/2
FastEthernet0/1 FastEthernet0/3
FastEthernet0/4 FastEthernet0/5
FastEthernet0/4 FastEthernet0/6
```

NOTE

The switches *previously* described always copy both *ingress* (incoming) and *egress* (outgoing) packets from monitored ports to a monitoring port. So, if a packet is switched between two monitored ports, it will be seen twice by an IDS—after it enters the switch and before it leaves the switch.

Configuring a 4000/6000 Series IOS-Based Switch

The configuration of 4000/6000 series IOS-based switches resembles the preceding configuration, but their SPAN features are more complicated and flexible. They differ from 2900/3500 spanning port configurations in two main ways:

- It is possible to have source ports not belonging to the same VLAN (that is, there is no rule that the monitor and all monitored ports should belong to one VLAN), and
- It is possible to configure a direction of the monitored traffic—for example, monitor only ingress packets or only egress or both.

A configuration of each SPAN session consists, in this case, of two tasks: designating source ports and destination ports. There are restrictions on how many SPAN destination ports a switch can have. For the 4000 series, it is two ingress sessions and four egress sessions. A session monitoring traffic in both directions counts as one ingress and one egress session. SPAN destination interfaces cannot receive any ingress traffic, so if you want to send anything from the IDS back to the network, you will need another connection on a non-spanning port.

SPAN source ports are configured using the command:

```
[no] monitor session session_number source interface type/num | vlan  
vlan_ID [rx | tx | both]
```

This command specifies source ports or whole source VLANs for a specific SPAN session and also the direction in which traffic from this source will be monitored. Parameter *rx* turns on monitoring for ingress packets, *tx* turns it on for egress packets, while *both* works for both directions. If no direction is entered in this command, then *both* is assumed. The prefix *no*, as usual, deletes an already configured source. For example:

```
Sw4000 (config) # monitor session 1 source interface fa2/1 tx
```



```
Sw4000 (config) # monitor session 1 source interface fa2/2 rx
Sw4000 (config) # monitor session 2 source vlan 1 rx
```

It is possible to use several VLAN IDs in one command, for example:

```
Sw4000 (config) # monitor session 2 source vlan 1, 5 - 7
```

You cannot mix source ports and source VLANs in one session—each session can have as a source either ports or VLANs, but not both. SPAN destinations are configured with the command:

```
[no] monitor session session_number destination interface type/num
```

For example,

```
Sw4000 (config) # monitor session 1 destination interface fa3/38
```

After source and destination ports for the session are configured, the switch starts to copy packets between the source port and a destination port.

There is a possibility to use a trunk interface as a SPAN source and then filter only traffic from specific VLANs you are interested in to the destination port. To accomplish this, first designate the trunk port as a source port for a session and then use the following command:

```
[no] monitor session session_number filter vlan vlan_ID
```

For example (if Fa2/1 is the trunk port):

```
Sw4000 (config) # monitor session 3 source interface fa2/1 tx
Sw4000 (config) # monitor session 3 filter vlan 3 - 5
```

It is not possible to have a source VLAN and a trunk port with filtering in the same session, although it is possible to have trunk and non-trunk ports in one session. To disable a specific session, use the following command:

```
no monitor session <session_number>
```

Finally, you can view the active SPAN configuration with the command:

```
show monitor session <session_nimber> {detail}
```

It displays SPAN sources, destinations, and filters. For example:

```
Sw400# show monitor session 3
Session 3
-----
Source Ports:
```

```

RX Only:      Fa2/1
TX Only:      Fa2/2
Both:         None
Source VLANs:
RX Only:      None
TX Only:      None
Both:         None
Destination Ports: Fa3/38
Filter VLANs: 3-5

```

This output describes a situation where session 3 is configured with source ports Fa2/1 (in ingress direction) and Fa2/2 (in egress direction) and the destination for this session is port Fa3/38. From the trunk port Fa2/1, only traffic belonging to VLANs 3 to 5 is monitored.

NOTE

Cisco documentation sometimes uses the abbreviations PSPAN and VSPAN. Their meaning is simple: PSPAN means Port-based SPAN—a case when sources for a session are ports, and VSPAN is a VLAN SPAN, when session sources are VLANs.

Configuring a SET-Based Switch for SPAN

CatOS-based switches like 4000, 5000, and 6000 series use a different command syntax. They are also sometimes called Set-based switches, because a lot of configuration work is done using the *set* command. A command for configuring SPAN on these switches is *set span*.

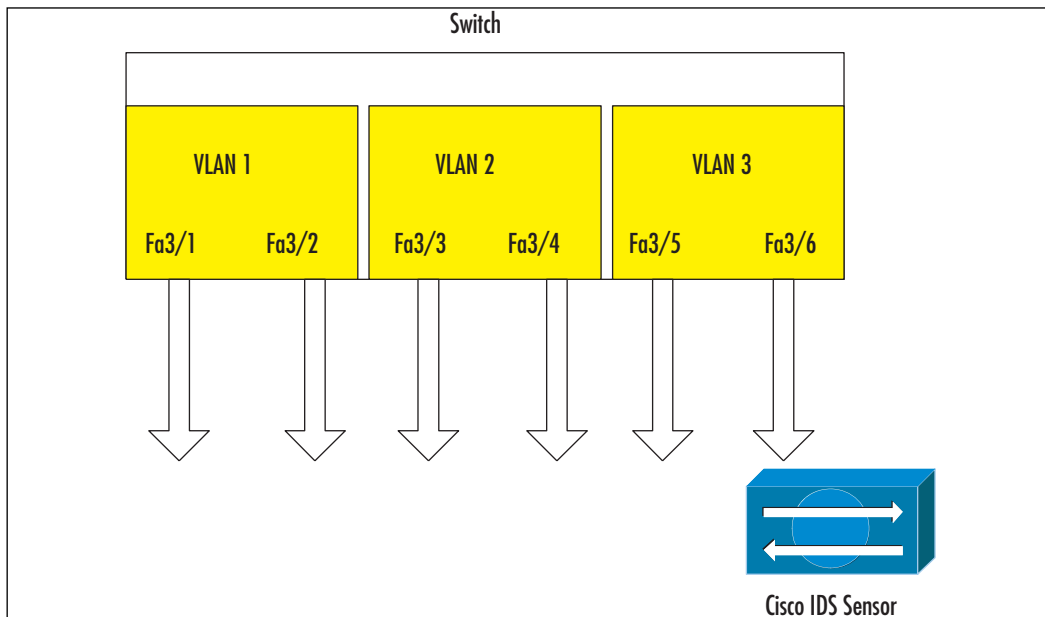
```

Sw6000 (enable) set span
Usage: set span disable [dest_mod/dest_port|all]
       set span <src_mod/src_ports...|src_vlans...|sc0>
              <dest_mod/dest_port> [rx|tx|both]
              [inpkts <enable|disable>]
              [learning <enable|disable>]
              [multicast <enable|disable>]
              [filter <vlans...>]
              [create]

```

We will use the following port configuration, as shown in Figure 9.5.

Figure 9.5 Example Switch Ports and VLANs



The simplest case is when you need to copy traffic from specific ports to a port where an IDS is attached (a destination port). For example, to monitor ports 3/1, 3/2, 3/3, and 3/5 using an IDS module attached to port 3/6, you need to enter the following command:

```
Sw6000 (enable) set span 3/1-3, 3/5 3/6
```

This command produces output describing a new span session similar to this:

```
Destination : Port 3/6
Admin Source : Port 3/1-3, 3/5
Oper Source : Port 3/1-3, 3/5
Direction : transmit/receive
Incoming Packets: disabled
Learning : enabled
Multicast : enabled
Filter : -
Status : active
switch (enable) 2003 Jun 19 08:34:36 %SYS-5-SPAN_CFGSTATECHG:local span
session active for destination port 3/6
```

The session becomes active immediately. The first parameter for a *set span* command in this case is a list of source ports (3/1–3 means 3/1 through 3/3), while the destination port 3/6 is the second parameter. This command also takes several optional switches, which specify more detailed features. As with the earlier IOS-based configurations, it is possible to select the direction of the captured traffic: only ingress traffic, only egress traffic, or traffic in both directions. The preceding example does not have any keyword describing the direction, so the *both* keyword is assumed. To monitor only ingress traffic, the command line could be

```
Sw6000 (enable) set span 3/1-3, 3/5 3/6 rx
2003 Jun 19 08:35:37 %SYS-5-SPAN_CFGSTATECHG:local span session inactive
    for destination port 3/6
Destination : Port 3/6
Admin Source : Port 3/1-3, 3/5
Oper Source  : Port 3/1-3, 3/5
Direction   : receive
Incoming Packets: disabled
Learning    : enabled
Multicast   : enabled
Filter      : -
Status      : active
switch (enable) 2003 Jun 19 08:35:37 %SYS-5-SPAN_CFGSTATECHG:local span
session active for destination port 3/6
```

The output produced by this command (assuming it was entered after the command from the previous example) shows that the previously configured span session was disabled and a new one created. By default, there is only one session active on a switch. In order to create a new session without disabling another one, use the keyword *create*:

```
Sw6000 (enable) set span 3/1 3/4 create
```

This command creates a second session on the switch, which you can check using the *show span* command:

```
Sw6000 (enable) show span
Destination : Port 3/6
Admin Source : Port 3/1-3, 3/5
Oper Source  : Port 3/1-3, 3/5
Direction   : receive
```

```
Incoming Packets: disabled
Learning : enabled
Multicast : enabled
Filter : -
Status : active
```

```
-----
Destination : Port 3/6
Admin Source : Port 3/1
Oper Source : Port 3/1
Direction : transmit/receive
Incoming Packets: disabled
Learning : enabled
Multicast : enabled
Filter : -
Status : active
Total local span sessions: 2
```

SPAN sessions can be disabled with the command

```
Sw6000 (enable) set span disable [ all | destination_port ]
```

The keyword *all* disables all configured sessions, and specifying a destination port disables the session monitored by this port only.

NOTE

For Catalyst switches with the IDSM module, the SPAN destination should be the first port on the corresponding slot. For example, if IDSM is module 6, then the corresponding destination will be 6/1.

By default, no packets are received by the switch on a SPAN destination port (this is what is generally needed when an IDS is connected to this port). If you want to allow switches to receive packets on a destination interface too, use the *inpkts enable* option, although this is not advisable, because it can cause bridging loops. Also, by default a destination port learns MAC addresses from incoming packets it receives. From the IDS point of view it is better to switch this feature off using the *learning disable* option, for example:

```
Sw6000 (enable) set span 3/1 3/4 inpkts disable learning disable create
```

Configuring & Implementing...

SPAN Ports and Bridging Loops

Let's consider a scenario where we have a VLAN distributed between several switches and we want to monitor its traffic from a remote location. In this case, the switches are connected to each other by trunks. One obvious approach would be to create a SPAN session monitoring traffic from the desired VLAN (VLAN 1, for example) on each switch and have their destination ports connected to the same switch or hub, where IDS is also connected. IDS will be able to see traffic from the whole VLAN 1. Unfortunately, if destination ports are working in both directions—not only transmitting but also receiving packets, they will be interchanging their traffic on the IDS switch and will thus create a bridging loop. Remember, SPAN destination ports do not run STP, which could have prevented this.

There is no way to fix this when using 2900/3500 series switches, so it is recommended not to use such configurations with them. In the case of 4000/6000, both running Integrated IOS and CatOS, destination ports are unidirectional by default, which prevents most of the problems that could arise.

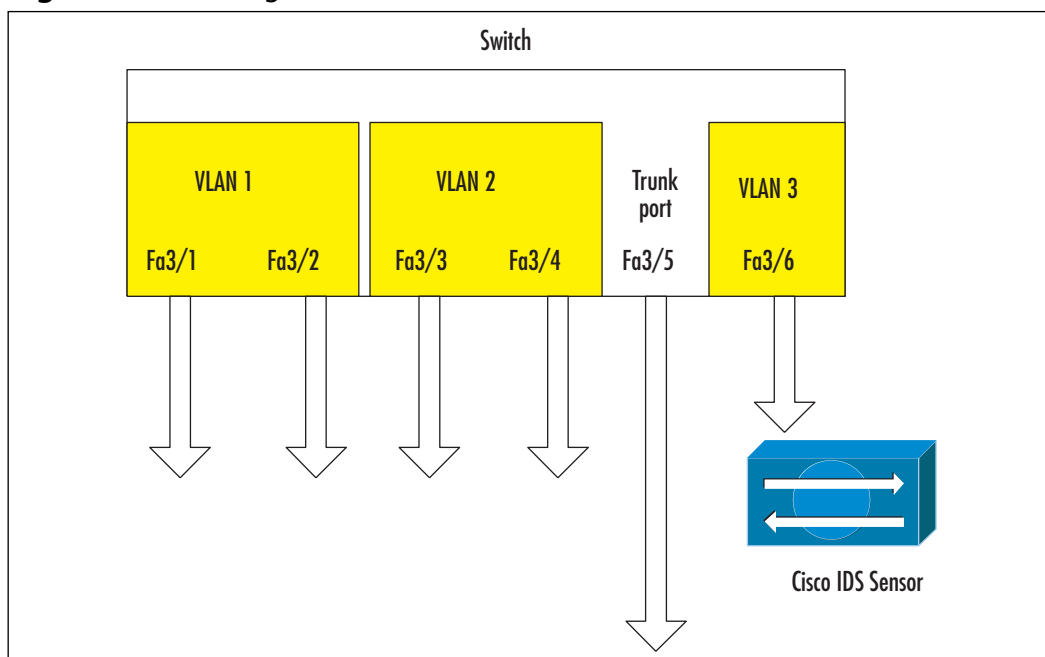
The best solution is to use RSPAN (Remote SPAN), which does exactly the job we are trying to do here: collect traffic from several switches and deliver it over trunk connections to one destination. Configuring RSPAN is described later in this chapter.

As with other models, it is possible to monitor not only specific ports, but whole VLANs. The command line remains the same except that sources are denoted by VLAN numbers instead of port names. For example:

```
Sw6000 (enable) set span 2,3 3/4
```

This creates a session monitoring traffic from VLANs 2 and 3 and then copying it to the port 3/4.

Consider a more complex situation: let's assume we have a switch with one trunk port and we want to monitor this switch traffic from the whole VLAN 1 (which is distributed), excluding one port, 3/1, as shown in Figure 9.6.

Figure 9.6 Filtering on a Trunk

This means we need to monitor all traffic from VLAN 1 coming from the trunk, and also from port 3/2, but not 3/1. The command

```
Sw6000 (enable) set span 1 3/6
```

will result in forwarding all VLAN 1 traffic to monitor port 3/6. Another possible solution

```
Sw6000 (enable) set span 3/2, 3/5 3/6
```

will get too much traffic—in other words, the whole trunk 3/5 instead of only VLAN 1 packets.

The required result is achieved by using the VLAN filtering feature.

```
Sw6000 (enable) set span 3/2, 3/5 3/6 filter 1
```

This gives us exactly what we need—only traffic from ports 3/2 and 3/5, which belongs to VLAN 1. The output from *show span* command indicates this:

```
Destination : Port 3/6
Admin Source : Port 3/2, 3/5
Oper Source : Port 3/2, 3/5
Direction : transmit/receive
```

```
Incoming Packets: disabled
Learning : enabled
Multicast : enabled
Filter : 1
Status : active
```

It is possible, of course, to filter on more than one VLAN ID, for example:

```
Sw6000 (enable) set span 3/5 3/6 filter 1,2
```

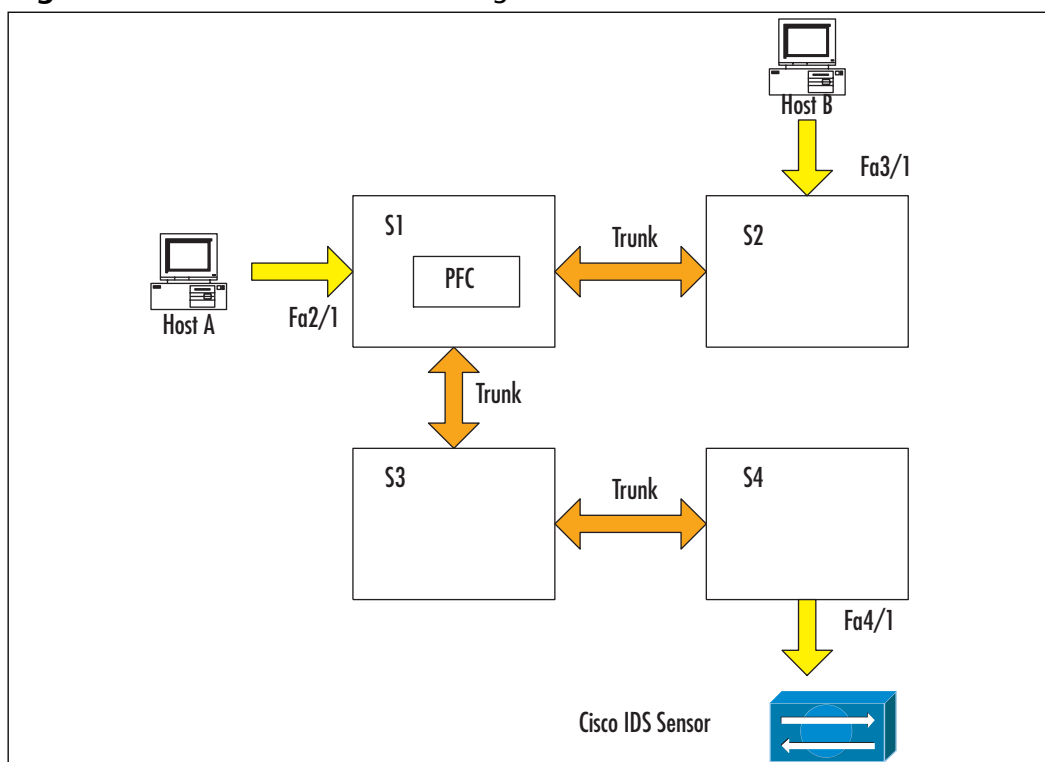
will copy from trunk port 3/5 to port 3/6 only traffic belonging to VLANs 1 and 2.

NOTE

VLAN filtering is possible on Catalyst 4000 and 6000 series switches. The Catalyst 5000 series switch does not support the filter option in the *set span* command.

Configuring RSPAN

The earlier “SPAN Ports and Bridging Loops” sidebar described a situation where in a distributed switch environment an administrator wants to monitor a set of ports or VLANs spread over several switches. While approaches described in a sidebar typically work, the best solution in this case is to use Remote SPAN feature (RSPAN). In short, this approach joins all ports to be monitored in a special RSPAN VLAN and traffic from this VLAN is transferred over trunk ports to the destination port, where an IDS is attached. See Figure 9.7.

Figure 9.7 RSPAN Traffic Forwarding

In Figure 9.7, switches S1 and S2 are called source switches. Currently, a switch can have only one RSPAN VLAN configured (this means it is not possible to have on the same switch two sources for two different RSPAN sessions).

Switch S3 is an intermediary switch. It does not have the preceding restrictions on a number of RSPAN VLANs, because it simply forwards the traffic. Switch S1 also acts as an intermediary switch, forwarding traffic from host B.

Finally, switch S4 is a destination switch. Some of its ports are configured as RSPAN destinations. Catalyst 6000 can currently have up to 24 destination ports for RSPAN sessions. All switches are connected via ISL trunks. STP is running, so loops will be prevented.

The configuration process consists of creating a RSPAN VLAN on source switches, configuring trunks on intermediary switches (if they are not already in place) and specifying destination ports on destination switches. Specific commands used for RSPAN configuration are different in cases of IOS-based and CatOS Catalyst 4000/6000 switches, so we will describe them separately.

Configuring an IOS-Based Switch for RSPAN

The process is different for source and destination switches. Intermediary switches do not need any additional configuration provided that trunking infrastructure is already in place.

A RSPAN VLAN is created first. This is done by creating a VLAN and then using the command *remote-span* in the *config-vlan* mode to specify that this VLAN is for Remote SPAN. For example:

```
R4000(config)# vlan 123
R4000(config-vlan)# remote-span
R4000(config-vlan)# end
```

configures a VLAN 123 for RSPAN. The command *no remote-span* turns off the RSPAN feature on this VLAN. This command is entered only on one switch and the knowledge about this VLAN is propagated using VTP to all other participating switches

Source Switch Configuration

Sources of traffic are configured similar to a local SPAN mode. In such cases, the destination of this session is set to a remote SPAN VLAN. For example, on switch S1:

```
R4000-1(config)# monitor session 1 source interface fa2/1 rx
R4000-1(config)# monitor session 1 destination remote vlan 123
```

On switch S2:

```
R4000-2(config)# monitor session 1 source interface fa3/1 rx
R4000-2(config)# monitor session 1 destination remote vlan 123
```

Destination Switch Configuration

On a destination switch, the configuration is somewhat reversed compared to the source switch. The source of a session is the RSPAN VLAN and a destination, the port to which IDS is connected. For example, on switch S4

```
R4000-4(config)# monitor session 1 source remote vlan 123
R4000-4(config)# monitor session 1 destination interface fa4/1
```

It is also possible to filter traffic further by using VLAN access-lists (VACLs), which is described later in this chapter.

Configuring a SET-Based Switch for RSPAN

Basic steps are the same as with IOS switches. Trunking structure is configured independently of RSPAN and has to be in place before RSPAN is configured. Basically, you need to use the same VTP domain on all switches and configure some ports as trunking-desirable. VTP negotiation will do the rest. For example, running the command:

```
Sw4000-1(enable) set vtp domain cisco
```

```
Sw4000-2(enable) set vtp domain cisco
```

on all switches, and additionally using the command

```
Sw4000-2> (enable) set trunk 5/1 desirable
```

on switch S2 will result in establishing trunking between them.

Then RSPAN VLANs are created. Using the same numbering as in previous sections, we need to configure the following on a VPT server switch:

```
Sw4000> (enable) set vlan 123 rspan
```

```
Vlan 123 configuration successful
```

```
Sw4000> (enable) show vlan
```

VLAN	DynCreated	RSPAN
---	-----	-----
1	static	disabled
2	static	disabled
3	static	disabled
99	static	disabled
123	static	enabled

Source Switch Configuration

In source switch configuration, source ports are again configured similarly to local SPAN sources, with the keyword *rspan* used instead of *span* and where a destination using the *set rspan* command is always an ID of an RSPAN VLAN. For example:

```
Sw4000-1> (enable) set rspan 2/1 123 rx
```

```
Rspan Type : Source
```

```
Destination : -
```

```
Rspan Vlan : 123
```

```
Admin Source : Port 2/1
```

```

Oper Source   : None
Direction    : receive
Incoming Packets: -
Learning     : -
Multicast    : enabled
Filter       : -

```

This configures ingress traffic from port 2/1 as a source for the RSPAN session associated with RSPAN VLAN 123.

NOTE

In this output, *admin source* are source ports or source VLANs configured from the console. The *Oper Source* field shows ports that are actually monitored—for example, if the administrative source includes a VLAN, then the operational source will list all ports belonging to this VLAN. The *Oper Source* field is not updated until the session is active and is never used for RSPAN sources.

It is also possible to use VLANs as sources for RSPAN, for example:

```

Sw4000-1> (enable) set rspan source 200 123 rx
Rspan Type      : Source
Destination     : -
Rspan Vlan      : 123
Admin Source    : VLAN 200
Oper Source     : None
Direction      : receive
Incoming Packets: -
Learning       : -
Multicast      : enabled
Filter         : -

```

Destination Switch Configuration

On a destination switch, the destination port is configured this way:

```

Sw4000-4> (enable) set rspan destination 4/1 123
Rspan Type : Destination

```

```

Destination : Port 4/1
Rspan Vlan : 123
Admin Source : -
Oper Source : -
Direction : -
Incoming Packets: disabled
Learning : enabled
Multicast : -
Filter : -

```

RSPAN sessions can be disabled on source switches by using:

```

Sw4000> (enable) set rspan disable source all
This command will disable all remote span source session(s).
Do you want to continue (y/n) [n]? y
Disabled monitoring of all source(s) on the switch for remote span.

```

Or, for a specific session, identified by RSPAN VLAN number:

```

Sw4000> (enable) set rspan disable source <vlan_number>

```

Sessions can also be disabled on destination switches using

```

Sw4000> (enable) set rspan disable destination all
This command will disable all remote span destination session(s).
Do you want to continue (y/n) [n]? y
Disabled monitoring of remote span traffic for all rspan destination ports.

```

Or, for a specific session identified by a port number:

```

Sw4000> (enable) set rspan disable destination <port_number>

```

Configuring VACLs

VLAN Access Control Lists (VACLs) is the tool for controlling redirection of traffic within VLANs—both bridged and Layer 3—switched. Packet filtering can be done based on Layer 2, 3, and 4 headers. VACLs are enforced in hardware and do not produce overhead. In general, they are similar to IOS access lists, the main difference is that VACLs are not direction-specific and capture both ingress and egress traffic. In order to use the VACL feature, you need to have a PFC (Policy Feature Card) installed.

VACLs allow for much more granular control over the selection of traffic forwarded for inspection by an IDS system. It is possible, for example, to capture traffic based on source or destination IP addresses, to filter it by TCP port numbers or capture only packets from established sessions. Furthermore, MSFC (Multilayer Switch Feature Card) can use flows to ensure that packets crossing the backplane between VLANs are not duplicated when captured. VACLs are especially useful when an IDS Module is installed on a Catalyst switch.

Configuring VACLs is more complicated than SPAN settings. The following steps need to be performed:

1. Create a VACL to capture interesting traffic.
2. Commit a VACL to switch hardware.
3. Map the VACL to specific VLANs.

After that, a monitoring port is selected and assigned as a VACL capture port. In the case of IDSM, it will be port 1 on the module.

NOTE

By default, port 1 on IDSM is set as a trunk port by default and will monitor traffic from all VLANs where appropriate VACLs are configured. If you want to monitor specific VLANs only, you need to clear the unwanted VLANs from this trunk. We show this in detail in Chapter 6.

As usual with high-end switches, configuration commands depend on which software runs on a switch. We will see how VACLs are configured on a CatOS switch and then compare this to an IOS-based one.

Configuring & Implementing...

Which Is Better—VACL-Based Capture or SPAN Ports?

Both technologies provide a means for capturing network traffic. Either can be more useful than the other, depending on the circumstances. SPAN sessions are much easier to configure, but they are limited in number (you can have two to six local SPAN sessions and up to 64 RSPAN destination sessions on one switch depending on a model) and can drop or duplicate packets in some cases.

VACLs can capture inter-VLAN traffic (they actually capture traffic based on IP flows instead of matching ports or VLAN names) and this capture is performed with a high degree of granularity. On the other hand, the VACL feature is available only on high-end switches with PFC cards installed. Furthermore, it is possible to have only one VACL per protocol, that is, you can configure only one VACL for IP traffic.

The sidebar “Using RSPAN and VACLs Together” describes how VACLs can be applied to RSPAN VLANs in order to filter traffic in the distributed capturing environment.

On a SET-based switch, VACLs are created using the *set security acl* command. Its syntax when it is used for capturing IP traffic is as follows:

```
set security acl ip <acl_name> permit <protocol> <src_ip_address>
    [operator port] <dest_ip_address> [operator port] [established]
    capture
```

The *protocol* field can be any IP protocol, or the abbreviations *tcp*, *udp*, or *icmp*. For example, this sequence of commands:

```
Sw6000> (enable) set security acl ip IDSCAP permit tcp 192.168.1.0 0.0.
    0.255 range 1024 32000 10.1.1.0 0.0.0.255 lt 1024 capture
IDSCAP editbuffer modified. Use 'Commit' command to apply changes
Sw6000> (enable) set security acl ip IDSCAP permit ip any any
IDSCAP editbuffer modified. Use 'Commit' command to apply changes
Sw6000> (enable)
```

creates a VACL which captures traffic with source IP addresses from network 192.168.1.0/24, source ports 1024-32000, and destinations in the network

10.1.1.0/24, as well as destination ports 1–1023. It also has a *permit any any* at the end, because there is an implicit *deny any any* at the end of each VACL, and we do not need to really drop any traffic, just select some of it for inspection.

The next stage is to commit the access list to hardware. This is done either for each list by its name or all of them at the same time using the command

```
commit security acl <acl_name> | all
```

For example,

```
Sw6000> (enable) commit security acl IDSCAP  
Hardware programming in progress...  
ACL IDSCAP is committed to hardware.  
Sw6000> (enable)
```

The final step in VACL configuration is mapping a created access-list to specific VLANs which have to be monitored. The command is as follows:

```
set security acl map <acl_name vlan>
```

NOTE

When mapping VLANs using the *set security* command, valid values for the VLANs are from 1 to 1005, and from 1025 to 4094.

For example, to map our IDSCAP access-list to VLANs 100 and 200, we would use the following set of commands:

```
Sw6000> (enable) set security acl map IDSCAP 100  
ACL IDSCAP mapped to vlan 100  
Sw6000> (enable) set security acl map IDSCAP 200  
ACL IDSCAP mapped to vlan 200
```

The preceding steps are common in VACL configuration, but in the case of VACLs with the *capture* feature, we also need to specify the destination of the captured traffic. This is done using the command

```
set security acl capture-ports mod/ports...
```

This command specifies a set of ports as capture destinations. For example, with the IDSM module installed in slot 5, the following command will forward captured traffic to the module (IDSM capture port is port 1, 5/1 in this case):


```
Sw6000> (enable) set security acl capture-ports 5/1
Successfully set 5/1 to capture ACL traffic.
```

On IOS based switches, different commands are used, although the same steps are followed. The preceding example would be implemented in the following way. First, an extended IP ACL would be created like so:

```
R6000 (config)# ip access-list 101 permit tcp 192.168.1.0 0.0.0.255 range
1024 32000 10.1.1.0 0.0.0.255 lt 1024
```

This list does not need a *permit any any* clause at the end, because it will not actually filter any traffic, only match a part of the traffic for capture. Then, a VLAN access map called IDSCAP is created and configured to match traffic based on IP access list 101 which then captures matched traffic:

```
R6000 (config)# vlan acces-map IDSCAP
R6000 (config-access-map)# match ip address 101
R6000 (config-access-map)# action forward capture
```

This map is applied to VLANs that have to be monitored by an IDS:

```
R6000 (config)# vlan filter IDSCAP vlan-list 100,200
```

Finally, a port on a switch (or on an IDSM module) is configured as a destination port for captured traffic.

```
R6000 (config)# interface gigabitEthernet 8/1
R6000 (config-if)# switchport capture
```

Designing & Planning...

Using RSPAN and VACL Together

As was noted in the section “Configuring RSPAN,” the task of capturing traffic in a distributed switch structure is difficult and requires a strategic approach. It is very easy to oversubscribe monitoring ports so that switches start dropping spanned packets. VACLs provide a neat way of controlling RSPAN-produced traffic.

You can configure a RSPAN VLAN with interesting traffic, and then further narrow traffic selection down, applying a VACL to this VLAN. The process is not easy though, depending on how complicated the infrastructure is, what software is used on switches (with CatOS configurations being generally more straightforward), and how complicated the conditions for the traffic selection are.

VACL are not compatible with some features of Cisco IOS Firewall for MSFC. You cannot apply VACLs to a VLAN in which there is an *ip inspect* rule. There is a workaround for this case, though—using the command

```
mls ip ids <acl_name>
```

This command matches incoming traffic against a specified extended IP access-list. If a packet is permitted by the ACL, it is captured. If a packet is denied, it is not captured. Thus, the packet is not actually permitted or denied—it is always forwarded to its destination. The example of configuration is shown next (these commands are executed on the MSFC):

```
R6000 (config)# ip access-list 101 permit tcp 192.168.1.0 0.0.0.255 range  
                  1024 32000 10.1.1.0 0.0.0.255 lt 1024  
R6000 (config)# interface vlan 100  
R6000 (config-if)# mls ip ids 101
```

After the capture destination is configured on the supervisor engine using the commands described earlier, either

```
set security acl capture-ports
```

or in the case of IOS-based switches

```
switchport capture
```

NOTE

For IDS Module to capture packets marked by the *mls ip ids* command, port 1 of the IDSM must be a member of all VLANs where these packets are routed.

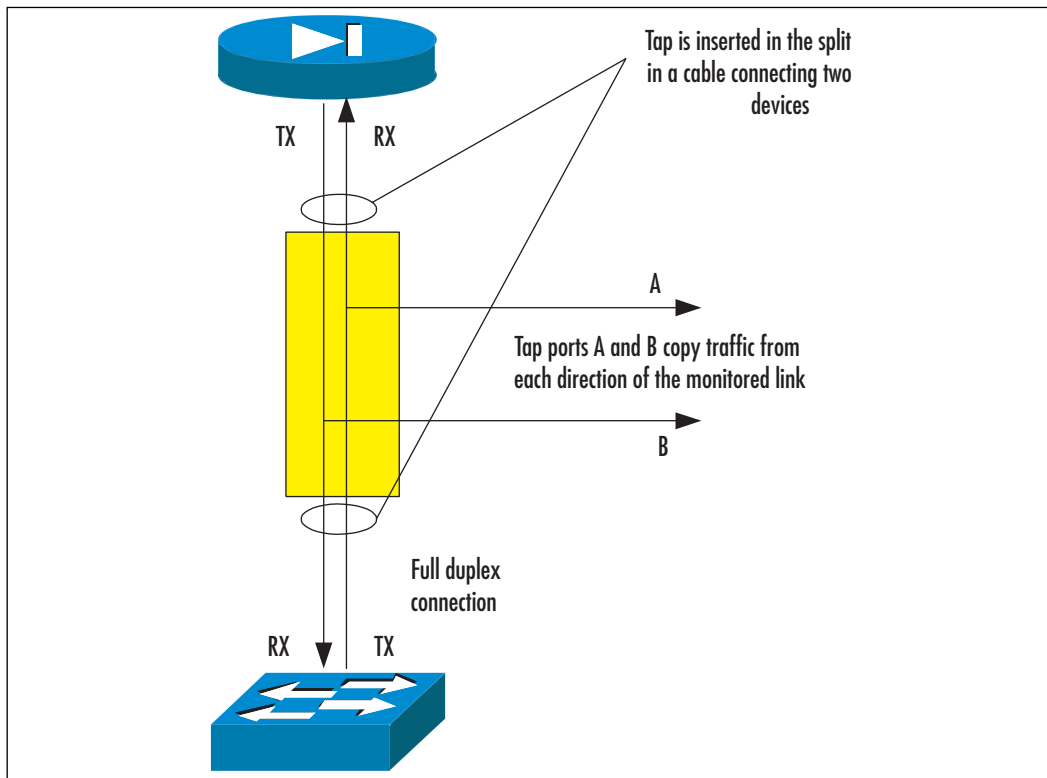
When using VACLs, the capture port of IDSM has to be a member of VLANs where monitored packets are internally routed.

Using Network Taps

As we saw earlier, in cases where monitoring is needed for a set of links widely distributed over different switches, configurations can get quite complicated where RSPAN, VACLs, and trunking are involved. There are also cases when features such as RSPAN are simply unavailable because they are not supported by hardware—for example, with 2900 series switches.

The other option for adding IDS systems to such environments is to use network taps. A network tap is a device that is inserted into the monitored link. This device usually has at least four ports—two for connecting a network cable of a monitored link and two output ports where the traffic is copied. When used on a full-duplex connection, the tap splits copied traffic into two—one monitoring port outputs traffic flowing in one direction and the second port tackles traffic flowing in the opposite direction (see Figure 9.8). One of the nice features of the tap compared to SPAN ports is that taps monitor all traffic, including incorrect or control frames, which are usually not copied to SPAN ports on switches. Some network taps allow traffic flow in one direction while others allow dual-direction traffic. Why would a network tap permit this, you ask? Because your IDS sensor may allow for something called TCP Resets where the IDS sensor can send an IP reset packet to break the connection of a suspected attacker. Without the ability to send traffic back through the TAP, this capability would be lost.

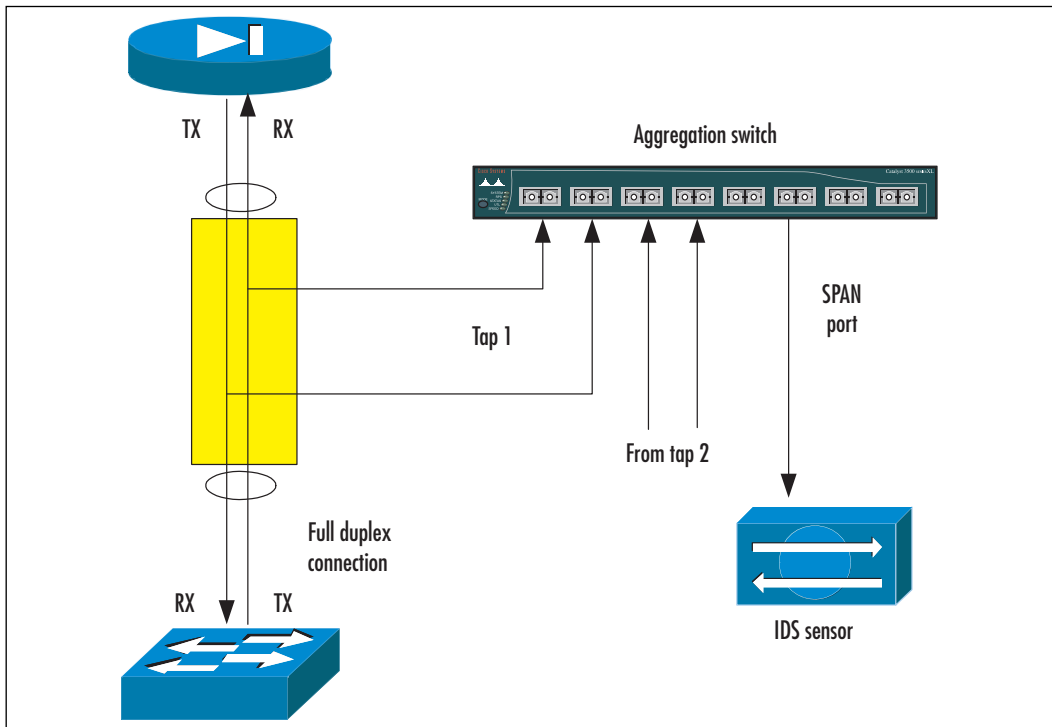
Figure 9.8 Network Tap Connections



There also exist multiport taps, which allow monitoring of a number of connections by the same device. Taps are different from small hubs—they are designed so that in case of a power failure they do not block traffic on a monitored line (they “fail open”), as a hub would. Some larger tap products may have internal load balancers to prevent packet loss—for example, it is possible to have a Gigabit Ethernet tap which outputs captured traffic into several monitor ports, where a set of IDS sensors is connected.

Taps do, however, pose some challenges from an implementation point of view. Most important is the fact that tap output is two data streams and IDS usually has only one monitoring interface. This means that tap outputs have to be connected to an aggregation device of some sort, where traffic is assembled. This device can be a hub or a switch, although hubs are not recommended—when both flows of a single full-duplex connection are plugged into the same hub, this will most likely result in a heavy collision rate, meaning an IDS will not be able to see much. Thus, it is more appropriate to use a switch. This switch can have many taps connected. The output port, connected to the IDS is usually a local SPAN port, configured to monitor all tap connections, as shown in Figure 9.9.

Figure 9.9 Aggregating Tap Traffic on a Switch



Multiport taps often come with an internal aggregation device, which outputs collected traffic into a designated “analysis” port.

NOTE

As usual, with multiple taps connected to the same switch it is possible to oversubscribe a SPAN port. This can be avoided, for example, by using switches that have Gigabit Ethernet ports for SPAN ports monitoring several 100-Mbps links.

The pros and cons of SPAN ports and network taps are shown in Tables 9.1 and 9.2.

Table 9.1 SPAN Port Pros and Cons

Advantages	Disadvantages
No extra cost for hardware	Packets go through the switch backplane and can be delayed or retimed.
Allows monitoring of many links simultaneously	Easy to oversubscribe the monitoring port in cases where many links are monitored, which leads to packet losses.
Generally easier to implement	Do not capture anomalous frames, because these are dropped by the switch logic. May sometimes affect switch performance. Moving an IDS to another location usually requires the heavy reconfiguration of switches.

Table 9.2 Network Tap Pros and Cons

Advantages	Disadvantages
Sees 100 percent of the packets on the monitored link	Extra hardware cost (may be very expensive for complex solutions).
IDS monitor can be moved without reconfiguring core network switches	Sees only one link at a time, full-duplex links are divided into two streams.

As a result, taps are often used on core links—inter-switch trunks, server farms, and so on. SPAN ports are commonly used in smaller networks, on the leaf nodes, and when planning IDS installation and testing, because they allow for easy drafting of IDS' place in the network infrastructure. Of course, with the Catalyst IDSM module, the situation is completely different than with external sensors, there is no need to use taps because IDSM is already connected to the switch backplane.

Two of the leading vendors of network taps are Finisair (www.finisair.com) and Netoptics (www.netoptics.com).

Using Advanced Capture Methods

Previous sections described how various methods of capturing traffic work and how each feature is configured on different models of hardware. This section applies configuration tips from earlier to some common cases of IDS installation and also describes the specifics of using either standalone IDS or Catalyst IDS Modules.

We will assume that IDSMs are installed in slots 5 and 6 of the Catalyst 6000 switch, and that the VLANs to be monitored are numbered 100, 200, and so on. Regarding the IDSM, we are generally interested in monitoring only Web traffic using VACLs. For external IDS modules, we assume they are connected to ports 3/1, 3/2, and so on.

Capturing with One Sensor and a Single VLAN

Capturing using one sensor and a single VLAN is the simplest case and should be easy to configure. If you are using an external sensor, simply create a SPAN session, either local or remote, for the VLAN you want to monitor and forward all traffic to the port where the sensor is connected. The same configuration can be used with IDSM, setting port 1 of the IDS module card as the SPAN destination.

The simple local SPAN for a 2900 series switch can be configured in this way (see Figure 9.10):

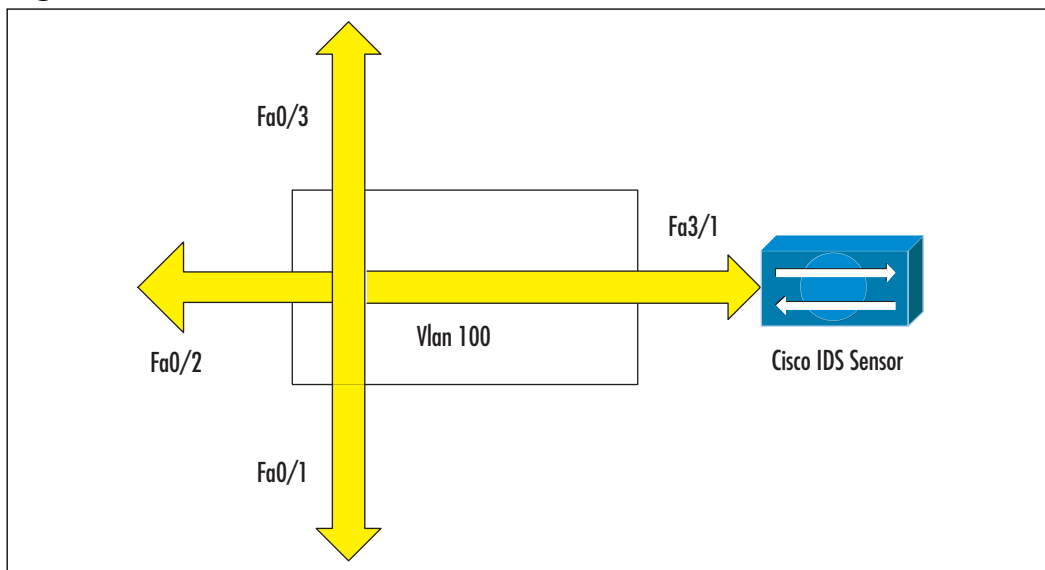
```
!  
interface FastEthernet3/1  
port monitor FastEthernet0/1  
port monitor FastEthernet0/2  
port monitor FastEthernet0/3  
switchport access vlan 100
```

```

!
interface FastEthernet0/1
switchport access vlan 100
!
interface FastEthernet0/2
switchport access vlan 100
!
interface FastEthernet0/3
switchport access vlan 100
!

```

Figure 9.10 Cisco 2900 Switch with One VLAN and One Sensor



On a Catalyst (with CatOS), similar results can be achieved with just one command:

```
Switch (enable) set span 100 5/1 rx create
```

An IOS-based Catalyst requires the following:

```
Switch(config)# monitor session 1 source vlan 100 rx
```

```
Switch(config)# monitor session 1 destination interface Fa5/1
```

VACL-based capture can be configured as follows:

```
switch>(enable) set security acl ip WEBCAP permit tcp any any eq 80 capture
switch>(enable) set security acl ip WEBCAP permit tcp any eq 80 any capture
switch>(enable) commit security acl WEBCAP
switch>(enable) set security acl map WEBCAP 100
switch>(enable) set security acl capture-ports 5/1
```

VACLs can also be configured on an IOS-based Catalyst switch, as described earlier. It is also worth noting that you can use trunking configuration commands to filter traffic reaching the sensor port of an IDS/IPS. This is more important when using several sensors monitoring multiple VLANs, because it helps distributing traffic. The monitoring interface of an IDS/IPS is set as trunk by default. We will use the following commands to filter traffic from all VLANs but 100:

```
switch>(enable) clear trunk 5/1 1-1024
switch>(enable) set trunk 5/1 100
switch>(enable) set vlan 100 5/1
```

Capturing with One Sensor and Multiple VLANs

A more complex example includes one sensor monitoring several VLANs. This is not possible to configure on low-end switches, so we start with the 4000 series and up.

SET-based configuration of SPAN ports (either for external sensors or for IDS/IPS) is straightforward:

```
Switch (enable) set span 100, 200 5/1 rx create
```

For IOS:

```
Switch(config)# monitor session 1 source vlan 100, 200 rx
Switch(config)# monitor session 1 destination interface Fa5/1
```

For VACL-based capturing of Web traffic only one extra line is needed compared to the previous example:

```
switch>(enable) set security acl ip WEBCAP permit tcp any any eq 80 capture
switch>(enable) set security acl ip WEBCAP permit tcp any eq 80 any capture
switch>(enable) commit security acl WEBCAP
switch>(enable) set security acl map WEBCAP 100, 200
switch>(enable) set security acl capture-ports 5/1
```

Trunking can be limited using a command set similar to the one before:


```
switch>(enable)clear trunk 5/1 1-1024
switch>(enable)set trunk 5/1 100, 200
switch>(enable)set vlan 100 5/1
```

However, in cases where only one IDS/IPS is installed, there is not much use in modifying trunking because its monitoring port receives only traffic matched by a VACL on VLANs to which the access-list is applied.

Capturing with Multiple Sensors and Multiple VLANs

The generic case of multiple sensors capturing traffic from a number of VLANs can be very complex depending on the switch infrastructure. The simplest implementation is when there are many 2900-type switches, and sensors are connected to corresponding switches. In reality, this case is just a multiplied “one VLAN, one sensor” case.

Assume now that we have a Catalyst 6000 with two IDS/IPSs installed and we need to capture VLANs 100 and 200 on the module in slot 5, and VLANs 300 and 400 on the module in slot 6. SPAN-based capture is again simpler to configure:

```
Switch (enable) set span 100, 200 5/1 rx create
Switch (enable) set span 300, 400 6/1 rx create
```

Traffic capture using VACLs is more complex in the configuration here, but it also produces the best results, as only interesting traffic is forwarded to monitoring ports of IDS/IPSs:

```
switch>(enable) set security acl ip WEBCAP permit tcp any any eq 80 capture
switch>(enable) set security acl ip WEBCAP permit tcp any eq 80 any capture
switch>(enable) commit security acl WEBCAP
switch>(enable) set security acl map WEBCAP 100, 200, 300, 400
switch>(enable) set security acl capture-ports 5/1, 6/1
```

Now we have the same VACL for IP traffic applied to VLANs 100, 200, 300, and 400. Remember that there can be only one VACL for each type of traffic, and captured traffic is forwarded to all designated capture ports. In order to separate the traffic between two IDS/IPSs, we need to set trunking on their monitoring ports:

```
switch>(enable)clear trunk 5/1 1-1024
```

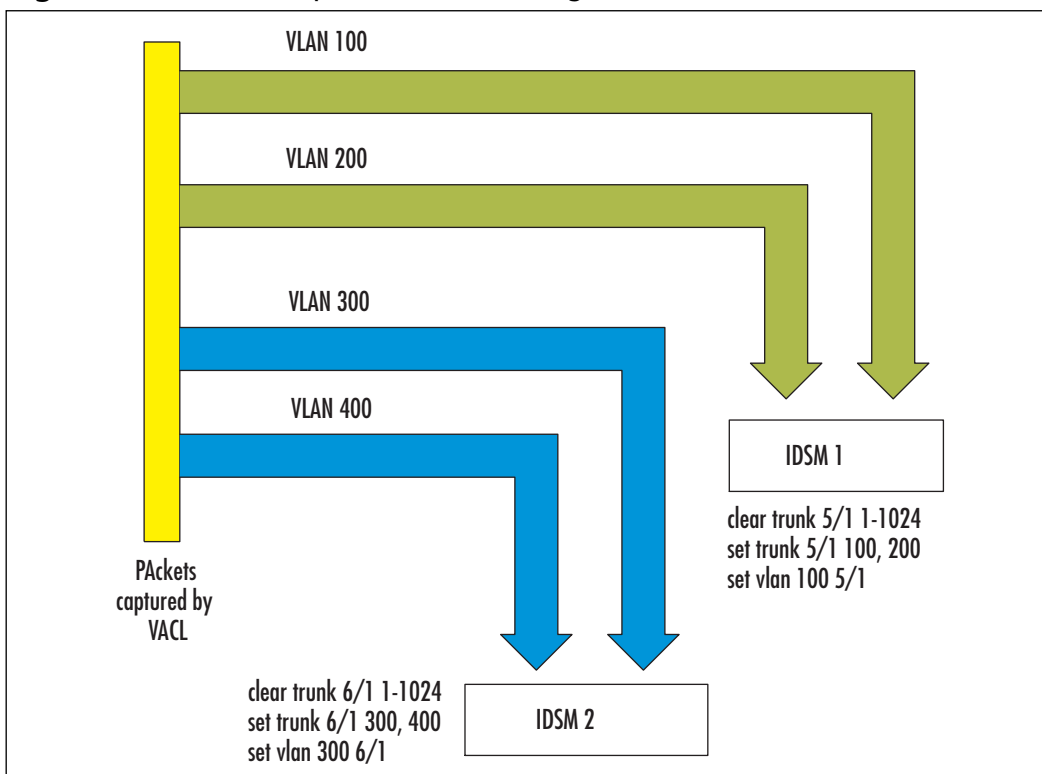
```

switch>(enable)set trunk 5/1 100, 200
switch>(enable)set vlan 100 5/1
switch>(enable)clear trunk 6/1 1-1024
switch>(enable)set trunk 5/1 300, 400
switch>(enable)set vlan 300 5/1

```

After you have configured this, the configuration can be changed to the one in the “one VLAN, one sensor” case by simply filtering out VLAN 100—for example, on the corresponding trunk. Thus, you can configure capturing for many VLANs in advance and then filter already captured traffic before it reaches IDS/IPS by using trunking commands (see Figure 9.11).

Figure 9.11 Traffic Capture, with Trunking as an Additional Filter



Dealing with Encrypted Traffic and IPv6

The last-but-not-least important problem of traffic capture is the spread of various traffic encryption mechanisms. Use of virtual private networks (VPNs),

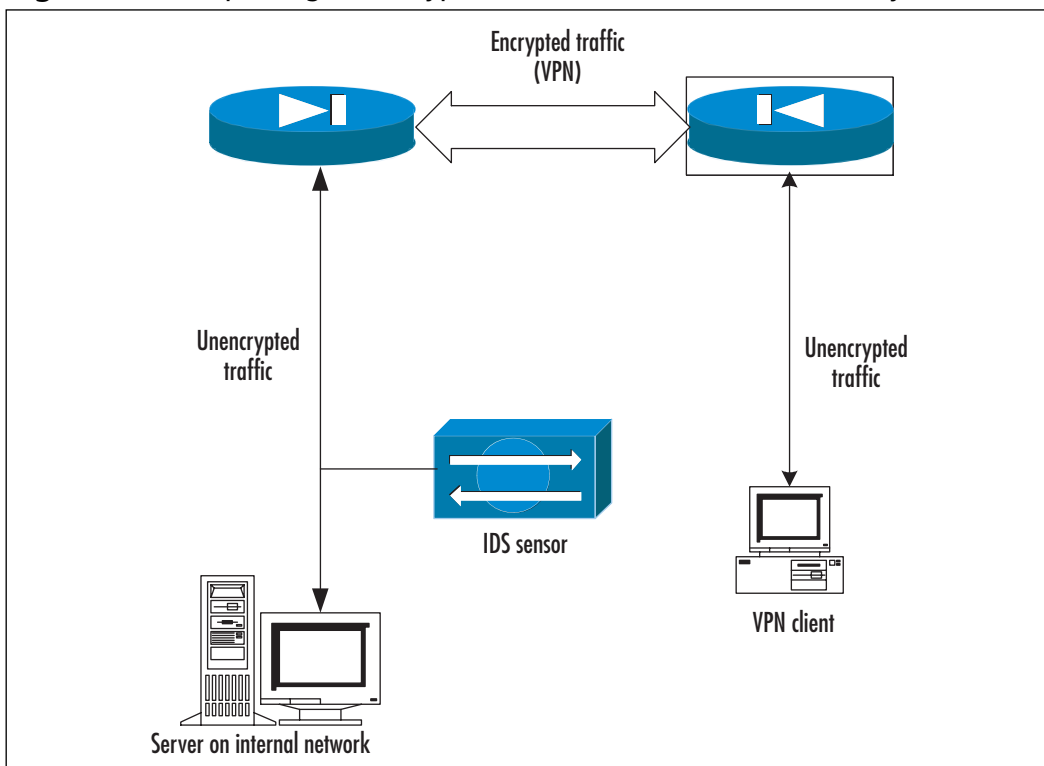
either IPSec-based or otherwise, HTTPS Web servers, and Secure Shell (SSH) became a common issue. From the point of view of parties that participate in the encrypted interchange, traffic sniffing is exactly what they try to avoid by using the encryption. And most of the encryption protocols in use are very good in achieving data confidentiality, so IDS is not able to look inside the encrypted interchange.

On the other hand, consider the following situation: you have a Web server, which is used for e-commerce. Web clients talk to the server over HTTPS connections, thus customer data is not exposed. But when an attacker connects to your server over SSL (using an HTTPS connection), he is able to do what he likes without any IDS noticing it, because all information exchange between a Web browser on an attacker's computer and a Web server (an Apache process, for example) is encrypted. IDS therefore becomes useless in a case like this.

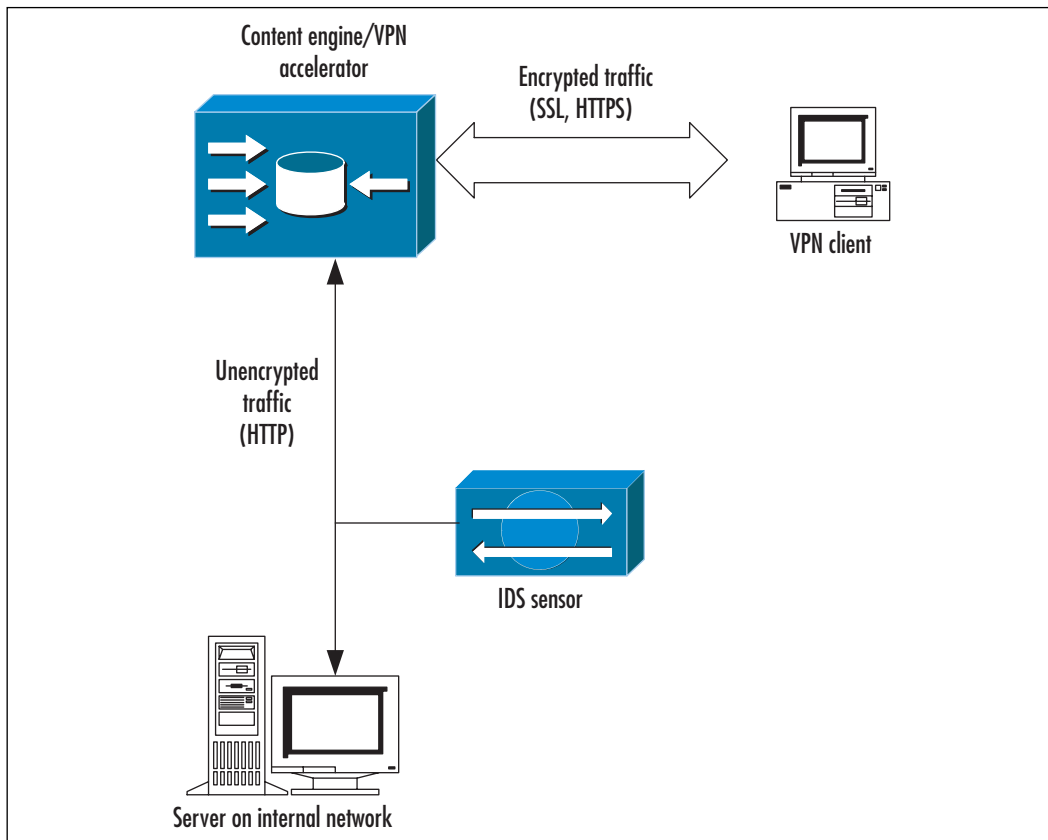
A similar situation arises when SSH is used for logging into hosts on the network. SSH does what it is meant to do—protects traffic, including passwords, from sniffing, disabling IDS capabilities from detecting any wrongdoing. The same goes for VPNs—all encapsulated traffic is usually encrypted between the client and a gateway or destination host.

Unfortunately, the situation cannot be helped much. There are two workarounds for this, though.

- **SSH** Nothing can be done to capture the contents of an SSH session. You can have signatures that will be triggered upon SSH-specific attacks (which use the SSH vulnerabilities, not the local interactive exploits), but you cannot see, for example, somebody running a *su* command in an interactive session.
- **Site-to site VPN** This case is a little easier to handle. All you need to do is capture traffic *behind* your VPN gateway, after it has been received and decrypted. Figure 9-12 illustrates this idea. If the VPN is of the host-to-host type, where encryption and decryption occurs only on connection endpoints, then we find ourselves in the same situation as with SSH and sniffing is not possible.

Figure 9.12 Capturing Unencrypted Traffic Behind a VPN Gateway

SSL connections also cannot be sniffed directly. You can put an SSL accelerator device before the Web server, terminate all incoming SSL connections on this device and let it interact with the server over plain HTTP. The traffic passing on this unencrypted link can then be redirected to the IDS. This setup is shown in Figure 9.13.

Figure 9.13 Capturing HTTPS Traffic

Finally, the increasing use of IP version 6 poses even more problems for IDS. In their current state, almost no IDS can look inside an IPv6 packet going through, and just a few of them can detect basic information such as source and destination IP addresses. One hopes, though, that once the use of IPv6 becomes really widespread, its detection and monitoring will be incorporated into the Cisco IDS solution.

Summary

During the last five or so years Ethernet networks have silently undergone a major change: before they were built using hubs, but now most of the time network infrastructure is based on switches. The main difference between a hub and a switch is that hubs forward each received packet to all their ports and switches—only to the port where the destination device is connected. The latter effectively prevents IDS from seeing any non-broadcast traffic on the switch.

The main solutions to the problem posed by the switching environment to IDS are

- SPAN or mirror ports
- Capturing traffic directly off the switch backplane
- Using network taps for monitoring crucial links

A SPAN port is a port to which traffic from other ports is copied. Many Cisco switches have a port spanning or port mirroring capability, although various models perform differently. Depending on the model, it may be possible to specify several different ports as a traffic source, one or more VLANs, and so on. Switches also have restrictions on how many SPAN sessions can run simultaneously. Sometimes their performance degrades when port spanning is turned on, although usually you do not need to worry about that. One of the more important points is that a SPAN destination port has to process combined traffic from several ports, thus possibly dropping some frames. Catalyst 4000/6000 not only allows for local traffic monitoring, but also has Remote SPAN (RSPAN) capabilities. It is not recommended to connect SPAN ports to other switches or similar network devices because this may cause bridging loops. In cases where such a connection is needed, you have to at least configure a switch so it doesn't receive any packets on a SPAN port.

In the case of Remote SPAN, traffic from designated ports or VLANs is collected on designated source switches in a so-called RSPAN VLAN. It is then passed through the trunking infrastructure to the destination switch where it is forwarded to the destination port and then connected to an IDS module.

The configuration of both local and remote SPAN features on high-end switches depends on the command set used on a switch. The configuration of IOS-based switches for SPAN usually takes more commands than that used for CATOS (so called SET-based) devices.

VACL or VLAN Access Control Lists is a feature available on Catalyst 6000. When enabled, the VACL controls the forwarding of traffic in or between VLANs based on specified criteria. It is also possible to capture some of the forwarded traffic and forward it either to a designated switch port or to an internal IDS module.

Only one VACL for each traffic type is allowed. This means that at any given time on a switch only one IP VACL may exist. If a switch has two IDSs, then all captured traffic will be forwarded to both modules. To separate traffic from different VLANs, trunk-clearing commands can be used, because monitor ports of IDSs are configured as trunks. Another interesting feature of VACLs is that when applied to RSPAN VLANs they help filter monitored traffic even for standalone IDS sensors.

A network tap is a passive device inserted in the monitored link. It copies traffic flowing in both directions onto two monitor ports. A full-duplex link thus usually becomes two links. In order to feed this traffic back into the IDS module with one monitoring interface, an aggregation device is needed. A switch with a SPAN port can be used as such a device. Network taps exist for almost any type of link and are designed in a “fail-open” way (if their power fails, they do not break the monitored link). The latter is a big advantage of taps when compared to using cheap hubs.

An additional problem for traffic capturing and analysis is posed by the increasing use of various tools for traffic encryption. SSH, SSL, virtual private networks and so on all encrypt the data in transit, with the side effect being that IDS cannot analyze the protected traffic. Not much can be done here other than putting the IDS at the point in the network where it can see the traffic already unencrypted, either behind a VPN gateway, or, in the case of SSL connections, between a Web server and an SSL accelerator.

IPv6 is a big problem for IDSs at this time, simply because they are not designed to do anything with traffic that IPv4 didn't do. One can hope, though, that with the spread of IPv6 usage, Cisco IDS will eventually be adapted to it.

Solutions Fast Track

Switching Basics

- ☑ Switches forward traffic only to destination ports, thus preventing IDS from seeing any non-broadcast packets.

- ☑ There are several ways of getting monitored traffic to an IDS sensor. Common ones include SPAN ports and network taps.
- ☑ Cisco Catalyst 6000 switches with embedded IDS modules allow traffic to be captured directly from the switch backplane.
- ☑ VLAN access-lists are used to select monitored traffic for IDSM.

Configuring SPAN

- ☑ The SPAN (switch port analyzer) port is a port configured to receive a copy of traffic passing through other ports on the switch.
- ☑ Low-end models of switches are limited in the number of SPAN sessions they support, and usually require that all monitored ports belong to the same VLAN.
- ☑ With Catalysts 4000/6000, it is possible to have up to six sessions on one switch. These switches allow you to combine sources from several VLANs in one session.
- ☑ Commands for configuring SPAN ports are different on IOS-based switches and CatOS (SET-based) switches.

Configuring RSPAN

- ☑ Cisco Catalyst 6000 series switches provide a Remote SPAN functionality, which allows the collection of monitored traffic across a distributed switch infrastructure.
- ☑ The configuration of RSPAN sessions on switches where traffic is collected (source switches) differs from that where traffic is fed into an IDS sensor (destination switches).
- ☑ To configure a RSPAN session, create a special RSPAN VLAN, and set this VLAN as the SPAN destination on source switches, while using the same VLAN as a source for a SPAN session on a destination switch.
- ☑ Switches on the route between source and destination switches do not have to be high-end switches. Only source and destination switches have to be Catalyst 6000.

Configuring VACLs

- ☑ VACLs are used on Catalyst 6000 for controlling the redirection of traffic within VLANs.
- ☑ VLAN access-lists are enforced in hardware and do not produce any overhead.
- ☑ It is possible to use VACLs to capture traffic permitted by them.
- ☑ Captured traffic then can be redirected to a specific port or IDS module.
- ☑ VACLs are useful for determining “interesting” traffic to monitor when traffic volume is high.

Using Network Taps

- ☑ Network taps are passive devices which split a monitored link into two data streams that are copied to a tap’s output ports.
- ☑ A tap does not disrupt the traffic flow of the link it is applied to, and in the case of power failure, it continues to pass traffic through itself.
- ☑ When traffic from one splitting tap or several taps is to be monitored by one IDS sensor, an aggregation switch is needed to collect this traffic and output it to one link.
- ☑ Bigger taps (or tap panels) have the reverse feature—they can split traffic from a high-speed link into several slower data streams, so that an array of IDS sensors can be attached to them.

Using Advanced Capture Methods

- ☑ Several approaches can be taken in complex environments, with Catalyst 6000 providing the most flexible options for traffic capture.
- ☑ SPAN-based configurations are easier to create in both external and internal IDS modules.
- ☑ VACL-based capture is useful for the granular selection of traffic, mainly in internal IDSMs.
- ☑ The Additional filtering of traffic can be done by clearing unwanted VLANs off the interswitch trunks or internal IDSM’s monitoring ports.

Dealing with Encrypted Traffic and IPv6

- ☑ Encrypted traffic, by design, cannot be understood by an IDS.
- ☑ Tools that use traffic encryption include various VPNs, SSH connections, and SSL for HTTPS servers.
- ☑ The usual workaround to let IDS understand this kind of traffic is to arrange for an IDS to see the traffic in its unencrypted form—for example, by capturing traffic behind a VPN gateway, or on a link between an SSL accelerator and the HTTP Web server.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: How many sessions can be running on a 2900/3500 switch at a time?

A: The number of SPAN sessions on a given switch depends only on the number of free ports that can be used as a SPAN destination. There is no internal limit for SPAN sessions.

Q: Does the Catalyst 2948G-L3 support SPAN?

A: This model is actually a router, not a switch. Thus, it does not support port monitoring.

Q: How do you configure SPAN ports on Catalyst 8540?

A: These switches contain SPAN features under the name of “port snooping.” Port snooping is configured as part of the interface configuration on the destination port, for example:

```
8500R# configure terminal
8500R(config)# interface fastethernet 1/0/1
8500R(config-if)# shutdown
8500R(config-if)# snoop interface fastethernet 0/0/1 direction both
8500R(config-if)# no shutdown
```

This will configure port 1/0/1 to monitor traffic on port 0/0/1 in both directions.

Q: Which cables—straight or crossover—are needed to connect a network tap?

A: This depends on the model, but generally to install the tap you need to unplug the link which you are going to monitor, plug this cable into one port of the tap, then use a straight cable to connect the other tap port to the link's destination (where the disconnected cable was plugged in before). Tap ports A and B then are connected to the aggregation switch using crossover cables.

Q: I have an asymmetric routing. Can I still monitor traffic with Cisco IDS?

A: Yes, to some extent. You can try to collect traffic from both egress and ingress routes and feed it into the same IDS sensor, using either taps or (R)SPAN sessions. It is recommended to turn off signatures which require keeping state of TCP connection and TCP stream reassembly will not work properly.

Q: Why is it not possible to capture corrupted packets using SPAN ports?

A: When a frame enters the switch, it is first analyzed for consistency and error-checked. If the frame is not well-formed, it is dropped and not forwarded anywhere, including the SPAN port. This should not be a problem for IDS, because it deals with the information at the network level and up. Should you want all the frames, you need to use a network tap or a hub.

Q: Are there any restrictions on the number of VLANs that an IDSM blade can span?

A: No, this number is virtually unlimited

Q: Which type of VLAN tagging does IDSM blade support?

A: IDSM-2 supports both 802.1q and ISL tagging headers.

Q: What other possibilities are there for traffic capturing and balancing?

A: There are several enterprise-level IDS load-balancing systems, which can distribute captured traffic among many connected IDS systems while preserving the state of connections. This means that all traffic from one session (that is, the “flow”) is sent to the same IDS engine. One of the most popular in this class is the AppSwitch family from Top Layer Networks (www.toplayer.com).

Cisco Enterprise IDS Management

Solutions in this Chapter:

- Understanding the Cisco IDS Management Center
 - Installing the Cisco IDS Management Center
 - Setting Up Sensors and Sensor Groups
 - Configuring Signatures and Alarms
 - Configuring Reports
 - Administering the Cisco IDS MC Server
-
- ☑ Summary
 - ☑ Solutions Fast Track
 - ☑ Frequently Asked Questions

Introduction

Successful attacks against enterprise networks typically require a substantial effort on the part of the attacker. Many large networks that realize they have been compromised only do so after discovering a discrepancy in activity or the log files traversing their network. Once the compromise is known, the network staff may backtrack and identify all of the activity that occurred *prior* to the compromise...or they may not. Attacks typically are characterized by three phases of activity:

- Reconnaissance
- Probing
- Exploitation

Reconnaissance involves identifying network address ranges, telephone numbers, performing DNS lookups (both forward and reverse), as well as *whois* searches to identify potential names and accounts to try on various target systems. Probing involves ping sweeps to identify potential targets as well as port scans to identify services active on the target systems. Finally, exploitation of a vulnerability (whether it be a buffer overflow in a running service or access due to poor password selections) is the culmination of an attack to gain access to the target network.

The probing and exploitation phases require the use of active tools to identify available services and potential exploit targets. It is this activity that intrusion detection systems (IDSs) are designed to identify. By monitoring traffic on the network and inspecting and analyzing packets, the IDS is able to determine if a network is under attack. If an attack is identified by the IDS, it can issue alerts to network and security operations personnel so they can respond appropriately to protect vital corporate assets. Additionally, many modern IDSs can execute response measures on their own accord, thus terminating the attacker's connection.

There are significant differences between managing a small handful of IDS sensors (on the order of one, two, or three sensors) and handling an enterprise-wide deployment of sensors. Tuning a single sensor to the traffic on a particular LAN may require one or more days simply for the actual tuning of IDS signatures. Once that has been completed, the sensor must be monitored for false positives and for any additional signature tuning required. This can take on the order of a week or more for a single sensor. When new signature packs are released containing additional attack signatures, they must be deployed and tuned as well. Clearly, once the number of sensors goes beyond a small handful, the administrative effort of configuring, monitoring, and updating sensors becomes a significant

burden. By using a tool that provides for managing all sensors through a single interface, the burden is dramatically reduced. This is where CiscoWorks2000 and, in particular, the IDS Management Console (MC) are meant to provide the greatest benefit. Scalable management of IDS sensors is needed to meet the needs of an enterprise network. The Cisco Intrusion Detection System Management Center is designed to provide the centralized sensor management required to protect large enterprise networks.

Understanding the Cisco IDS Management Center

The Cisco IDS Management Center serves four primary functions:

- It logs audit records pertaining to the intrusion detection system .
- It notifies IDS personnel when internal event thresholds are reached.
- It manages and distributes configurations to the sensors.
- It manages and distributes signatures to the sensors.

IDS MC and Security Monitor

Closely related to the Cisco IDS MC is the Cisco Monitoring Center for Security, also known as the Security Monitor. Although the Security Monitor is a separate and optional product, it is often packaged with the IDS MC. While the Security Monitor's primary purpose is to receive alarms from the Sensors, the IDS MC's primary purpose is to administer and manage the sensors.

The Security Monitor provides the following functions:

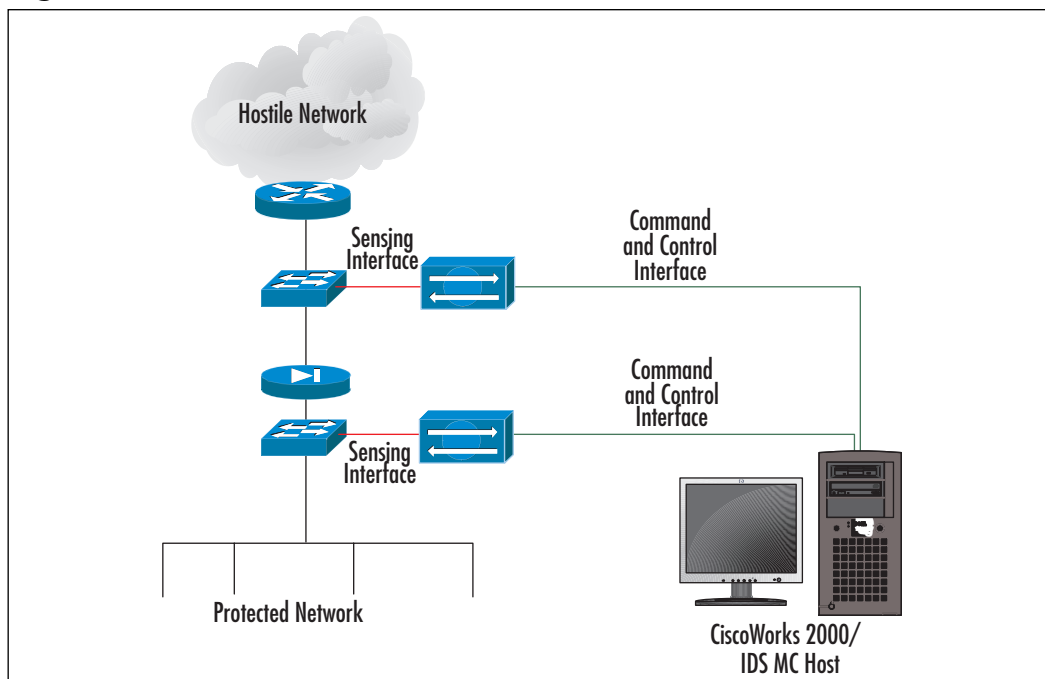
- Event collection
- Event rules and notifications
- Real-time event viewing
- Reporting (scheduled and on-demand).

The Event Viewer of the Security Monitor is used for the real-time display of alarms generated by the IDS sensors. While the Security Monitor may be installed on the same host platform as the IDS MC, often it is installed on a separate host platform for increased performance.

The IDS MC and Sensors

The Cisco IDS Management Center can manage up to approximately 300 sensors. In the example deployment shown in Figure 10.1, the sensor is deployed on the network perimeter or demilitarized zone (DMZ). Inside the protected network is a management host with the IDS MC installed.

Figure 10.1 The IDS MC and Sensor



The sensor monitors traffic inside the DMZ between the inner and outer firewall routers. The sensor has two interfaces: a control interface that is connected to the internal network and a monitoring interface connected to the DMZ network. The control interface provides for management and configuration of the sensor. The monitoring interface, operating in promiscuous mode, passively listens on the DMZ segment. When the sensor detects suspicious network traffic on its monitoring interface, it will send an alarm or event to the Security Monitor via the control interface. Through this same control interface, the IDS Management Center manages the sensor and updates its software versions and signature releases. The sensor uses the control interface to enable blocks or shuns in routers or PIX firewalls. When the sensor uses a TCP RST (reset) as a countermeasure against an attack it sends the TCP RST packets out through the monitoring interface.

IDS MC and Signatures

IDS sensor signatures are the representations of patterns that have certain characteristics of various attacks and other activities attackers may use against a network. The patterns or signatures will be used by the Cisco IDS sensors to detect malicious traffic and act on it. Upon detection of a suspected attack or reconnaissance, the IDS sensor can send an alarm to the Security Monitor or attempt to intervene through the use of shunning, blocking, or TCP resets (RSTs). The IDS MC provides many administrative services with regards to the maintenance of signatures. The MC can be used to enable or disable various signatures based on the administrator's determination of whether they are relevant to the network being monitored by a given sensor. Additionally, the IDS MC provides for the capability to define custom signatures that may not be part of the normal signature pack distributed in CIDS software or signature updates. This capability allows security staff to add to the sensor signature database. Managing, updating, and distributing these signatures are key administrative functions of the IDS Management Center.

IDS MC and Security Policy

From an enterprise perspective, it is important to note that sensor and signature management are merely tools used to implement your Corporate Security Policy. This policy will determine how you deploy your sensors and what signatures you will need.

Designing & Planning...

Cisco Security Wheel

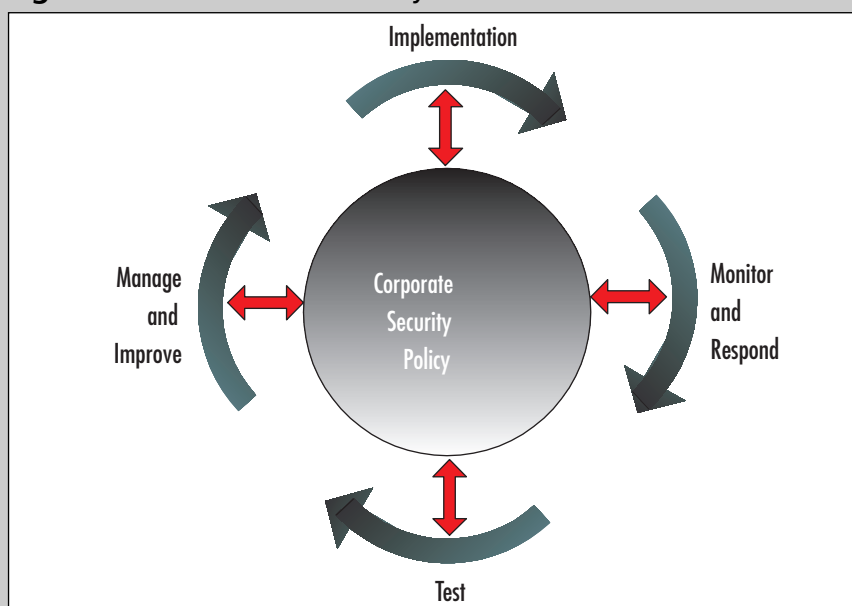
Network security methodology has become increasingly important in driving the overall security of a network. The "old-world" enterprise network security philosophy calls for the development of a security policy first and only then are security products deployed. Once the network is secured, it may be inspected once in a while to identify any potential issues. Any security incidents are then handled on a case-by-case basis.

The Security Wheel is a concept whereby the corporate security policy forms the hub around which all network security practices are based. This methodology evolved as an alternative to the traditional

Continued

approach to network security. As in the “old-world” approach, the security policy is developed first and then the network is secured according to the policy documents. Also, as in the “old-world” approach, the network is monitored for events and any incidents are handled appropriately. However, where the Security Wheel goes further than the traditional “old-world” approach is that the Security Wheel calls for the testing of network security and the results of that testing are then fed back into the security process to manage and improve the state of the network’s security posture. (This concept is shown in Figure 10.2.)

Figure 10.2 The Cisco Security Wheel



The security policy must clearly state the organization’s stance and objectives with regards to security issues. Typically, a security policy is not a single document but a group of documents that provide a high-level overview of security implementation in the network. The policy should document resources to protect and identify the network infrastructure and architecture in general. Finally, the security policy should clearly identify any critical resources that require additional protection. Intrusion detection can be seen as an *extension* of the network security

policy. In many respects, IDS can be considered the *enforcement* of that policy because it provides a continual audit of the network traffic. An in-depth discussion of the development of a security policy is beyond the scope of this chapter as well as this book. For a more detailed discussion of security policies and how to develop them, please refer to the bibliography at the end of the chapter.

Installing the Cisco IDS Management Center

The Cisco IDS MC is a component of the VPN/Security Management Solution (VMS) that, in turn, is part of the CiscoWorks2000 software package. The VMS software suite includes additional components such as CiscoWorks2000 Common Services, which provides for user roles and MC access privileges to be defined as well as for data storage. Additionally, it offers data storage for CiscoWorks client applications that use its services. Other components of VMS include the PIX Management Center, VPN Router Management Center, VPN Monitor, the Cisco Security Agent Management Center, and the Security Monitor. This discussion will focus primarily on the installation of the IDS MC; however, it is often helpful to understand the combined installation requirements of the required VMS bundle.

Server Hardware Requirements

CiscoWorks2000 and the VMS bundle can be installed and operated on either a Windows 2000 Server platform or a Sun Solaris platform. The hardware requirements for CiscoWorks2000 and VMS are specified in Table 10.1.

Table 10.1 Server Hardware Requirements

Component	Minimum Requirement	
Hardware	IBM PC Compatible with 1 GHz or faster Pentium CPU	Sun UltraSPARC 60 MP with 440 MHz or faster processor or Sun UltraSPARC III system (i.e., Sun Blade 2000 or Sun Fire 280R)
Operating System	Windows 2000 Professional, Server, and Advanced Server (Service Pack 3)	Solaris 8 with the following patches: 108528-13 108827-15 108528-13 108827-15

Continued

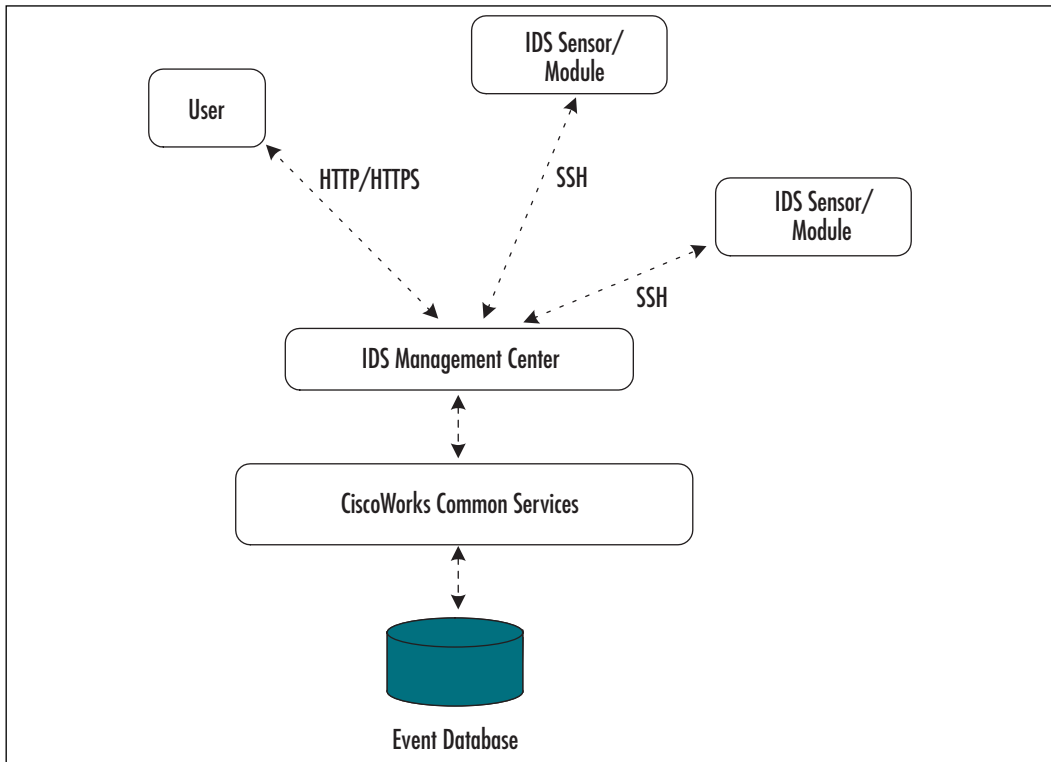
Table 10.1 Server Hardware Requirements

Component	Minimum Requirement
Additional Software	Microsoft ODBC Driver Manager 3.510 or later
File System	NTFS
Memory	1GB (minimum) experience shows 1.5GB to 2GB is more practical
Virtual Memory	1GB (minimum)
Hard Drive Space	9GB (minimum); this will increase depending on what reporting and logging is enabled. Be sure to monitor the log file size.
Java Version	Sun Java plug-in 1.3.1-b24

The VMS product suite should *not* be installed on a Microsoft Windows Server system that is a domain controller or a terminal server. The remainder of this chapter will focus on the installation of the IDS MC on a Windows 2000 system. For additional information regarding the installation of the IDS MC on a Solaris 8 server, please refer to Cisco's web site (www.cisco.com).

CiscoWorks Architecture Overview

The IDS MC architecture is shown in Figure 10.3. The MC itself relies upon the services provided by the CiscoWorks Common Services software. The Common Services component provides a comparable environment for all of the MCs. Some of these services include data storage and management, session management, a web interface, and user authentication and permission management. Before installing the Cisco IDS Management Center, it is important to understand related software that may be prerequisites for successful installation.

Figure 10.3 The IDS MC Architecture

The IDS MC provides a Web-based interface for managing and configuring Cisco IDS sensor appliances and the IDS module for the Catalyst chassis. The MC is built on top of the CiscoWorks framework, allowing it to leverage the ability to define user roles. These roles provide for the definition of user management privileges, including the ability to generate as well as deploy IDS configurations. The IDS MC requires the CiscoWorks Common Services component to provide the necessary base components, software libraries and other software packages. The CiscoWorks Common Services is comprised of the following components:

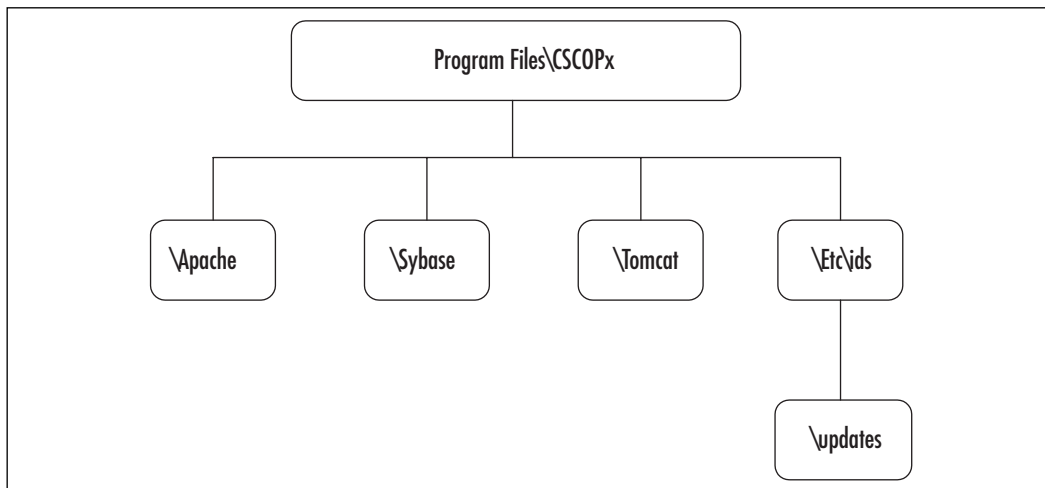
- **Data Storage and Management** The Common Services data store is provided by a Sybase SQL Anytime database. Data backup, and repair and restoration capabilities of the database, are also provided by the Common Services package.

- **Session Management** Allows multiple users to connect to the MC and perform configuration and management tasks without data corruption or loss.
- **User Management** Provides for authentication and authorization.
- **Web Interface** Provided by an Apache Web server allowing for connections to the MC system through a Web browser. Access to the CiscoWorks2000 server is done on a secure encrypted channel over TCP port 1741. Once the user has authenticated to the CiscoWorks2000 server, communication with the IDS MC is conducted over TCP port 443.

IDS MC Installation

The IDS MC software installs its components into the same directory as the CiscoWorks Common Services software components. This is typically in the directory: Program Files\CSCOPx. The directory structure is shown in Figure 10.4.

Figure 10.4 The IDS MC Directory Tree Structure



Cisco chose to use an open source program called Apache for the built-in Web server for CiscoWorks. The subdirectory \Apache is where the Apache Web Server is installed and from where Apache serves the Web pages that are displayed when using the IDS MC. The Sybase subdirectory is where the Sybase SQL Anytime database is installed as well as where all data from the IDS appliances and the IDSM sensors is stored. The Tomcat subdirectory is where the Tomcat

application server is installed. This server provides servlets to the IDS MC from the Common Services. The Etc\ids directory is where the IDS MC is actually stored. The updates subdirectory is where the signature update packs are stored for the MC to push out to the sensors or to the MC itself.

IDS MC Processes

The IDS MC is composed of the following system processes:

- IDS_Analyzer
- IDS_Backup
- IDS_DbAdminAnalyzer
- IDS_DeployDaemon
- IDS_Notifier
- IDS_Receiver
- IDS_ReportScheduler

The IDS_Analyzer defines event rules and requests user-specified notifications when appropriate. The IDS_Backup process provides for database backup and restore capabilities to the MC. The DbAdminAnalyzer applies various active database rules to the current state of the server. The IDS_DeployDemon provides for the deployment of configurations to IDS sensors. IDS_Notifier retrieves and performs MC subsystem notification requests. The IDS_Receiver receives alarms and syslog events from IDS appliance sensors and IDS modules for the Catalyst chassis and stores them in the Sybase database. As its name implies, the IDS_ReportScheduler handles the generation of reports in the MC.

VMS Component Compatibility

Most VMS components require CiscoWorks2000 Common Services to be installed on the same server. While it may seem more efficient to combine some of these VMS components on one server, this cannot always be done due to compatibility and performance reasons.

For example, both the IDS Management Center and the Security Monitor are delivered on the same CD-ROM package. Both require CiscoWorks 2000 Common Services. The IDS MC and the Security Monitor may be installed together or separately on different host servers. However, for optimal performance, separate installation of these two applications on different host servers is recommended.

Other VMS components that are not compatible on the same server as the IDS Management Center include the Cisco Secure Policy Manager (CSPM). To attempt this may result in the installation of a second instance of the post office process on the host server.

Client Installation Requirements

Accessing CiscoWorks2000 and IDS Management Center is accomplished through a Web interface. This allows clients to access the IDS Management Center by using a browser. The minimum system requirements for a client are specified in Table 10.2.

Table 10.2 Client System Requirements

Component	Minimum Requirement	
Hardware	IBM PC Compatible with minimum 300MHzPentium Processor	Sun Ultra 10 or Sun SPARCstation with a 333MHz processor
Software	Windows 2000 Server, or Professional Edition with Service Pack 3 Windows XP Professional, Service Pack 1 with Microsoft Virtual Machine	Solaris 8
Memory	256MB	
Virtual Memory	400MB	512MB
Browser	Microsoft Internet Explorer 6.0, Service Pack 1 for Windows operating systems with Microsoft Virtual Machine. Netscape Navigator 4.79 on any of the following: Windows 2000 Server Professional Edition with Service Pack 3 Windows XP Professional, Service Pack	Netscape Navigator 4.76 for Solaris

Installation Steps

Once the prerequisite components have been verified, the basic installation steps for the IDS MC are as follows:

1. Log in as the local administrator.
2. Insert the CD-ROM containing the “Monitoring Center for Security and Management Center for IDS Sensors” program. If the installation program does not start, select **Run** from the **Start** button. Browse for the **setup** program on the CD-ROM drive. Open the **Setup** program and click **OK**. If the installation program does start, click **Install** on the Installer page. Click **Next**.
3. The Software License Agreement page appears. Be sure you understand the Agreement, then click **Yes** to accept its terms..
4. The installation now begins. To install both the IDS MC and the Security Monitor, click the **Typical Installation** radio button. To install only the IDS MC or the Security Monitor, click the **Custom Installation** button, and select either the **IDS MC only** radio button or the **Security Monitor only** radio button. Click **Next**.
5. The System Requirements page appears. Verify that the system meets the minimum disk space and memory requirements. Click **Next**.
6. The Verification page appears. Verify the selected components. Click **Next**.
7. The Select Database Location page appears. By default, the IDS database is located in the directory where CiscoWorks Common Services is installed. To specify a different directory for the database, enter a directory path in the **Database File Location** field provided. Click **Next**.
8. The “Select Database Password” page appears. Enter the database password in both the **Password** and **Confirm Password** fields. Click **Next**.
9. Either the Select CW2000 Syslog Port page or the Restart page appears.
 - If the Security Monitor is installed, the **Select CW2000 Syslog Port** page appears. Specify the UDP port to be used by CiscoWorks. The default value of 52514 is recommended. Click **Next**. The **Configure Communications Properties** page appears. Enter the host ID, organization ID, IP address, hostname, and organization name into the appropriate fields. Click **Next**.

- If only the IDS MC is installed, the Restart page appears. On the **Restart** page, select **Yes** to restart the computer. Choose **No** to restart the computer at a later time. Select **Finish**. The computer must be restarted before it is possible to use the IDS MC or Security Monitor.

Getting Started

Access to the IDS MC is provided through the Apache Web server on the CiscoWorks2000 host. This provides for easy access through either a web browser meeting the requirements defined in Table 10.2. The CiscoWorks 2000 Apache Web server listens for incoming connections on TCP port 1741 of the CiscoWorks2000 host. To access the CiscoWorks2000 system, enter one of the following URLs:

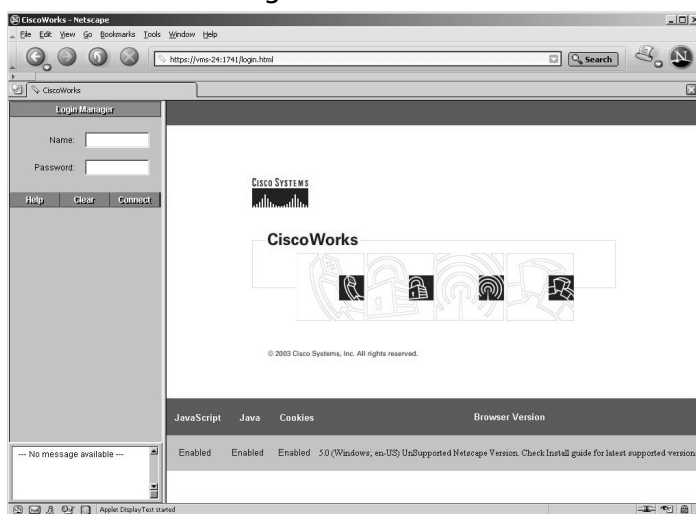
- **http://127.0.0.1:1741** Use this if the IDS MC server is the local machine
- **http://A.B.C.D:1741** Use this if A.B.C.D is the IP address of the IDS MC server.

Log into the CiscoWorks2000 Server Desktop, as shown in Figure 10.5. The default login name and password for the CiscoWorks2000 system include the following:

- Default login name: **admin**
- Default password: **admin**

After entering the login Name and Password, click **Connect**. Note, it is *highly recommended* that the password of the **admin** account be changed from the default value of **admin** immediately upon installation and configuration of CiscoWorks2000 in order to prevent unauthorized users from gaining administrative access to the CiscoWorks2000 software.

Figure 10.5 The CiscoWorks Login Screen



Authorization Roles

CiscoWorks provides for five different default types of accounts or authorization roles that can be created for IDS MC users. These authorization roles and their respective privileges are summarized in Table 10.3.

Table 10.3 Authorization Roles


Authorization Role	Privileges
	<p>View</p> <p>Create/Modify/Delete</p>
Help Desk	<p>View reports and alarms</p> <p>Cannot delete reports or alarms. Cannot generate reports.</p>
Approver	<p>View reports and alarms</p> <p>Approve configurations. Cannot delete reports or alarms. Cannot generate reports.</p>
Network Operator	<p>View reports and alarms</p> <p>Deploy configurations. Delete reports and alarms. Generate reports.</p>
Network Administrator	<p>View reports and alarms</p> <p>Edit devices and device groups.</p>
System Administrator	<p>View reports and alarms.</p> <p>Edit devices and device groups. Delete reports and alarms. Generate reports. Import lists (files) and notification scripts.</p>

Creating accounts with different authorization roles allows an administrator to delegate different responsibilities to different IDS Management Center users. Each account holder or user can be given the authority needed to carry out his responsibilities.

Installation Verification

To verify the successful installation of CiscoWorks 2000 and the IDS MC, select the **Server Configuration** entry on the CiscoWorks2000 Server Desktop, as shown in Figure 10.6. Then select **About the Server** and **Applications and Versions**.

Figure 10.6 Server Configuration



Name	Version	Install Date	Installed Patches	State
Apache	1.3.27	7-18-2003 18:49:50	none	ENABLED
Auto Update Server	1.1	7-21-2003 13:55:21	none	ENABLED
Client Application Manager	3.0	7-18-2003 18:49:50	none	ENABLED
CWCS SQL Components	7.1.3	7-18-2003 18:49:50	none	ENABLED
CiscoWorks Common Services	2.2	7-18-2003 18:49:50	none	ENABLED
Cisco Common Services Help	1.1	7-18-2003 18:49:50	none	ENABLED
CWCS Foundation	2.2	7-18-2003 18:49:50	none	ENABLED
CWCS java2 engine	1.2	7-18-2003 18:49:50	none	ENABLED
CWCS Web Desktop	2.2	7-18-2003 18:49:50	none	ENABLED
CWCS Utilities	1.1	7-18-2003 18:49:50	none	ENABLED
Management Center for Cisco Security Agents	4.0	7-30-2003 18:31:21	none	ENABLED
CiscoView	5.5	7-18-2003 18:49:50	none	ENABLED
Database package	4.2	7-18-2003 18:49:50	none	ENABLED
CiscoWorks Process Management package	3.5	7-18-2003 18:49:50	none	ENABLED
CWCS (includes CiscoView) Help	2.2	7-18-2003 18:49:50	none	ENABLED
CWCS Event Distribution System	3.2	7-18-2003 18:49:50	none	ENABLED
Event Services Software	2.0	7-18-2003 18:49:50	none	ENABLED

Verify that the following key CiscoWorks components are installed:

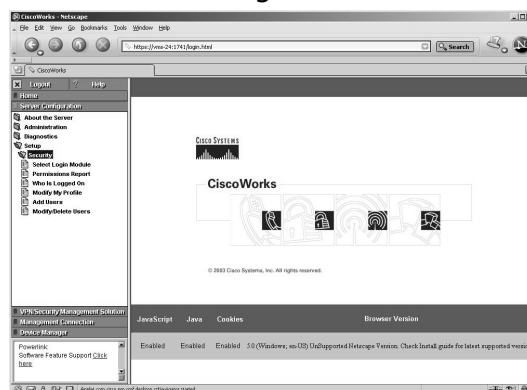
- **Apache** The Apache Web server provides the Web interface used by a client to access the IDS Management Center.
- **CWCS SQL Components** A Sybase SQL server is used to provide the database services required by the IDS Management Center.
- **Cisco Works Common Services (CWCS)** There are a multitude of services provided by CWCS that are required by the IDS Management Center.

Adding Users to CiscoWorks

Adding users to the CiscoWorks system is straightforward. To add a new user to the CiscoWorks2000 system:

1. Open the **Server Configuration** tab in the right side panel of the CiscoWorks interface.
2. Select **Setup** and then **Security**, as shown in Figure 10.7.

Figure 10.7 CiscoWorks Server Configuration Add User



3. Select the **Add Users** option.
4. Enter values for the setting listed in Table 10.4 and shown in Figure 10.8.

Table 10.4 CiscoWorks “Add Users” Information

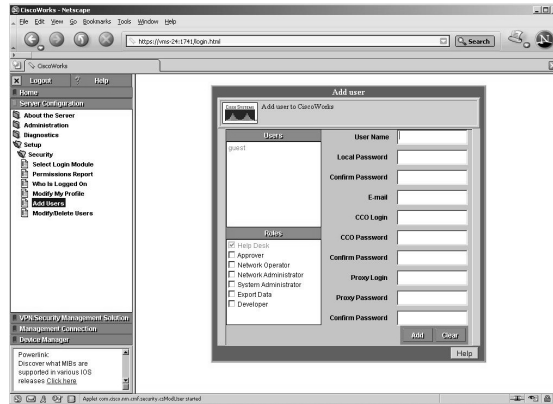
CiscoWorks2000 “Add Users” Setting	Information
Username	Name of new user account to add
Local Password	Account password
Confirm Password	Password confirmation
E-mail	User’s e-mail address (optional)
CCO Login	User’s CCO login account name (optional)
CCO Password (optional)	User’s CCO login account password
Confirm Password	CCO Password confirmation (optional)
Proxy Login	User’s proxy server login name (optional)

Continued

Table 10.4 CiscoWorks “Add Users” Information

CiscoWorks2000 “Add Users” Setting	Information
Proxy Password	User’s proxy server password (optional)
Confirm Password	Proxy password confirmation (optional)

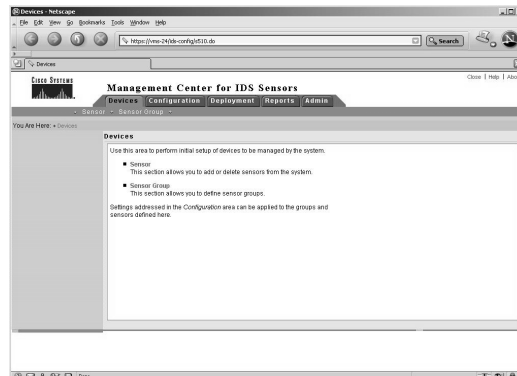
Figure 10.8 The CiscoWorks2000 Add User Web Page



The IDS MC

If the IDS MC installation is successful, an entry for the Management Center will appear. Selecting the **Management Center** entry will prompt the IDS Sensors entry to appear. Selecting the **IDS Sensors** entry brings up the Management Center IDS Sensors interface, shown in Figure 10.9.

Figure 10.9 The Management Center for IDS Sensors Page



The Devices tab of this page allows for the definition of sensor groups as well as the addition or deletion of sensors from the system as described in the next section.

Setting Up Sensors and Sensor Groups

Sensors are the “eyes and ears” of the Cisco IDS Management Center. They are placed strategically at the perimeter of the network and near key resources within the enterprise. Each of the sensors deployed in the network have been configured with a unique IP address. The IDS MC uses this IP address to communicate with the sensor. Once these sensors are deployed and assigned IP addresses, they can be configured and managed from within the MC.

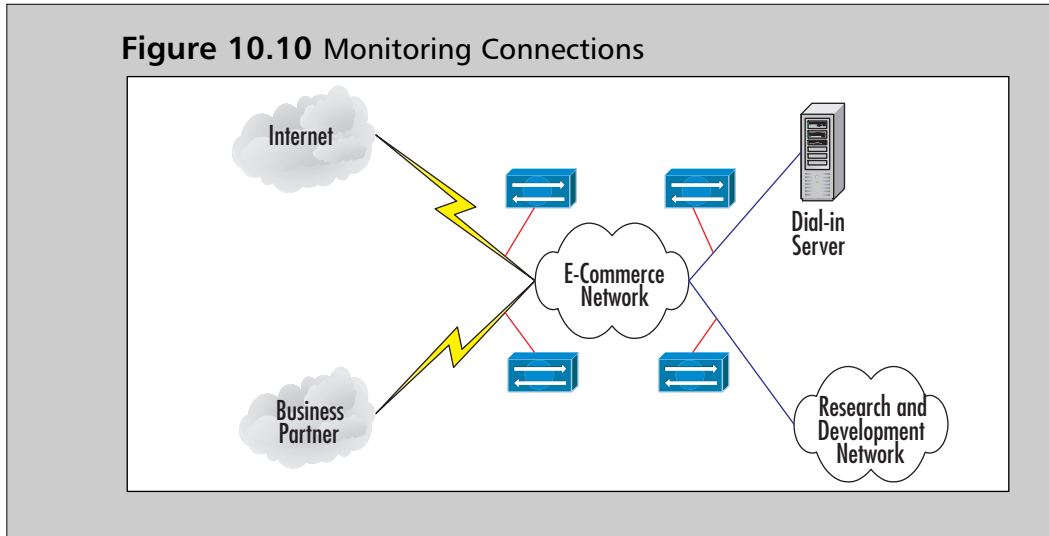
Configuring & Implementing...

Monitoring Connections

A sensor is commonly placed on a connection to monitor traffic between the network to be protected and other networks. In Figure 10.10, a protected enterprise network is comprised of two intranets: the E-Commerce network and the R&D network. Here, sensors have been deployed to monitor four different types of connections. Starting from the upper left, the sensors offer the following protection:

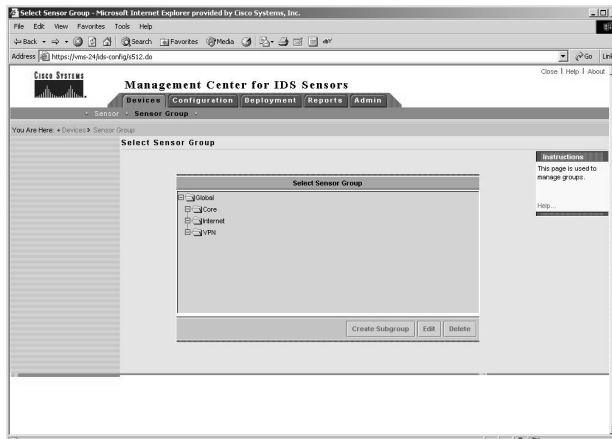
- **Perimeter Protection** The most common deployment for a sensor is to be placed between the network to be protected and the Internet. This is known as perimeter protection.
- **Remote Access Protection** A dial-in server is used only by employees but may still be vulnerable to external attack. A sensor is placed on the interior side of the dial-in server connection.
- **Intranet Protection** While the R&D network is an internal network, it may require a different level of security and hence a sensor is deployed between the two intranets.
- **Extranet Protection** A business partner may have similar network security policies but the level of protection may differ. A sensor is deployed between the two extranets.

Continued

Figure 10.10 Monitoring Connections

The IDS MC Hierarchy

The IDS MC maintains a hierarchy of sensors, sensor groups and sensor sub-groups. Groups provide the capability of managing multiple sensors performing similar functions. Rather than configuring each sensor individually, the IDS MC allows for the configuration of groups of sensors. This dramatically reduces the administrative burden on security personnel. Figure 10.11 illustrates an example of an IDS MC sensor group hierarchy. At the top of the group hierarchy is the Global group. There can be many levels of groups and sensors under the Global group. Each of the lower-level groups, subgroups, and sensors are added manually.

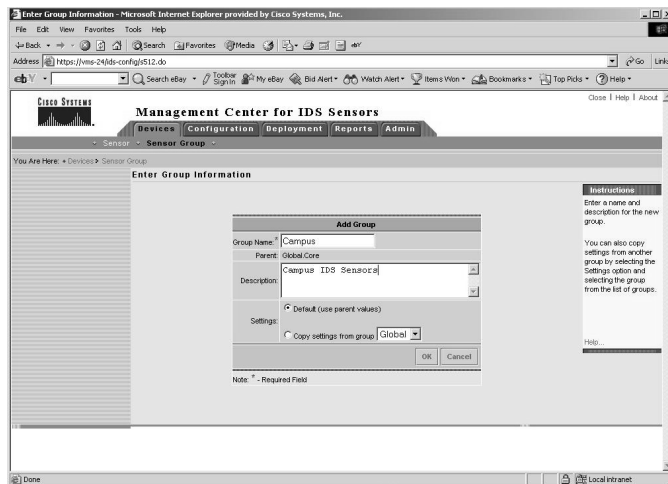
Figure 10.11 The IDS MC Hierarchy

Creating Sensor Subgroups

A sensor subgroup can be added to any group including the Global group. The following steps can be used to create a sensor subgroup:

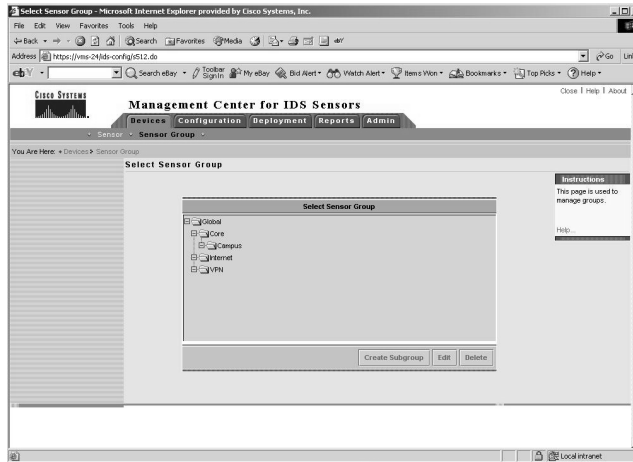
1. From the **Management Center for IDS Sensors** page (Figure 10.7), select the **Devices** tab, then choose **Sensor Group**. The Sensor Group page will appear, as shown in Figure 10.11.
2. The Sensor Group page displays a tree of multiple levels of sensor groups and sensors. At present, there is a Global group as well as three subgroups: Core, Internet, and VPN. Select the name of the group under which the new subgroup will appear. Click the **Create Subgroup** button.
3. The **Add Group** page appears, as shown in Figure 10.12. Enter the new subgroup's name in the **Group Name** field. Describe the new group in the **Description** field. Under settings, select the parent group's settings or copy the settings from a group in the pull-down menu.
4. Click **OK** to create the new subgroup.

Figure 10.12 The Add Group Page



The Sensor Group page reappears, containing the newly created group. In Figure 10.13, this new group is named Campus.

Figure 10.13 The Sensor Group Page with the New Subgroup

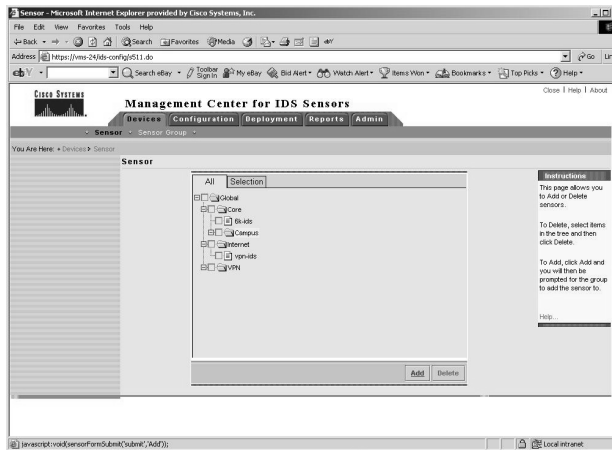


Adding Sensors to a Sensor Group

A sensor can be added to any group including the Global group. To add a sensor to the Global group or a subgroup, use the following procedure:

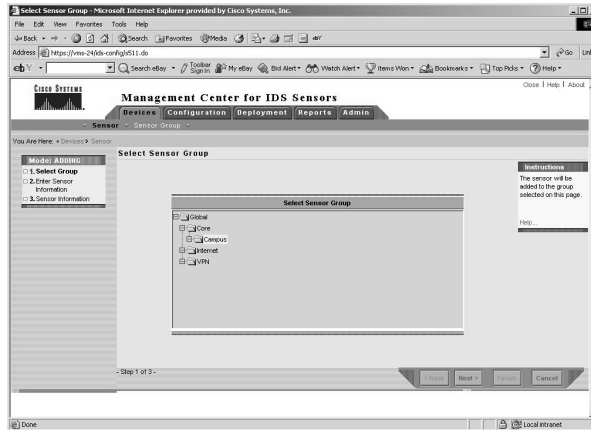
1. From the **Management Center for IDS Sensors** page (Figure 10.9), select the **Devices** tab, then choose **Sensors**.
2. The **Sensor** page will appear as shown in Figure 10.14. Click the **Add** button.

Figure 10.14 The Sensor Page



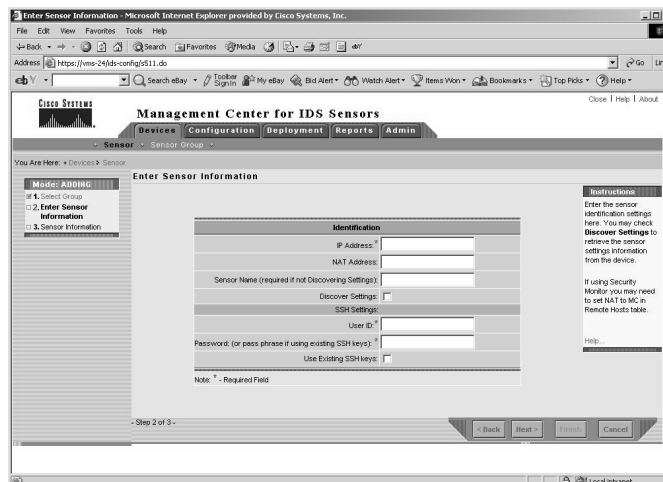
3. The **Select Group** page will appear, as shown in Figure 10.15. Select the Group to add the sensor to and click **Next**.

Figure 10.15 The Select Sensor Group Page



- The **Enter Sensor Information** page appears, as shown in Figure 10.16. Enter the **IP Address** of the sensor, the **NAT Address** of the sensor if one exists, and the **Sensor Name**. To retrieve sensor settings directly from the sensor, select the **Discover Settings** check box. Enter the **User ID** and **Password** for Secure Shell (SSH) communications. For sensor appliances and IDS modules, the default user ID is **cisco**. The default password for the account is **cisco**. It is also possible to authenticate to the IDS sensor using an SSH public/private key pair. To use existing SSH keys, check the **Use Existing SSH keys** check box. However, do not select this option if the sensor is to be used as a master blocking sensor. Once the information has been entered, click **Next** to move on to the final step.

Figure 10.16 The Enter Sensor Information Page



- The Sensor Information page appears, as shown in Figures 10.17 and 10.18. From the **Version** pull-down menu, select the sensor software version installed on the sensor. Enter a text **Comment**. For sensors running the IDS sensor software version 3.x, additional information needs to be entered. This information includes the sensor **Host ID**, which is typically the last octet of the sensor's IP address. Enter the **Org Name** using only lowercase letters. Enter the **Org ID**. The default is 100. Within a **Postoffice** domain, with no sensor or sensor group, the Org ID/Host ID pair must be unique. For Sensor software version 4.x and later, a text comment need only be entered in the **Comment** field. Click **Finish**.

Figure 10.17 The Sensor Information Page for Sensor OS Version 3.x

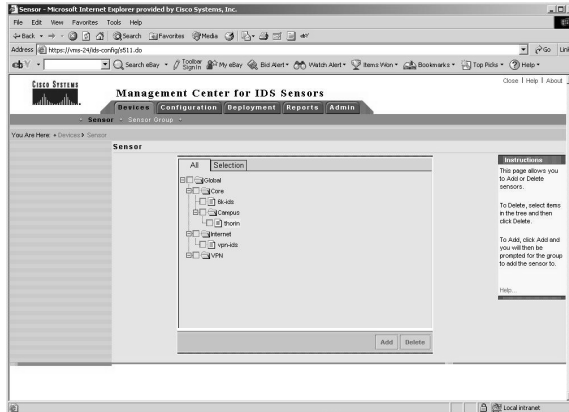
The screenshot shows the 'Sensor Information' page in the Management Center for IDS Sensors. The page is titled 'Sensor Information' and is part of a multi-step process (Step 3 of 3). The 'Version' dropdown menu is set to '3.0(1)S4'. The 'Comment' field is empty. Below the 'Comment' field, there are three input fields for 'Postoffice Settings': 'Host ID' (containing '45'), 'Org Name' (containing ' '), and 'Org ID' (containing '100'). A note below these fields states 'Note: * - Required Field'. The page includes navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'. The breadcrumb trail shows 'You Are Here: Devices > Sensor Group > Sensor'.

Figure 10.18 The Sensor Information Page for Sensor OS Version 4.x

The screenshot shows the 'Sensor Information' page in the Management Center for IDS Sensors for version 4.x. The 'Version' dropdown menu is set to '4.0(2)S43'. The 'Comment' field is empty. The 'Postoffice Settings' section is not visible, indicating that only the 'Comment' field is required for this version. The page includes navigation buttons: 'Back', 'Next', 'Finish', and 'Cancel'. The breadcrumb trail shows 'You Are Here: Devices > Sensor Group > Sensor'.

- The Sensor page reappears, updated with an entry for the new sensor you have added, as shown in Figure 10.19.

Figure 10.19 The Updated Sensor Page

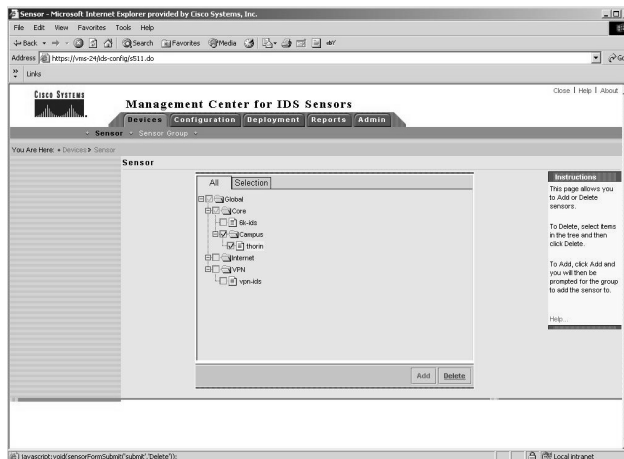


Deleting Sensors from a Sensor Group

A sensor can be deleted from any group including the Global group. Use the following steps to delete a sensor from a subgroup:

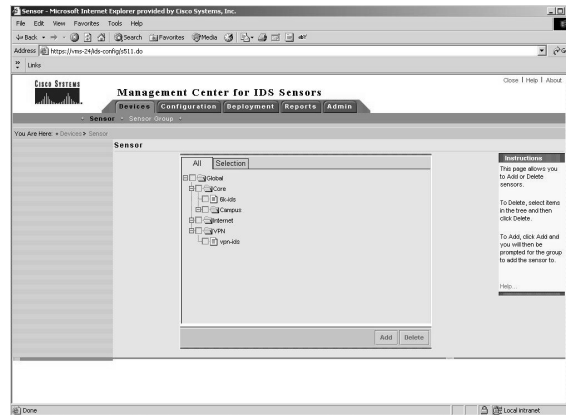
- From the **Management Center for IDS Sensors** page (Figure 10.9), select the **Devices** tab and choose **Sensors**.
- The **Sensor** page appears, as shown in Figure 10.20. Check the box in front of the entry for the sensor to delete. In this case, the sensor to be deleted is call **thorin**. Click the **Delete** button.

Figure 10.20 The Sensor Page



- The Sensor tree page appears, as shown in Figure 10.21. Note that the sensor named thorin has been removed from the tree.

Figure 10.21 The Sensor Tree Page

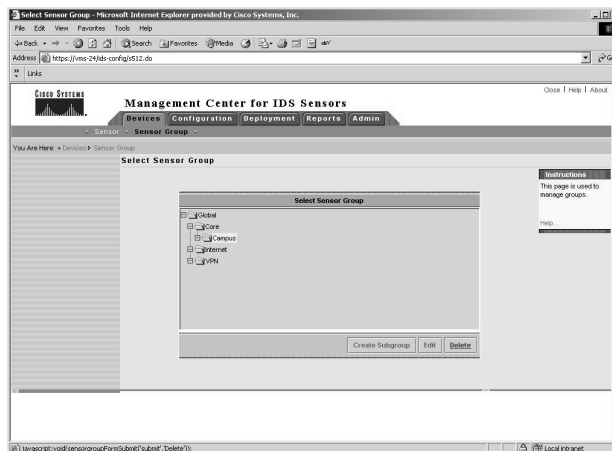


Deleting Sensor Subgroups

As with sensors, sensor subgroups can be deleted from any group including the Global group. Use the following steps to delete a sensor subgroup:

- From the **Management Center for IDS Sensors** page (Figure 10.9), select the **Devices** tab, and choose **Sensor Group**.
- The **Sensor Group** page appears, as shown in Figure 10.22. In the tree, select the subgroup to delete and click the **Delete** button.

Figure 10.22 The Select Sensor Group Page



Configuring Signatures and Alarms

Network intrusions are scans, attacks upon, or misuses of the network resources. To detect network intrusion, the Cisco IDS sensors use a signature-based technology. Every network attack has an order or a pattern to the bytes in the traffic stream between the attacking system and the target. These bytes represent a “fingerprint” or “signature” of the attack. By comparing the pattern of bytes in a given traffic stream between two hosts against a database containing various known signatures for network attacks, the IDS is able to determine when an attack has occurred. Each signature specifies the type of attack the sensor detects and reports. As a sensor scans the network packets, the rules allow it to detect patterns that match a known attack.

The IDS MC allows the operator to specify which signatures should be enabled. Additionally, the response action the IDS sensor initiates, whether it is simply raising an alarm on the Security Monitor console or initiating a TCP RST, is also determined based on what is specified in the signature. Tuning IDS signatures is one of the more important features of the IDS MC. Improperly tuned IDS sensors account for the great majority of false positive alarms (alarms raised by the IDS in response to benign network traffic) and result in potential mistrust of the IDS system by security personnel.

Configuring Signatures

Signatures are divided into six groups:

1. General (embedded)
2. TCP connection
3. UDP connection
4. String-Matching
5. Access Control List (ACL)
6. Custom

To provide an example of how to configure and tune signatures, we will use a general signature for a configuration and tuning exercise.

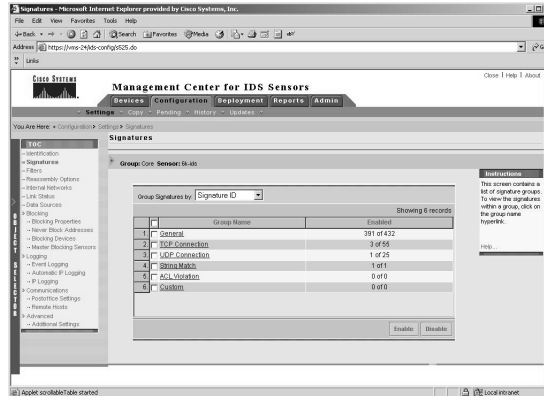
Configuring General Signatures

General signatures are signatures that are embedded in the sensor software itself. IDS end users cannot add or delete general signatures, but the end user can

enable or disable them and configure the response to attacks that fit the general signatures. The following steps can be used to configure a general signature:

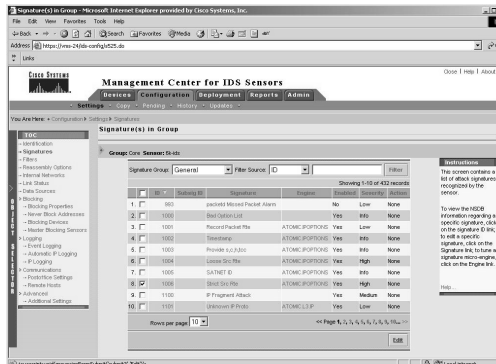
1. From the **Management Center for IDS Sensors** page, select **Configuration | Settings**.
2. A **Table of Contents** page appears. Select the **Object Selector** handle.
3. In the Object Selector, select the sensor containing the general signature to configure. The Object Selector will close and redisplay the Table of Contents.
4. In the **Table of Contents**, select **Signatures | General**. The general Signatures page will appear, as shown in Figure 10.23.

Figure 10.23 The General Signatures Page



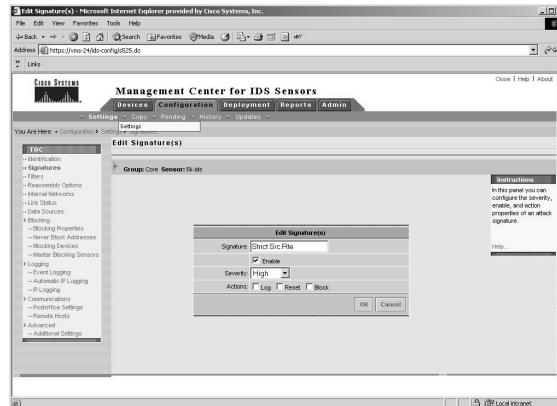
5. Click the link for the signature group to be modified. This results in the display of the Signature(s) in Group page listing all of the signatures within the selected group, as shown in Figure 10.24.

Figure 10.24 The Signature(s) in Group Page



6. Select the signature to configure by checking the corresponding box and clicking **Edit**.
7. The **Edit Signature(s)** window appears (as shown in Figure 10.25) and shows the name of the signature to configure. To enable or disable the signature, check or uncheck the **Enable** box.

Figure 10.25 The Edit Signature(s) Page



Configuring Alarms

The severity of an alarm, as well as the actions to be taken when an event matches a signature, can be specified by editing the signature.

1. To change the severity of an attack that matches this signature, select a **Severity** from the pull-down menu:
 - **Info** Indicates an event that results from normal activity.
 - **Low** Indicates an attack that is mild in severity. The Security Monitor Event Viewer will display this type of attack with a green icon.
 - **Medium** Indicates an attack that is moderately severe. The Security Monitor Event Viewer will display this type of attack with a yellow icon.
 - **High** Indicates an attack that is highly severe. The Security Monitor Event Viewer will display this type of attack with a red icon.
2. Note the options to the right of the **Actions** label. Depending on the signature, you may specify one or more of the following actions to be taken when a signature matches an event:

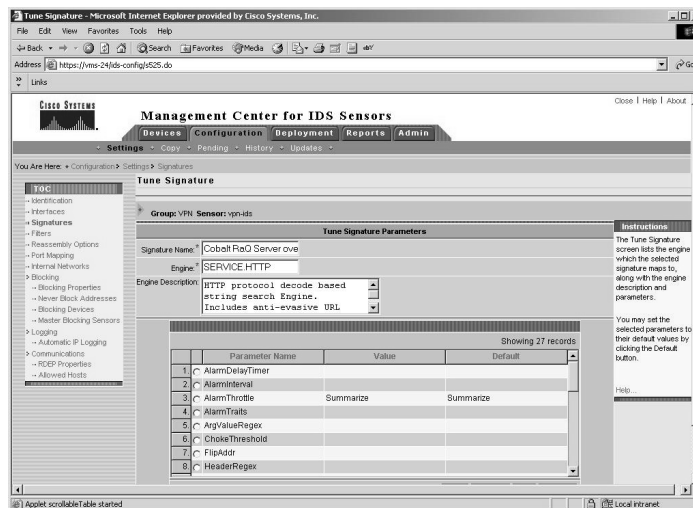
- **Log** Stands for IP Log, and generates an IP session log with information about the attack.
- **Reset** Stands for TCP Reset, and resets the TCP session in which the attack signature was detected.
- **Block** Causes the sensor to issue a command to a PIX firewall or Cisco router. That firewall or router will block packets from the attacking host or network and keep them from entering the protected network.

Tuning General Signatures

Signatures are tuned to minimize false alarms or “false positives.” False positives are alarm indicators of an attack where either benign or standard activity is present. A false positive may result from normal network activity in which a network management station polls or scans network devices to ascertain their status. This polling activity is similar to the scanning employed by hackers against a targeted network. Additionally, a false positive may occur when an attacker attempts to use an exploit against a host whose software is not vulnerable to that exploit (for example, using a Microsoft IIS exploit against an Apache Web server).

To tune a signature, return to the general Signature(s) page shown in Figure 10.23. For the signature to be tuned, select the signature link in the **Engine** column of the table. This brings up the **Tune Signature** page, as shown in Figure 10.26.

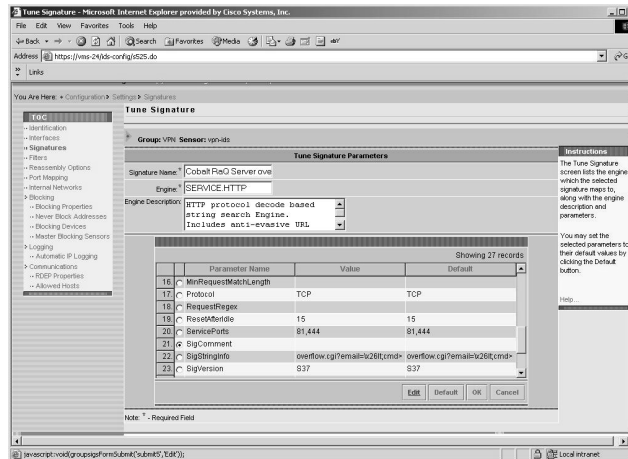
Figure 10.26 The Tune Signature Page



There are three columns in the Tune Signature Parameters table: Parameter Name, Value, and Default. Each one can be modified to an appropriate, desired value. Use the following procedure to tune a given parameter in a procedure:

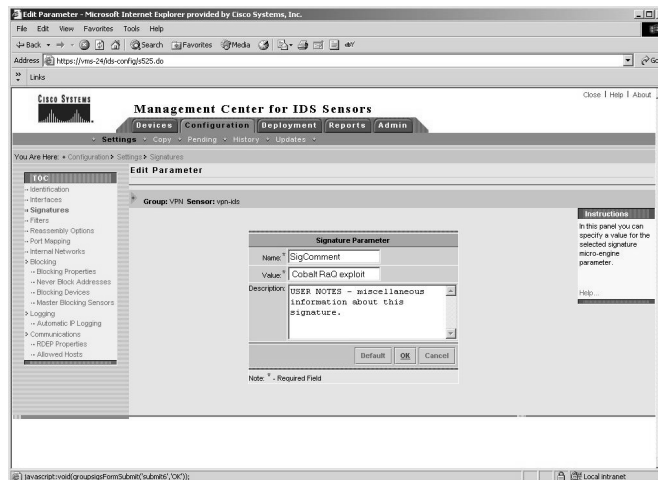
1. Select the radio button for the parameter to be tuned in the **Parameter Name** column, then select **Edit**, as shown in Figure 10.27.

Figure 10.27 The Tune Signature Parameters Page



2. Enter a value for the parameter in the **Value** field, as shown in Figure 10.28.
3. Enter an optional description for the signature parameter in the **Description** field.

Figure 10.28 The Signature Parameter Page



- To accept the changes, click the **OK** button. The Tune Signature page will redisplay.

On the **Tune Signature** page, click **OK** to accept the changes. The general Signature(s) page will reappear.

How to Generate, Approve, and Deploy IDS Sensor Configuration Files

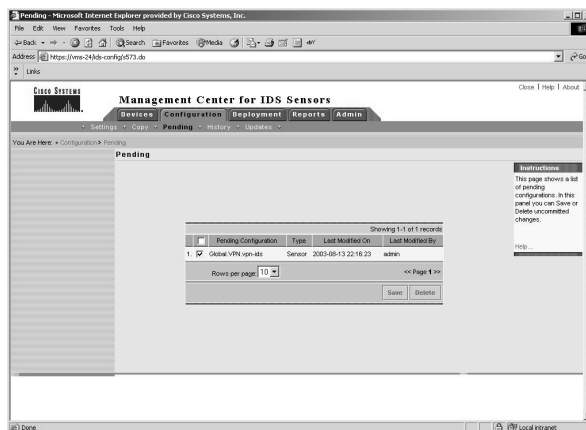
The previous section, “Configuring Signatures and Alarms,” covered how to select the proper values for the sensor settings and signature settings. The next step in using the IDS MC is to review and generate the configuration files that contain those settings. Once the configuration files for the IDS sensors have been generated, they need to be reviewed by the appropriate personnel and then deployed to the sensors. This section, covers how to review and generate the IDS sensor configuration files as well as how to approve and deploy the configuration files to the sensors.

Reviewing Configuration Files

Changes to file settings are placed in a pending status before they are committed to the IDS Database. The following steps can be used to review the pending changes and commit them to the database:

- From the **Management Center of IDS Sensors** page in Figure 10.9, select **Configuration | Pending**. The Pending configurations page appears, as shown in Figure 10.29.

Figure 10.29 The Pending Configurations Page



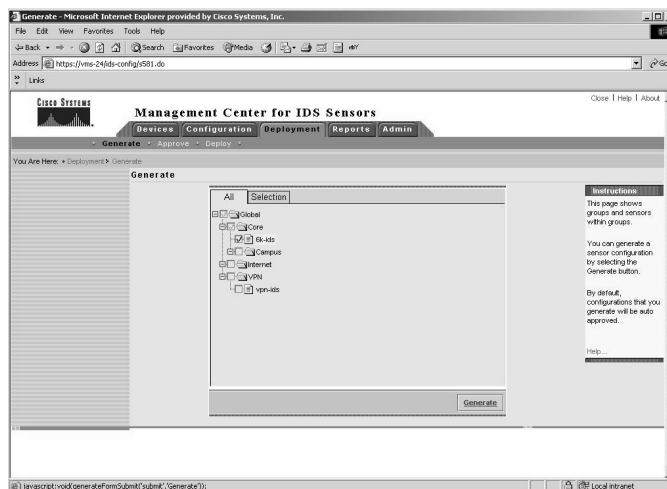
2. Check the box associated with the sensor whose configuration is to be saved in the IDS Database.
3. Click **Save** to save the configuration in the IDS Database or click **Delete** to delete it.

Generating Configuration Files

To generate a configuration file is to take a file of sensor configuration settings that is stored in the IDS Database and prepare it for deployment to the sensor itself. Generating a configuration file starts with the Management Center of IDS Sensors page, shown in Figure 10.9.

1. From the **Management Center of IDS Sensors** page shown in Figure 10.9, select **Deployment | Generate**.
2. The Generate page appears, as shown in Figure 10.30. To generate a configuration file for a specific sensor, select that sensor from the tree and click **Generate**. Once the configuration file has been generated, it is now ready for the approval process.

Figure 10.30 The Generate Page



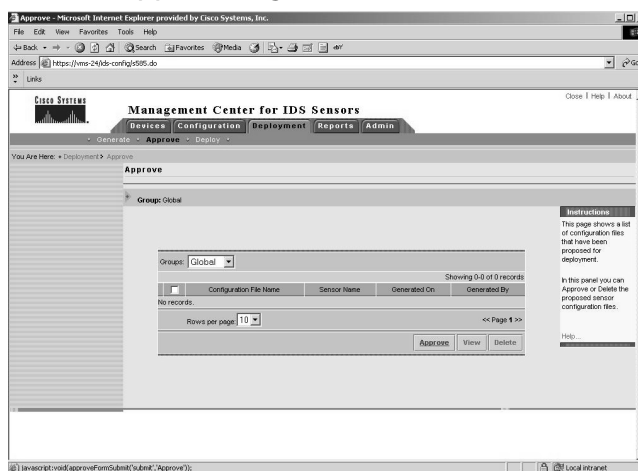
Approving Configuration Files

CiscoWorks2000 allows for a separation of duties among user roles. This makes it possible to assign the approval of configuration files and other actions to a specific account. By separating various functions among different accounts, CiscoWorks2000 allows for a “checks-and-balance” system whereby administrators

are able to verify configurations for network equipment. This is especially important in IDS because an error in the configuration file for an IDS sensor may result in the sensor not identifying an attack.

1. From the **Management Center of IDS Sensors** page in Figure 10.9, select **Deployment | Approve**.
2. The **Approve** page appears, as shown in Figure 10.31. To approve the configuration generated, check the corresponding box and click the **Approve** button.

Figure 10.31 The Approve Page



3. To view a selected IDS configuration file before approving it, check the corresponding box to the right of the configuration file name and click the **View** button.
4. To delete an IDS configuration without approving it, check the corresponding box to the right of the configuration file name and select the **Delete** button.

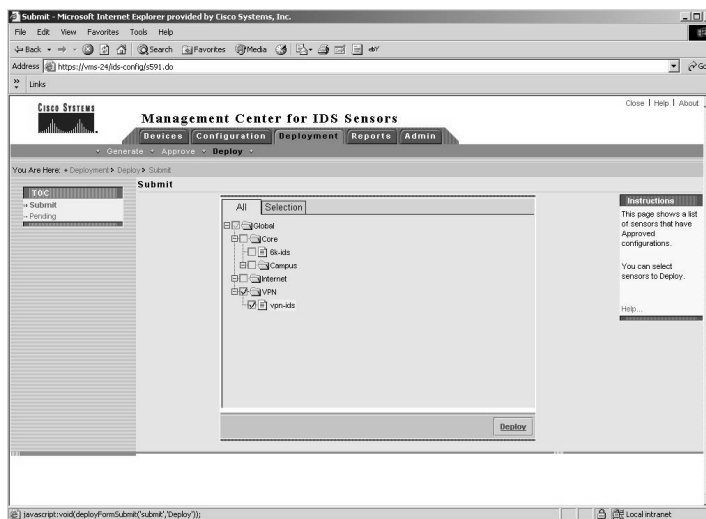
Deploying Configuration Files

To deploy a configuration file is to send an approved file of sensor configuration settings from the IDS Database to the sensor itself. Use the following steps to deploy a configuration file:

1. From the **Management Center for IDS Sensors** page, select **Deployment | Deploy**. Select **Submit** from the Table of Contents.

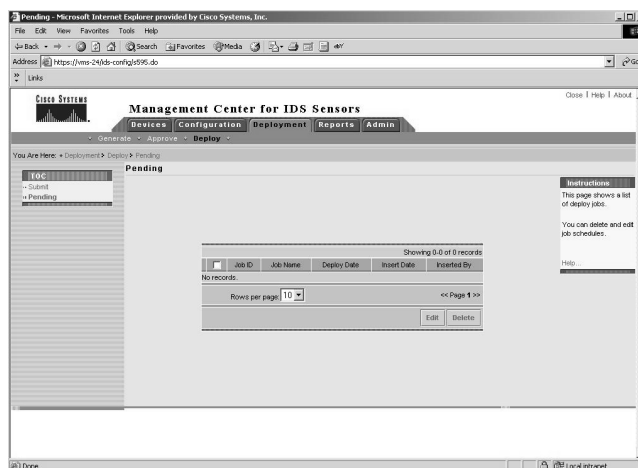
- The **Submit** page appears, as shown in Figure 10.32. From the tree, check the box next to the sensor name where the configuration file is to be deployed.

Figure 10.32 The Submit Page



- The **Select Configuration** page appears. Select a sensor configuration by checking the corresponding box and click **Next**.
- The **Enter Job Properties** page appears. Under **Schedule Type**, enter the name of the job from the **Job Name** field.
- The job will deploy the configuration to the selected sensor. To start the job immediately, click the **Immediate** button. To schedule the job to execute at a later time, click the **Scheduled** radio button and select the desired options.
- Click the **Finish** button.
- The Submit page appears. To verify the scheduled job return to the **Management Center for IDS Sensors** page, as shown in Figure 10.9. Select **Deployment | Deploy**. From the **Table of Contents**, select **Pending**. The Pending jobs page appears, as shown in Figure 10.33. On this page, it is possible to edit a pending deployment or delete it by using the Edit and Delete buttons.

Figure 10.33 The Pending Jobs Page



Configuring Reports

Reports provide a summarization of the various activity and configuration of the deployed IDS sensors as well as the IDS Management Center itself. This is crucial when managing and monitoring an enterprise-wide deployment of IDS since it becomes impractical to query each IDS sensor manually in order to determine its status. The IDS Management Center can produce reports, known as audit reports, which provide information about network configuration activities managed with the Cisco IDS MC. These reports can be generated from the **Reports** tab of the Management Center for IDS Sensors page shown in Figure 10.9.

Additional reports are available from the Security Monitor. The Security Monitor is a closely related but separate product that receives real-time communications from the sensors. When the IDS Management Center and the Security Monitor are installed in the same host system, the audit report templates are shared between the two products.

Audit Reports

There are six types of audit reports available from the IDS Management Center:

- The Subsystem Report
- The Sensor Version Import Report
- The Sensor Configuration Import Report
- The Sensor Configuration Deployment Report

- The Console Notification Report
- The Audit Log Report

The following sections examine each report in detail.

The Subsystem Report

The Cisco Intrusion Detection System has many subsystems. These subsystems include the Management Center, the Security Monitor, and other subsystems. The Subsystem Report shows audit records separated and ordered by subsystem. The entries in the Subsystem Report can be filtered by event severity, date/time, and subsystem.

The Sensor Version Import Report

The IDS Management Center tracks the version identifier of each sensor. When the version identifier of a sensor is imported to the IDS MC, an audit record is generated. The audit record indicates the success or failure of the import operation. The entries in the Sensor Version Import Report can be filtered by device, event severity, and date/time.

The Sensor Configuration Import Report

IDS sensor configurations are often imported into the IDS Management Center for viewing or editing. Audit records are generated when this import operation is executed. The audit record indicates the success or failure of the import operation. The entries in the Sensor Configuration Import Report can be filtered by device, event severity, and date/time.

The Sensor Configuration Deployment Report

File configurations containing new settings are often deployed to the sensors. Audit records are generated when this deployment operation is executed. These records can indicate successful deployment or provide error messages. The entries in the Sensor Configuration Deployment Report can be filtered by device, event severity, and date/time.

The Console Notification Report

The IDS Notification subsystem generates console notification audit records. The entries in the Console Notification Report can be filtered by event severity and date/time.

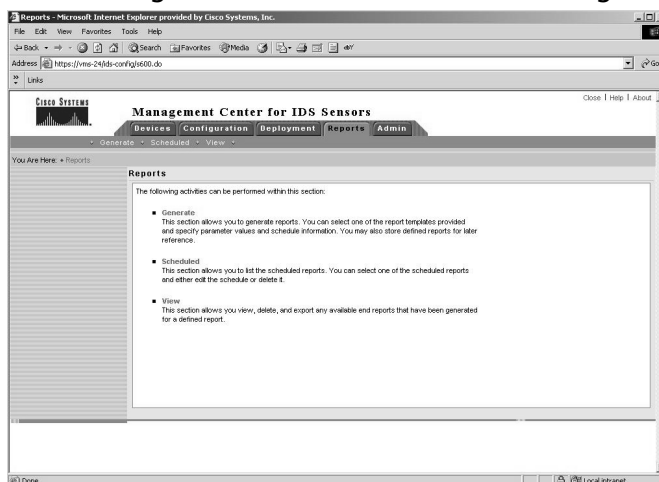
The Audit Log Report

The Audit Log Report displays audit records by the IDS server and by the IDS application. This report template provides a broad, non-task-specific view of audit records in the database. The entries in the Audit Log Report can be filtered by task type, event severity, date/time, subsystem, and application.

Generating Reports

Reports can be generated immediately or scheduled at a later time. We can generate a report by starting from the IDS Management Center for IDS Sensors page and selecting the Reports tab. The resulting page is shown in Figure 10.34.

Figure 10.34 The Management Center for IDS Sensors Page



To generate a report, follow these steps:

1. From the **Reports** page, select **Generate**.
2. The **Select Report** page appears. Choose the type of report to generate and click **Select**.
3. The **Report Filtering** page appears. Enter the report parameters for the report selected and click **Next**.
4. The **Schedule Report** page appears. In the **Report Title** field, specify a name for the report. Select a radio button to schedule the report:
 - *Run Now* will generate the report immediately.
 - *Schedule for Later* will allow the specification of when the report will be generated, including the generation of reports on regular intervals.

5. The Email Report To field allows the specification of an e-mail address of a report recipient. Click **Finish**.
6. To view the reports scheduled for generation, from the **Management Center for IDS Sensors** page, select **Reports | Scheduled**.

Viewing Reports

To view a generated report, start from the Management Center for IDS Sensors page and do the following:

1. Select **Reports | View**.
2. The **Choose Completed Report** page appears. Check the box corresponding to the title of the report to view and click **View**.

Exporting Reports

To export a generated report to an HTML file, start from the Management Center for IDS Sensors page and perform the following steps:

1. Select **Reports | View**.
2. The **Choose Completed Report** page appears. Check the box corresponding to the title of the report you want to view and click **Open in Window**.
3. Depending on the browser that appears, select **File | Save As** or **Save File**. Browse to the location where the file is to be saved, enter a file name and click **Save**.

Deleting Generated Reports

To delete a generated report, start from the “Management Center for IDS Sensors” page and do the following:

1. Select **Reports | View**.
2. The **Choose Completed Report** page appears. Check the boxes corresponding to the titles of the reports to delete and click **Delete**.

Editing Report Parameters

To edit the schedule for a report or the parameters for a scheduled report, start from the Management Center for IDS Sensors page and perform the following steps:

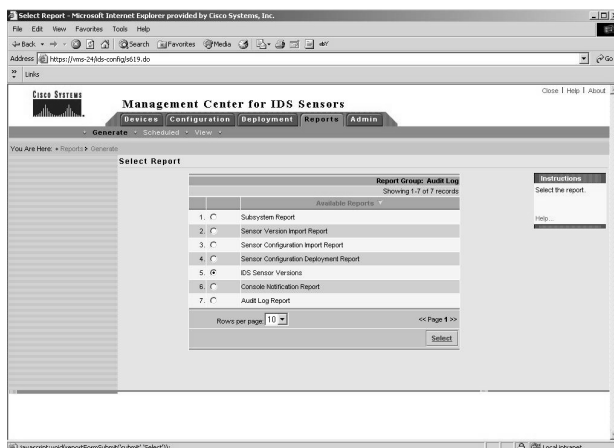
1. Select **Reports | Scheduled**.
2. The Edit Scheduled Reports page appears. Check the box corresponding to the title of the report template to edit and click **Edit**.
3. A new page appears displaying the report parameters. Change any report parameter and click **Finish**.

Example of IDS Sensor Versions Report Generation

This section details the generation of an example report. Use the following procedure to generate and view reports:

1. Select **Reports | Generate** to select the type of report to be generated from the **Select Report** page.
2. In the **Select Report** page, choose one of the report types desired (as shown in Figure 10.35) and click **Select**.

Figure 10.35 The Select Report Page

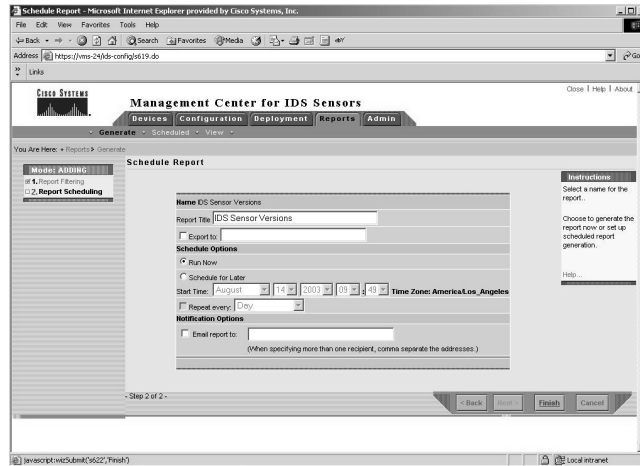


3. The next step is to schedule the report. In the **Schedule Report** page (shown in Figure 10.36), the report generation can be scheduled to

occur immediately, with the **Schedule Options | Run Now** option, or for some later period (**Schedule Options | Schedule for Later**).

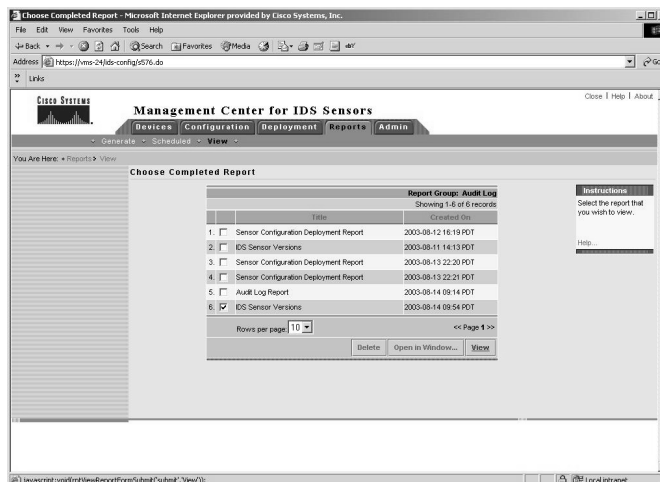
4. Select the **Finish** button to generate the report.

Figure 10.36 The Schedule Report Page



5. Once the report generation is complete, the report title will appear in the list of completed reports. Select the check box (or check boxes) of the report (or reports) to view, and then select **View** (as shown in Figure 10.37).

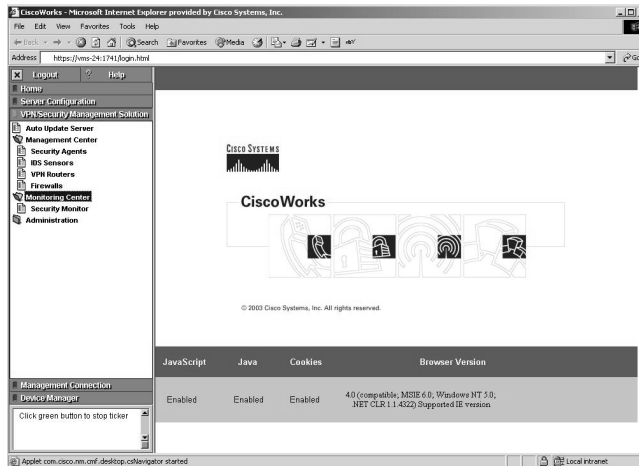
Figure 10.37 The Choose Completed Report Page



Security Monitor Reports

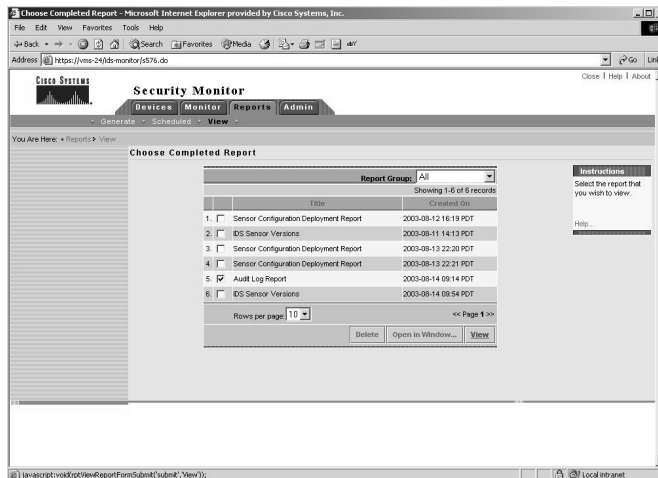
While the IDS Management Center can provide audit log reports, information about network activities detected by the IDS Sensors are usually provided by the Security Monitor. To access the Security Monitor from the CiscoWorks2000 Desktop, select the **Monitoring Center** and then the **Security Monitor**, as shown in Figure 10.38.

Figure 10.38 The Security Monitor



To access reports provided by the Security Monitor, select the **Reports** tab and then the **View** entry. This will bring up the **Completed Reports** menu, as shown in Figure 10.39.

Figure 10.39 The Security Monitor Completed Reports



To select a report for viewing, check the box next to the report and click the **View** button.

Administering the Cisco IDS MC Server

The administration of the Cisco IDS MC server is comprised of tasks associated with the IDS Database and other global tasks. This encompasses:

- Operations with database rules
- Updating sensor software and signature release levels
- Defining the e-mail server settings
- Setting the configuration file approval method

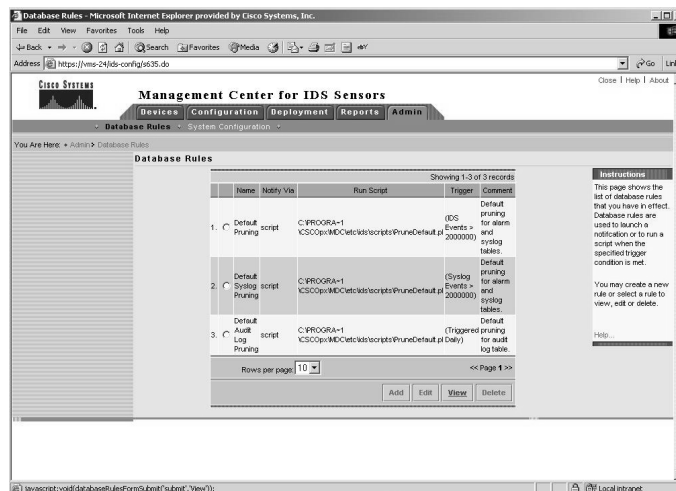
Database Rules

Database rules are used to configure the Cisco IDS Management Center to take an action at daily intervals or when a database threshold has been reached. These actions to be taken may include: sending an e-mail notification, logging a console notification event, or executing a script.

Adding a Database Rule

To add a database rule, start from the Management Center for IDS Sensors page, select the **Admin** tab and **Database Rules** (as shown in Figure 10.40), and perform the following steps:

Figure 10.40 The Database Rules Page



1. Select **Admin | Database**.
2. The **Database Rules** page appears. Click **Add**.
3. The **Specify the Trigger Conditions** page appears. Specify the threshold to trigger Security Monitor to take an action. The following triggers can be specified with check boxes:
 - **Database used space greater than (megabytes)** This will trigger an action when the database reaches a size in megabytes that is specified in the next field.
 - **Database free space less than (megabytes)** This will trigger an action when the database free space drops to a size in megabytes that is specified in the next field.
 - **Total IDS events** This will trigger an action when the total number of IDS events in the database reaches the number specified in the next field.
 - **Total SYSLOG events** This will trigger an action when the total number of SYSLOG events in the database reaches the number specified in the next field.
 - **Total events** This will trigger an action when the total number of events in the database reaches the number specified in the next field.
 - **Daily beginning** This will trigger an action to occur daily beginning on the date and time specified.

In the **Comment** field, you may enter a description of the Database Rule. Click **Next**.
4. The **Choose the Actions** page appears. More than one action can be selected via the following check boxes:
 - Notify via Email
 - Log a Console Notification Event
 - Execute a Script
5. Click **Finish**.

Editing a Database Rule

To edit a database rule, start from the Management Center for IDS Sensors page (as shown in Figure 10.29) and follow these steps:

1. Select **Admin | Database**.
2. The **Database Rules** page appears. Select the radio button corresponding to the rule to edit and click **Edit**.
3. The **Specify the Trigger Conditions** page appears. Select the radio button corresponding to the rule to edit and click **Edit**. Change the field to be revised and click **Next**.
4. The **Choose the Actions** page appears. Make the desired changes and click **Finish**.

Viewing a Database Rule

To view a database rule, start from the Management Center for IDS Sensors page (as shown in Figure 10.29) and follow these steps:

1. Select **Admin | Database**.
2. The Database Rules page appears. Select the radio button corresponding to the rule to view and click **View**.
3. The **View Database Rule** page appears. In the text box is detailed information about the rule. To return to the Database Rules page, click **OK**.

Deleting a Database Rule

To delete a database rule, start from the Management Center for IDS Sensors page (as shown in Figure 10.29) and follow these steps:

1. Select **Admin | Database**.
2. The **Database Rules** page appears. Select the radio button corresponding to the rule you want to delete and click **Delete**. The database rule is deleted from the IDS Management Center.

Updating Sensor Software and Signatures

Cisco Systems is constantly providing new sensor software versions and signature release levels. These new versions and release levels are provided in files known as Service Pack update files and Signature update files.

The procedures to update the sensor software and the signatures are complex. To be informed of the latest update files by e-mail, you can subscribe to the Cisco IDS Active Update Notification.

Defining the E-mail Server Settings

You can specify the e-mail server that the Cisco IDS Management Center uses for event notification. To specify the server, follow these steps:

1. Start from the Management Center for IDS Sensors page as shown in Figure 10.29 and select **Admin | System Configuration**. Select **Email Server** in the Table of Contents.
2. The **E-mail Server** page appears. Enter the e-mail server name in the **Server Name** box. Click **Apply**. The e-mail server specified will be used for event notification.

Summary

Sensors cannot be used efficiently as standalone devices in the enterprise network. When a network and its sensors grow in size and number, the administrative overhead of the sensors becomes an ever-increasing burden. When deployed in large numbers on an enterprise network, the sensors require the IDS Management Center to provide the group management functions needed for scalable operations. The IDS Management Center can group together sensors with similar configurations so that the same operations can be performed on all sensors within a group. Similarly, the IDS MC can efficiently update the sensor software version and the signature release level of all, or selected, sensors in one operator action. The IDS MC is integrated with an IDS Database where the configuration and signature settings of all the sensors are stored. This database permits the IDS MC operator to easily review, edit, approve, and deploy configuration settings and signature parameters for each and every sensor.

The MC contains report generation features that can be automated. Reports can be scheduled for generation at periodic intervals and can be viewed online, exported to an HTML file or posted on a company intranet. Finally, the IDS MC has various self-administration capabilities, including the capability to log audit records of its own internal functions. It can even be configured to take action when certain event thresholds are reached such as the IDS database size growing beyond a configured limit.

The following sources should prove useful for further research:

- Barman, Scott, *Writing Information Security Policies*, (2nd Ed), New Riders, Indianapolis, IN., 2002
- Pfleeger, Charles P., *Security in Computing*, (2nd Ed), Prentice Hall PTR, Upper Saddle River, NJ., 1997
- SANS – Security Policy Project, www.sans.org/resources/policies/
- NIST – “Guidelines on Firewalls and Firewall Policy,” NIST, <http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf>
- National State Auditors Association and U.S. General Accounting Office – “Management Planning Guide for Information Systems Security Auditing,” www.gao.gov/special.pubs/mgmtpln.pdf

Solutions Fast Track

Understanding the Cisco IDS Management Center

- ☑ The IDS MC logs internal audit records pertinent to the intrusion detection system.
- ☑ The IDS MC can manage approximately 300 sensors.
- ☑ Sensor and signature configuration are key functions performed by the IDS MC.
- ☑ Maintaining current sensor software and signature releases are functions of the IDS MC.

Installing the Cisco IDS Management Center

- ☑ Prerequisite products include Windows 2000 and Cisco Works Common Services.
- ☑ A related product is the Security Monitor that displays real-time alarms from the sensors.

Setting Up Sensors and Sensor Groups

- ☑ Sensors should be placed at entry points to the network and between sub-networks of different security levels.
- ☑ Sensors with similar configuration settings can be placed in the same sensor group or subgroup.
- ☑ A sensor can be placed behind a filtering router so the sensor can issue a blocking command to the router when an attack is detected.

Configuring Signatures and Alarms

- ☑ There are six classifications of signatures: general, TCP, UDP, string-matching, ACL, and custom.
- ☑ Signature settings can be configured and tuned by the IDS MC.

- ☑ The IDS MC can generate, approve, and deploy sensor configuration files.

Configuring Reports

- ☑ The IDS MC has six audit log reports: subsystem, sensor version import, sensor configuration import, sensor configuration deployment, console notification, and audit log.
- ☑ Reports can be generated immediately, scheduled at a later time, or scheduled at regular intervals.
- ☑ The generated reports can remain online for viewing or be deleted.
- ☑ The generated reports can be exported into an HTML file.
- ☑ The scheduled report parameters can be edited.

Administering the Cisco IDS MC Server

- ☑ Database Rules are designed to trigger actions when specified database event thresholds are reached.
- ☑ The IDS MC can be used to update sensor software versions and signature releases.
- ☑ An e-mail server can be specified for the IDS MC to use to distribute e-mail notifications.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Where in my enterprise network should I deploy my sensors?

A: In your enterprise network, the sensors should be deployed at entry points to your network and between sub-networks that require different levels of protection. This does not pertain to just Internet connections but to any connection to a vendor’s network, whether it be by VPN or another connection type.

Q: How can I make sure my sensors have signatures for the latest threats?

A: To be informed of the latest update files by e-mail, you can subscribe to the Cisco IDS Active Update Notification.

Q: Will the IDS MC protect my network from Denial-of-Service (DoS) attacks?

A: The IDS MC itself will not protect your network from DoS attacks. However, you can use the MC to configure your IDS sensors to warn you of an attack and allow you to take appropriate action to filter the attack packets. Second, the IDS can configure the sensors to order the blocking router to block the attack.

Q: Can the IDS MC manage honeypots?

A: Through the use of a sensor on the honeypot network, the IDS MC can detect attacks directed against honeypots and notify you so you can take appropriate action to determine the source and nature of the activity.

Q: Does the IDS Management Center display real-time intrusion alarms?

A: No. The Security Monitor will display real-time intrusion alarms through its Event Viewer. SecMon will send e-mail notification when certain event thresholds are reached.

Q: Using IDS MC, can I update the configurations on several sensors at one time?

A: Yes. If all the sensors in the same group require the same configuration update, the configuration updates can be deployed with the same operator action.

Q: Can the IDS MC manage sensors outside of my firewall?

A: Yes. The IDS MC only requires a TCP connection to the sensor it manages. It is not even necessary that the sensor be in the same network as the IDS MC.

Q: Can a large network be managed by multiple IDS MCs?

A: Yes. Different portions of a large network may have different security policies and it may be more advantageous from an administrative perspective to manage it with more than one IDS MC.

Q: How can I minimize false alarms?

A: By tracking and analyzing false positives that are generated, you can determine the optimal settings for your signatures to minimize false alarms. This is usually done by tuning the Micro-Engine Parameters in your signatures or by excluding certain internal networks as triggers of alarms.

Cisco Firewall/IDS IOS

Solutions in this chapter:

- Understanding Cisco IOS-Based IDS
- Configuring the IOS-Based IDS
- Configuring IOS-Based IDS Signatures
- Responses from the IOS-Based IDS
- Verifying the IOS-IDS configuration

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

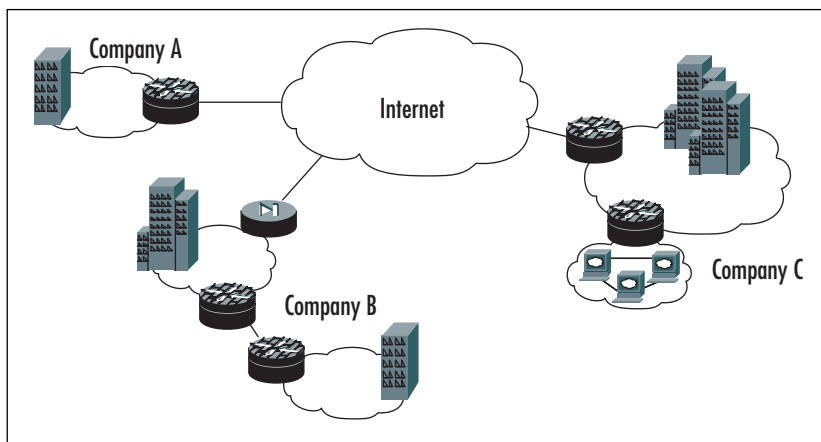
Introduction

When you start implementing intrusion detection in the corporate LAN, it isn't necessary to spend a lot on IDS sensors or IDSM blades. This is even truer for networks in small offices, which don't have the budgets of larger corporations. An affordable start with intrusion detection can be made using the Firewall/IDS feature set of IOS, which a growing number of Cisco router platforms now support. Because IOS-IDS runs on existing network hardware and uses Syslog for alarm notification, it complements the existing security infrastructure without the need for new hardware and Director software. The downside of using IOS-based IDS is that the capabilities of IOS-IDS are limited if you compare them with the IDS sensors or IDSM. The performance of the router may suffer under the processing load of IDS and the number of signatures supported is limited.

In this chapter, we will discuss these performance issues and look at the limitations of IOS-IDS, as well as explore which router platforms are capable of running IOS-IDS and the number of signatures the IOS identifies. We will learn how to configure IOS-based IDS, see how IDS takes action when under attack, and learn how to verify and monitor an IDS configuration.

In Figure 11.1, we see some of the ways Cisco IOS-IDS can be employed within your network. Company A is using Cisco IOS-IDS to protect its LAN from attacks originating on the Internet. Company B has put IOS-IDS to use to protect a Frame-Relay link to one of its branches. Company C is using Cisco IOS-IDS to protect the LAN from attacks originating on the Internet, but is also using IOS-IDS to protect a cluster of intranet web servers from attacks.

Figure 11.1 Cisco IOS-IDS Employment



Understanding Cisco IOS-Based IDS

Understanding Cisco IOS-based IDS starts with realizing that it is a different kind of IDS than previously seen. There are differences in hardware, software, performance, and signatures. To get a better understanding of IOS-based IDS, we will discuss the following issues:

- Supported router platforms
- Performance
- Signatures
- Intrusion Response options

Supported Router Platforms

One of the major benefits of using IOS-based IDS is that you can add intrusion detection functionality to your network, using your existing router hardware. Not all Cisco routers have support for the Firewall IDS feature set of IOS; their number however is growing. IDS has been available in IOS since version 12.0(5)T. IOS has built-in IDS support for the following router platforms:

- Cisco 1700 Series
- Cisco 2600 Series
- Cisco 3600 Series
- Cisco 3700 Series
- Cisco 7100 Series
- Cisco 7200 Series
- Cisco 7400 Series
- Cisco 7500 Series

Performance

A router configured for IDS can be classified as an inline processing network sensor. The router sits in the packets' path, analyzes each packet that passes through and compares it to the signature base. For some packets, the router needs to maintain state, and even application state, information. Thus, you should

understand that maintaining this information will have some impact on IDS performance, and that you should always test the configuration, if possible, before network deployment. Even once it is deployed, the old configuration should be on hand as a backup. Some good tools to measure CPU performance include: MRTG and the CPU Monitor from Solarwinds.net. An explanation of how to use the free MRTG to monitor the CPU utilization for a Cisco router can be found at <http://slowest.net/docs/howtos/mrtg/mrtg-cisco-cpu.html>

As discussed earlier in this book, atomic signatures are triggered by a single packet that matches the signature. Auditing these kinds of signatures don't influence performance much. Compound signatures, on the other hand, are triggered by multiple packets, and IOS-IDS has to allocate memory to maintain the state of each session. IOS-IDS further allocates memory to the configuration database and for internal caching.

Signatures

Originally, Cisco IOS-IDS supported 59 signatures, but starting with 12.2(11)YU and the latest 12.2T IOS releases, IOS-IDS supports a total of 100 signatures. These signatures are a cross-section of the signatures available to the Cisco IDS Sensor that supports over 300 signatures and are selected to identify the most common network attacks and information gathering scans.

In contrast to the traditional Cisco IDS Sensor where signatures are updated via special files on a regular basis, signatures on IOS-IDS are not frequently updated. Signatures on an IOS-IDS can only be updated by installing a new IOS image on all IDS routers.

As we will see later in this chapter, an IOS-IDS can only use a Director to send alarm notifications. It is therefore not possible to create a custom signature for an IOS-IDS on the Director in case of a new threat for which no signature is available yet, such as the recent SQL Slammer Worm.

NOTE

Be aware that the current test material of the Cisco Secure Intrusion Detection Systems Exam (CSIDS 9E0-100) still refers to a total number of 59 signatures that Cisco IOS-IDS supports.

Intrusion Response Options

A router configured as an IOS-IDS sensor will track and audit the packet flow through the router. When a packet or a number of packets match a certain signature, IOS-IDS will respond to that match in the way you have configured it to respond. The router can be configured to perform one or more of the following actions:

- **Send an alarm** An IOS-IDS sensor can be configured to send an alarm when a signature is matched. An alarm can be sent to a Syslog server, a Director, or an IDS Sensor. The router will forward the offending packet if no other actions are configured.
- **Drop the packet** If this feature is configured, an IOS-IDS sensor will drop offending packets immediately when a signature is matched.
- **Reset a TCP session** An IOS-IDS sensor resets a TCP session in which unauthorized activity takes place if this action is configured. It will do so by sending a packet with the Reset (RST) flag set, to both the offender and the victim. If no other actions are configured on the IOS-IDS, the offending packet will still be forwarded to the victim. The best practice is to use the drop and reset actions together, as it will completely terminate the attack.

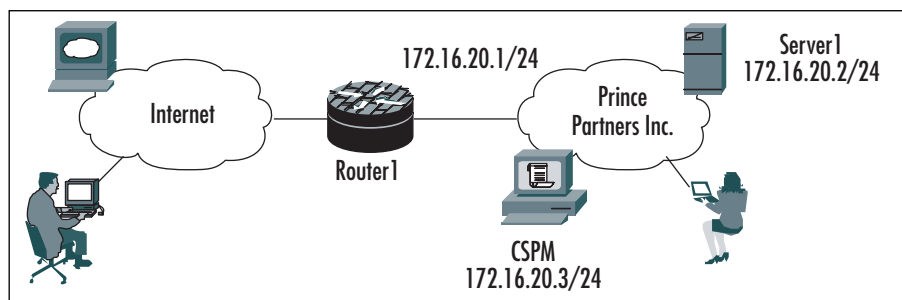
Configuring the IOS-Based IDS

The IOS Firewall/IDS image and a Cisco router that supports the Firewall/IDS feature set is all you need to start configuring the IOS-based IDS. Configuring an IOS-based IDS is a six-step process. In these six steps, the IDS will be enabled, communications with a Syslog or Director will be configured and the IOS-IDS will learn what to do when it's attacked. The six configuration steps include:

- Configuring the notification type
- Configuring local and remote PostOffice parameters
- Configuring protected networks
- Changing the size of the notification queue
- Setting the default signature actions
- Creating and applying audit rules

We will walk through the configuration process and discuss each step using the following example. Figure 11.2 shows the network of Prince Partners Inc. Router1 connects Prince Partners LAN with the Internet via a T1, and is configured to act as a Firewall and NAT gateway. Mr. Hague, the CEO of Prince Partners wants you to configure IDS on this IOS-based router to complement the already existing security infrastructure with intrusion detection functionality. On the LAN, a Syslog server and a CSPM server are available, and Mr. Hague wants you to use both. Let's begin configuring IOS-based IDS.

Figure 11.2 The Prince Partners Inc. LAN



Configuring the Notification Type

The first step in configuring IOS-based IDS is specifying the notification type your IDS should use for alarm notification. IOS supports two types of alarm notification: log and nr-director. Using the log notification type will make the router send its messages in Syslog format to the local Syslog service or to a remote Syslog server; the nr-director notification type will let the router send its messages in PostOffice format to a Director or IDS sensor. An important note to make is that it is not necessary to have a Director, like the CSIDD or CSPM, to manage the alarms; a Syslog server is sufficient to manage the alarms from a Cisco IOS-IDS.

In the following example, we will configure Router1 to send its alarms to both the Syslog server named Server1 and the Director named CSPM. We do this by using the *ip audit notify log* command.

To enable Router1 to use Syslog for notification, apply the following commands:

```
Router1(config)#ip audit notify log
Router1(config)#logging host 172.16.20.2
```

The first command enables Router1 to send alarm notifications to a Syslog server. The second command configures a remote Syslog server, and is only needed in case a remote Syslog server has not yet been configured on your router. If you only use the *ip audit notify log* command on a router and do not configure a remote Syslog server, the router will send alarm notifications to its local Syslog service.

Next, we configure Router1 to also send alarm notifications to the Director.

```
Router1(config)#ip audit notify nr-director
```

Configuring the *nr-director* notification type will make Router1 send alarms to the Director in PostOffice format, whereas the *log* notification type will make the router send its alarms in syslog format.

Configuring Local and Remote PostOffice Parameters

Enabling the *nr-director* notification type is the first step in configuring an IOS-based IDS to send its alarm notifications to a Director. Router1 from our example will communicate with the Director using the PostOffice protocol, and for Router1 to know how to communicate with the Director we will have to configure local and remote PostOffice parameters on Router1. Note that configuring local and remote PostOffice parameters is only needed when using a Director or IDS sensor for alarm notification, when using only a Syslog server, this step in the configuration process can be skipped.

The local and remote PostOffice parameters are used to uniquely identify a router as an IOS-based IDS in the network and to tell the router where the Director is located in the network. The two local PostOffice parameters to be configured on a router are the *host-id* and the *org-id*. The *host-id* is a unique number between 1 and 65535 that is used to identify the host in PostOffice communications; the *org-id* is a number between 1 and 65535 that is used to identify the group to which the host belongs in PostOffice communications. The default value of the local PostOffice parameters is 1 for both *host-id* and *org-id*.

Now we will configure the local PostOffice parameters on Router1 in the Prince Partners network. We do this by using the *ip audit po local* command in global configuration mode.

```
Router1(config)#ip audit po local hostid 2 orgid 100
```

We have now configured Router1 with a hostid of 2 and an orgid of 100. To be able to successfully communicate with the Director, Router1 also needs to know where to find the Director. We configure this using the *ip audit po remote* command in global configuration mode.

```
Router1(config)#ip audit po remote hostid 1 orgid 100 rmtaddress
172.16.20.3 localaddress 172.16.20.1
```

What we have done here is tell Router1 that the Director can be found at IP address 172.16.20.3 and that the PostOffice parameters of the Director are 1 for the host-id and 100 for the org-id. We chose the host-id and org-id values from the same integer range of 1-65535 that we used for the local PostOffice parameters. Important to see is that we used the same org-id but a unique host-id. The org-id defines the group to which the Director and IOS-IDS sensor belong and therefore has to have the same value for both. The host-id defines the host within the group and has to be unique. Finally, at the end of the preceding command we have said that PostOffice communications will originate from IP address 172.16.20.1 locally.

If we had chosen not to configure the remote PostOffice parameters, Router1 would have used its default settings. Like with the local PostOffice parameters, the default setting for host-id and org-id is 1.

Other parameters that can be configured using the *ip audit po remote* command are the heartbeat timeout value for PostOffice communications, the UDP port that the Director uses to listen for alarm notifications, the alarms receiving application and the preference. Let's take a look at the last two items. Using the application parameter, you tell an IOS-based IDS that the remote PostOffice device it is talking to is either a Director or an IDS sensor. This is done using the *director* or *logger* keyword. The default value for this parameter is *director* so there is no need for us to configure this on Router1.

To specify an IDS sensor as the receiving PostOffice device use the command:

```
aRouter(config)#ip audit po remote hostid 5 orgid 150 rmtaddress
192.6.30.5 localaddress 192.6.30.3 port 45000 preference 1 timeout
5 application logger
```

To specify a Director as the receiving PostOffice application use the following command. Since this is the default value for this parameter there is no need to add this parameter to the *ip audit po remote* command.

```
aRouter(config)# ip audit po remote hostid 5 orgid 150 rmtaddress
192.6.30.5 localaddress 192.6.30.3 port 45000 preference 1 timeout
5 application director
```

The last item is the preference parameter. Using preference, redundancy can be provided by setting up multiple paths for a router to deliver its alarm notifications to a Director or IDS sensor. The following is an example of how to configure this feature:

```
aRouter(config)#ip audit po remote hostid 5 orgid 150 rmtaddress
192.6.30.5 localaddress 192.6.30.3 port 45000 preference 1
aRouter(config)#ip audit po remote hostid 5 orgid 150 rmtaddress
192.6.31.5 localaddress 192.6.31.3 port 45000 preference 2
```

The first command specifies what the PostOffice parameters are for the Director and at what subnet it is located; the second command specifies the same Director but is connected to another subnet. The preference parameter tells the router which route is preferred for communication with the Director. If the first route goes down, a second route is still available to provide redundancy.

Configuring Protected Networks

When a router sends an alarm, it will add an IN or OUT designator to the alarm if the victim's address is part of a configured, protected network. This flag will define an IP address as inside or outside the protected network. This means that if no protected network is configured at all, the router will consider all addresses as outside the protected network. Configuring a protected network will not have an impact on IDS performance or functionality as it only adds the flag discussed earlier.

To configure the LAN of Prince Partners Inc. in Figure 11.2 as a protected network, we use the command *ip audit po protected*, as shown next.

```
Router1(config)#ip audit po protected 172.16.20.1 to 172.16.20.254
```

To remove all protected IP addresses, you can use the following command in global configuration mode.

```
Router1(config)#no ip audit po protected
```


Configuring & Implementing...

Changes in IOS

IOS-IDS commands have been added and changed since IDS first became available in IOS 12.0(5)T. New debug commands like *debug ip audit dns* and *debug ip audit http* have been added which provide more troubleshooting power to the engineer. The command that has changed is the *ip audit po protected* command. From IOS 12.2 on, this command is known as the *ip audit protected* command.

Changing the Size of the Notification Queue

A router has a limited persistent storage facility and for every alarm the router tracks, it uses 32KB of memory. For these reasons, a notification queue exists. The default size of this queue is 100 alarms and can be changed, but you have to be careful in doing so for the reasons just mentioned. As the queue fills up and reaches its maximum size, the router starts losing alerts on a first-in first-out (FIFO) basis.

We have upgraded memory in Router1 and will change the size of the notification queue to hold a maximum of 200 alarms. We enter the following command in global configuration mode to accomplish this goal.

```
Router1(config)#ip audit po max-events 200
```

You can see that the size of the notification queue has changed by using the *show ip audit configuration* command. The following is part of the output generated by this command.

```
PostOffice:HostID:2 OrgID:100 Msg dropped:0
      :Curr Event Buf Size:100 Configured:200
```

If, in the future, you decide to reset the size of the notification queue to 100 alarms, you can do so by issuing the following command as a global configuration command.

```
Router1(config)#no ip audit po max-events
```

Setting the Default Signature Actions

Before we create an audit rule, we will set the default action for the router to take when either information or attack signatures are triggered. These settings will apply to all traffic that doesn't fall under the audit rules created with the *ip audit name* command in the next step of the configuration process.

We configure the default action on Router1 as follows:

```
Router1(config)#ip audit info action alarm
Router1(config)#ip audit attack action alarm drop reset
```

The first command sets the default action to send an alarm when an informational signature is triggered. The second command sets the default action to send an alarm, drop the packet, and reset the TCP session when an attack signature is being triggered.

Creating and Applying Audit Rules

The last step in setting up IDS on IOS is creating and applying audit rules. Using the *ip audit name* command, we will create audit rules for both information and attack signature groups on Router1. We will then apply the audit rule to the interface(s) and specify if the rule should apply to inbound or outbound traffic.

We configure the audit rules on Router1 as follows:

```
Router1(config)#ip audit name idstest info action alarm
Router1(config)#ip audit name idstest attack action alarm drop reset
Router1(config)#interface Serial 0/0
Router1(config-if)#ip audit idstest in
```

This will tell Router1 to track all inbound traffic on interface Serial 0/0 for intrusions and send an alarm when an informational signature gets triggered. When an attack signature is triggered, Router1 sends an alarm, drops the offending packet, and resets the session if it involved a TCP session.

This completes the IDS configuration of Router1 for Prince Partners Inc. It might however be the case that a certain host is causing a lot of false positives or you just don't want it to take part in intrusion detection. Audit rules can be configured to exclude a certain host or network from taking part in intrusion detection, this is configured using access-lists. Let's take a look at our example in Figure 11.2. We want to prevent Server1 from taking part in intrusion detection and that all other traffic is tracked for intrusions. We use the following commands in global and interface configuration mode to accomplish these goals.

```
Router1(config)#ip audit name idstest info list 73 action alarm
Router1(config)#ip audit name idstest attack list 73 action alarm drop
reset
```

```
Router1(config)#ip access-list standard 73
Router1(config-std-nacl)#deny host 172.16.20.2
Router1(config-std-nacl)#permit any
```

```
Router1(config)#interface Serial 0/0
Router1(config-if)#ip audit idstest in
```

Configuring IOS-Based IDS Signatures

IOS-IDS will trigger an alarm when a packet matches a certain behavior defined in a signature. It is critical that no alarms are generated for an event that will not be harmful for the network. A large number of these so-called false positives will drain resources and can become very costly to a company. It is also critical that alarms are generated when needed. False negatives occur when an IDS fails to detect an intrusion. You can prevent false negatives by making sure you have the IOS with the latest signatures available installed. Fine-tuning an IDS configuration by disabling or excluding signatures will help prevent a large number of false positives in your network.

In this section, we look at how we can manage signatures to prevent false positives. We do so by doing the following:

- Disabling signatures globally
- Excluding signatures by host or network
- Configuring the spam signature threshold

Disabling Signatures Globally

All signatures available in IOS are enabled by default when IDS is configured on a router. A number of these signatures are application- or operating system-specific, and might not pose a threat to your network. Still, intrusions occur and keep your Operations Department busy. You might for instance have no UNIX-based servers in your network, yet alarms keep getting triggered for the mountd Portmap Request signature and fill up your management GUI. In such cases, you

want to disable a certain signature, and by doing so lower the administrative burden that results from these false positives.

We disable the mountd Portmap Request signature and the Majordomo Execute Attack signature by entering the following command in global configuration mode:

```
aRouter(config)#ip audit signature 6155 disable
aRouter(config)#ip audit signature 3107 disable
```

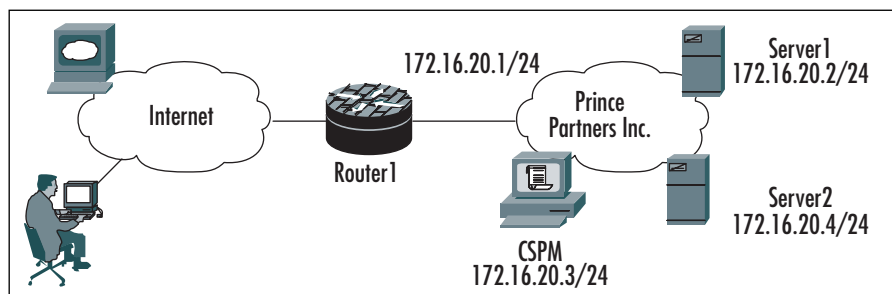
When the need arises to track traffic for signature 6155 (Portmap) again, you can enable this signature using the *no* keyword in front of the *ip audit signature* command. Doing so may look like you're disabling the signature, but that's not the case. It simply enables the signature. Here's an example:

```
aRouter(config)#no ip audit signature 6155
```

Excluding Signatures by Host or Network

One of the major disadvantages of disabling signatures globally is that network traffic is no longer being tracked for intrusions with that specific signature. This is especially true when you have a mixed environment. You may be receiving a lot of false positives from a Windows-based host, but disabling the signature globally will make those UNIX-based hosts vulnerable to attack. Excluding signatures by host or network will ensure that if an attack takes place on a UNIX-based host, it is detected and actions are taken. The following example shows how to exclude a signature for a certain host.

We are back at Prince Partners Inc. In Figure 11.3, we see Router1 connecting Prince Partners LAN to the Internet. Router1 is acting as a Firewall/IDS device, has all signatures enabled, and is protecting network 172.16.20.0/24. Server1 is a Windows 2000 Exchange server, and traffic to that server is creating false positives for signature 6155, the mountd Portmap Request signature. The alarms generated are false positives because Server1 does not have mountd running and is therefore not vulnerable for this intrusion. Server2 is a UNIX server that might be vulnerable for this attack and traffic to this server must still be tracked for signature 6155.

Figure 11.3 The Prince Partners Inc. LAN

To prevent false positives on Server1 from happening we will exclude Server1 from this signature, and do so by using the following commands:

```
Router1(config)#ip audit signature 6155 list 10
Router1(config)#ip access-list standard 10
Router1(config-std-nacl)#deny host 172.16.20.3
Router1(config-std-nacl)#permit any
Router1(config-std-nacl)#end
Router1#
```

In this example, we see that the *ip audit signature* command refers to a standard access-list that specifies which hosts are to be excluded when tracking network traffic for signature 6155. Here, we have excluded Server1 and permitted all other hosts. Remember that at the end of an access-list there is an implicit deny. If we had not used the *permit any* statement in the access-list, all hosts would have been excluded from this signature.

You can make the signature available again for tracking traffic to Server1 by using the following commands:

```
Router1(config)#no ip audit signature 6155 list 10
Router1(config)#no access-list 10
Router1(config)#end
```

Using the Spam Signature

The *ip audit smtp spam* command is used to change the recipient threshold of the spam signature (Signature 3106). This signature will detect an e-mail message whose number of recipients exceeds the threshold and take appropriate action. The default value of this threshold is 250 recipients. Depending on your existing

mail environment and traffic, you can change this value to a higher or lower number. Be careful you don't set this value so low that e-mail gets lost.

Let's take another look at Figure 11.3. Prince Partners Inc. is using an Exchange 2000 e-mail environment. Server1 is the only mail server at this moment. The server administrator has come to you requesting something be done about the spam mail appearing on Server1. You decide to lower the value of the spam signature, so more spam is detected. You do this by entering the following command at the Router1 prompt:

```
Router1(config)#ip audit smtp spam 150
```

The spam signature value can be set to its default of 250 again by using the following command. Although it looks like you are disabling the spam signature, you are, in fact, resetting the threshold value back to 250 recipients.

```
Router(config)#no ip audit smtp
```

The spam signature can be disabled using the *ip audit signature* command.

```
Router1(config)#ip audit signature 3106 disable
```

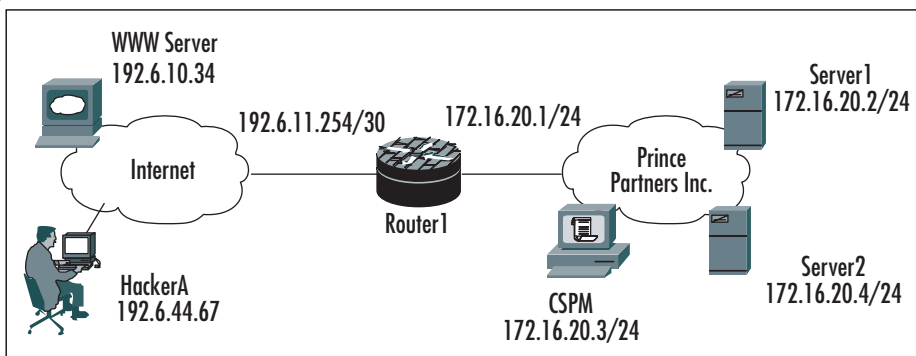
The *show ip audit configuration* command will show you that signature 3106 is disabled. The default threshold value still shows up in the output, but IDS will now ignore e-mails with recipient lists over 250 recipients. The following shows part of the *show ip audit configuration* command output after disabling the spam signature.

```
Default threshold of recipients for spam signature is 250  
Signature 3106 disable
```

Responses from the IOS-Based IDS

At this point, we have seen how to configure IOS-based IDS and in the next section we will see how to verify and monitor a configuration. What we haven't seen so far is Cisco IOS-based IDS in action. What happens when the router sends an alarm, performs a TCP reset, or drops a packet? We will walk through a number of examples to see how IOS-based IDS responds to intrusions.

Figure 11.4 shows how the LAN of Prince Partners Inc. is connected to the Internet via Router1.

Figure 11.4 The Prince Partners Inc. LAN

Router1 is configured for IDS and has all IDS signatures enabled. A Syslog server is used to send the alarm notification messages; there is no Director or IDS sensor in play. The following shows the IDS configuration of RouterA. We will use this setup in all examples. To get a good view on what the router does when detecting an intrusion, we will use debug commands when necessary. Router1 will send all messages, including debug messages, to the Syslog server on 172.16.20.2.

```

!
ip audit info action alarm
ip audit attack action alarm drop reset
ip audit notify log
ip audit po max-events 100
ip audit protected 172.16.20.1 to 172.16.20.254
ip audit name idstest info action alarm
ip audit name idstest attack action alarm drop reset
!
interface Serial 1/0
 ip address 192.6.11.254 255.255.255.252
 ip audit idstest in
!
interface Ethernet 0/0
 ip address 172.16.20.1 255.255.255.0
!
logging trap debugging
logging 172.16.20.2
!

```

In Figure 11.4, a WWW server is connected to the Internet. When we execute a PING from Server1 to this WWW server, we expect to get a reply back. Router1 will track this reply and compare it with its signature base. An echo reply triggers informational signature 2000. Since this is an informational signature, Router1 only sends an alarm notification to the Syslog server.

```
%IDS-4-ICMP_ECHO_REPLY_SIG: Sig:2000:ICMP Echo Reply - from 192.6.10.34 to
172.16.20.2
```

In Figure 11.4, we also see HackerA connected to the Internet. HackerA is trying to get more information on Server1 and is performing a UDP port scan. A hacker uses a UDP port scan to determine which UDP ports are open on a host. It works by sending 0-byte UDP packets to each port on the target host. If the hacker receives an ICMP port unreachable message, then the port is closed. Otherwise, he will assume the port is open.

To get more information from the router about the action it takes after detecting the attack, we enable the debug command *debug ip audit detailed*. Be careful when using this command in a production environment. While HackerA is running his UDP port scan, what we see in Syslog is the following:

```
May 31 13:45:43 75607: IDS UDP Signature - UDP IOS BOMB (Sig: 76)
May 31 13:45:43 75608: %IDS-4-UDP_IOS_BOMB_SIG: Sig:4600:UDP IOS Bomb -
from 192.6.10.241 to 172.16.20.2
May 31 13:45:43 75609: IDS* Interface Ethernet1/0 Pak 0x816295D0 audit
(on input) completed, dropping
May 31 13:45:43 75611: IDS UDP Signature - UDP IOS BOMB (Sig: 76)
May 31 13:45:43 75612: IDS* Interface Ethernet1/0 Pak 0x816295D0 audit
(on input) completed, dropping
May 31 13:45:43 75614: IDS UDP Signature - UDP IOS BOMB (Sig: 76)
May 31 13:45:43 75615: IDS* Interface Ethernet1/0 Pak 0x816295D0 audit
(on input) completed, dropping
```

Router1 has detected the port scan and identifies it as signature 4600, a UDP IOS Bomb. The router then sends an alert to the Syslog server and starts dropping the offending UDP packets.

While the scan is taking place we use the *show ip audit statistics* command to get a better idea of what is going on and see how many times a signature has been triggered.

```
Router1#show ip audit statistics
```



```
Signature audit statistics [process switch:fast switch]
  signature 1101 packets audited: [0:98]
  signature 2004 packets audited: [0:11]
  signature 4600 packets audited: [0:720]
Interfaces configured for audit 1
Session creations since subsystem startup or last reset 845
Current session counts (estab/half-open/terminating) [0:0:0]
Maxever session counts (estab/half-open/terminating) [0:4:0]
Last session created 00:09:39
Last statistic reset never
```

```
Host ID:2, Organization ID:100, SYN pkts sent:218,
ACK pkts sent:3, Heartbeat pkts sent:14085, Heartbeat ACK pkts sent:7114,
Duplicate ACK pkts received:0, Retransmission:0, Queued pkts:0
```

The output of the *show ip audit statistics* command shows that the IOS-IDS has audited 720 IOS/UDP Signature related packets so far.

Designing & Planning...

Increasing Security and Availability

To provide an increased level of security and to ensure notification of an attack, you can configure the Cisco IOS-IDS to send its alarms to multiple Directors or Syslog servers. This way, if there's a DoS attack or an outage of the Director or Syslog server, you can be sure that alarms are being received at an alternate location and the administrator is notified of an intrusion.

Verifying the IOS-IDS Configuration

A working and well-tested IDS can be very important for the continuity of your business. It ensures all attacks IOS has a signature for are being detected and that alerts are sent to the right place. In this section, we discuss how you can verify and test an IOS-based IDS configuration. We will see examples of commands

you can use to verify the working of your IDS. In addition, we look at how to troubleshoot an IDS configuration. The commands and items we will discuss include:

- *show ip audit interfaces*
- *show ip audit configuration*
- *show ip audit statistics*
- *show ip audit session*
- *show ip audit debug*
- *clear ip audit statistics*
- *clear ip audit configuration*
- debug commands

show ip audit interfaces

The *show ip audit interfaces* EXEC command is used to display the interface configuration. Figure 11.5 shows an example of the output of this command.

Figure 11.5 The *show ip audit interfaces* Command

```
Router#show ip audit interfaces
Interface Configuration
Interface Ethernet1/0
  Inbound IDS audit rule is idstest
    info actions alarm
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
```

In Figure 11.5, the *audit rule idstest* is applied to interface Ethernet1/0 on an inbound direction. When an informational signature is triggered by certain activity, the router sends an alarm to the configured Syslog or Director. When an attack signature is triggered, an alarm is sent, the packet is dropped, and in case of a TCP session, the session is reset. There is no audit rule applied in an outbound direction.

show ip audit configuration

The *show ip audit configuration* EXEC command is used to display an overview of configuration information. It includes information not shown using the *show running-config* command, like the default values of certain parameters. Figure 11.6 shows an example of the output of this command.

Figure 11.6 The *show ip audit configuration* Command

```
Router#show ip audit configuration
Event notification through syslog is enabled
Event notification through Net Director is disabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm drop reset
Default threshold of recipients for spam signature is 250
PostOffice:HostID:0 OrgID:0 Msg dropped:0
      :Curr Event Buf Size:0  Configured:100
Post Office is not enabled - No connections are active
Audit Rule Configuration
  Audit name idstest
    info actions alarm
    attack actions alarm drop reset
```

Figure 11.6 is an example of how the output of the *show ip audit configuration* command looks when only the log notification type is used and no PostOffice parameters are configured. As you can see, event notification through the Director is disabled, and PostOffice communications is not enabled.

Figure 11.7 shows the command output of another IOS-IDS sensor.

Figure 11.7 The *show ip audit configuration* Command

```
Router#show ip audit configuration
Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm drop reset
Default threshold of recipients for spam signature is 300
Signature 1107 disable
PostOffice:HostID:1 OrgID:100 Msg dropped:0
```

Continued

Figure 11.7 The *show ip audit configuration* Command

```

:Curr Event Buf Size:100  Configured:100
Host ID:2, Organization ID:100, SYN pkts sent:1,
ACK pkts sent:2, Heartbeat pkts sent:82, Heartbeat ACK pkts sent:49,
Duplicate ACK pkts received:0, Retransmission:0, Queued pkts:0
ID:1 Dest:172.16.20.2:45000 Loc:172.16.20.1:45000 T:5 S:ESTAB *

Audit Rule Configuration
  Audit name idstest
    info actions alarm
    attack actions alarm drop reset

```

The first thing we see when looking at Figure 11.7 is that event notification through Syslog and Director are both enabled. This means that each time a signature is triggered an alarm is sent to both locations. The default actions for informational and attack signatures are set and the threshold of recipients for the spam signature has been set to 300. We also see that signature 1107 has been disabled.

In the next section of output, we find PostOffice settings, the current configured notification queue size, and statistics on packets sent between the IOS-IDS sensor and the Director. Using this data, you can verify the communication between the IOS-IDS sensor and the Director. The line ending with the word ESTAB * tells you that a session between IOS-IDS sensor and Director has been established. If you find the word SYN SENT at the end of this line, it means the IOS-IDS sensor tried to set up a session but the Director is not answering, or that the set up of the session has not yet been completed. Figure 11.8 shows an example of the output of the *show ip audit configuration* command in this situation.

Figure 11.8 The *show ip audit configuration* Command

```

Router#show ip audit configuration
Event notification through syslog is enabled
Event notification through Net Director is enabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm drop reset
Default threshold of recipients for spam signature is 300
Signature 1107 disable
PostOffice:HostID:1 OrgID:20 Msg dropped:0
:Curr Event Buf Size:100  Configured:100
Host ID:2, Organization ID:100, SYN pkts sent:573,

```

Continued

Figure 11.8 The *show ip audit configuration* Command

```
ACK pkts sent:0, Heartbeat pkts sent:0, Heartbeat ACK pkts sent:0,
Duplicate ACK pkts received:0, Retransmission:0, Queued pkts:0
  ID:1 Dest:172.16.20.2:45000 Loc:172.16.20.1:45000 T:5 S:SYN SENT
```

```
Audit Rule Configuration
  Audit name idstest
    info actions alarm
    attack actions alarm drop reset
```

The output of the *show ip audit configuration* command ends with the audit rule configuration on the router. In Figure 11.6 through 11.8, we see that an audit rule with the name *idstest* has been configured on this router, plus what actions have been configured for the information and attack signatures under that rule.

show ip audit statistics

The *show ip audit statistics EXEC* command displays the number of packets audited plus the number of alarms sent. Figure 11.9 shows an example of the output of this command.

Figure 11.9 The *show ip audit statistics* Command

```
Router#show ip audit statistics
Signature audit statistics [process switch:fast switch]
  signature 1101 packets audited: [0:98]
  signature 2004 packets audited: [0:11]
  signature 3050 packets audited: [145:0]
  signature 4050 packets audited: [0:720]
Interfaces configured for audit 1
Session creations since subsystem startup or last reset 2729
Current session counts (estab/half-open/terminating) [0:0:0]
Maxever session counts (estab/half-open/terminating) [0:4:0]
Last session created 00:09:39
Last statistic reset never

Host ID:2, Organization ID:100, SYN pkts sent:218,
ACK pkts sent:3, Heartbeat pkts sent:14085, Heartbeat ACK pkts sent:7114,
Duplicate ACK pkts received:0, Retransmission:0, Queued pkts:0
```

Figure 11.9 shows a number of intrusions detected by IOS-IDS. For instance, signature 3050 has been triggered several times, meaning a half-open SYN attack has been detected. Further, there are some session counters and statistics on PostOffice communications.

show ip audit sessions

The *show ip audit sessions* EXEC command is used to display the current sessions on the IOS-IDS sensor. This command can be helpful when troubleshooting or verifying the working of the IDS. Figure 11.10 shows the output of the command at the moment a user is checking some POP3 e-mail accounts.

Figure 11.10 The *show ip audit sessions* Command

```
Router#show ip audit sessions
Established Sessions
  Session 813635E4 (172.16.20.2:4071)=>(192.6.6.40:110) tcp SIS_OPEN

Terminating Sessions
  Session 81363CC8 (172.16.20.2:4070)=>(192.6.196.44:110) tcp SIS_CLOSING

Router#show ip audit sessions
Terminating Sessions
  Session 81363CC8 (172.16.20.2:4070)=>( 192.6.196.44:110) tcp SIS_CLOSING
  Session 813635E4 (172.16.20.2:4071)=>(192.6.6.40:110) tcp SIS_CLOSING
```

show ip audit debug

The *show ip audit debug* EXEC command is used to display the debug commands that have been enabled on the router. An example of the output of this command is shown in Figure 11.11.

Figure 11.11 The *show ip audit debug* Command

```
Router#show ip audit debug
IDS Detailed Debug debugging is on
IDS TCP Audit debugging is on
```

The same result can be attained using the *show debug* command, but that will show all debug commands enabled on the router, while the *show ip audit debug* command displays only the ip audit debug commands enabled. An example of the *show debug* command output is shown in Figure 11.12.

Figure 11.12 The *show debug* Command

```
Router#show debug
Generic IP:
    IP packet debugging is on for access list 101
IDS Audit:
    IDS Detailed Debug debugging is on
    IDS TCP Inspection debugging is on
```

clear ip audit statistics

The *clear ip audit statistics* EXEC command is used to reset statistics on packets that have been audited and the number of alarms sent. To perform this action, type the command at the router prompt as follows:

```
Router#clear ip audit statistics
```

This command becomes useful when troubleshooting an IDS configuration and you want to start with fresh statistics.

clear ip audit configuration

The *clear ip audit configuration* EXEC command can be used to disable IOS-based IDS. The command removes all IDS configuration entries and releases dynamic resources IDS has in use. To clear the existing IP audit configuration, type the command at the router prompt as follows:

```
Router#clear ip audit configuration
```

Debug Commands

A number of debug commands are available to troubleshoot and test your IDS configuration. A combination of alarms sent by the sensor and certain debug commands is very helpful in testing the quality of your IDS configuration. We saw an example of this earlier in the section, “Responses from the IOS-Based

IDS,” where we combined alarms with the *debug ip audit detailed* command. The following list shows the available ip audit debug commands in Cisco IOS; the last two commands are new in IOS 12.2.

- ***debug ip audit detailed*** The *debug ip audit detailed* command enables IDS detailed debugging. Using this command, we see how IDS handles a packet: Does it forward or drop the packet? In the previous section, we saw an example of this command in action. It can also be used in combination with other debug ip audit commands to get additional information.
- ***debug ip audit ftp-cmd*** This command enables IDS FTP command and response debugging. The output of this command shows messages about IDS-audited FTP command and response events.
- ***debug ip audit ftp-token*** This command enables IDS FTP tokens debugging and is best used in combination with the *debug ip audit ftp-cmd* command. It enables tracing of the ftp tokens parsed.
- ***debug ip audit function-trace*** Using this command enables IDS function trace debugging, and creates a lot of output. The messages displayed relate to software functions called by IDS.
- ***debug ip audit icmp*** The *debug ip audit icmp* command enables IDS ICMP packet debugging. The output of the command shows ICMP echo requests and replies.
- ***debug ip audit ip*** This command enables IDS IP packet debugging
- ***debug ip audit object-creation*** Using this command enables IDS Object Creations debugging. The command’s output shows messages about software objects created by IDS. Object creation refers to the beginning of an IDS-audited session.
- ***debug ip audit object-deletion*** The *debug ip audit object-deletion* command enables IDS Object Deletions debugging. The command’s output shows messages about software objects deleted by IDS. Object deletion refers to the closing of IDS-audited sessions.
- ***debug ip audit rpc*** This command enables IDS RPC Inspection debugging. The command’s output shows messages about IDS-audited RPC events, including details about RPC packets.
- ***debug ip audit smtp*** Using this command enables IDS SMTP Inspection debugging and the output shows messages about IDS-audited

SMTP events. One of these events is the check for the spam signature, where IDS checks the number of recipients and thereupon permits or denies the message.

- ***debug ip audit tcp*** The *debug ip audit* command enables IDS TCP Inspection debugging. The command's output displays messages about IDS-audited TCP events, including details about TCP packets. It shows every ACK and SYN that passes through.
- ***debug ip audit tftp*** This command enables IDS TFTP Inspection debugging. The output of this command displays messages about IDS-audited TFTP events.
- ***debug ip audit timers*** The *debug ip audit timers* event enables the debugging of IDS timer event.
- ***debug ip audit udp*** This command enables IDS UDP Inspection debugging. The output of this command shows messages about IDS-audited UDP events, including details about UDP packets.
- ***debug ip audit dns*** Using this command enables IDS DNS Inspection debugging. Output of this command displays messages about IDS-audited DNS events.
- ***debug ip audit http*** The *debug ip audit http* command enables IDS HTTP Inspection debugging. The output of this command shows messages about IDS-audited HTTP events.



WARNING

Use these debug commands with caution on a production system. Some of the commands generate a lot of output and consume available CPU cycles, possibly causing a router to hang.

Summary

In this chapter, we learned how Cisco IOS can support intrusion detection using the IOS Firewall/IDS code. An IOS-IDS sensor tracks and audits all traffic that flows through the router. The number of signatures enabled, the type of signatures enabled, the quantity of traffic that flows through the router, and the router platform itself all influence the performance of IOS-based IDS.

The Firewall/IDS feature set of Cisco IOS is supported by router platforms like the 1700, 2600, 3600, 3700, 7100, 7200, 7400, and 7500 Series routers. Cisco IOS currently supports 100 signatures in the latest IOS releases. Originally, Cisco IOS supported 59 signatures. Signatures can be updated by loading the latest IOS release on the router. Custom signatures are not supported. A Cisco IOS-IDS sensor responds to an intrusion using one or more of the configured actions, like sending an alarm, dropping the offending packet, or resetting a TCP session. Cisco IOS-based IDS is configured using a six-step process:

- Configuring the notification type
- Configuring local and remote PostOffice parameters
- Configuring protected networks
- Changing the size of the notification queue
- Setting the default signature action
- Creating and applying audit rules

Local and remote PostOffice parameters only need to be configured when the nr-director notification type is used. A Cisco IOS-IDS sensor can send its alarm notifications to a Syslog server, a Director, or an IDS sensor. The signatures can be disabled; host or networks can be excluded from a signature to improve performance of the IOS-IDS sensor and to reduce the chance of false positives being triggered. A number of show, clear, and debug commands are available to verify and test the Cisco IOS-IDS configuration.

Solutions Fast Track

Understanding Cisco IOS-Based IDS

- ☑ The Firewall/IDS feature set of IOS is now supported on the 3700 and 7400 Series routers.
- ☑ Compound signatures have a bigger influence on IDS performance than atomic signatures.
- ☑ An IOS-IDS sensor uses 32KB of memory for every alarm.
- ☑ Testing an IDS configuration for overall performance and to verify intrusion response actions before going into production is recommended.

Configuring the IOS-Based IDS

- ☑ IOS-based IDS can send alarm notifications to Syslog, Director, or IDS sensor.
- ☑ When using Syslog as a notification type, it is not necessary to configure the local and remote PostOffice parameters.
- ☑ The host-id must be a unique number within the group to which the host belongs and is used to identify the host in PostOffice communications.
- ☑ Configuring a protected network is optional. If you don't configure a protected network, the router will consider all addresses as outside the protected network.
- ☑ The notification queue size determines the amount of alarms a router can track at a given time.

Configuring IOS-Based IDS Signatures

- ☑ False positives take place when an IDS reports specific friendly activity as malicious, requiring human intervention to diagnose the event.
- ☑ Fine-tuning your IDS configuration using the *ip audit signature* commands reduces the burden of administrators.

- ☑ Signatures are updated by loading the latest available IOS release.
- ☑ Spam can be identified by the 3106 – Mail Spam signature.

Responses from the IOS-Based IDS

- ☑ Use the debug commands with caution in a production environment.
- ☑ A combination of debug and show commands, plus the alarm notifications sent by the router, will give you an overview of how the router responds to an attack.
- ☑ Configuring NTP on a Cisco IOS-IDS is recommended.

Verifying the IOS-IDS Configuration

- ☑ Use the `show ip audit configuration` command to troubleshoot the PostOffice communications between the IOS-IDS sensor and the Director.
- ☑ When you want to know how many intrusive packets have been audited, use the `show ip audit statistics` command.
- ☑ Using the `clear ip audit configuration` will disable IDS on the router.
- ☑ Use the `show ip audit sessions` command to see what sessions are being audited.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to www.syngress.com/solutions and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

- Q:** Is it possible to create a custom signature for IOS-IDS in case of a new threat for which no signature has yet been released?
- A:** No, this is not possible for IOS-IDS. IOS identifies a number of signatures that are taken up in the IOS image, by updating the IOS image to the latest version. This will still be a small number of signatures, however, compared to those available for the IDS sensor and IDSM.
- Q:** Does Cisco IOS-based IDS provide shunning functionality?
- A:** No, only the 4200 Series IDS Sensor and the IDSM support this. IOS-based IDS, however, sits in the packet's path and can therefore simply drop the offending packet(s).
- Q:** Will my IOS-IDS sensor identify the 6051 – DNS Zone Transfer signature?
- A:** It depends on the IOS version your router is running. Originally, IOS identified 59 signatures, but starting with IOS release 12.2(11)YU and the latest 12.2T IOS releases, IOS identifies 100 signatures, including signature 6051. The Cisco IOS-IDS Signature List can be found on Cisco.com at www.cisco.com/warp/public/cc/pd/iosw/prodlit/idssl_ds.pdf.
- Q:** How do I verify if my IOS-IDS is communicating with the Director?
- A:** Use the *show ip audit configuration* command. If the command's output shows the keyword *ESTAB **, a session has been established with the Director. If no session has been established, the keyword *SYN SENT* is displayed, as shown in Figure 11.8.
- Q:** I'm having trouble correlating Syslog messages I receive from my Cisco IOS-IDS sensors, what should I do?

- A:** It is very important that each IOS-IDS sensor is set with the same time. That way when an intrusion occurs that triggers alarms or other messages, the Syslog messages are sent with the correct timestamp, making it easier to trace the attacker. You can use the Network Time Protocol (NTP) for that purpose, the IOS-IDS sensor will retrieve its time from a time source and set its clock accordingly.
- Q:** I can't inspect my e-mail properly. What could be the problem?
- A:** CBAC can inspect SMTP but not extended SMTP (ESMTP). You can verify if the e-mail server uses ESMTP by connecting with telnet to the e-mail server on port 25 and viewing the header information.
- Q:** When I enable HTTP on the router, can I use HTTPS to connect?
- A:** No, the command *ip http server* only supports http, not https.
- Q:** I have a web server running on port 8080. The server has been a victim of a half-open SYN attack but I cannot find any record of it on my CSPM server. How is this possible?
- A:** Detection of a half-open SYN attack with signature 3050 is currently only possible for attacks aimed at a number of common TCP ports like ports 21, 23, 25, and 80.

Cisco IDS Sensor Signatures

IP Signatures 1000 Series

The 1000 series signatures examine IP options, IP fragmentation, and bad IP packets. IP headers are examined for correct IP options and fire alarms based on the content of the IP header. If the data contained within the IP header does not meet the requirements for IP headers these signatures fire an alarm. IP fragmentation signatures examine the fragments of a packet for suspicious activity. Bad IP packets focus on invalid or crafted packets.

- 1001-IP Options-Record Packet Route: This signature fires when an IP datagram is received with the IP option 7, Record Packet Route, set in the datagram.
- 1002-IP Options-Timestamp: This signature fires when an IP datagram is received with the IP option 4, Timestamp, set in the datagram.
- 1004-IP Options-Loose Source Route: This signature fires when an IP datagram is received with the IP option *Loose Source Route* (option 3) is set in the datagram.
- 1006-IP Options-Strict Source Route: This signature fires when an IP datagram is received with the IP option *Strict Source Routing* (option 2) is set in the datagram.
- 1100-IP Fragment Attack: This signature fires when IP datagrams are received with a offset value greater than 0 but less than 5 in the offset field.
- 1101-Unknown IP Protocol: This signature fires when an IP datagram is received with the protocol field set to 134 or greater.
- 1102-Impossible IP Packet: This signature fires when an IP packet arrives with source equal to destination address.
- 1103-IP Fragments Overlap: This signature is fired when two fragments contained within the same IP datagram have offsets that indicate that they share positioning within the datagram.
- 1104-IP Localhost Source Spoof: This signature fires when an IP packet with a address of 127.0.0.1 is detected.
- 1105-Broadcast Source Address: This signature fires when an IP packet with a source address of 255.255.255.255 is detected.

- 1106–Multicast IP Source Address: This signature fires when an IP packet with a source address of 224.x.x.x is detected.
- 1107–RFC 1918 Addresses Seen: Legitimate network traffic may cause this signature to fire. Verify if valid RFC1918 address ranges are in use on your internal networks.
- 1108–IP Packet with Protocol 11: This signature alarms upon detecting IP traffic with the protocol set to 11.
- 1109–Cisco IOS Interface DoS: This alarm will fire upon detecting a “specially crafted packet” that may wedge the IOS input queue if that IOS image is vulnerable.
- 1200–IP Fragmentation Buffer Full: This signature is fired when there is an extraordinary amount of incomplete fragmented traffic detected on the protected network.
- 1201–IP Fragment Overlap: This signature is fired when two fragments contained within the same IP datagram have offsets that indicate that they share positioning within the datagram.
- 1202–IP Fragment Overrun - Datagram Too Long: This signature fires when a reassembled fragmented datagram would exceed the declared IP data length or the maximum datagram length. Alarm level 5.
- 1203–IP Fragment Overwrite - Data is Overwritten: This signature fires when an IP fragment that overlaps a previous fragment. This behavior is consistent with the ‘Ping of Death’.
- 1204–IP Fragment Missing Initial Fragment: This signature fires when a datagram can not be reassembled due to missing initial data.
- 1205–IP Fragment Too Many Datagrams: This signature is fired when there is an excessive number of incomplete fragmented datagrams detected on the network.
- 1206–IP Fragment Too Small: This signature fires when any fragment other than the final fragment is less than 400 bytes, indicating that the fragment is likely intentionally crafted.
- 1207–IP Fragment Too Many Fragmentss: This signature is fired when there is an excessive number of fragments for a given datagram. This is most likely either a denial of service (DOS) attack or an attempt to bypass security measures.

- 1208-IP Fragment Incomplete Datagram: This signature fires when a datagram can not be fully reassembled due to missing data.
- 1220-Jolt2 Fragment Reassembly DoS Attack: This alarm will fire when multiple fragments are received, all claiming to be the last fragment of an IP datagram.
- 1300-TCP Segment Overwrite: This signature fires when one or more TCP segments in the same stream overwrite data from a one or more segments located earlier in the stream.

ICMP Signatures 2000 Series

The 2000 signature series applies to all, or most of, the traffic that is ICMP. ICMP is used for troubleshooting purposes. Although they are of use to administrators they pose a threat to the network if not monitored closely. Inbound ICMP traffic should be scrutinized and disabled if not specifically required.

- 2000-ICMP Echo Reply: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 0 (Echo Reply).
- 2001-ICMP Host Unreachable: This signature fires when an IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 3 (Host Unreachable).
- 2002-ICMP Source Quench: This signature fires when an IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 4 (Source Quench).
- 2003-ICMP Redirect: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 5 (Redirect).
- 2004-ICMP Echo Request: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8 (Echo Request).
- 2007-ICMP Timestamp Request: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13 (Timestamp Request).

- 2008-ICMP Timestamp Reply: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 14 (Timestamp Reply).
- 2011-ICMP Address Mask Request This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17 (Address Mask Request).
- 2012-ICMP Address Mask Reply: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 18 (Address Mask Reply).
- 2100-ICMP Network Sweep with Echo: This signature fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8 (Echo Request).
- 2101-ICMP Network Sweep with Timestamp: This signature fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13 (Timestamp Request).
- 2102-ICMP Network Sweep with Address Mask: This signature fires when IP datagrams are received directed at multiple hosts on the network with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17 (Address Mask Request).
- 2150-Fragmented ICMP Traffic: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and either the more fragments flag is set to 1 (ICMP) or there is an offset indicated in the offset field.
- 2151-Large ICMP Traffic: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the IP length is less than 1024.
- 2152-ICMP Flood: This signature fires when multiple IP datagrams are received directed at a single host on the network with the protocol field of the IP header set to 1 (ICMP).
- 2153-Smurf: This fires when a large number of ICMP Echo Replies are targeted at a machine.

- 2154–Ping of Death Attack: This signature fires when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP), the Last Fragment bit is set, and $(IP\ offset * 8) + (IP\ data\ length) > 65535$ that is to say, the IP offset (which represents the starting position of this fragment in the original packet, and which is in 8 byte units) plus the rest of the packet is greater than the maximum size for an IP packet. This indicates a DOS attack.
- 2155–Modem DoS: This signature fires when a series of three pluses (+) in an ICMP packet.

TCP Signatures 3000 Series

TCP signatures are specific to TCP activity. TCP requires a three-way-handshake and several of the signatures are compared to the TCP traffic on the network. Other activity that is examined is scans, sweeps, and attacks that attempt to make connections to systems using TCP over specific ports. Some of these signatures even take into consideration bad or abnormal TCP packets.

- 3001–TCP Port Sweep: This signature fires when a series of TCP connections to a number of different privileged ports (having port number less than 1024) on a specific host have been initiated.
- 3002–TCP SYN Port Sweep: This signature fires when a series of TCP SYN packets have been sent to a number of different destination ports on a specific host.
- 3003–TCP Frag SYN Port Sweep: This signature fires when a series of fragmented TCP SYN packets are sent to a number of different destination ports on a specific host.
- 3005–TCP FIN Port Sweep: This signature fires when a series of TCP FIN packets have been sent to a number of different privileged ports (having port number less than 1024) ports on a specific host.
- 3006–TCP Frag FIN Port Sweep: This signature fires when a series of fragmented TCP FIN packets have been sent to a number of different privileged ports (having port number less than 1024) destination ports on a specific host.

- 3010-TCP High Port Sweep: This signature fires when a series of TCP connections to a number of different high-numbered ports (having port number greater than 1023) on a specific host have been initiated.
- 3011-TCP FIN High Port Sweep: This signature fires when a series of TCP FIN packets have been sent to a number of different destination high-numbered ports (having port number greater than 1023) on a specific host.
- 3012-TCP Frag FIN High Port Sweep: This signature fires when a series of fragmented TCP FIN packets have been sent to a number of different destination high-numbered ports (having port number greater than 1023) on a specific host.
- 3015-TCP Null Port Sweep: This signature fires when a series of TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to a number of different destination ports on a specific host.
- 3016-TCP Frag Null Port Sweep: This signature fires when a series of fragmented TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to a number of different destination ports on a specific host.
- 3020-TCP SYN FIN Port Sweep: This signature fires when a series of TCP packets with both the SYN and FIN flags set have been sent to a number of different destination ports on a specific host.
- 3021-TCP Frag SYN FIN Port Sweep: This signature fires when a series of fragmented TCP packets with both the SYN and FIN flags set have been sent to a number of different destination ports on a specific host.
- 3030-TCP SYN Host Sweep: This signature fires when a series of TCP SYN packets have been sent to the same destination port on a number of different hosts.
- 3031-TCP Frag SYN Host Sweep: This signature fires when a series of fragmented TCP SYN packets have been sent to the same destination port on a number of different hosts.
- 3032-TCP FIN Host Sweep: This signature fires when a series of TCP FIN packets have been sent to the same destination port on a number of different hosts.

- 3033–TCP Frag FIN Host Sweep: This signature fires when a series of TCP FIN packets have been sent to the same destination port on a number of different hosts.
- 3034–TCP NULL Host Sweep: This signature fires when a series of TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to the same destination port on a number of different hosts.
- 3035–TCP Frag NULL Host Sweep: This signature fires when a series of fragmented TCP packets with none of the SYN, FIN, ACK, or RST flags set have been sent to the same destination port on a number of different hosts.
- 3036–TCP SYN FIN Host Sweep: This signature fires when a series of TCP packets with both the SYN and FIN flags set have been sent to the same destination port on a number of different hosts.
- 3037–TCP Frag SYN FIN Host Sweep: This signature fires when a series of TCP packets with both the SYN and FIN flags set have been sent to the same destination port on a number of different hosts.
- 3038–Fragmented NULL TCP Packet: This signature fires when a single fragmented TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host.
- 3039–Fragmented Orphaned FIN Packet: This signature fires when a single fragmented orphaned TCP FIN packet is sent to a privileged port (having port number less than 1024) on a specific host.
- 3040–NULL TCP Packet: This signature fires when a single TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host.
- 3041–SYN/FIN Packet: This signature fires when a single TCP packet with the SYN and FIN flags are set and is sent to a specific host.
- 3042–Orphaned FIN Packet: This signature fires when a single orphaned TCP FIN packet is sent to a privileged port (having port number less than 1024) on a specific host.
- 3043–Fragmented SYN/FIN Packet: This signature fires when a single fragmented TCP packet with the SYN and FIN flags are set and is sent to a specific host.

- 3045-Queso Sweep: This signature fires after having detected a FIN, SYN-FIN, and a PUSH sent from a specific host bound for a specific host.
- 3046-NMAP OS Fingerprint: This signature looks for a unique combination of TCP packets that the NMAP tool uses to fingerprint a remote operating system.
- 3050-Half-open SYN Attack: This signature fires when multiple TCP sessions have been improperly initiated on any of several well-known service ports.
- 3100-Smail Attack: This signature fires on the very common smail attack against e-mail servers.
- 3101-Sendmail Invalid Recipient: This signature fires on any mail message with a pipe (|) symbol in the recipient field.
- 3102-Sendmail Invalid Sender: This signature fires on any mail message with a pipe (|) symbol in the From: field.
- 3103-Sendmail Reconnaissance: This signature fires when expn or vrfy commands are issued to the SMTP port.
- 3104-Archaic Sendmail Attacks: This signature fires when wiz or debug commands are sent to the SMTP port.
- 3105-Sendmail Decode Alias: This signature fires on any mail message with decode@ in the header.
- 3106-Mail Spam: Counts number of Rcpt to: lines in a single mail message and alarms after a user-definable maximum has been exceeded. The user default is 250 recipients.
- 3107-Majordomo Execute Attack: A bug in the Majordomo program will allow remote users to execute arbitrary commands at the privilege level of the server.
- 3108-MIME Overflow Bug: This signature fires when an SMTP mail message has a MIME “Content-” field that is excessively long.
- 3109-Long SMTP Command: This signature fires when an attempt is made to pass an overly long command string to a mail server
- 3110-Suspicious Mail Attachment: A suspicious mail attachment was found in a mail message.

- 3111-W32 Sircam Malicious Code: Alarms when SirCam virus e-mail attachment is sent.
- 3111:1-W32 Sircam Malicious Code: Alarms when SirCam virus e-mail attachment is received.
- 3112-Lotus Domino Mail Loop DoS: Alarms when a To: field in the mail is detected greather than 100 characters
- 3114-FetchMail Arbitrary Code Execution: Alarms when an e-mail command containing a list of large integers is encountered.
- 3115-Sendmail Data Header Overflow: Alarms when an e-mail command containing a list of large integers is encountered.
- 3116-Netbus: Alarm fires upon detecting a Netbus communications channel setup.
- 3117-KLEZ Worm: The alarm fires when a filename gn.exe is found as a audio/x-wav attachment to an e-mail.
- 3118-rwhoisd Format String: This sig fires upon detecting a 'soa' command sent to a rwhois server with a large argument.
- 3119-WS_FTP STAT Overflow: This signature fires when a stat command with an argument that is greater than 450 characters.
- 3120-ANTS virus: The alarm fires when a e-mail is found with the attachment ants3set.exe
- 3121-Vintra MailServer EXPN DoS: This signature fires when '*@' is detected as the argument to the SMTP command *expn*.
- 3122-SMTP EXPN Root Recon: This signature fires when an attempt to expand the e-mail alias of the 'root' user with SMTP command *expn* is detected.
- 3123-NetBus Pro Traffic: Alarm fires upon detecting a Netbus Pro communications channel setup.
- 3124-Sendmail Prescan Memory Corruption: This signature looks for an abnormally long (1000+ characters). The subsignatures are:
 - SubSig 0: MAIL FROM
 - SubSig 1: RCPT TO

- 3150-FTP Remote Command Execution: This signature fires when someone tries to execute the `Ftp site` command.
- 3151-FTP SYST Command Attempt: This signature fires when someone tries to execute the `FTP SYST` command.
- 3152-FTP CWD ~root: This signature fires when someone tries to execute the `CWD ~root` command.
- 3153-FTP Improper Address Specified: This signature fires if a port command is issued with an address that is not the same as the requesting host.
- 3154-FTP Improper Port Specified: This signature fires if a port command is issued with a data port specified that is less than 1024 or greater than 65535.
- 3155-FTP RETR Pipe Filename Command Execution: The ftp client can be tricked into running arbitrary commands supplied by the remote server.
- 3156-FTP STOR Pipe Filename Command Execution: The ftp client can be tricked into running arbitrary commands supplied by the remote server.
- 3157-FTP PASV Port Spoof: Possible attempt has been made to open connections through a firewall to a protected FTP server to a non-FTP port.
- 3158-FTP SITE EXEC Format String: Affected versions of Wu-ftp are missing some character-formatting arguments in several function calls that implement the `site exec` command functionality.
- 3159-FTP PASS Suspicious Length: In order to exploit some Wu-ftpd vulnerabilities (sig3158), a malicious user must supply shell code in the password field of the ftp login.
- 3160-Cesar FTP Buffer Overflow: Alarms when a `HELP` command is followed by 200 or more characters
- 3161-FTP realpath Buffer Overflow: This signature fires when an attempt is detected to create or delete a directory during a FTP session using a path argument containing executable machine code, also known as shellcode.
- 3162-glFtpD LIST DoS: This signature fires when an abnormally long FTP `list` command is detected with an argument that is composed only of the character `'*'`.
- 3163-wu-ftpd Heap Corruption Vulnerability: This signature fires when an unbalanced `'{'` is detected in FTP traffic.

- 3164- Instant Server Mini Portal Directory Traversal: This signature fires when .../ is detected in a FTP connection.
- 3165-FTP SITE EXEC: This alarms when a SITE EXEC command is attempted within FTP traffic. There is a potential danger if the SITE EXEC command is allowed when FTP servers are incorrectly configured.
- 3166-FTP USER Suspicious Length: The signature fires when a longer than normal username is detected during an FTP session. This could cause a buffer overflow.
- 3167-Format String in FTP Username: This signature fires when a percent sign (%) is detected as a username argument of an ftp login. A percent signs indicate a format string attack when part of the username.
- 3168-FTP SITE EXEC Directory Traversal: This signature fires when a SITE EXEC command is attempted with arguments of a directory traversal (../) within the FTP traffic. There is a potential danger if the SITE EXEC command is allowed when ftp servers are incorrectly configured. Directory traversal attempts are indicators of command execution attacks.
- 3169-FTP SITE EXEC tar: This signature fires when a SITE EXEC command is attempted with arguments of an piped tar command in the FTP traffic. There is a potential danger if the SITE EXEC command is allowed when FTP servers are incorrectly configured. Piped tar command attempts are indicators of malicious traffic.
- 3170-WS_FTP SITE CPWD Buffer Overflow: This signature fires when it detects a SITE CPWD command with an argument greater than 100 characters in length.
- 3171-FTP Privileged Login: The signature fires when it detects an FTP login for a privileged user (root or administrator). Ftp activity with privileged users is dangerous because passwords are sent in the clear (plaintext) across the network.
- 3172-FTP CWD Overflow: This signature fires when it detects the FTP command CWD with abnormally long argument. This is a good sign of a buffer overflow attack.
- 3173-Long FTP Command: Normal FTP commands may cause false positives. If you receive false positives, you can tune the signature by increasing

the default value of the *MinMatchLength* parameter until false positives are eliminated.

- 3174-SuperStack 3 NBX FTP Dos: This signature fires when the FTP command *cel* is received with more than 2048 bytes of arguments.
- 3175-ProFTPD STAT DoS: This signature fires when a FTP STAT command has several ‘/*’ contiguous character combinations. This is a sign of a denial of service attack.
- 3176-Cisco ONS FTP DoS: This signature fires when a long “CEL” FTP command is detected.
- 3200-WWW phf Attack: This signature fires when the phf attack is detected. This is an indicator that an attempt has been made to illegally access system resources.
- 3201-Unix Password File Access Attempt: These alarms fire when any cgi-bin script attempts to retrieve password files on various operating systems. Examples of such password files are:
 - /etc/passwd (Sub ID 1)
 - /etc/shadow (Sub ID 2)
 - /etc/master.passwd (Sub ID 3)
 - /etc/master.shadow (Sub ID 4)
 - /etc/security/passwd (Sub ID 5)
 - /etc/security/opasswd (Sub ID 6)

Signature 3201 is a good indicator that illegal attempts are being made to access system resources.

- 3202-WWW .URL File Requested: This signature fires when a user attempts to get any .URL file. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .URL files are accessed using the HTTP GET command.
- 3203-WWW .LNK File Requested: This signature fires when a user attempts to get any .LNK file. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .LNK files are accessed using the HTTP GET command.

- 3204-WWW .BAT File Requested: This signature fires when a user attempts to get any .BAT file. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .BAT files are accessed using the HTTP GET command.
- 3205-HTML File Has .URL Link: This signature fires when a file has a .URL link. This signature sends a warning to the user before he/she can click on the damaging link. Signature 3202 will fire on any attempts to click on the link, but it can cause damage before defensive measures are taken. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .URL files are accessed using the HTTP GET command.
- 3206-HTML File Has .LNK Link: This signature fires when a file has a .LNK link. This signature sends a warning to the user before he/she can click on the damaging link. Signature 3203 will fire on any attempts to click on the link, but it can cause damage before defensive measures are taken. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .LNK files are accessed using the HTTP GET command.
- 3207-HTML File Has .BAT Link: This signature fires when a file has a .BAT link. This signature sends a warning to the user before they can click on the damaging link. Signature 3204 will fire on any attempts to click on the link, but it can cause damage before defensive measures are taken. There is a flaw in Microsoft Internet Explorer that could allow illegal access to system resources when .BAT files are accessed using the HTTP GET command.
- 3208-WWW Campas Attack: This signature fires when attempts are made to pass commands to the CGI program campas. A problem in the CGI program campas, included in the NCSA Web Server distribution, allows attackers to execute commands on the host machine. These commands will execute at the privilege level of the HTTP server.
- 3209-WWW Glimpse Server Attack: This signature fires when attempts are made to pass commands to the perl script GlimpseHTTP. These could allow attackers to execute commands on the host machine. The GlimpseHTTP is an interface to the Glimpse search tool.
- 3210-WWW IIS View Source Attack: If a request to a Microsoft IIS server is formatted in a certain way, executable files are read instead of being executed. Passwords, scripts, and database information can be revealed. Analysis

of the scripts could turn up vulnerabilities. This signature fires when a request is made to an HTTP server attempting to view the source.

- 3211-**WWW IIS Hex View Source Attack**: If a request to a Microsoft IIS server is formatted in a certain way, executable files are read instead of being executed. Passwords, scripts, and database information can be revealed. Analysis of the scripts could turn up vulnerabilities. This signature fires when a request is made to an HTTP server with an embedded escape code, %2E, in place of a ".". This is a sign someone is trying to view the source of a protected web page script.
- 3212-**WWW NPH-TEST-CGI Attack**: This signature fires when attempts are made to view directory listings with the script `nph-test-cgi`. Some but not all HTTP servers include this script. The script can be used to list directories on a server. This script is for testing purposes and should be removed on production servers.
- 3213-**WWW TEST-CGI Attack**: This signature fires when attempts are made to view directory listings with the script `test-cgi`. Some but not all HTTP servers include this script. The script can be used to list directories on a server. This script is for testing purposes and should be removed on production servers.
- 3214-**IIS DOT DOT VIEW Attack**: This signature fires on attempts to view files above the chrooted directory using Microsoft IIS. The result of this attack is the viewing of files not intended for public access. The chroot directory is supposed to be the topmost directory to which HTTP clients have access.
- 3215-**IIS DOT DOT EXECUTE Attack**: Fires on attempts to cause Microsoft IIS to execute commands. Valid URL requests can cause false positives. Verify the target system from where the signature is firing to see if it is vulnerable.
- 3216-**WWW Directory Traversal ../..:** This signature fires when attempts to traverse directories on the web server using "../.." are detected. This is a sign attempts are being made to gain access to files and directories outside the root directory of the Web server.

- 3217-**WWW PHP View File Attack**: This signature fires when someone attempts to use the PHP cgi-bin program to view a file. This is an indicator illegal attempts are being made to access system resources.
- 3218-**WWW SGI Wrap Attack**: This signature fires attempts to view or list files using a program called wrap. This was distributed with the IRIX Web Server. There could be legitimate uses that cause false positives. Validate its use.
- 3219-**WWW PHP Buffer Overflow**: This signature fires when an oversized query is sent to the PHP cgi-bin program. This is an indicator of a buffer overflow attack to gain system access.
- 3220-**IIS Long URL Crash Bug**: This fires when a large URL is sent to a Web server in attempts to crash the system.
- 3221-**WWW cgi-viewsource Attack**: This signature fires when someone attempts to use the cgi-viewsource script to view files above the HTTP root directory.
- 3222-**WWW PHP Log Scripts Read Attack**: This signature fires when someone attempts to use the PHP scripts mlog or mylog to view files on a machine.
- 3223-**WWW IRIX cgi-handler Attack**: This signature fires when someone attempts to use the cgi-handler script to execute commands.
- 3224-**HTTP WebGais**: This signature fires when someone attempts to use the webgais script to run arbitrary commands.
- 3225-**WWW websendmail File Access**: This signature fires when unauthorized attempts are made to read a file using the websendmail CGI program.
- 3226-**WWW Webdist Bug**: This signature fires when attempts are made to use the webdist program. False positive alarms will fire from legitimate use of the webdist program.
- 3227-**WWW Htmlscript Bug**: This signature fires when attempts are made to view files above the HTML root directory.
- 3228-**WWW Performer Bug**: This signature fires when attempts are made to view files above the HTML root directory.

- 3229–Website Win-C-Sample Buffer Overflow: This signature fires when attempts are made to access the win-c-sample program in the Web site server distribution. Testing new Web site servers or upgrades using the win-c-sample program can cause false positives. This script is for testing purposes and should be removed on production servers.
- 3230–Web Site Uploader: This signature fires when attempts are made to access the uploader program in the Web site server distribution.
- 3231–Novell Convert: This signature fires when a user has attempted view files illegally using the convert.bas program included with Novell web server distribution.
- 3232–WWW finger attempt: This signature fires when an attempt is made to run the finger.pl program using the http server. Legitimate use can cause false positives. Unneeded CGI scripts should be removed from the cgi-bin directory.
- 3233–WWW count-cgi Overflow: This signature fires when attempt are made to cause a buffer overflow in the cgi count program.
- 3250–TCP Hijack: This signature fires when both data streams of a TCP connection indicate that TCP hijacking has occurred. TCP Hijacking is used to gain illegal access to system resources. False positives are possible.
- 3251–TCP Hijacking Simplex Mode: This signature fires when both data streams of a TCP connection indicate that TCP hijacking has occurred. TCP Hijacking is a method used to gain illegal access to system resources. Simplex mode means that only one command is sent, followed by a connection RESET packet. This is the discriminating factor between signature 3251 and 3250. False positives are possible. The most common network event that may trigger this signature is an idle telnet session. The TCP Hijack attack is a low-probability, high level-of-effort event. If it is successfully launched it could lead to serious consequences, including system compromise. The source of these alarms should be investigated thoroughly before any actions are taken. Recommend security professional consultation to assist in the investigation.
- 3300–NetBIOS OOB Data: This signature fires when an attempt to send data Out Of Band to port 139 is detected. This can be used to crash Windows machines.

- 3303–Windows Guest Login: This signature fires when a client establishes a connection to an SMB server (WinNT or Samba), it provides an account name and password for authentication. If the server does not recognize the account name, it may log the user in as a guest. This is optional behavior by the server and guest privileges should be limited. As a general security precaution, users should not be allowed access as guest.
- 3305–Windows Password File Access: This signature fires when a client attempts to access a .PWL on Windows 95 or other servers. The .PWL files is the password file.
- 3306–Windows Registry Access: This signature fires when a client attempts to access the registry on the Windows server. False positives are possible because every attempt to access the registry will cause an alarm to fire.
- 3307–Windows RedButton Attack: This signature fires when the RedButton tool is run against a server. The tool is use to show the security flaw in Windows NT 4.0 that allows remote registry access without a valid user account.
- 3308–Windows LSARPC Access: This signature fires when an attempt has been made to access the LSARPC service on a Windows system. When the source is from an external source, the traffic should be considered suspect. LSARPC can be used to gather system information that would be useful in launching subsequent attacks.
- 3309–Windows SRVSVC Access: This signature fires when an attempt is made to access the SRVSVC on a Windows system. SRVSVC may be used to gather system information that would be useful in launching subsequent attacks.
- 3310–Netbios Enum Share DoS: This signature fires when a malformed netbios enum share packet.
- 3311–SMB: Remote SAM Service Access Attempt: This signature fires when an attempt has been made to access the SAM security service on a Windows system. This service may be used to gather system information that would be useful in launching subsequent attacks. This is normal traffic on Windows networks and is included as an informational signature.

NOTE

Signature 3311 is only available in Cisco IDS versions 4.0 and newer.

- 3312-SMB .EML E-mail File Remote Access: This signature fires on any attempt to create or open a remote file with a .EML file extension. The NIMDA worm and variants drop files with the .EML e-mail file extension on open remote shares.

NOTE

Signature 3312 is only available in Cisco IDS versions 4.0 and newer.

- 3313-SMB Suspicious Password Usage: This signature fires because the client portion of an SMB login or authentication transaction uses passwords in the clear.

NOTE

Signature 3313 is only available in Cisco IDS versions 4.0 and newer.

- 3314-Windows Locator Service Overflow: This signature fires when attempts are made to pass an extremely long name to the Windows Locator service. This is a sign of a buffer overflow attack. Normal SMB traffic can cause false positives. In most cases only domain controllers are vulnerable.
- 3320-SMB: ADMIN\$ Hidden Share Access Attempt: This signature fires when attempts are made to connect to the hidden windows administration share ADMIN\$. This share point does not appear in normal browsing and may access attempts are indicators that an attempt to break into the system is occurring.

NOTE

Signature 3320 is only available in Cisco IDS versions 4.0 and newer.

- 3321-SMB: User Enumeration: A Microsoft Remote Procedure Call (MSRPC) system call has been made to enumerate the users on the target machine. This is normal Windows NT/2000/XP network activity. It should be considered suspect if it occurs from a source outside of your network.

NOTE

Signature 3321 is only available in Cisco IDS versions 4.0 and newer.

- 3322-SMB: Windows Share Enumeration: A remote network call has been made to Microsoft Windows' built-in resource enumeration interface. This interface is used to browse or otherwise enumerate resources being advertised to the network. Normal Windows browsing will cause false positives. It should be considered suspect if it occurs from a source outside of your network.

NOTE

Signature 3322 is only available in Cisco IDS versions 4.0 and newer.

- 3323-SMB: RFPoison Attack: This signature fires when a specially malformed share enumeration request is made. The attacker can cause the Service Control Manager (Server service) to misbehave and access illegal memory areas. The result is the server service being terminated, creating a denial of service in the loss of remote services to the affected machine including services that use named pipes.

NOTE

This signature is only available in Cisco IDS versions 4.0 and newer.

- 3324-SMB NIMDA infected file transfer: The NIMDA worm creates a file name desktop.eml on remote accessible shares as a means of propagation. This signature fires when an attempt to create or open remote file with the specific name of desktop.eml. False positives can be generated only when a remote file with the name desktop.eml is accessed.

NOTE

Signature 3324 is only available in Cisco IDS versions 4.0 and newer.

- 3325-Samba call_trans2open Overflow: This signature fires when a buffer overflow attempt to exploit the call_trans2open function of Samba is detected.
- 3326-Windows Startup Folder Remote Access: This signature fires when SMB access to the Windows startup folder is accessed. Many Internet worms copy themselves into the startup folder as a way to propagate themselves. A good indicator that a machine is infected with an Internet worm is if the particular machine is generating a lot of alarms.
- 3327-Windows RPC DCOM Overflow: This signature fires when a potential buffer overflow attempt against a Windows DCOM RPC service is detected. This could be an indicator there has been a system compromise. SubSig 0: \00\<400 chars>\ port 135tcp SubSig 1: \00\<400 chars>\ port 135udp SubSig 2: RPC over SMB, overflow packet port 139 SubSig 3: RPC over SMB, overflow packet port 445
- 3328-Windows SMB/RPC NoOp Sled: This signature fires when 10 or more consecutive hexadecimal “90” characters (Intel NoOp assembly instructions) are seen in TCP-based Windows SMB / RPC traffic. This activity is an indicator of a buffer overflow attack.

- 3400-Sunkill: This signature fires when an attempt is made to cause the telnetd server to lock up. This will catch the program known as sunkill.
- 3401-Telnet-IFS Match: Fires on when an attempt to change the IFS to / is done during a telnet session. This is an indicator an attempt is made to gain unauthorized access to system resources.
- 3402-BSD Telnet Daemon Buffer Overflow: This signature fires when an abnormally long 'New Environment Variable' telnet option is detected. Telnet daemons derived from the BSD source contain a buffer overflow in the handling of telnet options.
- 3403-Telnet Excessive Environment Options: This signature fires when an excessive number of environment variables are exchanged during a telnet session.
- 3404-SysV /bin/login Overflow: This signature fires when an excessive number of environment variables are sent to the 'login' program during a telnet session.
- 3405- Avirt Gateway Proxy Buffer Overflow: This signature fires when a string over 400 bytes is detected containing LoadLibraryRef call in a Telnet session.
- 3406-Solaris TTYPROMPT /bin/login Overflow: This signature fires when the environmental variable TTYPROMPT is detected during the negotiation of telnet options. This variable should not be seen on the network and should be considered an indicator of a buffer overflow attack.
- 3450-Finger Bomb: This signature fires when it detects a finger bomb attack. This particular attack attempts to crash a finger server by issuing a finger request that contains multiple "@" characters. If the finger server allows forwarding, then the multiple @s will cause the finger server to recursively call itself and use up system resources.
- 3451-BearShare Directory Traversal: This signature fires if a directory traversal (.....) is sent on the TCP port of 6346.
- 3452-gopherd halidate Overflow: This signature fires when a request "halidate <600+characters>" is sent to a gopher server.
- 3453-MS NetMeeting RDS DoS: This signature fires when a large number of NULL bytes are detected being sent to the Microsoft NetMeeting

Remote Desktop Sharing server port (TCP 1720). Legitimate traffic could cause false positives.

NOTE

HTTP traffic is the normal cause for this signature to misfire, but other protocols can also cause it to fire. This issue will be corrected in version 4.0 of the sensor.

- 3454-Check Point Firewall Information Leak: This signature fires when a TCP request to port 256 or 264 is detected with topologyrequest. Authenticated requests can also cause the signature to fire.
- 3455-Java Web Server Cmd Exec: This signature fires if /servlet/com.sun.server.http.pagecompile.jsp92.jspervlet is accessed. Administrators can cause false positives by accessing this file.
- 3456- Solaris in.fingerd Information Leak: This signature fires when an attempt to retrieve excessive information using the finger protocol is detected. SubSig 0: 'a b c d e f g h'@sunhost SubSig 1: 0@sunhost
- 3457-Finger Root Shell: This alarm will fire upon detecting the string cmd_rootsh in finger traffic. cmd_rootsh is a backdoor known to run on the finger port.
- 3458-AIM Game Invite Overflow: This signature alarms upon detecting an unusually long online game invite using AOL instant messenger.
- 3459-ValiCert Forms.exe Overflow: This signature fires upon detecting a large argument value sent to the file forms.exe on port 13333.
- 3460-AVTronics InetServer Buffer Overflow: Alarms when a TCP String containing "Authentication Basic" is followed more than 125 characters
- 3461-Finger Probe: This signature alarms upon detecting a zero '0' sent to a finger port. This type of activity is indicative of finger probing. Since finger is a useful recon tool for attackers a finger probe is commonly sent to detect active finger daemons.

- 3462-Finger Redirect: This signature alarms upon detecting an '@' sign in a finger request. An '@' in a finger request means a finger redirect is occurring. A finger redirect shouldn't be seen on today's modern networks as finger is a dangerous recon tool for attackers.
- 3463-Finger Root: This signature fires when root is fingered. This type of activity is a good indicator that an attacker is trying to gather recon information for use in future attacks.
- 3464-File Access in Finger: This signature fires upon detecting the string /etc/ on the finger port. There is no reason /etc/ would be seen in normal finger usage. This indicates backdoor activity on the finger port.
- 3465-Finger Activity: This signature fires upon detecting network traffic using the finger service.
- 3500-Rlogin -froot Attack: This signature fires when an attempt to rlogin with the arguments -froot has been made. A flaw in some rlogin processes allow unauthorized root access and a system compromise could be the result.
- 3501-Rlogin Long TERM Variable: This signature fires when an excessively long TERM environment variable is detected during the negotiation of an rlogin session.
- 3502-rlogin Activity: This signature fires upon detecting network activity destined to the rlogin port (513).
- 3525-IMAP Authenticate Buffer Overflow: This signature fires on receipt of packets bound for port 143 that are indicative of an attempt to overflow a buffer in the IMAP daemon. This is an indicator of an attempt to gain unauthorized access to system resources.
- 3526-Imap Login Buffer Overflow: This signature fires on receipt of packets bound for port 143 that are indicative of an attempt to overflow the imapd login buffer. This is an indicator of an attempt to gain unauthorized access to system resources.
- 3530-Cisco Secure ACS Oversized TACACS+ Attack: This signature fires when an oversized TACACS+ packet is sent to certain Cisco Secure ACS for NT versions and causes the server to crash. False positives can occur when hosts using the pluggable authentication module (PAM) pam_tacacs for authentication is used.

- 3540–Cisco Secure ACS CSAdmin Attack: This signature fires when a large request is made to the CSAdmin service which listens on TCP port 2002.
- 3550–POP Buffer Overflow: This signature fires on receipt of packets bound for port 110. This is an indicator an attempt to overflow the POP daemon user buffer is occurring. This is an indicator of an attempt to gain unauthorized access to system resources.
- 3551–POP User Root: This signature will fire when ‘ROOT’ is used as the user name to authenticate with POP3 mail server.
- 3575–INN Buffer Overflow: This signature fires when an attempt is made to overflow a buffer in the Internet News Server.
- 3576–INN Control Message Exploit: This signature fires when an attempt is made to execute arbitrary commands using the control message.
- 3600–IOS Telnet Buffer Overflow: This signature fires on receipt of packets bound for port 23 of a Cisco router that are indicative of attempt to crash the router by overflowing an internal command buffer. This is an indicator of an attempt to gain unauthorized access to system resources.
- 3601–IOS Command History Exploit: This signature fires on an attempt to force a Cisco router to reveal prior users command history.
- 3602–Cisco IOS Identity: This signature fires if someone attempts to connect to port 1999 on a Cisco router. This port is not enabled for access.
- 3603–IOS Enable Bypass: This signature fires when a successful attempt to gain privileged access to a Cisco Catalyst switch has been detected. Verify the configuration on the switch in question and ensure that the latest IOS release is installed.
- 3604–Cisco Catalyst CR DoS: This signature fires upon detecting a carriage return as the first character sent to TCP port 7161.
- 3650–SSH RSAREF2 Buffer Overflow: A buffer overflow is present in versions of SSH1, up to and including 1.2.27 that are compiled using —with-rsaref option. During key exchange, the RSAREF2 library does not bounds check the key length. A buffer overflow can occur on either client or server.
- 3651–SSH CRC32 Overflow: This signature fires upon detecting a crc overflow attempt.

- 3652-SSH Gobblers: This signature fires when a Gobblers implementation of the openSSH vulnerability is detected.
- 3700-CDE dtspcd overflow: This signature will fire if a buffer overflow attack to the CDE sub-process control daemon (**dtspcd**) on TCP port 6112 is detected.
- 3701-Oracle 9iAS Web Cache Buffer Overflow: This signature fires when an excessively long HTTP GET request is detected bound for the default Oracle Web Cache port. Legitimate traffic can cause false positives.

NOTE

HTTP traffic is the normal cause for this signature to misfire, but other protocols can also cause it to fire. This issue will be corrected in version 4.0 of the sensor.

- 3702-Default sa account access: This signature fires upon when an attempt to login to a MSSQL server with the default sa account is detected.
- 3703-Squid FTP URL Buffer Overflow: This signature fires when attempt malicious username and password arguments are detected being supplied as part of a proxied FTP request.
- 3704-IIS FTP STAT Denial of Service: This signature will fire if a FTP 'STAT' command with an unusually long argument is detected.
- 3705-Tivoli Storage Manager Client Acceptor Overflow: This signature fires when an excessively long URL request destined for TCP port 1581 is detected. Legitimate traffic can cause false positives.

NOTE

HTTP traffic is the normal cause for this signature to misfire, but other protocols can also cause it to fire. This issue will be corrected in version 4.0 of the sensor.

- 3706–MIT PGP Public Key Server Overflow: This signature fires when an excessively long search parameter is detected being sent to a PGP key server on TCP port 11371. It can cause false positives from a web session using port 11371 as its ephemeral port.
- 3707–Perl fingerd Command Exec: This signature fires when shell meta-characters are detected in a finger request.
- 3708–AnalogX Proxy Socks4a DNS Overflow: This signature fires upon detecting a SOCKS4 proxy request with an overflow in the DNS field.
- 3709–AnalogX Proxy Web Proxy Overflow: This signature fires upon detecting a web proxy request with an overflow in the URI field sent to port 6588.
- 3710–Cisco Secure ACS Directory Traversal: This signature fire upon detecting two or more slashes (//) in an HTTP request sent to port 9090.
- 3711–Informer FW1 auth replay DoS: This signature fires on 32 ASCII zeros, followed by the string ‘rand’, an 0x01 byte, and the string ‘sign’.
- 3714–Oracle TNS ‘Service_Name’ Overflow: This signature fires upon detecting an abnormally long value sent to the parameter Service_Name on the Oracle TNS Listener port (1521t).
- 3728–Long pop username: This signature fires upon detecting a long USER argument (80+ chars) sent to a pop server
- 3729–Long pop password: This signature fires upon detecting a long USER argument sent to a pop server.
- 3730–Trinoo (TCP): This signature fires upon detecting the string “trinoo” or “betaalmostdone” on any well-known Trinoo TCP ports. SubSig 0: Traffic to trinoo service SubSig 1: Traffic from trinoo service SubSig 2: Traffic to trinoo service SubSig 3: Traffic from trinoo service.

NOTE

SubSigs 2 and 3 are IDS 3.1 version sensor signatures and only detect the string “betaalmostdone”.

- 3731-IMail HTTP Get Buffer Overflow: This signature fires when an HTTP get request is made to port 8383 with a URI longer than 96 bytes.
- 3732-MSSQL xp_cmdshell Usage: This signature fires when an attempt to use the MSSQL 'xp_cmdshell' stored procedure is detected. This is an indicator that an attempt has been made to execute unauthorized commands on a MSSQL server. Administrators using the 'xp_cmdshell' stored procedure can cause false positives.
- 3990-BackOrifice BO2K TCP Non Stealth: This signature fires when non-stealth traffic of the BO2K toolkit is detected.
- 3991-BackOrifice BO2K TCP Stealth 1: Stealth type 1 indicates XOR encryption is being used and the signature fires when stealth mode, covert or sneaky activity, on the part of an attacker is detected. Administrators can generate this alarm but the activity should always be considered suspect.
- 3992-BackOrifice BO2K TCP Stealth 2: Stealth type 2 indicates an encryption other than XOR is being used and causes the signature to fire when stealth mode, covert or sneaky activity, on the part of an attacker is detected. Administrators can generate this alarm but the activity should always be considered suspect.

UDP signatures 4000 series

The 4000 series is specific to UDP. Just to refresh your memory, UDP is an unreliable protocol. They are a “send and pray” type of packet. You never know if they made it to their destination or not. Many of these signatures can cause enormous amounts of logs. Cisco has disabled most of these by default. Make sure you analyze your traffic before enabling them.

- 4001-UDP Port Sweep: This signature fires when a series of UDP connections to a number of different destination ports on a specific host have been initiated. This is an indicator of a reconnaissance sweep of your network. Be wary of potentially more serious attacks.
- 4002-UDP Flood
- 4003-Nmap UDP Port Sweep: This signature fires when a series of UDP connections to several different privileged ports (port number < 1024) on a

specific host have been initiated. This is an indicator of a reconnaissance sweep of your network. Be wary of potentially more serious attacks.

- 4050-UDP Bomb: This signature fires when the UDP length specified is less than the IP length specified. This malformed packet type is associated with a denial of service attempt. Remember there is not any legitimate use for malformed packets.
- 4051-Snork: This signature fires when a UDP packet with a source port of either 135, 7, or 19 and a destination port of 135 is detected. If you have Windows applications that are using port 135, they should be excluded from firing this signature.
- 4052-Chargen DoS: This signature fires when a UDP packet is detected with a source port of 7 and a destination port of 19.
- 4053-Back Orifice: This signature fires when the IDS detect traffic coming from the Back Orifice server that is running on the network.

NOTE

Back Orifice is a “backdoor” program that can be installed on a Microsoft Windows 95 or Windows 98 system allowing remote control of the system.

- 4054-RIP Trace: This signature fires when TRACEON or TRACEOFF commands are enabled for the packet.
- 4055-BackOrifice BO2K UDP: BO2K UDP mode is a basic configuration of BackOrifice. Seeing this traffic indicates a non-stealth use of the BO2K toolkit.
- 4056-NTPd readvar overflow: This signature will fire if a readvar command is seen with ntp data that is too large for the ntp daemon to capture.
- 4058-UPnP LOCATION Overflow: This signature alarms upon detecting a large location request sent to a UPnP device.
- 4060-Back Orifice Ping: Alarms when a BO Ping detector is used to scan a network.
- 4061-Chargen Echo DoS: This signature detects packets destined for the port 7 UDP which is the echo port with the chargen service port 19 as the

source. This results in the contents of the packet being “echoed” back to the source IP address, which may be spoofed.

- 4100-Tftp Passwd File: Fires on an attempt to access the passwd file using TFTP. This signature is a good indicator that an attempt to gain unauthorized access to system resources is occurring.
- 4101-Cisco TFTP Directory Traversal: Alarms when a TFTP request is made by appending ../ to the pathname.
- 4150-Ascend Denial of Service: This signature fires when an attempt has been made to send a maliciously malformed command to an ascend router in an attempt to crash the router.
- 4500-Cisco IOS Embedded SNMP Community Names: Certain versions of Cisco IOS contain embedded community names that could possibly allow a remote attacker to view, modify, or both, SNMP MIB variables. This could lead to a denial-of-service attack or total system compromise. There are two different Cisco product advisories concerning the community names. Make sure you review those for more information.

NOTE

The first embedded community name “ILMI” is a read-write community name that allows access to the MIB-II System MIB and various ATM related MIBS. Remote users can modify SNMP variables such as the system name, contact, and location, and many of the ATM interface variables.

The second embedded community name “cable-docsis” is a read-write community string that was introduced as part of the support for the DOCSIS cable-industry standard. It allows a remote user to modify or view any SNMP variable on the affected system, including being able to retrieve the system configuration.

- 4501-Cisco CVCO/4K Remote Username/Password return: This signature detects attempts to access the list of system usernames and passwords on a Cisco Virtual Central device using SNMP. The passwords are encrypted with a triusesl encoding scheme. This signature fires when an SNMP OID fragment 1.3.6.1.886.1.1.1.1 is detected.

- 4502-SNMP Password Brute Force Attempt: This signature detects attempts to brute-force guess community names. A threshold (default of 5) is set and fires when more than this threshold of unique community names between a source and destination in a specified time interval is detected.
- 4503-SNMP NT Info Retrieve: This signature fires when an attempt to gain access to sensitive information about a certain Windows NT system is made. There are two SubSigIds associated with signature 4503. SubSigId 0 fires when an attempt is made to enumerate the list of usernames with SNMP OID .1.3.6.1.4.1.77.1.2.25. SubSigId 1 fires when an attempt is made to enumerate the list of network shares with SNMP OID .1.3.6.1.4.1.77.1.2.27.
- 4504-SNMP IOS Configuration Retrieval: This signature fires when an attempt to retrieve the configuration from a Cisco IOS device. This signature fires when the SNMP OID contains the pattern .1.3.6.1.4.1.9.2.1.55 as a prefix.
- 4505-SNMP VACM MIB Access: This signature fires when SNMP OID fragment .1.3.6.1.6.3.16.1.2.1.3 is matched in an attempt to access the SNMP v2 View-based Access Control MIB (VACM) table. The SNMP v2 View-based Access Control MIB (VACM) table contains all of the SNMP community names in clear-text.
- 4506-D-Link Wireless SNMP Plain Text Password: This signature fires when MIB OID 1.3.6.1.4.1.937.2.1.2.2.0 is accessed with community string “public”.
- 4507-SNMP Protocol Violation: This signature fires when an error in decoding the SNMP protocol is detected.
- 4508-Non SNMP Traffic: This signature fires when non-SNMP traffic is detected destined for port 161UDP.

NOTE

This signature is only available in Cisco IDS versions 4.0 and newer.

- 4509-HP Openview SNMP Hidden Community Name: This signature fires when the SNMP community name 'snmpd' is detected in a SNMP request.
- 4510-Solaris SNMP Hidden Community Name: This signature fires when the SNMP community name 'all private' is detected in a SNMP request.
- 4511-Avaya SNMP Hidden Community Name: This signature fires when the SNMP community name 'all private' is detected in a SNMP request.
- 4600-IOS UDP Bomb: This signature fires when improperly formed SYSLOG transmissions bound for port 514UDP are detected.
- 4601:0-CheckPoint Firewall RDP Bypass: This signature fires when traffic, destined for port 259UDP with the following patterns is detected:
 - SubSig 0: 0x80 0x00 0x00 0x96
 - SubSig 1: 0x80 0x00 0x00 0x80
 - SubSig 2: 0x80 0x00 0x00 0x64
 - SubSig 3: 0x80 0x00 0x00 0x65.
- 4601:1-CheckPoint Firewall RDP Bypass: Alarms when following command is sent to port 259UDP “\x80\x00\x00\x64”.
- 4601:2-CheckPoint Firewall RDP Bypass: Alarms when the following command is sent to port 259UDP “\x80\x00\x00\x96”.
- 4601:3-CheckPoint Firewall RDP Bypass: Alarms when the following command is sent to port 259UDP “\x80\x00\x00\x80”.
- 4603-DHCP Discover: This signature fires when DHCP discovery attempts from clients are made. This is an indicator of unauthorized attempts to connect to the network. Legitimate DHCP discovery attempts can cause this signature to fire an alarm.
- 4604-DHCP Request: This signature fires when DHCP client requests are detected. This is an indicator of unauthorized attempts to connect to the network. Legitimate DHCP discovery attempts can cause this signature to fire an alarm.
- 4605-DHCP Offer: This fires when DHCP lease offers from a DHCP server are made This is an indicator of unauthorized attempts to connect to the network. Legitimate DHCP offers can cause this signature to fire an alarm.

- 4606–Cisco TFTP Long Filename Buffer Overflow: This signature fires when a TFTP request for a file with an abnormally long name is detected. This is an indicator of a buffer overflow.
- 4607–Deep Throat Response: This signature fires when the string “My Mouth is Open” is detected in a UDP packet sent on well-known Deep Throat UDP ports.
- 4608–Trinoo (UDP): This signature fires when the string “trinoo” is detected on any UDP port known to have Trinoo traffic.
- 4609–Orinoco SNMP Info Leak: This signature fires when a specially crafted packet is detected with a destination of UDP port 192. This is a good indicator that attempts are being made to retrieve the SNMP community names from the target.
- 4610–Kerberos 4 User Recon: This signature fires a null character sent to UDP port 750 is detected. This is a good indicator that a Kerberos user recon attack may be occurring.
- 4611–D-Link DWL-900AP+ TFTP Config Retrieve: This signature fires when a TFTP request for the file ‘config.img’ is detected. This is an indicator of an attempted reconnaissance probe. If you are running this D-Link appliance normal administrative work can cause this alarm.
- 4612–Cisco IP Phone TFTP Config Retrieve: This signature fires when a TFTP request for a Cisco IP Phone configuration file is detected. This may indicate an attempted reconnaissance attack.
- 4613–TFTP Filename Buffer Overflow: This signature fires when a TFTP read or write request with a filename containing a non-printable character is detected. This may be an indication of a buffer overflow attack.
- 4614–DHCP request overflow: This signature fires upon detecting a large dhcp request to port 67. The typical dhcp request is quite small in size and shouldn’t fire this signature. If this signature fires, the traffic needs to be investigated.
- 4701–MS-SQL Control Overflow: This signature fires when a buffer overflow attempt to the MS-SQL control port (UDP 1434) is made. This is an indicator the “Slammer” worm is present.

Web/HTTP signature series 5000

The 5000 series of signatures is the largest group. The signatures focus on different types of Web attacks. Buffer overflows, directory traversal, and illegal uploading and downloading of files are just a few examples.

- 5034-**WWW IIS newdsn attack**: This signature fires when attempts are made to run the newdsn.exe command from the http server. This could be indicative of a remote denial of service attack attempt. This particular command could be used to fill up the target host's file system.
- 5035-**HTTP cgi HylaFAX Faxsurvey**: This signature fires when an attempt is made to pass commands to the CGI program faxsurvey. A problem in the CGI program faxsurvey, included with the HylaFAX package from SGI, allows an attacker to execute commands on the host machine. These commands will execute at the privilege level of the HTTP server. There are no legitimate reasons to pass commands to the faxsurvey command. This signature indicates abuse and the source should be shunned.
- 5036-**WWW Windows Password File Access Attempt**: This alarm is fired when an attempt is made to retrieve either the current or backup copy of the NT password file through a web server.
 - Sub ID: 1: Backup copy
 - Sub ID: 2: Current
- 5037-**WWW SGI MachineInfo Attack**: This alarm is fired when an attempt is made to retrieve either the current or backup copy of the NT password file through a web server.
 - Sub ID: 1: Backup Copy
 - Sub ID: 2: Current
- 5038-**WWW wwwsql file read Bug**: This signature fires when an attempt is made to read files in the cgi-bin directory by the www-sql script. This could indicate that a remote attacker is trying to download cgi-bin scripts and access otherwise protected directories under DocumentRoot.
- 5039-**WWW finger attempt**: This signature fires when an attempt is made to run the finger program using the http server. It is recommended that all unnecessary programs be removed from the cgi-bin directory.

- 5040-**WWW Perl Interpreter Attack**: This signature fires when someone attempts to pass and execute Perl commands on the server through a perl interpreter. These commands will execute with the privilege level of the Web Server. If successful, an attacker may gain unauthorized access and remotely execute commands. This can lead to further system access (including root access) and malicious activity. The source address for this signature should be shunned.
- 5041-**WWW anyform attack**: This signature fires when an attacker attempts to execute arbitrary commands through the anyform cgi-bin script. The source address for this attack should be shunned.
- 5042-**WWW CGI Valid Shell Access**: This signature fires when attempts are made to access a valid shell or interpreter on the targeted system. Shells include:
 - Sub ID: 1: bash
 - Sub ID: 2: tcsh
 - Sub ID: 3: ash, bsh, csh, ksh, jsh, or zsh
 - Sub ID: 4: sh
 - Sub ID: 5: Java interpreter
 - Sub ID: 6: Python interpreter
- 5043-**WWW Cold Fusion Attack**: This signature fires when attempts are made to access example scripts that are shipped with the Cold Fusion Servers. The source address for this signature should be shunned.
 - Sub ID 1: indicates an attempt to access the openfile script. This script allows an attacker to upload files to the target host or server.
 - Sub ID 2 indicates an attempt to access displayopenedfile.cfm. This could indicate that a remote attacker is trying to access files on the target host or server.
 - Sub ID 3 indicates an attempt to upload files to a Cold Fusion server through the exprcalc.cfm script. This can be used to overwrite files on the target server or host.
- 5044-**WWW Webcom.se Guestbook attack**: This signature fires when an attacker attempts to execute arbitrary commands through Webcom.se's

rguest.exe or wguest.exe cgi-bin script. The source address for this attack should be shunned.

- 5045-**WWW xterm display attack**: This signature fires when any cgi-bin script attempts to execute the command `xterm -display`. This is an indicator someone is trying to login to your network illegally. There is not a legitimate use for someone to execute `xterm -display`. Any hosts attempting this command should be shunned.
- 5046-**WWW dumpenv.pl recon**: This signature fires when an attempt is made to display information about the targeted host with the `dumpenv.pl` script. Some web servers include this script, which is intended to show environmental information about the server. External attempts should be scrutinized thoroughly. In most cases the source should be shunned.
- 5047-**WWW Server Side Include POST attack**: This signature fires when attempts are made to embed a server side include (SSI) in an http POST command. This is an indicator someone is trying to access system resources without authorization.
- 5048-**WWW IIS BAT EXE attack**: This signature fires when an attempt is made to execute remote commands on a Microsoft IIS 1.0-2.0b web server. This may indicate an attempt to illegally access system resources.
- 5049-**WWW IIS showcode.asp access**: This signature fires whenever an attempt is made to access the `showcode.asp` Active Server Page. This script allows for arbitrary access to any file on the targets file system. Hosts that attempt to access this file, especially from outside your network, should be shunned.
- 5050-**WWW IIS .htr Overflow Attack**: This signature fires when an .htr buffer overrun attack is detected, indicating a possible attempt to execute remote commands, or cause a denial of service against the targeted Windows NT IIS server. Hosts that attempt to cause this type of alarm, especially from outside your network, should be shunned.
- 5051-**IIS Double Byte Code Page**: IIS contains a vulnerability that could allow a web site visitor to view the source code for selected files on the server. However, this is based on the servers default language. The vulnerability only applies to default languages set to Chinese, Japanese or Korean.

- 5052-FrontPage Extensions PWD Open Attempt: This signature fires when attempts are made to open a configuration file on a Microsoft Personal Webserver (for Windows) or FrontPage extensions (for UNIX) web server.
- 5053-FrontPage _vti_bin Directory List Attempt: This signature fires when attempts are made to list the directory of binaries from a Microsoft Personal Webserver (for Windows) or FrontPage extensions (for UNIX) web server.
- 5054-WWWBoard Password: This signature fires when CGI scans are detected looking for WWWBoard services. WWWBoard has several vulnerabilities and should be used with great care.
- 5055-HTTP Basic Authentication Overflow: This signature fires when extremely large usernames and passwords are detected during authentication. This can cause a buffer overflow.
- 5056-WWW Cisco IOS %% DoS: This signature fires when attempts to crash a Cisco IOS-based product using the HTTP management interface is detected. Certain versions of IOS incorrectly interpret the characters “%%” when sent to the HTTP management interface. This can result in a router crashing, causing the need for the power to be cycled to restore normal operation.

The affected operating system versions are: Cisco IOS 11.3AA,11.3DB,12.0x,11.3,11.2SA,12.0T,12.0W5,12.0XA,12.0XE,12.0XH,12.0XJ,12.1,12.1AA,12.1DA,12.1DB,12.1DC,12.1E,12.1EC,12.1T,12.1XA,12.1XB,12.1XC,12.1XD,12.1XE,12.1XF,12.1XG,12.1XH,12.1XI,12.1XJ,12.1XL,12.1XP,11.2P,11.2,11.1,11.0,11.1CC, and 12.0.

The affected software versions are: Cisco IOS 11.2SA,12.0T,12.0W5,12.0XA,12.0XE,12.0XH,12.0XJ,12.1,12.1AA,12.1DA,12.1DB,12.1DC,12.1E,12.1EC,12.1T,12.1XA,12.1XB,12.1XC,12.1XD,12.1XE,12.1XF,12.1XG,12.1XH,12.1XJ,12.1XL,12.1XP,11.2P,11.2,11.1,11.3(1.2),11.3(1.2)T,11.3,11.2(10)P,11.1(14)CA, and 11.1CC.

The affected services are: HTTP Web on ports 80/TCP and 8080/TCP>

- 5057-WWW Sambar Samples: This signature fires when an attempt has been made to access certain CGI programs that contain known vulnerabilities shipped with the Sambar web server. Those programs are **echo.bat** and **hello.bat**.

- 5058-*WWW info2www Attack*: This signature fires when an attempt is made to execute commands with the **info2www** CGI program.
- 5059-*WWW Alibaba Attack*: This signature fires when an attempt is made to execute commands using certain CGI programs shipped with the Alibaba web server. Those programs are **get32.exe**, **alibaba.pl**, and **tst.bat**.
- 5060-*WWW Excite AT-generate.cgi Access*: This signature fires when an attempt is made to access the CGI program **AT-generate.cgi**. Administrator passwords for the Excite Web Server application could be changed. If you feel your system has been subject to this type of activity have your system administrator verify the administrator passwords.
- 5061-*WWW catalog_type.asp Access*: This signature fires when an attempt is made to access the vulnerable sample ASP file **catalog_type.asp**.
- 5062-*WWW classifieds.cgi Attack*: This signature fires when an attempt has been made to execute commands with the CGI program **classifieds.cgi**.
- 5063-*WWW dmbparser.exe Access*: This signature fires when an attempt is made to access the CGI program **dmbparser.exe**.
- 5064-*WWW imagemap.cgi Attack*: This signature fires when an attempt is made to cause a buffer overflow in the CGI program **imagemap.cgi**.
- 5065-*WWW IRIX infosrch.cgi Attack*: This signature fires when an attempt is made to execute commands using the IRIX CGI program **infosrch.cgi**.
- 5066-*WWW man.sh Access*: An attempt has been to access the CGI shell script **man.sh**.
- 5067-*WWW plusmail Attack*: This signature fires when an attempt has been made to change the **PlusMail** administrator password. The attacker could possibly gain full control of the PlusMail program. If this is suspected have the system administrator verify the password.
- 5068-*WWW formmail.pl Access*: This signature fires when an attempt is made to access the CGI program **formmail.pl**.
- 5069-*WWW whois_raw.cgi Attack*: This signature fires when an attempt is made to access to possibly execute commands using the CGI program **Cdomain whois_raw.cgi**.

- 5070-*WWW msadcs.dll Access*: This signature fires when an attempt is made to access the CGI program **msadcs.dll**. This is an indicator a reconnaissance session is occurring for a possible later attack to exploit the IIS RDS vulnerability. The affected operating system versions are Windows NT Server 4.0. The affected software and program versions are IIS 4.0 and 3.0. The affected services are: HTTP Web 80/TCP 8080/TCP, HTTPS Web 443/TCP
- 5071-*WWW msadcs.dll Attack*: This signature fires when an attempt is made to execute commands or view secured files, with privileged access. This type of activity should be scrutinized closely and administrators should audit and validate the system from which the activity has been detected. This is a very common attack used to deface websites.
- 5072-*WWW bizdb1-search.cgi Attack*: An attempt has been made to execute commands or view files with the privileges of the web server using the CGI program **bizdb1-search.cgi**.
- 5073-*WWW EZshopper loadpage.cgi Attack*: An attempt has been made to execute commands or view files with the privileges of the web server using the CGI program **loadpage.cgi**.
- 5074-*WWW EZshopper search.cgi Attack*: An attempt has been made to execute commands or view files with the privileges of the web server using the CGI program **EZshopper search.cgi**.
- 5075-*WWW IIS Virtualized UNC Bug*: An attempt has been made to view the source of an ASP file. A bug exists in certain versions of Microsofts IIS web server which allow an attacker to view of the source of ASP, and other files if the IIS virtual directory they reside in has been mapped to a UNC share.
- 5076-*WWW webplus bug*: An attempt was made to gain access to files outside the web server directories using the CGI program **webplus**.
- 5077-*WWW Excite AT-admin.cgi Access*: An attempt has been made to access the CGI program **AT-admin.cgi**.
- 5078-*WWW Piranha passwd attack*: An attempt has been made to access the vulnerable cgi script **passwd.php3** with suspicious arguments. This is found in the *piranha/secure/* directory.

- 5079-**WWW PCCS MySQL Admin Access:** The PCCS PHP-based MySQL administration tool contains a file with the databases administrator's username and password. This may not seem like much of a problem except it can be accessed remotely.
- 5080-**WWW IBM WebSphere Access:** This signature fires when someone attempts to access a JSP file using a URL like `http://server/servlet/file/login.jsp` potentially revealing the JSP source code.
- 5081-**WWW WinNT cmd.exe Access:** This signature fires when the use of the Windows NT `cmd.exe` is detected in a URL.
- 5083-**WWW Virtual Vision FTP Browser Access:** This signature fires when an attempt to traverse directories in a URL like `http://server/cgi-bin/ftp/ftp.pl?dir=../..` etc is detected.
- 5084-**WWW Alibaba Attack 2:** This signature fires when a pipe (`|`) character is detected in a URL like `http://server/cgi-bin/|post32.exe` or `http://server/cgi-bin/|index2.bat`.
- 5085-**WWW IIS Source Fragment Access:** This signature fires when a URL ending in `+.htr` is detected.
- 5086-**WWW WEBactive Logfile Access:** This signature fires when an attempt to access the WEBactive logfile is detected.
- 5087-**WWW Sun Java Server Access:** This signature fires when an attempt to access URL's like `http://server/pservlet.html` or `http://server/servlet/sunexamples.RealmDumpServlet` are detected.
- 5088-**WWW Akopia MiniVend Access:** This signature fires when an attempt to access a URL like `http://server/view_page.html` is detected.
- 5089-**WWW Big Brother Directory Access:** This signature fires when an attempt to traverse directories with the Big Brother CGI program `bb-hostsvc.sh` has been detected.
- 5090-**WWW FrontPage htmimage.exe Access:** This signature fires when the FrontPage CGI program is accessed with a filename argument ending with `"0,0"`.
- 5091-**WWW Cart32 Remote Admin Access:** This signature fires when an attempt is made to access the vulnerable `cart32.exe` cgi script with suspi-

cious arguments: /cart32.exe/cart32clientlist or /c32web.exe/changeadmin-password.

- 5092-**WWW CGI-World Poll It Access**: This signature fires when an attempt is made to access the Poll-It CGI using an internal script variable name “data_dir” as an argument in the HTTP request.
- 5093-**WWW PHP-Nuke admin.php3 Access**: An attempt has been made to access the vulnerable PHP-Nuke admin.php3 cgi script using suspicious arguments.
- 5095-**WWW CGI Script Center Account Manager Attack**: This signature fires when an attempt to change the administrator password of the CGI Script Center Account Manager is detected.
- 5096-**WWW CGI Script Center Subscribe Me Attack**: This signature fires when an attempt to change the administrative password of the CGI Script Center Subscribe package is detected.
- 5097-**WWW FrontPage MS-DOS Device Attack**: This signature fires when a URL is requested using the shtml.exe component of FrontPage that includes an MS-DOS device name. A denial of service can result from this URL request.
- 5099-**WWW GWScripts News Publisher Access**: This signature fires when attempt to add an author to the GWScripts News Publisher interface is detected.
- 5100-**WWW CGI Center Auction Weaver File Access**: This signature fires when an attempt to access normally inaccessible files using the CGI script auctionweaver.pl has.
- 5101-**WWW CGI Center Auction Weaver Attack**: This signature fires when an attempt to execute an unauthorized command using the auctionweaver.pl CGI script is detected.
- 5102-**WWW phpPhotoAlbum explorer.php Access**: This signature fires when unauthorized attempt to access files using the explorer.php CGI script is detected.
- 5103-**WWW SuSE Apache CGI Source Access**: This signature fires when an attempt to access the /cgi-bin-sdb directory of a web server is detected.

- 5104-**WWW YaBB File Access**: This signature fires when an attempt to read unauthorized files using the YaBB.pl CGI bulletin board program is detected.
- 5105-**WWW Ranson Johnson mailto.cgi Attack**: This signature fires when an attempt to execute system commands using the mailto.cgi program is detected.
- 5106-**WWW Ranson Johnson mailform.pl Access**: This signature fires when an attempt to access the “mailform.pl” has been detected.
- 5107-**WWW Mandrake Linux /perl Access**: This signature fires when an attempt to access the URL path /perl directly has been detected.
- 5108-**WWW Netegrity Site Minder Access**: This signature fires when an unauthorized attempt to access protected content on a website managed by Netegrity Site Minder using an authentication bypass method is detected. Looks for strings like “/\$/**somefile.ccc**” in a URL.
- 5109-**WWW Sambar Beta search.dll Access**: This signature fires when an unauthorized attempt to access files or directories using the Sambar Server search.dll CGI program is detected.
- 5110-**WWW SuSE Installed Packages Access**: This signature fires when an attempt to access the URL /doc/packages is detected.
- 5111-**WWW Solaris Answerbook 2 Access**: This signature fires when an attempt to add a user to the AnswerBook interface is detected.
- 5112-**WWW Solaris Answerbook 2 Attack**: This signature fires when attempt to execute an unauthorized command using the access / error rotation feature of the administrative interface of AnswerBook 2 is detected.
- 5113-**WWW CommuniGate Pro Access**: This signature fires when an unauthorized attempt to access files using the CommuniGate Pro web interface is detected.
- 5114-**WWW IIS Unicode Attack**: This signature fires when an attempt to exploit the Unicode ../ directory traversal vulnerability is detected. Looks for the commonly exploited combinations which are included in publicly available exploit scripts.
- 5115-**Netscape Enterprise Server with ?wp Tags**: This signature fires when certain Netscape Enterprise Server 3.x HTML tags are detected in use.

These tags allow remote users to view the contents of directories on the web servers. In most cases they should be disabled if not in use.

Each of the HTML tags are as follows:

- SubSigId 0 - ?wp-cs-dump
 - SubSigId 1 - ?wp-ver-info
 - SubSigId 2 - ?wp-html-rend
 - SubSigId 3 - ?wp-usr-prop
 - SubSigId 4 - ?wp-ver-diff
 - SubSigId 5 - ?wp-verify-link
 - SubSigId 6 - ?wp-start-ver .
- 5116-Endymion MailMan Remote Command Execution: This signature fires when the perl function open() is used on Endymion MailMan. This allows user-supplied input containing shell metacharacters to be executed as shell commands with the privilege level of the CGI script.
 - 5117-phpGroupWare Remote Command Exec: phpGroupWare is a multi-user groupware suite that is freely distributed. There exists a problem in the software could allow users to remotely execute malicious code by exploiting a vulnerable include() command.
 - 5118-eWave ServletExec 3.0C File Upload: UploadServlet is a servlet that ServletExec contains in its server side classes. UploadServlet, when invoked with a special formed HTTP or GET request, allows an attacker to upload any file to any directory on the server. The uploaded file may have code that can later be executed on the server, leading to remote command execution.
 - 5119-CGI Script Center News Update Admin Passwd Change: Newsup, a cgi script from the CGI Script Center allows password changes to the administrator account without proper verification. Every time a person changes a news update administrator password this signature will trigger.
 - 5120-Netscape Server Suite Buffer Overflow: This signature will fire if the value of the “content” variable sent to the CGI program “search is longer than 1000 bytes. The Netspace Server administrative interface is installed on TCP port 24326 by default.

- 5121-iPlanet .shtml Buffer Overflow: This signature fires if a request with more than 180 characters between slashes (/ or) is received with a .shtml suffix.
- 5122-Nokia IP440 Denial of Service: This signature will fire if more than 6000 characters are sent with a specifically formed request on a web port.
- 5123- WWW IIS Internet Printing Overflow: There are two subsignatures associated with this signature.
 - SubSig 0: This alarm will fire if web traffic is detected sending an abnormally large GET request with a large 'Host' field. Both are
 - SubSig 1: This signature fires upon detecting **.printer** in a URI argument field with a large argument field length.
- 5124-IIS CGI Double Decode: This signature fires when a doubly obfuscated attempt to traverse the directory structure of a web server is detected. Certain versions of the IIS web server perform a second pass decode of the arguments sent to a CGI program. During this second pass decode, the IIS server erroneously reevaluates the already decoded path portion of the URL. An attacker can manipulate the path portion of a URL in such a way as to hide characters, such as ../, which would normally be filtered out during the first pass decode of the URL.

This signature will alarm if the following characters are found in a deobfuscated HTTP request:

 - SubSig 0 - %2e (.)
 - SubSig 1 - %2f (/)
 - SubSig 2 - %5c ()
- 5125-PerlCal Directory Traversal: This alarm will fire if a '../' is present in a HTTP request to the CGI script **'make_cal.pl'**.
- 5126-WWW IIS .ida Indexing Service Overflow: This vulnerability will alarm if web traffic is detected with the ISAPI extension of **.ida?** and a data size of greater than 200 chars.
- 5127-WWW viewsrc.cgi Directory Traversal: This alarm will fire if a ../ is used while requesting viewsrc.cgi using the web.

- 5128-WWW nph-maillist.pl Cmd Exec: This alarm will fire if the cgi script nph-maillist.pl is used with the parameter e-mailaddress having a semicolon (;) in its argument.
- 5129-IOS HTTP Unauth Command Execution: This signature fires when a HTTP attempt to bypass router authentication to execute privileged (level 15) commands is detected. The HTTP request looks like: http:///level/XX/exec/... where XX is 16 - 99.
There are two subsignatures IDs:
 - SubSig 0 fires when XX is between 16 and 19 inclusive.
 - SubSig 1 fires when XX is between 20 and 99 inclusive.
- 5130-Bugzilla globals.pl: This signature fires when an HTTP request for the file '**globals.pl**' is detected.
- 5131-talkback.cgi Directory Traversal: This signature fires when an HTTP access to talkback.cgi attempting to traverse outside the normal directory structure is detected.
- 5132-VirusScan catinfo Buffer Overflow: This signature fires when an abnormally long request is made to the CGI script '**catinfo**', which is part of the Interscan VirusWall management interface.
- 5133-Net.Commerce Macro Path Disclosure: This signature fires when a HTTP request to '**macro.d2w**', with NOEXISTINGHTMLBLOCK appended to the end of the path, is detected.
- 5134-MacOS PWS DoS: This signature fires when an abnormally long HTTP request like `"/?aaaa..."` is detected.
- 5138-Oracle Application Server Shared Library Overflow: Alarms when a URL containing more than 2050 characters is sent to a Oracle server.
- 5140-Net.Commerce Macro Denial of Service: This signature fires when an abnormally long HTTP request has been made to the CGI script '**macro.d2w**', which causes the server to crash.
- 5141-NCM content.pl SQL Query Vulnerability: Alarms when content.pl is detected in the URL with '**<**' or '**>**' characters.
- 5142-DCShop File Disclosure: This signature fires when an HTTP request to one of two files is detected.

- SubSigId 0 - /DCShop/Orders/orders.txt
- SubSigId 1 - /DCShop/Auth_data/auth_user_file.txt
- 5143-Microsoft Media Player ASX Overflow: Alarms when detects a large string in BANNER.HREF field.
- 5146-MS-DOS Device Name DoS: This is referred to as the “DOS Device in Path Name” vulnerability. Microsoft Windows 95, 98, and 98SE will allow an attacker to cause a DoS by using a pathname that includes file device names. The DOS device names are reserved words, and cannot be used as folder or file names.

The following subsignatures IDs correspond to the reserved DOS device names:

- Subsig 0 alarms when /aux is detected in the URL.
- Subsig 1 alarms when /CON is detected in the URL.
- Subsig 2 alarms when /NUL is detected in the URL
- Subsig 3 alarms when /PRN is detected in the URL
- Subsig 4 alarms when /LPT1 through /LPT9 is detected in the URL
- Subsig 5 alarms when /COM1 through /COM9 is detected in the URL
- Subsig 6 alarms when /CLOCK\$ is detected in the URL
- Subsig 7 alarms when /CONFIG\$ is detected in the URL
- Subsig 8 alarms when /XMSXXXX0 is detected in the URL
- Subsig 9 alarms when /\$MMXXXX0 is detected in the URL
- Subsig 10 alarms when /MSCD000 is detected in the URL
- Subsig 11 alarms when /DBLBUFF\$ is detected in the URL
- Subsig 12 alarms when /EMMXXXX0 is detected in the URL
- Subsig 13 alarms when /IFS\$HLP\$ is detected in the URL
- Subsig 14 alarms when /SETVERXX is detected in the URL
- Subsig 15 alarms when /SCSIMGR\$ is detected in the URL
- Subsig 16 alarms when /DBLSBIN\$ is detected in the URL
- Subsig 17 alarms when /MS\$MOUSE is detected in the URL.

- 5147–Arcadia Internet Store Directory Traversal Attempt: This signature fires when an attempt is made to pass `../..` as a template argument to the `tradecli.dll` for the Internet Directory Store program.
- 5148–Perception LiteServe Web Server CGI Script Source Code Disclosu: Alarms when a MS-DOS style CGI directory name is contained in a web request.
- 5149–Trend Micro Interscan Viruswall Configuration Modification: Alarms when **interscan.dll** is accessed.
- 5150–InterScan VirusWall RegGo.dll Buffer Overflow: Alarms when **RegGo.dll** is sent a buffer greater than 820 bytes in length.
- 5151–WebStore Admin Bypass: Detects when an attempt to bypass the administrative authentication of the WebStore application is made.
- 5152–WebStore Command Exec: This signature fires when an attempt to execute unauthorized commands with WebStore application is detected.
- 5154–WWW uDirectory Directory Traversal: Alarms when `udirectory.pl` is called with an arguments that contains a `'../?'`.
- 5155–WWW SiteWare Editor Directory Traversal: Alarms if **SWEditServlet** is called with a `'../'` as an argument.
- 5156–WWW Microsoft fp30reg.dll Overflow: Alarms if **fp30reg.dll** is detected with a argument size that is greater than 258 bytes.
- 5157–Tarantella TTAWebTop.CGI Directory Traversal Bug: This signature fires when an attempt is made to pass `../..` as a value for the `pg` argument to the `ttawebtop.cgi` program.
- 5158–iPlanet Proprietary Method Overflow: This alarm will fire if a supported method is requested with arguments of greater than 2000 characters. Unless an iPlanet web server is being used on your network this alarm should be disabled. Many web forms contain GET/POST methods when used with large sets of arguments could cause this signature to fire.

The following subsignatures IDs correspond to the iPlanet proprietary methods:

- Sub Sig 0 DELETE
- Sub Sig 1 INDEX

- Sub Sig 2 PUT
- Sub Sig 3 MOVE
- Sub Sig 4 MKDIR
- Sub Sig 5 POST
- Sub Sig 6 COPY
- Sub Sig 7 EDIT
- Sub Sig 8 UNEDIT
- Sub Sig 9 SAVE
- Sub Sig 10 LOCK
- Sub Sig 11 UNLOCK
- Sub Sig 12 REVLABEL
- Sub Sig 13 REVLOG Sub
- Sig 14 REVADD
- Sub Sig 15 REVNUM
- Sub Sig 16 SETATTRIBUTE
- Sub Sig 17 GETATTRIBUTE
- Sub Sig 18 GETATTRIBUTENAMES
- Sub Sig 19 GETPROPERTIES
- Sub Sig 20 STARTREV
- Sub Sig 21 STOPREV
- 5159-phpMyAdmin Cmd Exec: The trigger will fire upon detecting access to **sql.php** with the arguments '**goto**' and '**btnDrop=No**'.
- 5160-Apache ? indexing file disclosure bug: This signature fires when attempts to view directories on web servers with certain strings in the URLs. The URL types are:
 - Sub Sig 0: /directory/?M=A
 - Sub Sig 1: /directory/?S=D.

- 5160:1-Apache ? indexing file disclosure bug: This signature fires on attempts to view directories on web servers with certain strings in the URLs. The URL types are:
 - Sub Sig 0: /directory/?M=A
 - Sub Sig 1: /directory/?S=D
- 5161-SquirrelMail Command Exec: This signature fires when an attempt to insert malicious PHP code in to the CGI script '**options_order.php**' is detected in a HTTP request.
- 5162-Active Classifieds Command Exec: This signature fires when attempt is detected to insert arbitrary Perl code into an HTTP request to '**admin.cgi**'.
- 5163-Mambo SiteServer Administrative Password ByPass: Alarms when a request with index2.php is detected with a UID of administrator.
- 5164-PHPBB Remote SQL Query Manipulation: Alarms when an user_level 4 is sent to prefs.php.
- 5165-php-nuke article.php sql query: This signature will fire when it detects a web request to '**article.php**' with the arguments of **mainfile** and **prefix**. Valid requests can cause false positives.
- 5166-php-nuke modules.php DoS: This will fire when it detects access to **modules.php** with an argument's value of '../'.
- 5167-phpMyAdmin Cmd Exec 2: This signature fires when attempt to execute unauthorized PHP commands using **phpMyAdmin** is detected. The following subsignatures are associated with their PHP commands:
 - SubSig 0: illegitimate use of the CGI script 'tbl_copy.php'.
 - SubSig 1: illegitimate use of the CGI script 'tbl_rename.php'.
- 5168-Snapstream PVS Directory Traversal Bug: Fires on an attempt to use '../' to traverse the directory tree on a webserver listening on port 8129.
- 5169-SnapStream PVS Plaintext Password Vulnerability: This signature fires when an attempt to touch the **ssd.ini** file is detected on port 8129.
- 5170-NUL byte in URI: This signature fires when a URL request ending in the character '%00' is detected.
- 5171-NC-Book book.cgi Cmd Exec: This signature fires when '**book.cgi**' is accessed with arguments that contain pipes (|). The CGI script is located in */ncbook/*.

- 5172-WinWrapper Admin Server Directory Traversal: Alarms when the classic directory traversal ‘../’ is detected on port 4096.
- 5173-Directory Manager Cmd Exec: This signature fires if edit_image.php is called with the parameter ‘**userfile_name**’ that contains a semicolon (;). The server does not filter these out. As long as the user is required to authenticate on the webserver this vulnerability is eliminated.
- 5174-phpmyexplorer directory traversal: This signature fires when index.php is accessed with a parameter of ‘**chemin**’ whose value contains a ‘../’.
- 5175-Hassan Shopping Cart Command Exec: This signature fires when an attempt to execute unauthorized commands using the CGI script ‘**shop.pl**’ is detected.
- 5176-Exchange Address List Disclosure: This signature fires when an attempt to retrieve addresses from the Global Address Book using the Exchange Outlook Web Access interface is detected. False positives can occur because of legitimate queries to the Exchange server.
- 5178-MS Index Server File/Path Recon: This signature fires when the ‘**SQLQHit.asp**’ file is accessed with a certain argument, ‘**CiColumns**’ containing a wildcard (*).
- 5179-PHP-Nuke File Upload: This signature fires when an attempt to upload a file using the ‘**admin.php**’ CGI script is detected.
- 5180-sgiMerchant Directory Traversal: This signature fires when the ‘**view_item**’ file is accessed with a certain value, ‘../’, in the parameter **html_file**.
- 5181-MacOS Apache File Disclosure: This signature fires when certain patterns are detected at the end of HTTP requests. The following is a list of subsignatures and their associated patterns:
 - SubSig 0 - ‘/.DS_Store’
 - SubSig 1 - ‘/.FBCIndex’
- 5181:1-MacOS Apache File Disclosure: This signature fires when ‘**../FBCIndex**’ is detected in a URL.
- 5182-WebDiscount’s eShop Arbitrary Command Exec: This signature fires when certain shell meta-characters are detected as part of the input to the Perl script **eShop.pl**. The characters are (;) and (|).

- 5183-PHP File Inclusion Remote Exec: This signature fires when there is an attempt made by a PHP script to retrieve a file using HTTP for execution. Legitimate use of PHP scripts can cause false positives.
- 5184-Apache Authentication Module ByPass: This signature fires upon detecting a select statement on the Authorization line of an HTTP header.
- 5188-HTTP Tunneling: This signature fires when HTTP Tunneling tools are detected in use. These tunneling tools allow users inside your private network to bypass the firewall to access services such as ftp, chat etc. This would be considered in violation of most security policies and pose a real threat to internal networks and should not be allowed.
 - SubSig 0: This signature, GotomyPC, fires when a computer connects to the GotomyPC site.
 - SubSig 1: This signature, FireThru, fires when an attempt is made to use /cgi-bin/proxy is detected. The cgi-bin/proxy is used to tunnel connections to other ports using web ports.
 - SubSig 2: This signature, HTTP Port, fires when a connection is made to exectech-va.com. The site runs a server, which connects a requested resource and returns the information using web ports.
 - SubSig 3: This signature, httptunnel, fires when '/index/html?crap' is detected on POST request.
- 5191-Active Perl PerlIS.dll Buffer Overflow: The Signature fires when a filename greater than 300 characters is seen in a URL with the '.pl' extension.
- 5194-Apache Server .ht File Access: This signature fires when an HTTP request to specific files is detected. The files are:
 - SubSig 0: .htaccess
 - SubSig 1: .htpasswd
 - SubSig 2: .htgroup
- 5195-AS/400 '/' attack: This signature fires when a GET request with '.jsp/' on the end is detected. Unless you are running an IBM AS/400 web server you should disable this signature. This signature can cause false positives.
- 5196-Red Hat Stronghold Recon attack: This signature fires when a HTTP request is detect to specific files. Those files are:

- SubSig 0: stronghold-info
- SubSig 1: stronghold-status
- 5197-Network Query Tool command Exec: This signature fires when attempts are made to pass shell metacharacters to the '**nqt.php**' or '**network_query.php**' variables.
- 5199-W3Mail Command Exec: This signature fires if an attempt to execute commands in a HTTP request to the CGI program '**sendmessage.cgi**' is detected.
- 5200-IIS Data Stream Source Disclosure: This signature fires when attempts are made to access a file using HTP with the '**::\$DATA**' extension. This extension looks peculiar itself and any sightings should be scrutinized thoroughly.
- 5201-PHP-Nuke Cross Site Scripting: Cross site scripting occurs when web applications gather malicious data from a user. This data is gathered in the form of a hyperlink that contains the malicious content within it. The subsignatures associated with PHP-Nuke Cross Site Scripting are:
 - SubSig 0: This signature fires if '**user.php**' is accessed and the parameter **uname** contains a HTML script directive.
 - SubSig 1: This signature fires when '**modules.php**' is accessed and the parameter title contains a HTML script directive.
 - SubSig 2: This signature fires when '**phptonuke.php**' is accessed and the parameter '**filenavn**' contains a HTML script directive.
- 5202- PHP-Nuke File Copy / Delete: This signature fires when attempts are made to either copy or delete files using the PHP-Nuke administrator filemanager. The subsignatures associated with this signature are:
 - SubSig 0: This signature fires when attempts are made to copy a file a using the PHP-Nuke administrator filemanager module.
 - SubSig 1: This signature fires when attempts are made to delete a file a using the PHP-Nuke administrator filemanager module.
- 5203- Hosting Controller File Access and Upload: This signature fires when directory traversal attempts are made using the script '**filemanager.asp**'. This is a good indicator of uploading or downloading from a web server is taking place.

- 5204-AspUpload Sample Scripts: This signature fire when certain sample scripts are detected as being used. Sample scripts should be removed from all production servers.
 - SubSig 0: This signature fires if directory traversal attempts to use the sample script “**UploadScript11.asp**” are detected.
 - SubSig 1: This signature fires if attempts to use the sample script “**DirectoryListing.asp**” are detected.
- 5205-Apache php.exe File Disclosure: This signature fires when a MS-DOS drive letter is detected as an argument to the script ‘**php.exe**’. This is a good indicator that unauthorized attempts to retrieve files off the Apache web server are occurring.
- 5206-Horde IMP Session Hijack: This signature fires if ‘**status.php3**’ is accessed and the message parameter includes a script HTML directive.
- 5207-Entrust GetAccess directory traversal: This signature fires when a directory traversal ‘**../**’ is sent as a argument value to the script ‘**aboutbox.gas.bat**’.
- 5208-Network Tools shell metacharacters: This signature fires when a shell metacharacter is sent as an argument to the **Network_Tool**.
- 5209-Agora.cgi Cross Site Scripting: This signature fire when HTML tags are detected as arguments sent to the Agora shopping cart application.
- 5210-FAQManager.cgi directory traversal: This signature fires when a web request to FAQManager.cgi with a hard-coded path to a file outside of the web directory is detected.
- 5211-zml.cgi File Disclosure: This signature fires when an argument - file, containing **../** is sent to zml.cgi script.
- 5212-Bugzilla Admin Authorization Bypass: This signature fires when an unauthorized attempt is made to add a user to the administrative group of Bugzilla.
- 5213-Bugzilla Command Exec: This signature fires if an attempt is made to add an unauthorized command to Bugzilla.
- 5214-FAQManager.cgi null bytes: This signature fires if a web request to FAQManager.cgi with a null byte appended to the request is detected.

- 5215-lastlines.cgi cmd exec/traversal: This signature fires when an HTTP request for lastlines.cgi with arguments is detected. The subsignatures with the associated arguments are:
 - SubSig 0: ../
 - SubSig 1: Shell Metacharacters
- 5216-PHP Rocket Directory Traversal: This signature fires when an HTTP request to '**PHProcketadmin.php**' or '**index.php**' with a value for the parameter page of '../' is detected.
- 5217-Webmin Directory Traversal: This signature fires when an HTTP request to '**edit_action.cgi**' with an argument of '../' is detected.
- 5218-Boozt Buffer Overflow: The signature fires when '**Index.cgi**' in Boozt package is sent a name containing 1000+ characters.
- 5219-Lotus Domino database DoS: This signature fires when '/..' is detected in the URL.
- 5220-CSVForm Remote Command Exec: This signature fires when the script '**CSVForm.pl**' is sent a file argument containing a pipe "|" character.
- 5221-Hosting Controller Directory Traversal: This signature fires when an http request to a hosting controller file with certain arguments for the fail-path is detected. False positives are possible if an administrator issues certain web requests. The subsignatures and the associated files are:
 - SubSig 0 statsbrowse.asp
 - SubSig 1 servubrowse.asp
 - SubSig 2 browsedisk.asp
 - SubSig 3 browsewebalizerexe.asp
 - SubSig 4 sqlbrowse.asp
- 5223-Pi3Web Buffer Overflow: This signature fires when a long HTTP request to the CGI program '**hello.exe**' is detected.
- 5224-SquirrelMail SquirrelSpell Command Exec: This signature fires when attempts are made to execute commands using the SquirrelSpell feature of SquirrelMail is detected.

- 5227- AHG Search Engine Command Exec: This signature fires when shell metacharacters ‘;|’ are detected as input to the script ‘**search.cgi**’.
- 5229- DCP Portal Root Path Disclosure: This signature fires when a request to access `add_user.php` is detected.
- 5230- Lotus Domino Authentication Bypass: The alarm fires when a `.nsf` file is accessed with URL longer than 230 bytes.
- 5231- MRTG Directory Traversal: This signature will fire if directory traversal attempts using MRTG CGI scripts are detected.
- 5232- URL with XSS: This signature will alarm upon detecting a URL with script in it. This is a common way to execute a XSS. This is also known as cross site scripting. Cross site scripting occurs when web applications gather malicious data from a user. This data is gathered in the form of a hyperlink that contains the malicious content within it.
- 5233- PHP fileupload Buffer Overflow: This signature fires when an abnormal and long file name arguments are being sent to an HTTP form.
- 5234- pforum sql-injection: This signature will fire when a sql-injection attempt to ‘**logincheck.php**’ is detected.
- 5236- Xoops sql-injection: This signature will fire upon detecting a request to `userinfo.php` that contains a sql-injection attack in a parameter.
- 5237- HTTP CONNECT Tunnel: The signature fires when the HTTP CONNECT method is detected. Attackers may try to exploit vulnerabilities in HTTP proxies to help hide their locations. Internal users accessing proxies can cause false positives.
- 5238- EZNET Ezboard Buffer Overflow: The alarm fires when access to scripts ‘**Ezboard.cgi**’, ‘**Ezman.cgi**’, or ‘**Ezadmin.cgi**’ is detected. The HTTP header must be greater than 350 characters to make this signature fire.
- 5239- Sambar cgitest.exe Buffer Overflow: This signature fires when an unusually long argument is detected being sent to the CGI program “/cgitest.exe”.
- 5240- Marcus Xenakis Shell Command Exec: The alarm fires when shell metacharacters are detected as argument to the script ‘**directory.php**’.

- 5241-Avenger System Command Exec: The alarm fires when a directory traversal or shell metacharacters are input to ans.pl script.
- 5243-CS .cgi Script Cmd Exec: This signature will alarm upon detecting the use of a possible command exec statement in the argument list. The subsignatures and the associated scripts are:
 - SubSig 0: - csSearch.cgi
 - SubSig 1: - csMailto.cgi
 - SubSig 2: - csGuestbook.cgi
 - SubSig 3: - csLiveSupport.cgi
 - SubSig 4: - csNewsPro.cgi
 - SubSig 5: - csChatRBox.cgi
- 5244- PhpSmsSend Command Exec: This signature fires when attempts are made to execute unauthorized commands using the CGI program **'phpsmsend.php'** are detected.
- 5245- HTTP 1.1 Chunked Encoding Transfer: This signature fires when HTTP 1.1 chunked encoding transfer activity is detected. False positives are possible. Any detect should be scrutinized closely.
- 5246-IIS ISAPI Filter Buffer Overflow: This signature fires when an unusually long argument sent to the CGI program 'shtml.exe' is detected.
- 5247-IIS ASP SSI Buffer Overflow: This signature fires when a HTTP request for an Active Server Page (ASP) document has an unusually large 'Content-Length' value.
- 5248-IIS HTR ISAPI Buffer Overflow: This signature fires when an unusually long HTTP request for a HTR document with an ASP file as an argument is detected.
- 5249-IDS Evasive Encoding: This signature looks for special characters such as Null %00, New Line %0a, Carriage Return %0d, Period "." %2e, Forward Slash "/" %2f, and Back Slash "\" %5c in the URL of a HTTP request that have been encoded in hexadecimal vice the actual character. This is a technique used to evade detection of an attack. This signature is fired if any of the before mentioned characters are detected as being encoded as part of the URL:

- 5250-IDS Evasive Double Encoding: This signature looks for special characters such as Null %00, New Line %0a, Carriage Return %0d, Period “.” %2e, Forward Slash “/” %2f, and Back Slash “\” %5c in the URL of a HTTP request that have been encoded in hexadecimal vice the actual character in the URL of a HTTP request that have been “doubly” encoded. This is a technique used to evade detection of an attack. This signature is fired if any of the before mentioned characters are detected as being doubly encoded as part of an URL
- 5251-Allaire JRun // Directory Disclosure: This signature will fire if an unauthorized attempt to display directory listings for the Allaire JRun web server is detected.
- 5252-Allaire JRun Session ID Recon: This signature will fire if the system detects that a remote user tries to access the sample servlet files in Allaire JRun web server in order to get sensitive information.
- 5253-Axis StorPoint CD Authentication Bypass: This signature will fire if the system detects that a remote user tries to use the “dot dot” (..) attack to access the server’s administration pages without authentication.
- 5254-Sambar Server CGI Dos Batch File: This signature will fire if the system detects that a remote user tries to run MS-DOS batch files that are in server’s cgi-bin directory.
- 5255-Linux Directory traceroute / nslookup Command Exec: This signature fires when an unauthorized attempt to execute commands using the CGI script “nslookup.pl” or “traceroute.pl” is detected.
- 5256-Dot Dot Slash in URI: This signature will when a “dot dot slash” (../) is detected in a URI.
- 5257-PHPNetToolpack traceroute Command Exec: This signature fires when an unauthorized attempt to execute commands using the “nettools.php” CGI script is detected.
- 5258-Script source disclosure with CodeBrws.asp: This signature fires upon detecting a request to the sample script CodeBrws.asp with arguments of ‘../’. You should never see a ‘../’ request to this script.

- 5259–Snitz Forums SQL injection: This signature will fire upon detecting a HTTP request to members.asp that includes the character ‘ as a value sent to the parameter M_NAME.
- 5260–Xpede sprc.asp SQL Injection: This signature will alarm upon detecting an HTTP request to sprc.asp with an argument that contains an apostrophe (‘). This would be indicative of a SQL insertion attack.
- 5261–BackOffice Server Web Administration Access: This signature fires upon detecting access to Backoffice/Services.asp. This script has been known to be vulnerable to an authentication bypass attack.
- 5262–Large number of Slashes URL: This signature will fire when a large number of slashes (“/”) in URL are detected.
- 5263–ecware.exe Access: This signature fires when a HTTP request for ‘ecware.exe’ is detected.
- 5265–RedHat cachemgr.cgi Access: This signature fires when unauthorized remote access to ‘**cachemgr.cgi**’ file is detected. False positives are possible with normal access to the ‘**cachemgr.cgi**’ file.
- 5266–iCat Carbo Server File Disclosure: This signature will fire when a http request contains carbo.dll in the url and ../ in the icatcommand parameter is detected.
- 5268–Cisco Catalyst Remote Command Execution: This signature will fire when a http request contains /exec/ in the URL is detected. A vulnerability exists in the webserver configuration interface of Cisco Catalyst 3500 XL will allow a remote attacker to execute arbitrary commands. Legitimate access to the GUI of the Catalyst switch can cause false positives.
- 5269–ColdFusion CFDOCS Directory Access: This signature will fire when unauthorized remote access to ‘**/CFDOCS**’ directory is detected. Normal access to the ‘**/CFDOCS**’ can cause false positives.
- 5270–EZ–Mall order.log File Access: This signature fires when an HTTP request for attempt is ‘**/mall_log_files/order.log**’ is detected.
- 5271–search.cgi Directory Traversal: This signature fires when ‘../’ is found in the ‘letter’ argument to the CGI script ‘search.cgi’.
- 5272–count.cgi GIF File Disclosure: This signature fires when ‘../’ is found in the ‘image’ argument to the CGI script ‘count.cgi’.

- 5273-Bannermatic Sensitive File Access: This signature fires upon detecting an HTTP request to certain Bannermatic files. Bannermatic allows a web master to build his own banner exchange service without having to purchase, install, or operate special software because it functions exclusively online. The subsignatures and associated files are:
 - SubSig 0 - ban.log
 - SubSig 1 - ban.bak
 - SubSig 2 - ban.dat
 - SubSig 3 - banmat.pwd
- 5274-Netpad.cgi Directory Traversal/Command Exec: This signature fires upon detecting an attack to the known vulnerable script '**netpad.cgi**'. The subsignatures associated with this signature are:
 - SubSig 0 - Command Exec Attempt
 - SubSig 1 - Directory Traversal Attempt.
- 5275-Phorum Remote Command Exec: This signature fires upon detecting an attempted remote script execution on certain files that are part of the '**Phorum**' package. These files and corresponding subsignatures are:
 - SubSig 0 - admin.php
 - SubSig 1 - plugin.php
- 5276-cart.cgi Command Execution: This signature fires when argument '**3fdj939jf**' is used with the cart.cgi script, which is the backdoor remote-execution argument.
- 5276:1-cart.cgi vars,env,db Recon: This signature fires when argument 'vars', 'env', or 'db' is used with the cart.cgi script, which reveals configuration settings of the application. False positives are possible if arguments ending in '**vars**', '**env**' or '**db**' is used with the script '**cart.cgi**'.
- 5276:2-cart.cgi Backdoor: This signature fires when argument '**usmbu7777**' is used with the cart.cgi script, which is the e-mail backdoor argument.
- 5277- dfire.cgi Command Exec: This signature fires when dfire.cgi is executed with a pipe or semicolon in the 'ipinc' or 'ipone' argument.

- 5278-VP-ASP shoptest.asp access: This signature will fire upon detecting access to a dangerous default script of VP-ASP /demo400/shopdbtest.asp.
- 5279-JJ CGi Cmd Exec: This signature fires when an unauthorized attempt to execute commands using the 'jj' CGI script is detected.
- 5280-IIS idq.dll Directory Traversal: This signature will fire if an unauthorized attempt to view files on web server using idq.dll is detected.
- 5281-Carello add.exe Access: This signature will fire when unauthorized remote access to /carello/add.exe file is detected. Legitimate access to '/carello/add.exe' file can cause this signature to fire.
- 5282-IIS ExAir advsearch.asp Access: This signature will fire if the direct remote access to '/ExAir/search/advsearch.asp' page is detected.
- 5282:1-IIS ExAir search.asp Access: This signature will fire if the direct remote access to '/ExAir/search/search.asp' page is detected.
- 5282:2-IIS ExAir query.asp Access: This signature will fire if the direct remote access to '/ExAir/search/query.asp' page is detected.
- 5283-info2www CGI Directory Traversal: This signature will fire when unauthorized remote access to 'info2www' CGI script is detected.
- 5284- IIS webhits.dll Directory Traversal: This signature will fire if an unauthorized attempt to view files on web server using '**webhits.dll**' is detected.
- 5285-PHPEventCalendar Cmd Exec: This signature will fire upon detecting a shell metacharacter in the argument value of '**userfile**' inside an HTTP request for '**index.php**'.
- 5286-WebScripts WebBBS Cmd Exec: This signature will fire upon detecting a shell metacharacter in the argument value of '**followup**' inside an HTTP request for '**webbbs_post.pl**'.
- 5287-SiteServer AdSamples SITE.CSC File Access: This signature will fire when unauthorized remote access to '/adsamples/config/site.csc' file is detected. Legitimate access to the '**site.csc**' can cause false positives.
- 5288-Verity search97 Directory Traversal: This signature will fire when an unauthorized attempt to access files on the server using search97 CGI script is detected.

- 5289-SQLXML ISAPI Buffer Overflow: This signature will fire if an attempt to overflow the “contenttype” argument in a HTTP request is detected.
- 5290-Apache Tomcat DefaultServlet File Disclosure: This signature fires when an attempt is made to access org.apache.catalina.servlets.DefaultServlet uses an HTTP request.
- 5291-WEB-INF Dot File Disclosure: This signature fires when a HTTP request includes a “.” character appended to “WEB-INF”. This may indicate an attempt to view the contents of directories and files under the “/WEB-INF” subdirectory on the web server.
- 5292-SalesCart shop.mdb File Access: This signature will fire if an HTTP request for ‘shop.mdb’ is detected. This may indicate the possible disclosure of sensitive customer information.
- 5293-robots.txt File Access: This signature fires when the file “robots.txt” is accessed on a web server.
- 5294-BearShare File Disclosure: This signature fires on “\.\.” appearing in an HTTP request on port 6346 after deobfuscation has been applied. Remember, deobfuscation is the process of clarifying or unobscuring the traffic.
- 5295-finger CGI Recon: Fires on an HTTP request for a URI containing “/finger”.
- 5296-Netscape Server PageServices Directory Access: Fires on an HTTP request for a URI containing “?PageServices”.
- 5297-order_log.dat File Access: This signature fires when the file “/orders/order_log.dat” is accessed on a web server.
- 5298-shopper.conf File Access: This signature fires when the file “/PDG_Cart/shopper.conf” is accessed on a web server.
- 5299-quickstore.cfg File Access: This signature fires when the file “/quickstore.cfg” is accessed on a web server.
- 5300-reg_echo.cgi Recon: Fires on any HTTP access to ‘reg_echo.cgi’. False positives are possible from legitimate use of ‘reg_echo.cgi’.

- 5301-~/consolehelp/ CGI File Access: Fires on any HTTP access to ‘/consolehelp/’.
- 5302-~/file/ WebLogic File Access: Fires on any HTTP containing ‘/file/’ in the URL. False positives are likely if any URL contains the ‘/file/’ string.
- 5303-pfdisply.cgi Command Execution: Fires on an HTTP access containing ‘pfdisplay.cgi’ followed by an argument containing a pipe (‘|’) or a semicolon (‘;’). Legitimate use of ‘pfdisplay.cgi’ can cause false positives.
- 5304-files.pl File Access: Fires on any HTTP access to ‘files.pl’. Verify the files in question.
- 5305-.bash_history File Access: This signature will fire when unauthorized remote access to ‘.bash_history’ file is detected. False positives can be caused from legitimate use of the file.
- 5305:1-.sh_history File Access: This signature will fire when unauthorized remote access to ‘.sh_history’ file is detected. False positives can be caused from legitimate use of the file.
- 5305:2-.history File Access: This signature will fire when unauthorized remote access to ‘.history’ file is detected. False positives can be caused from legitimate use of the file.
- 5305:3-.zhistory File Access: This signature will fire when unauthorized remote access to ‘.zhistory’ file is detected. False positives can be caused from legitimate use of the file.
- 5306-SoftCart storemgr.pw File Access: This signature will fire when unauthorized remote access to ‘/pw/storemgr.pw’ file is detected. False positives can be caused from legitimate use of this file.
- 5308-rpc-nlog.pl Command Execution: This signature fires when a URL containing the string “/*.jsp/” or “/*.jhtml/” is accessed on a web server. False positives can be caused from legitimate use of the ‘rpc-nlog.pl’ script.
- 5309- handler CGI Command Execution: This signature fires when “/handler” is accessed on a web server with a pipe or semicolon as an argument. False positives can be caused from legitimate use of the ‘handler’ script.
- 5310-INDEX / directory access: This signature fires when an INDEX request is made to a web server. False positives can be caused from legitimate INDEX requests.

- 5311-8.3 file name access: This signature fires when an 8.3-style abbreviested file name (such as “MICROS~1”) is accessed on a web server. False positives can be caused from legitimate access to files containing tildes.
- 5312-*.jsp/*.jhtml Java Execution: This signature fires when a URL containing the string “/*.jsp/” or “/*.jhtml/” is accessed on a web server.
- 5313-order.log File Access: This signature fires when the file “/admin_files/order.log” is accessed on a web server.
- 5314- windmail.exe Command Execution: This signature fires when “/windmail.exe” is accessed on a web server.
- 5315-changedisplay.pl WWWthreads Privilege Elevation: This signature fires when “/changedisplay.pl” is accessed on a web server with an argument of U_STATUS or U_SECURITY.
- 5316-BadBlue Admin Command Exec: This signature fires when a request is made to the BadBlue web administration interface to map a directory on the web server’s filesystem to a virtual directory on the web server. False positives can be caused from legitimate mapping of virtual directories.
- 5317-Tivoli Endpoint Buffer Overflow: This signature detects an excessive long request to the Tivoli Management Framework Endpoint web server on TCP port 9495 is detected.
- 5318-Tivoli ManagedNode Buffer Overflow: This signature fire when an excessive long request to the Tivoli Management Framework ManagedNode web server on TCP port 94 is detected. This may indicate a buffer overflow attack.
- 5319-SoftCart orders Directory Access: This signature will fire when unauthorized remote access to ‘/orders’ directory is detected. False positives can be caused by legitimate access to the ‘/orders’ directory.
- 5320-ColdFusion administrator Directory Access: This signature will fire when unauthorized remote access to ‘/cfide/administrator’ directory is detected. False positives can be caused by legitimate access to the ‘/cfide/administrator’ directory.

- 5321-Guest Book CGI access: This will trigger on any HTTP access to '/cgi-bin/guestbook'. False positives will be caused by any type of access to the '/cgi-bin/guestbook'.
- 5322-Long HTTP Request: This signature fires when a long HTTP request using the GET, HEAD, or POST method is detected. This signature must be tuned to reduce the number of false positives generated.
- 5323-Cisco Router http exec command: This alarm will fire upon detecting a /exec/ in the URI portion of an http request. An /exec/ usually indicates a privileged command in being executed uses the web interface on a Cisco router.
- 5323-midicart.mdb File Access: This alarm will fire upon detecting a ?/ in a URI portion of an http request.
- 5324-Cisco IOS Query (?/): This alarm will fire upon detecting a ?/ in a URI portion of an http request.
- 5325-Contivity cgiproc DoS: This alarm will fire upon detecting a shell meta-character as an argument to an http request to /cgi/cgiproc.
- 5326-Root.exe access: The signature alarms upon detecting a http request for root.exe.
- 5327-Tilde in URI: This signature fires upon detecting a tilde (~) in an http request.
- 5328- Cisco IP phone DoS: This signature will fire upon detecting a specially crafted HTTP request that will reboot a Cisco IP phone.
- 5329-Apache/mod_ssl Worm Probe: This signature fires when a probe by the Apache/mod_ssl worm is detected.
- 5330-Apache/mod_ssl Worm Buffer Overflow: This signature fires when a buffer overflow attack by the Apache/mod_ssl worm to the HTTPS (TCP port 443) is detected.

NOTE

The Apache/mod_ssl worm attempts to execute a buffer overflow attack to vulnerable web servers using the HTTPS port TCP443. If the worm can infect the host, it will propagate and begin to scan for new hosts to attack. A

backdoor on port UDP2002 is also installed in order to perform distributed DoS attacks.

- 5331-Image Javascript insertion: This signature fires upon detecting an HTML IMG tag that tries to inject javascript inside of it.
- 5332-Wordtrans-web Command Exec: This signature fires when attempt to execute unauthorized commands using the Wordtrans-web script 'webtrans.php' is detected.
- 5333-FUDForum File Disclosure: This signature fires when an attempt to view files using FUDForum is detected. SubSig 0 looks for access to the file 'tmp_view.php'. SubSig 1 looks for access to the file 'admbrowse.php'.
- 5334- DB4Web File Disclosure: This signature fires when an unauthorized attempt to view files using the DB4WEB webserver script 'db4web_c' or 'db4web_c.exe' is detected.
- 5335-DB4WEB Proxy Scan: This signature fires when an attempt to connect to a remote host using the DB4WEB web server as a proxy to scan for open TCP ports is detected. This is a good indicator of a reconnaissance attack.
- 5336- Abyss Web Server File Disclosure: This signature fires when a HTTP request ends in a '+' character. This may indicate an attempt the view the source of the requested file.
- 5337-Dot Dot Slash in HTTP Arguments: This signature fires upon detecting a directory traversal attempt (../) in the argument field of an HTTP request.
- 5338-Front Page Admin password retrieval: This signature fires upon detecting a access attempt to administrators.pwd uses HTTP traffic.
- 5339-SunONE Directory Traversal: This signature fires upon detecting a directory traversal attempt (../) sent to ports 6015-6018 TCP.
- 5340-Killer Protection Credential File Access: This signature fires upon detecting an HTTP request that contains 'vars.inc'.

- 5341-HP Procurve 4000M Switch DoS: This signature fires when a HTTP request for the URL '/sw2/cgi/device_reset' is detected. This may indicate a denial of service attack against a HP Procurve switch.
- 5342-Invision Board phpinfo.php Recon: This signature fire when a HTTP request for the URL 'phpinfo.php' is detected. This may indicate an attempted reconnaissance probe.
- 5343-Apache Host Header Cross Site Scripting: This signature fires when an HTTP Host: header is received containing a percent or less-than character. This signature is disabled by default. This signature is known to impact performance.
- 5344-IIS MDAC RDS Buffer Overflow: This signature fires when a buffer overflow attempt using the Remote Data Services (RDS) component of Microsoft Data Access Components (MDAC) is detected.
- 5345-HTTPBench Information Disclosure: This signature fires when the ezhttpbench.php is requested with an AnalyseSite parameter starting with a slash ('/') character.
- 5346-BadBlue Information Disclosure: This signature fires on an HTTP access to soinfo.php.
- 5347-Xoops WebChat SQL Injection: This signature fires when an HTTP request is made for the script 'index.php' with the 'roomid' argument containing a single-quote or semicolon character.
- 5348-Cobalt RaQ Server overflow.cgi Cmd Exec: This signature fires upon detecting to a HTTP request on port 81 or 444 to overflow.cgi with parameter name of 'e-mail'. False positives can be caused from legitimate activity.
- 5349-Polycom ViewStation Admin Password: This signature fires when the file "a_security.htm" is accessed uses HTML. This may indicate an attempt the retrieve sensitive information.
- 5350-PHPnuke e-mail attachment access: This signature fires upon detecting direct access to PHPnuke e-mail attachments from a web browser.
- 5351-MS IE Help Overflow: This signature fires when a buffer overflow attempt is detected in Active X instructions coming from a web server.
- 5352-H-Sphere Webshell Buffer Overflow: This signature fires when an HTTP request for '/cgi-bin/webshell' is detected with an excessively long multi-part boundary header.

- 5353-H-Sphere Webshell 'mode' URI exec: This signature fires when the CGI executable '/cgi-bin/webshell' is accessed with shell escape characters (| ; \$ `) in the 'mode' parameter.
- 5354-H-Sphere Webshell 'zipfile' URI exec: This signature fires when the CGI executable '/cgi-bin/webshell' is accessed with shell escape characters (| ; \$ `) in the 'zipfile' parameter.
- 5355-DotBr exec.php3 exec: This signature is fired by a URI which accesses the script '/admin/exec.php3' with a parameter of 'cmd='.
- 5356-DotBr system.php3 exec: This signature is fired by a URI which accesses the script '/admin/system.php3' with a parameter of 'cmd='.
- 5357-IMP SQL Injection: This signature will fire upon detecting an sql-injection attempt to mailbox.php.
- 5358-Psunami.CGI Remote Command Execution: This signature will fire when a http request contains psunami.cgi in the url and '|' character in the 'topic' parameter is detected.
- 5359-Office Scan CGI Scripts Access: This signature will fire when a http request contains /officescan/cgi/ in the url is detected. False positives can be caused by normal access to '/officescan/cgi/'.
- 5360-Frontpage htimage.exe Buffer Overflow: This signature will fire when a http request contains htimage.exe in the url and more than 700 characters in the argument field is detected.
- 5362-FrontPage dvwssr.dll Buffer Overflow: This signature will fire when a http request contains dvwssr.dll in the url and more than 2000 characters in the argument field is detected.
- 5363-Frontpage imagemap.exe Buffer Overflow: This signature will fire when a http request contains imagemap.exe in the url and more than 700 characters in the argument field is detected.
- 5364-IIS WebDAV Overflow: This signature fires when a long HTTP request (65000+ chars) is detected with a HTTP header option of 'Translate:'. This indicates the use of an attack to exploit a weakness in the WebDAV component of the IIS web server.
- 5365-Long WebDAV Request: This signature fires when a long WebDAV request(65000+ chars) is detected. This may indicate an attempted buffer

overflow attack. For performance reasons, Cisco IDS 3.x only implements checks for the WebDAV methods SEARCH (SubSig 0) and LOCK (SubSig 1). Public exploits are available which utilize these methods.

- 5366-Shell Code in HTTP URL / Args: This signature fires when a non-printable ASCII character (128-255) is detected in either the URL or arguments of the HTTP request. The URL and arguments of a HTTP request should not contain any non-printable characters, which may indicate the presence of shell code used in buffer overflow attacks. This signature is disabled in version 3.x of the sensor software. The subsignatures break this into two alarms:
 - subSig 0: URL
 - SubSig 1: Arguments of the HTTP request.
- 5367-Apache CR / LF DoS: This signature fires when a long sequence of consecutive carriage return / linefeed characters (`\x0D\x0A`) to web server ports is detected. This may indicate a denial of service of attack.
- 5368-Cisco ACS Windows CSAdmin Overflow: This signature fires when an long username is sent to the 'login.exe' CGI program on TCP port 2002. This may indicate a buffer overflow attack.
- 5369-Win32 Apache Batch File CmdExec: This signature fires upon detecting a metacharacter used as an argument to a .bat file request. This indicates someone is trying to execute a command uses a request to the .bat file.
- 5370-HTDig File Disclosure: This signature fires upon detecting access to an htdig script with a back tick (`) in the argument field.
- 5371-bdir.htr Access: This signature fires upon detecting access to the file bdir.htr. False positives can be caused by legitimate use of an IIS versions 3.0 server.
- 5372-ASP %20 source disclosure: This signature fires upon detecting .asp%20 sent to an argument named CiWebHitsFile.
- 5373-IIS 5 Translate: f Source Disclosure: This signature fires upon detecting a field of Translate: F in the HTTP header request.
- 5374-IIS Executable File Command Exec: This signature fires upon detecting a crafted web request sent to a .bat file.

- 5375–Apache mod_dav Overflow: This signature fires upon detecting an XML document within an HTTP request that contains a WebDav method with a large argument.

NOTE

This signature is only available in Cisco IDS versions 4.0 and newer.

- 5376–iisPROTECT Admin SQL Injection: This signature fires when an attempt to inject arbitrary SQL statements into the arguments of an HTTP request to iisPROTECT administration interface is detected. This may be an unauthorized attempt to view or manipulate data or execute commands on the database server.
- 5377–xp_cmdshell in HTTP args: This signature fires when an attempt to use the MSSQL ‘xp_cmdshell’ stored procedure is detected in the arguments of a HTTP request. This may represent a SQL insertion attack attempting to execute unauthorized commands on a MSSQL server.
- 5378–Vignette TCL Injection Command Exec: This signature fires when attempt to inject TCL scripting code into a HTTP request to a Vignette template is detected.
- 5379–Windows Media Services Logging ISAPI Overflow: This signature fires when a long HTTP request is sent to the Windows Media Services DLL. This may indicate a buffer overflow attack.
- 5380–phpBB SQL injection: This signature is fired when an HTTP request is made for the CGI script ‘viewtopic.php’ with argument ‘topic_id’ containing either the word ‘union’ or a semicolon.
- 5381–VPASP SQL injection: This signature is fired when a request is made for the CGI script ‘shopexd.asp’ with the argument ‘id’ containing a semicolon.
- 5382– Xpressions SQL Admin Bypass: This signature fires when an attempt to bypass authentication controls to gain administrative access to a Xpressions Interactive application by injecting special-crafted SQL commands into a HTTP request.

- 5383-Cyberstrong eShop SQL Injection: This signature fires when an attempt to insert unauthorized SQL queries into a HTTP request to a Cyberstrong eShop script.

Cross Protocol signature series 6000 series

Cross protocol signatures detect attacks that span multiple protocols. For example, RPC services utilize both TCP and UDP. DNS and authentication failures are some of the other activity covered in the 6000 series.

- 6001-Normal SATAN Probe: This is a supersignature that is fired when a port sweep pattern produced by the SATAN tool is detected.
- 6002-Heavy SATAN Probe: This is a supersignature that is fired when a port sweep pattern produced by the SATAN tool is detected.
- 6050-DNS HINFO Request: This signature fires on an attempt to access HINFO records from a DNS server.
- 6051-DNS Zone Transfer: This signature fires on normal DNS zone transfers, in which the source port is 53.
- 6052-DNS Zone Transfer from High Port: This signature fires on an illegitimate DNS zone transfer, in which the source port is not equal to 53.
- 6053-DNS Request for All Records: This signature fires on a DNS request for all records. Similar to a zone transfer in that it provides a method for transferring DNS records from a server to another requesting host.
- 6054-DNS Version Request: This signature fires when a request for the version of a DNS server is detected.
- 6055-DNS Inverse Query Buffer Overflow: This signature fires when an IQUERY request arrives with a data section that is larger than 255 characters.
- 6056-DNS NXT Buffer Overflow: This signature fires when a DNS server response arrives that has a long NXT resource where the length of the resource data is > 2069 bytes OR the length of the TCP stream containing the NXT resource is > 3000 bytes.
- 6057-DNS SIG Buffer Overflow: This signature fires when a DNS server response arrives that has a long SIG resource where the length of the

resource data is > 2069 bytes OR the length of the TCP stream containing the SIG resource is > 3000 bytes.

- 6058-DNS SRV DoS: Alarms when a DNS query type SRV and DNS query class IN is detected with more than ten pointer jumps in the SRV resource record.
- 6059-DNS TSIG Overflow: Alarms when a DNS query type TSIG is detected and the domain name is greater than 255.
- 6060-DNS complain overflow: Alarms when an NS record is detected with a domain name greater than 255 and the IP address is 0.0.0.0, 255.255.255.255 or a multicast of form 224.X.X.X.
- 6061-DNS infoleak: Alarms when a DNS IQUERY is detected with a record data Length greater than 4 and Class IN.
- 6062-DNS authors request: Alarms when a DNS query type TXT class CHAOS is detected with string "Authors.Bind". This is not case sensitive.
- 6063-DNS Incremental zone transfer: Alarms when a DNS query type of 251 is detected.
- 6064-BIND Large OPT Record DoS: This signature will fire if a DNS request with a OPT resource record containing a large UDP payload length is detected.
- 6100-RPC Port Registration: This signature fires when attempts are made to register new RPC services on a target host.
- 6101-RPC Port Unregistration: This signature fires when attempts are made to unregister existing RPC services on a target host.
- 6102-RPC Dump: This signature fires when an RPC dump request is issued to a target host.
- 6103-Proxied RPC Request: This signature fires when a proxied RPC request is sent to the portmapper of a target host.
- 6104-RPC Set Spoof: This signature fires when an RPC set request with a source address of 127.x.x.x is detected.
- 6105-RPC Unset Spoof: This signature fires when an RPC unset request with a source address of 127.x.x.x is detected.

- 6110-RPC RSTATD Sweep: This signature fires when RPC requests are made to many ports for the RSTATD program.
- 6111-RPC RUSERSD Sweep: This signature fires when RPC requests are made to many ports for the RUSERSD program.
- 6112-RPC NFS Sweep: This signature fires when RPC requests are made to many ports for the NFS program.
- 6113-RPC MOUNTD Sweep: This signature fires when RPC requests are made to many ports for the MOUNTD program.
- 6114-RPC YPPASSWDD Sweep: This signature fires when RPC requests are made to many ports for the YPPASSWDD program.
- 6115-RPC SELECTION_SVC Sweep: This signature fires when RPC requests are made to many ports for the SELECTION_SVC program.
- 6116-RPC REXD Sweep: This signature fires when RPC requests are made to many ports for the REXD program.
- 6117-RPC STATUS Sweep: This signature fires when RPC requests are made to many ports for the STATUS program.
- 6118-RPC ttdb Sweep: This signature fires on an attempt to access the tooltalk database daemon on multiple ports on a single host.
- 6150-ypserv Portmap Request: This signature fires when a request is made to the portmapper for the YP server daemon (ypserv) port.
- 6151-ybind Portmap Request: This signature fires when a request is made to the portmapper for the YP bind daemon (ybind) port.
- 6152-yppasswdd Portmap Request: This signature fires when a request is made to the portmapper for the YP password daemon (yppasswdd) port.
- 6153-ypupdated Portmap Request: This signature fires when a request is made to the portmapper for the YP update daemon (ypupdated) port.
- 6154-ypxfrd Portmap Request: This signature fires when a request is made to the portmapper for the YP transfer daemon (ypxfrd) port.
- 6155-mountd Portmap Request: This signature fires when a request is made to the portmapper for the mount daemon (mountd) port.
- 6175-rexd Portmap Request: This signature fires when a request is made to the portmapper for the remote execution daemon (rex) port.

- 6180–rexrd Attempt: This signature fires when a call to the rexd program is made. The remote execution daemon is the server responsible for remote program execution.
- 6188–statd dot dot: This signature alarms upon detecting a dot dot slash (../) sequence sent to the statd RPC service.
- 6189–statd automount attack: This signature alarms upon detecting a statd bounce attack on the automount process.

NOTE

Signatures 6188 and 6189 are only available in Cisco IDS versions 4.0 and newer.

- 6190–statd Buffer Overflow: This signature fires when a large statd request is sent. This could be an attempt to overflow a buffer and gain access to system resources.
- 6191–RPC.tooltalk buffer overflow: This signature fires when an attempt is made to overflow an internal buffer in the tooltalk rpc program.
- 6192–RPC mountd Buffer Overflow: This signature fires on an attempt to overflow a buffer in the RPC mountd application.
- 6193–RPC CMSD Buffer Overflow: This signature fires when an attempt is made to overflow an internal buffer in the Calendar Manager Service Daemon, rpc.cmsd.
- 6194–sadmind RPC Buffer Overflow: This signature fires when a call to RPC program number 100232 procedure 1 with a UDP packet length > 1024 bytes is detected.
- 6195–RPC amd Buffer Overflow: Signature 6195 will detect the exploitation of the RPC AMD Buffer Overflow vulnerability.
- 6196–snmpXdmid Buffer Overflow: This signature fires when an abnormally long call to the RPC program 100249 (snmpXdmid) and procedure 257 is detected.

- 6197-rpc yppaswdd overflow: This alarm fires when an overflow attempt is detected when sent to yppaswdd RCP-based application.
- 6198-rwalld String Format: This signature fires if an unusually long message is detected being sent to the RPC service rwalld.
- 6199-cachefs Overflow: This alarm fires when an overflow attempt is detected when sent to cachefs, an RCP-based application.
- 6200-Ident Buffer Overflow: This signature fires when a server returns an IDENT reply that is too large.
- 6201-Ident Newline: This signature fires when a server returns an IDENT reply that includes a newline followed by more data.
- 6210-LPRng format String Overflow: Alarms when the first lpr command in a datastream is invalid (first byte != 1-9 ascii) and the length to the first LF is greater than 256.
- 6250-FTP Authorization Failure: This signature fires when a user has failed to authenticate three times in a row, while trying to establish an FTP session.
- 6251-Telnet Authorization Failure: This signature fires when a user has failed to authenticate three times in a row, while trying to establish a telnet session.
- 6252-Rlogin Authorization Failure: This signature fires when a user has failed to authenticate three times in a row, while trying to establish an rlogin session.
- 6253-POP3 Authorization Failure: This signature fires when a user has failed to authenticate three times in a row, while trying to establish a POP3 session.
- 6255-SMB Authorization Failure: This signature fires when a client fails Windows NTs (or Smbas) user authentication three or more consecutive times within a single SMB session.
- 6256- HTTP Authorization Failure: This signature fires when a user has failed to authenticate three times in a row, while trying to log into a secured HTTP website.
- 6275-SGI fam Attempt: This signature detects accesses to the SGI fam RPC daemon. Attackers can use this service to gain information about files on the vulnerable system.

- 6276-TooltalkDB overflow: This signature will alarm upon detecting an rpc connection to rpc program number 100083 using procedure 103 with an buffer greater than 1024.
- 6277-Show Mount Recon: This signature alarms upon detecting an RPC call to show all mounts on an NFS server.
- 6300-Loki ICMP Tunneling: Loki is a tool designed to run an interactive session that is hidden within ICMP traffic.
- 6302-General Loki ICMP Tunneling: This signature fires when an imbalance of ICMP echo replies to echo requests is detected.
- 6350-SQL Query Abuse: This signature fires if a select query is issued using the OPENROWSET() function with an ad hoc exec statement in it.
- 6500-RingZero Trojan: The RingZero Trojan consists of an information transfer (ITS) agent and a port scanning (PST) agent.
- 6501-TFN Client Request: TFN clients and servers by default, communicate using ICMP echo reply packets. This signature looks for ICMP echo reply packets containing potential TFN commands sent from a TFN CLIENT —TO-> a SERVER.
- 6502-TFN Server Reply: TFN clients and servers by default, communicate using ICMP echo reply packets. This signature looks for ICMP echo reply packets containing potential TFN commands sent from a TFN SERVER —TO-> CLIENT.
- 6503-Stacheldraht Client Request: Stacheldraht clients and servers by default, communicate using ICMP echo reply packets. This signature looks for ICMP echo reply packets containing potential commands sent from a Stacheldraht CLIENT —TO—> SERVER.
- 6504-Stacheldraht Server Reply: Stacheldraht clients and servers by default, communicate using ICMP echo reply packets. This signature looks for ICMP echo reply packets containing potential commands sent from a Stacheldraht SERVER —TO—> CLIENT.
- 6505-Trinoo Client Request: Trinoo clients communicate by default on UDP port 27444 using a default command set.
- 6506-Trinoo Server Reply: Trinoo servers reply to clients by default on UDP port 31335 using a default command set.

- 6507-TFN2K Control Traffic:TFN2K is a Distributed Denial of Service tool.
- 6508-Mstream Control Traffic:This signature identifies the control traffic between both the attacker <-> client (aka handler), and between the client (aka handler) <-> server (aka agent or daemon).
- 6901-Net Flood ICMP Reply:This signature fires when a configurable threshold for ICMP Type 0 (Echo Reply) traffic is crossed.
- 6902-Net Flood ICMP Request:This signature fires when a configurable threshold for ICMP Type 8 (Echo Request) traffic is crossed.
- 6903-Net Flood ICMP Any:This signature fires when a configurable threshold for all ICMP traffic is crossed.
- 6910-Net Flood UDP:This signature fires when a configurable threshold for all UDP traffic is crossed.
- 6920-Net Flood TCP:This signature fires when a configurable threshold for all TCP traffic is crossed.

NOTE

By default, signatures 6901, 6902, 6903, 6910, and 6920 are disabled. To use either or all of these signatures first enable them, set the "Rate" parameter to zero, and run for a period of time. This is what is called diagnostic mode. They are a tremendous resource hog and should not be left on.

ARP signature series 7000 series

The 7000 series covers all ARP type traffic. Do not look for any of these in software versions prior to 4.0.

- 7101-ARP Source Broadcast:The sensor saw ARP packets with an ARP payload Source MAC broadcast address.
- 7102-ARP Reply-to-Broadcast:The sensor saw an ARP Reply packet with its payload Destination MAC containing a broadcast address.

- 7104-ARP MacAddress-Flip-Flop-Response: The sensor saw a set of ARP response packets where the ARP payload Mac-to-Ip mapping changed more than MacFlip number of times.
- 7105-ARP Inbalance-of-Requests: The sensor saw many more requests than it saw replies for an IP address out of the ARP payload.

NOTE

The 7000 series signatures are only available in Cisco IDS versions 4.0 and newer.

String Matching signature series 8000 series

These signatures are highly configurable. They allow you to look for specific strings in the payload of a packet. If an attack is underway and there is not already a signature for it, a temporary string match can be put in place to help mitigate some of the risk.

- 8000:2101-FTP Retrieve Password File: This signature fires on string passwd issued during an FTP session.
- 8000:2302-Telnet-/etc/shadow Match: This signature fires on string /etc/shadow issued during a telnet session.
- 8000:2303-Telnet-+ +: This signature fires on string + + issued during a telnet session.
- 8000:51301-Rlogin-IFS Match: This signature fires when an attempt to change the IFS to / is done during a rlogin session.
- 8000:51302-Rlogin-/etc/shadow Match: This signature fires on string /etc/shadow issued during a rlogin session.
- 8000:51303-Rlogin-+ + : This signature fires on string + + issued during a rlogin session.

Back Door signature series 9000 series

Back door signatures are specific to well-known back doors. These signatures fire off of activity that is targeting the known ports and protocols of the backdoor. Any alarms from these signatures should be investigated closely. The ports can be used in valid applications.

- 9000-Back Door Probe (TCP 12345): This signature fires when a TCP SYN packet to port 12345 which is a known trojan port for NetBus as well as the following: Adore sshd, Ashley, cron / crontab, Fat Bitch trojan, GabanBus, icmp_client.c, icmp_pipe.c, Mypic, NetBus Toy, Pie Bill Gates, ValvNet, Whack Job, X-bill.
- 9001-Back Door Probe (TCP 31337): This signature fires when a TCP SYN packet to port 31337 which is a known trojan port for BackFire, Back Orifice, DeepBO, ADM worm, Baron Night, Beeone, bindshell, BO client, BO Facil, BO spy, BO2, cron / crontab, Freak88, Freak2k, Gummo, Linux Rootkit, Sm4ck, Sockdmini.
- 9002-Back Door Probe (TCP 1524): This signature fires when a TCP SYN packet to port 1524 which is a common backdoor placed on machines by worms and hackers.
- 9003-Back Door Probe (TCP 2773): This signature fires when a TCP SYN packet to port 2773 which is a known trojan port for SubSeven.
- 9004-Back Door Probe (TCP 2774): This signature fires when a TCP SYN packet to port 2774 which is a known trojan port for SubSeven.
- 9005-Back Door Probe (TCP 20034): This signature fires when a TCP SYN packet to port 20034 which is a known trojan port for Netbus Pro as well as NetRex and Whack Job.
- 9006-Back Door Probe (TCP 27374): This signature fires when a TCP SYN packet to port 27374 which is a known trojan port for SubSeven as well as Bad Blood, EGO, Fake SubSeven, Lion, Ramen, Seeker, The Saint, Ttfloader and Webhead.
- 9007-Back Door Probe (TCP 1234): This signature fires when a TCP SYN packet to port 1234 which is a known trojan port for SubSeven is detected.
- 9008-Back Door Probe (TCP 1999): This signature fires when a TCP SYN packet to port 1999 which is a known trojan port for SubSeven.

- 9009-Back Door Probe (TCP 6711): This signature fires when a TCP SYN packet to port 6711 which is a known trojan port for SubSeven.
- 9010-Back Door Probe (TCP 6712): This signature fires when a TCP SYN packet to port 6712 which is a known trojan port for SubSeven.
- 9011-Back Door Probe (TCP 6713): This signature fires when a TCP SYN packet to port 6713 which is a known trojan port for SubSeven.
- 9012-Back Door Probe (TCP 6776): This signature fires when a TCP SYN packet to port 6776 which is a known trojan port for SubSeven.
- 9013-Back Door Probe (TCP 16959): This signature fires when a TCP SYN packet to port 16959 which is a known trojan port for SubSeven.
- 9014-Back Door Probe (TCP 27573): This signature fires when a TCP SYN packet to port 27573 which is a known trojan port for SubSeven.
- 9015-Back Door Probe (TCP 23432): This signature fires when a TCP SYN packet to port 23432 which is a known trojan port for asylum.
- 9016-Back Door Probe (TCP 5400): This signature fires when a TCP SYN packet to port 5400 which is a known trojan port for back-construction.
- 9017-Back Door Probe (TCP 5401): This signature fires when a TCP SYN packet to port 5401 which is a known trojan port for back-construction.
- 9018-Back Door Probe (TCP 2115): This signature fires when a TCP SYN packet to port 2115 which is a known trojan port for bugs.
- 9019-Back Door (UDP 2140): This signature fires when a UDP packet to port 2140 which is a known trojan port for deep-throat.
- 9020-Back Door (UDP 47262): This signature fires when a UDP packet to port 47262 which is a known trojan port for delta-source.
- 9021-Back Door (UDP 2001): This signature fires when a UDP packet to port 2001 which is a known trojan port for the Apache/chunked-encoding worm.
- 9022-Back Door (UDP 2002): This signature fires when a UDP packet to port 2002 which is a known trojan port for the Apache/mod_ssl worm.
- 9023-Back Door Probe (TCP 36794): This signature fires when a TCP SYN packet to port 36794 which is a known trojan port for NetBus as well as the following: Bugbear

- 9024-Back Door Probe (TCP 10168): This signature fires when a TCP SYN packet to port 10168 which is a known trojan port for lovegate.
- 9025-Back Door Probe (TCP 20168): This signature fires when a TCP SYN packet to port 20168 which is a known trojan port for lovegate.
- 9026-Back Door Probe (TCP 1092): This signature fires when a TCP SYN packet to port 1092 which is a known trojan port for lovegate.
- 9027-Back Door Probe (TCP 2018): This signature fires when a TCP SYN packet to port 2018 which is a known trojan port for fizzer.
- 9028-Back Door Probe (TCP 2019): This signature fires when a TCP SYN packet to port 2019 which is a known trojan port for fizzer.
- 9029-Back Door Probe (TCP 2020): This signature fires when a TCP SYN packet to port 2020 which is a known trojan port for fizzer.
- 9030-Back Door Probe (TCP 2021): This signature fires when a TCP SYN packet to port 2021 which is a known trojan port for fizzer.
- 9200-Back Door Response (TCP 12345): This signature fires when a TCP SYN/ACK packet from port 12345 which is a known trojan port for NetBus as well as the following: Adore sshd, Ashley, cron / crontab, Fat Bitch trojan, GabanBus, icmp_client.c, icmp_pipe.c, Mypic, NetBus Toy, Pie Bill Gates, ValvNet, Whack Job, X-bill.
- 9201-Back Door Response (TCP 31337): This signature fires when a TCP SYN/ACK packet from port 31337 which is a known trojan port for BackFire, Back Orifice, DeepBO, ADM worm, Baron Night, Beeone, bind-shell, BO client, BO Facil, BO spy, BO2, cron / crontab, Freak88, Freak2k, Gummo, Linux Rootkit, Sm4ck, Sockdmini.
- 9202-Back Door Response (TCP 1524): This signature fires when a TCP SYN/ACK packet from port 1524 which is a common backdoor placed on machines by worms and hackers.
- 9203-Back Door Response (TCP 2773): This signature fires when a TCP SYN/ACK packet from port 2773 which is a known trojan port for SubSeven.
- 9204-Back Door Response (TCP 2774): This signature fires when a TCP SYN/ACK packet from port 2774 which is a known trojan port for SubSeven.

- 9205-Back Door Response (TCP 20034): This signature fires when a TCP SYN/ACK packet from port 20034 which is a known trojan port for Netbus Pro as well as NetRex and Whack Job.
- 9206-Back Door Response (TCP 27374): This signature fires when a TCP SYN/ACK packet from port 27374 which is a known trojan port for SubSeven as well as Bad Blood, EGO, Fake SubSeven, Lion, Ramen, Seeker, The Saint, Ttflooder and Webhead.
- 9207-Back Door Response (TCP 1234): This signature fires when a TCP SYN/ACK packet from port 1234 which is a known trojan port for SubSeven.
- 9208-Back Door Response (TCP 1999): This signature fires when a TCP SYN/ACK packet from port 1999 which is a known trojan port for SubSeven.
- 9209-Back Door Response (TCP 6711): This signature fires when a TCP SYN/ACK packet from port 6711 which is a known trojan port for SubSeven.
- 9210-Back Door Response (TCP 6712): This signature fires when a TCP SYN/ACK packet from port 6712 which is a known trojan port for SubSeven.
- 9211-Back Door Response (TCP 6713): This signature fires when a TCP SYN/ACK packet from port 6713 which is a known trojan port for SubSeven.
- 9212-Back Door Response (TCP 6776): This signature fires when a TCP SYN/ACK packet from port 6776 which is a known trojan port for SubSeven.
- 9213-Back Door Response (TCP 16959): This signature fires when a TCP SYN/ACK packet from port 16959 which is a known trojan port for SubSeven.
- 9214-Back Door Response (TCP 27573): This signature fires when a TCP SYN/ACK packet from port 27573 which is a known trojan port for SubSeven.
- 9215-Back Door Response (TCP 23432): This signature fires when a TCP SYN/ACK packet from port 23432 which is a known trojan port for asylum.

- 9216-Back Door Response (TCP 5400): This signature fires when a TCP SYN/ACK packet from port 5400 which is a known trojan port for back-construction.
- 9217-Back Door Response (TCP 5401): This signature fires when a TCP SYN/ACK packet from port 5401 which is a known trojan port for back-construction.
- 9218-Back Door Response (TCP 2115): This signature fires when a TCP SYN/ACK packet from port 2115 which is a known trojan port for bugs.
- 9223-Back Door Response (TCP 36794): This signature fires when a TCP SYN/ACK packet from port 36794 which is a known trojan port for NetBus as well as the following: Bugbear
- 9224-Back Door Response (TCP 10168): This signature fires when a TCP SYN/ACK packet from port 10168 which is a known trojan port for love-gate.
- 9225-Back Door Response (TCP 20168): This signature fires when a TCP SYN/ACK packet from port 20168 which is a known trojan port for love-gate.
- 9226-Back Door Response (TCP 1092): This signature fires when a TCP SYN/ACK packet from port 1092 which is a known trojan port for love-gate.
- 9227-Back Door Response (TCP 2018): This signature fires when a TCP SYN/ACK packet from port 2018 which is a known trojan port for fizzer.
- 9228-Back Door Response (TCP 2019): This signature fires when a TCP SYN/ACK packet from port 2019 which is a known trojan port for fizzer.
- 9229-Back Door Response (TCP 2020): This signature fires when a TCP SYN/ACK packet from port 2020 which is a known trojan port for fizzer.
- 9230-Back Door Response (TCP 2021): This signature fires when a TCP SYN/ACK packet from port 2021 which is a known trojan port for fizzer.

Policy Violation signature series 10000 series

The policy violation signatures apply to ACL violations. If you are not utilizing ACLs these alarms may or may not be utilized. Before you can use these the router(s) and sensor(s) need to be configured accordingly.

- 10000:1000-IP-Spoof Interface 1: This signature fires on notification from the NetSentry device that an IP datagram has been received in which an IP address that is behind the router has been used as a source address in front of the router.
- 10000:1001-IP-Spoof Interface 2: This signature fires on notification from the NetSentry device that an IP datagram has been received in which an IP address that is behind the router has been used as a source address in front of the router.
- 11000-KaZaA v2 UDP Client Probe: Kazaa is a peer-to-peer (P2P) file sharing application distributed by Sharman Networks.
- 11001-Gnutella Client Request: This signature fires when a peer-to-peer client program based on the gnutella protocol sending out a connection request.
- 11002-Gnutella Server Reply: This signature fires when a peer-to-peer server program based on the gnutella protocol replying to a connection request.
- 11003-Qtella File Request: This signature fires when the Qtella peer-to-peer file sharing client request a file from a sever.
- 11004-Bearshare file request: This signature fires when the BearShare peer-to-peer file sharing client request a file from a sever.
- 11005-KaZaA GET Request: The signature fires when a client request to the default KazaA server port (TCP 1214) is detected.
- 11006-Gnucleus file request: This signature fires when the Gnucleus peer-to-peer file sharing client request a file from a sever.
- 11007-Limewire File Request: This signature fires when the LimeWire peer-to-peer file sharing client request a file from a sever.
- 11008-Morpheus File Request: This signature fires when the Morpheus peer-to-peer file sharing client request a file from a sever.

- 11009-Phex File Request: This signature fires when the Phex peer-to-peer file sharing client request a file from a sever.
- 11010-Swapper File Request: This signature fires when the Swapper peer-to-peer file sharing client request a file from a sever.
- 11011-XoloX File Request: This signature fires when the BearShare peer-to-peer file sharing client request a file from a sever.
- 11012-GTK-Gnutella File Request: This signature fires when the GTK-Gnutella peer-to-peer file sharing client request a file from a sever.
- 11013-Mutella File Request: This signature fires when the Mutella peer-to-peer file sharing client request a file from a sever.
- 11014-Hotline Client Login: This signature is fired when a Hotline client logs into a hotline server.
- 11015-Hotline File Transfer: This signature is fired when a Hotline file transfer is initiated.
- 11016-Hotline Tracker Login: This signature is fired when a Hotline client contacts a Hotline tracker server.
- 11200-Yahoo Messenger Activity: This signature fires when a Yahoo Messenger client login attempt to the default TCP port 5050 is detected.
- 11201-MSN Messenger Activity: This signature fires when an MSN new connection attempt to the default TCP port 1863 is detected.
- 11202-AOL / ICQ Activity: This signature fires when an AOL / ICQ new connection attempt to the default TCP port 5190 is detected.
- 11203- IRC Channel Join: This signature fires when an attempt to join an IRC (Internet Relay Chat) channel is detected.
- 11204-Jabber Activity: This signature fires when a Jabber client login attempt to the default TCP port is detected.

Sensor Status Alarms

Sensor status alarms are used to monitor the health of the sensor daemons. Events like daemons going down and daemons unstartable appear when sensor services fail or cannot be started or restarted. These give health and status of the sensor and communication between the sensor and director.

- 993-Missed Packet Count: This signature is fired when the sensor is dropping packets and the percentage dropped can be used to help you tune the traffic level you are sending to the sensor. For example, if the alarms show that there is a low count of dropped packets or even zero, the sensor is monitoring the traffic without being overutilized. On the other hand, if 993 alarms show a high count dropped packets, the sensor may be oversubscribed.
- 994-Traffic Flow Started: This signature fires when traffic to the sensing interface is detected for the first time or resuming after an outage. SubSig 1 fires when initial network activity is detected. SubSig 2 fires when the link (physical) layer becomes active.
- 995-Traffic Flow Stopped: subsignature 1 is fired when no traffic is detected on the sensing interface. You can tune the timeout for this using the TrafficFlowTimeout parameter. SubSignature 2 is fired when a physical link is not detected.
- 993-Missed Packet Count: This signature is fired when the sensor is dropping packets and the percentage dropped can be used to help you tune the traffic level you are sending to the sensor. For example, if the alarms show that there is a low count of dropped packets or even zero, the sensor is monitoring the traffic without being overutilized. On the other hand, if 993 alarms show a high count dropped packets, the sensor may be oversubscribed.
- 994-Traffic Flow Started: This signature fires when traffic to the sensing interface is detected for the first time or resuming after an outage. SubSig 1 fires when initial network activity is detected. SubSig 2 fires when the link (physical) layer becomes active.
- 995-Traffic Flow Stopped: subsignature 1 is fired when no traffic is detected on the sensing interface. You can tune the timeout for this using the TrafficFlowTimeout parameter. SubSignature 2 is fired when a physical link is not detected.
- 996 - Route Up: This signifies that traffic between the sensor and director has started. When the services on the director and/or sensor are started this alarm will appear in the event viewer.
- 997 - Route Down: This signifies that traffic between the sensor and director has stopped. When the services on the director and/or sensor are started this alarm will appear in the event viewer.

- 998 - Daemon Down: One or more of the IDS sensor services has stopped.
- 999 - Daemon Unstartable: One or more of the IDS sensor services is unable to be started.

IDS signatures grouped by software release version

For configuration management purposes, the following list of signatures is grouped by the software release version from which it was publicly released. For more information regarding these signatures refer to the signature descriptions above or go to www.cisco.com.

- **Release version S49**

- 3327-Windows RPC DCOM Overflow

- 3328-Windows SMB/RPC NoOp Sled

- **Release version S48**

- 1109-Cisco IOS Interface DoS

- 5380-phpBB SQL injection:

- 5382- Xpressions SQL Admin Bypass

- 5383-Cyberstrong eShop SQL Injection

- 6256- HTTP Authorization Failure

- **Release version S47**

- 5375-Apache mod_dav Overflow

- 5376-iisPROTECT Admin SQL Injection

- 5377-xp_cmdshell in HTTP args

- 5378-Vignette TCL Injection Command Exec

- 5379-Windows Media Services Logging ISAPI Overflow

- 11204-Jabber Activity

- **Release version S46**

- 3123-NetBus Pro Traffic

3124-Sendmail prescan Memory Corruption
3176-Cisco ONS FTP DoS
3326-Windows Startup Folder Remote Access
5369-Win32 Apache Batch File CmdExec
5370-HTDig File Disclosure
5371-bdir.htr Access
5372-ASP %20 source disclosure
5373-IIS 5 Translate: f Source Disclosure
5374-IIS Executable File Command Exec
9025-Back Door Probe (TCP 20168)
9026-Back Door Probe (TCP 1092)
9027-Back Door Probe (TCP 2018)
9028-Back Door Probe (TCP 2019)
9029-Back Door Probe (TCP 2020)
9030-Back Door Probe (TCP 2021)
9225-Back Door Response (TCP 20168)
9226-Back Door Response (TCP 1092)
9227-Back Door Response (TCP 2018)
9228-Back Door Response (TCP 2019)
9229-Back Door Response (TCP 2020)
9230-Back Door Response (TCP 2021)
11014-Hotline Client Login
11015-Hotline File Transfer
11016-Hotline Tracker Login
11200-Yahoo Messenger Activity
11201-MSN Messenger Activity

■ **Release version S44**

1300-TCP Segment Overwrite

3325-Samba call_trans2open Overflow
3732-MSSQL xp_cmdshell Usage
5367-Apache CR / LF DoS
5368-Cisco ACS Windows CSAdmin Overflow
9024-Back Door Probe (TCP 10168)
9224-Back Door Response (TCP 10168)
11001-Gnutella Client Request
11002-Gnutella Server Reply
11003-Qtella File Request
11004-Bearshare file request
11005-KaZaA GET Request
11006-Gnucleus file request
11007-Limewire File Request
11008-Morpheus File Request
11009-Phex File Request
11010-Swapper File Request
11011-XoloX File Request
11012-GTK-Gnutella File Request

■ **Release version S43**

3311-SMB: remote SAM service access attempt
3312-SMB .eml e-mail file remote access
3313-SMB suspicious password usage
3320-SMB: ADMIN\$ hidden share access attempt
3321-SMB: User Enumeration
3322-SMB: Windows Share Enumeration
3323-SMB: RFPoison Attack
3324-SMB NIMDA infected file transfer

4003-Nmap UDP Port Sweep
5360-Frontpage htimage.exe Buffer Overflow
5363-Frontpage imagemap.exe Buffer Overflow
5364-IIS WebDAV Overflow
5365-Long WebDAV Request
5366-Shell Code in HTTP URL / Args
6188-statd dot dot
6189-statd automount attack

■ **Release version S42**

5362-FrontPage dvwssr.dll Buffer Overflow

■ **Release version S41**

3115-Sendmail Data Header Overflow
5351-MS IE Help Overflow
5352-H-Sphere Webshell Buffer Overflow
5353-H-Sphere Webshell 'mode' URI exec
5354-H-Sphere Webshell 'zipfile' URI exec
5355-DotBr exec.php3 exec
5356-DotBr system.php3 exec
5357-IMP SQL Injection
5358-Psunami.CGI Remote Command Execution
5359-Office Scan CGI Scripts Access

■ **Release version S40**

3314-Windows Locator Service Overflow
4614-DHCP request overflow
9200-Back Door Response (TCP 12345)
9201-Back Door Response (TCP 31337)
9202-Back Door Response (TCP 1524)

9203-Back Door Response (TCP 2773)
9204-Back Door Response (TCP 2774)
9205-Back Door Response (TCP 20034)
9206-Back Door Response (TCP 27374)
9207-Back Door Response (TCP 1234)
9208-Back Door Response (TCP 1999)
9209-Back Door Response (TCP 6711)
9210-Back Door Response (TCP 6712)
9211-Back Door Response (TCP 6713)
9212-Back Door Response (TCP 6776)
9213-Back Door Response (TCP 16959)
9214-Back Door Response (TCP 27573)
9215-Back Door Response (TCP 23432)
9216-Back Door Response (TCP 5400)
9217-Back Door Response (TCP 5401)
9218-Back Door Response (TCP 2115)
9223-Back Door Response (TCP 36794)

■ **Release version S39**

4701-MS-SQL Control Overflow

■ **Release version S38**

5349-Polycom ViewStation Admin Password

5350-PHPnuke e-mail attachment access

6064-BIND Large OPT Record DoS

■ **Release version S37**

3174-SuperStack 3 NBX FTP DOS

3175-ProFTPD STAT DoS

3652-SSH Gobbles

4508-Non SNMP Traffic
4613-TFTP Filename Buffer Overflow
5343-Apache Host Header Cross Site Scripting
5345-HTTPBench Information Disclosure
5346-BadBlue Information Disclosure
5347-Xoops WebChat SQL Injection
5348-Cobalt RaQ Server overflow.cgi Cmd Exec
7101-ARP Source Broadcast
7102-ARP Reply-to-Broadcast
7104-ARP MacAddress-Flip-Flop-Response
7105-ARP Inbalance-of-Requests
11000-KaZaA v2 UDP Client Probe

■ **Release version S36**

5344-IIS MDAC RDS Buffer Overflow

■ **Release version S35**

4611-D-Link DWL-900AP+ TFTP Config Retrieve
4612-Cisco IP Phone TFTP Config Retrieve
5294-BearShare File Disclosure
5339-SunONE Directory Traversal
5340-Killer Protection Credential File Access
5341-HP Procurve 4000M Switch DoS
5342-Invision Board phpinfo.php Recon

■ **Release version S34**

3173-Long FTP Command
3465-Finger Activity
3502-rlogin Activity
3604-Cisco Catalyst CR DoS

5337-Dot Dot Slash in HTTP Arguments

5338-Front Page Admin password retrieval

■ **Release version S33**

5331-Image Javascript insertion

5333-FUDForum File Disclosure

5334- DB4Web File Disclosure

5335-DB4WEB Proxy Scan

5336- Abyss Web Server File Disclosure

9023-Back Door Probe (TCP 36794)

■ **Release version S32**

5330-Apache/mod_ssl Worm Buffer Overflow

9021-Back Door (UDP 2001)

9022-Back Door (UDP 2002)

■ **Release version S31**

3121-Vintra MailServer EXPN DoS

3122-SMTP EXPN root Recon

3165-FTP SITE EXEC

3168-FTP SITE EXEC Directory Traversal

3169-FTP SITE EXEC tar

3170-WS_FTP SITE CPWD Buffer Overflow

3171-Ftp Priviledged Login

3172-Ftp Cwd Overflow

3310-Netbios Enum Share DoS

3406-Solaris TTYPROMPT /bin/login Overflow

3457-Finger root shell

3461-Finger probe

3462-Finger Redirect

3463-Finger root
3464-File access in finger
3551-POP User Root
3711-Informer FW1 auth replay DoS
4061-Chargen Echo DoS
4509-HP Openview SNMP Hidden Community Name
4510-Solaris SNMP Hidden Community Name
4511-Avaya SNMP Hidden Community Name
4609-Orinoco SNMP Info Leak
4610-Kerberos 4 User Recon
5321-Guest Book CGI access
5322-Long HTTP Request
5323-midicart.mdb File Access
5327-Tilde in URI
5328- Cisco IP phone DoS
6277-Show Mount Recon

■ **Release version S30**

2155-Modem DoS
3730-Trinoo (TCP)
3731-IMail HTTP Get Buffer Overflow
4606-Cisco TFTP Long Filename Buffer Overflow
4607-Deep Throat Response
4608-Trinoo (UDP)
5310-INDEX / directory access
5311-8.3 file name access
5323-Cisco Router http exec command
5324-Cisco IOS Query (?/)

5325-Contivity cgiproc DoS

5326-Root.exe access

6275-SGI fam Attempt

6276-TooltalkDB overflow

■ **Release version S29**

3728-Long pop username

3729-Long pop password

4603-DHCP Discover

4604-DHCP Request

4605-DHCP Offer

5305-`.bash_history` File Access

5305:1-`.sh_history` File Access

5305:2-`.history` File Access

5305:3-`.zhistory` File Access

5306-SoftCart `storemgr.pw` File Access

5308-`rpc-nlog.pl` Command Execution

5309- `handler CGI` Command Execution

5312-`*.jsp/*.jhtml` Java Execution

5313-`order.log` File Access

5316-BadBlue Admin Command Exec

5317-Tivoli Endpoint Buffer Overflow

5318-Tivoli ManagedNode Buffer Overflow

5319-SoftCart `orders` Directory Access

5320-ColdFusion `administrator` Directory Access

■ **Release version S28**

3167-Format String in FTP username

3708-AnalogX Proxy Socks4a DNS Overflow

- 3709–AnalogX Proxy Web Proxy Overflow
- 3710–Cisco Secure ACS Directory Traversal
- 5282–IIS ExAir advsearch.asp Access
- 5282:1–IIS ExAir search.asp Access
- 5282:2–IIS ExAir query.asp Access
- 5287–SiteServer AdSamples SITE.CSC File Access
- 5288–Verity search97 Directory Traversal
- 5289–SQLXML ISAPI Buffer Overflow
- 5291–WEB-INF Dot File Disclosure
- 5292–SalesCart shop.mdb File Access
- 5293–robots.txt File Access
- 5295–finger CGI Recon
- 5296–Netscape Server PageServices Directory Access
- 5297–order_log.dat File Access
- 5298–shopper.conf File Access
- 5299–quikstore.cfg File Access
- 5300–reg_echo.cgi Recon
- 5301–/consolehelp/ CGI File Access
- 5302–/file/ WebLogic File Access
- 5303–pfdispaly.cgi Command Execution
- 5304–files.pl File Access
- 5314– windmail.exe Command Execution
- **Release version S27**
 - 1108–IP Packet with Proto 11
 - 5279–JJ CGi Cmd Exec
 - 5280–IIS idq.dll Directory Traversal
 - 5281–Carello add.exe Access

5283-info2www CGI Directory Traversal

5284- IIS webhits.dll Directory Traversal

5285-PHPEventCalendar Cmd Exec

5286-WebScripts WebBBS Cmd Exec

■ **Release version S26**

3707-Perl fingerd Command Exec

3714-Oracle TNS 'Service_Name' Overflow

5243-CS .cgi Script Cmd Exec

5275-Phorum Remote Cmd Exec

5276-cart.cgi Command Execution

5276:1-cart.cgi vars,env,db Recon

5276:2-cart.cgi Backdoor

5277- dfire.cgi Command Exec

5278-VP-ASP shoptest.asp access

9015-Back Door Probe (TCP 23432)

9016-Back Door Probe (TCP 5400)

9017-Back Door Probe (TCP 5401)

9018-Back Door Probe (TCP 2115)

9019-Back Door (UDP 2140)

9020-Back Door (UDP 47262)

■ **Release version S25**

3705-Tivoli Storage Manager Client Acceptor Overflow

3706-MIT PGP Public Key Server Overflow

5251-Allaire JRun // Directory Disclosure

5262-Large number of Slashes URL

5263-ecware.exe Access

5265-RedHat cachemgr.cgi Access

5266-iCat Carbo Server File Disclosure
5268-Cisco Catalyst Remote Command Execution
5269-ColdFusion CFDOCS Directory Access
5270-EZ-Mall order.log File Access
5271-search.cgi Directory Traversal
5272-count.cgi GIF File Disclosure
5273-Bannermatic Sensitive File Access
5274-Netpad.cgi Directory Traversal/Cmd Exec

■ **Release version S24**

3702-Default sa account access
5249-IDS Evasive Encoding
5250-IDS Evasive Double Encoding
5252-Allaire JRun Session ID Recon
5253-Axis StorPoint CD Authentication Bypass
5254-Sambar Server CGI Dos Batch File
5255-Linux Directory traceroute / nslookup Command Exec
5256-Dot Dot Slash in URI
5257-PHPNetToolpack traceroute Command Exec
5258-Script source disclosure with CodeBrws.asp
5259-Snitz Forums SQL injection
5260-Xpede sprc.asp SQL Injection
5261-BackOffice Server Web Administration Access

■ **Release version S23**

6199-cachefsd Overflow

■ **Release version S22**

6198-rwalld String Format
9007-Back Door Probe (TCP 1234)

9008-Back Door Probe (TCP 1999)

9009-Back Door Probe (TCP 6711)

9010-Back Door Probe (TCP 6712)

9011-Back Door Probe (TCP 6713)

9012-Back Door Probe (TCP 6776)

9013-Back Door Probe (TCP 16959)

9014-Back Door Probe (TCP 27573)

■ **Release version S21**

3704-IIS FTP STAT Denial of Service

5244- PhpSmsSend Command Exec

5245- HTTP 1.1 Chunked Encoding Transfer

5246-IIS ISAPI Filter Buffer Overflow

5247-IIS ASP SSI Buffer Overflow

5248-IIS HTR ISAPI Buffer Overflow

■ **Release version S20**

5240-Marcus Xenakis Shell Command Exec

5241-Avenger System Command Exec

9000-Back Door Probe (TCP 12345)

9001-Back Door Probe (TCP 31337)

9002-Back Door Probe (TCP 1524)

9003-Back Door Probe (TCP 2773)

9004-Back Door Probe (TCP 2774)

9005-Back Door Probe (TCP 20034)

9006-Back Door Probe (TCP 27374)

■ **Release version S19**

3166- FTP USER Suspicious Length

3703-Squid FTP URL Buffer Overflow

5232-URL with XSS

5234-pforum sql-injection

5236-Xoops sql-injection

5237-HTTP CONNECT Tunnel

5238-EZNET Ezboard Buffer Overflow

5239-Sambar cgitest.exe Buffer Overflow

■ **Release version S18**

3164- Instant Server Mini Portal Directory Traversal

3405- Avirt Gateway proxy Buffer Overflow

3701-Oracle 9iAS Web Cache Buffer Overflow

5227- AHG Search Engine Command Exec

5229- DCP Portal Root Path Disclosure

5230- Lotus Domino Authentication Bypass

5231- MRTG Directory Traversal

5233-PHP fileupload Buffer Overflow

■ **Release version S17**

4507-SNMP Protocol Violation

5223-Pi3Web Buffer Overflow

5224-SquirrelMail SquirrelSpell Command Exec

■ **Release version S16**

4506-D-Link Wireless SNMP Plain Text Password

5197-Network Query Tool command Exec

5201-PHP-Nuke Cross Site Scripting

5203- Hosting Controller File Access and Upload

5205-Apache php.exe File Disclosure

5209-Agora.cgi Cross Site Scripting

5210-FAQManager.cgi directory traversal

5211-zml.cgi File Disclosure
5212-Bugzilla Admin Authorization Bypass
5213-Bugzilla Command Exec
5214-FAQManager.cgi null bytes
5215-lastlines.cgi cmd exec/traversal
5216-PHP Rocket Directory Traversal
5217-Webmin Directory Traversal
5218-Boozt Buffer Overflow
5219-Lotus Domino database DoS
5220-CSVForm Remote Command Exec
5221-Hosting Controller Directory Traversal

■ **Release version S15**

3700-CDE dtspcd overflow

■ **Release version S14**

3404-SysV /bin/login Overflow
3458-AIM game invite overflow
3459-ValiCert forms.exe overflow
4058-UPnP LOCATION Overflow
5202- PHP-Nuke File Copy / Delete
5204-AspUpload Sample Scripts
5206-Horde IMP Session Hijack
5207-Entrust GetAccess directory traversal
5208-Network Tools shell metacharacters

■ **Release version S13**

3117-KLEZ worm
3118-rwhoisd format string
3119-WS_FTP STAT overflow

- 3120-ANTS virus
- 3163-wu-ftpd heap corruption vulnerability
- 3403-Telnet Excessive Environment Options
- 3456- Solaris in.fingerd Information Leak
- 3501-Rlogin Long TERM Variable
- 5183-PHP File Inclusion Remote Exec
- 5191-Active Perl PerlIS.dll Buffer Overflow
- 5194-Apache Server .ht File Access
- 5195-AS/400 '/' attack
- 5196-Red Hat Stronghold Recon attack
- 5199-W3Mail Command Exec
- 5200-IIS Data Stream Source Disclosure
- **Release version S12**
 - 1107-RFC 1918 Addresses Seen
 - 3116-Netbus
 - 3651-SSH CRC32 Overflow
 - 5184-Apache Authentication Module ByPass
 - 5188-HTTP Tunneling
- **Release version S11**
 - 5178-MS Index Server File/Path Recon
 - 5179-PHP-Nuke File Upload
 - 5180-sgiMerchant Directory Traversal
 - 5181-MacOS Apache File Disclosure
 - 5181:1-MacOS Apache File Disclosure
 - 5182-WebDiscount's eShop Arbitrary Command Exec
- **Release version S10**
 - 3112-Lotus Domino Mail Loop DoS
 - 3460-AVTronics InetServer Buffer Overflow

4060-Back Orifice Ping

5173-Directory Manager Cmd Exec

5174-phpmyexplorer directory traversal

5175-Hassan Shopping Cart Command Exec

5176-Exchange Address List Disclosure

■ **Release version S9**

3114-FetchMail Arbitrary Code Execution

3162-glFtpD LIST DoS

3455-Java Web Server Cmd Exec

4101-Cisco TFTP Directory Traversal

4601-CheckPoint Firewall RDP Bypass

5170-NULL byte in URI

5171-NC-Book book.cgi Cmd Exec

5172-WinWrapper Admin Server Directory Traversal

6197-rpc yppaswdd overflow

■ **Release version S8**

5163-Mambo SiteServer Administrative Password ByPass

5164-PHPBB Remote SQL Query Manipulation

5165-php-nuke article.php sql query

5166-php-nuke modules.php DoS

5167-phpMyAdmin Cmd Exec 2

5168-Snapstream PVS Directory Traversal Bug

5169-SnapStream PVS Plaintext Password Vulnerability

■ **Release version S7**

3111-W32 Sircam Malicious Code

3111:1-W32 Sircam Malicious Code

3454-Check Point Firewall Information Leak

4601:1-CheckPoint Firewall RDP Bypass
4601:2-CheckPoint Firewall RDP Bypass
4601:3-CheckPoint Firewall RDP Bypass
5158-iPlanet Proprietary Method Overflow
5159-phpMyAdmin Cmd Exec
5160-Apache ? indexing file disclosure bug
5160:1-Apache ? indexing file disclosure bug
5161-SquirrelMail Command Exec
5162-Active Classifieds Command Exec

■ **Release version S6**

3161-FTP realpath Buffer Overflow
3402-BSD Telnet Daemon Buffer Overflow
3453-MS NetMeeting RDS DoS
5134-MacOS PWS DoS
5142-DCShop File Disclosure
5147-Arcadia Internet Store Directory Traversal Attempt
5148-Perception LiteServe Web Server CGI Script Source Code Disclosu
5149-Trend Micro Interscan Viruswall Configuration Modification
5150-InterScan VirusWall RegGo.dll Buffer Overflow
5151-WebStore Admin Bypass
5152-WebStore Command Exec
5154-WWW uDirectory Directory Traversal
5155-WWW SiteWare Editor Directory Traversal
5156-WWW Microsoft fp30reg.dll Overflow
5157-Tarantella TTAWebTop.CGI Directory Traversal Bug

■ **Release version S5**

993-Missed Packet Count
994-Traffic Flow Started

995-Traffic Flow Stopped
3451-BearShare Directory Traversal
3452-gopherd halidate overflow
5124-IIS CGI Double Decode
5125-PerlCal Directory Traversal
5126-WWW IIS .ida Indexing Service Overflow
5127-WWW viewsrc.cgi Directory Traversal
5128-WWW nph-maillist.pl Cmd Exec
5129-IOS HTTP Unauth Command Execution
5130-Bugzilla globals.pl
5131-talkback.cgi Directory Traversal
5132-VirusScan catinfo Buffer Overflow
5133-Net.Commerce Macro Path Disclosure
5138-Oracle Application Server Shared Library Overflow
5140-Net.Commerce Macro Denial of Service
5141-NCM content.pl SQL Query Vulnerability
5143-Microsoft Media Player ASX Overflow
5146-MS-DOS Device Name DoS

■ **Release version S4**

4056-NTPd readvar overflow
5120-Netscape Server Suite Buffer Overflow
5121-iPlanet .shtml Buffer Overflow
5122-Nokia IP440 Denial of Service
5123- WWW IIS Internet Printing Overflow
6196-snmPXdmid Buffer Overflow
6901-Net Flood ICMP Reply
6902-Net Flood ICMP Request

6903-Net Flood ICMP Any

6910-Net Flood UDP

6920-Net Flood TCP

■ **Release version S3**

3046-NMAP OS Fingerprint

3158-FTP SITE EXEC Format String

3159-FTP PASS Suspicious Length

4500-Cisco IOS Embedded SNMP Community Names

4501-Cisco CVCO/4K Remote Username/Password return

4502-SNMP Password Brute Force Attempt

4503-SNMP NT Info Retrieve

4504-SNMP IOS Configuration Retrieval

4505-SNMP VACM MIB Access

5115-Netscape Enterprise Server with ?wp Tags

5116-Endymion MailMan Remote Command Execution

5117-phpGroupWare Remote Command Exec

5118-eWave ServletExec 3.0C File Upload

5119-CGI Script Center News Update Admin Passwd Change

6058-DNS SRV DoS

6059-DNS TSIG Overflow

6060-DNS complain overflow

6061-DNS infoleak

6062-DNS authors request

6063-DNS Incremental zone transfer

6210-LPRng format String Overflow

6350-SQL Query Abuse

■ **Release version 2.2.1.6**

- 1220-Jolt2 Fragment Reassembly DoS attack
- 3530-Cisco Secure ACS Oversized TACACS+ Attack
- 3540-Cisco Secure ACS CSAdmin Attack
- 5079-WWW PCCS MySQL Admin Access
- 5080-WWW IBM WebSphere Access
- 5081-WWW WinNT cmd.exe Access
- 5083-WWW Virtual Vision FTP Browser Access
- 5084-WWW Alibaba Attack 2
- 5085-WWW IIS Source Fragment Access
- 5086-WWW WEBactive Logfile Access
- 5087-WWW Sun Java Server Access
- 5088-WWW Akopia MiniVend Access
- 5089-WWW Big Brother Directory Access
- 5090-WWW FrontPage htmimage.exe Access
- 5091-WWW Cart32 Remote Admin Access
- 5092-WWW CGI-World Poll It Access
- 5093-WWW PHP-Nuke admin.php3 Access
- 5095-WWW CGI Script Center Account Manager Attack
- 5096-WWW CGI Script Center Subscribe Me Attack
- 5097-WWW FrontPage MS-DOS Device Attack
- 5099-WWW GWScripts News Publisher Access
- 5100-WWW CGI Center Auction Weaver File Access
- 5101-WWW CGI Center Auction Weaver Attack
- 5102-WWW phpPhotoAlbum explorer.php Access
- 5103-WWW SuSE Apache CGI Source Access
- 5104-WWW YaBB File Access

5105-WWW Ranson Johnson mailto.cgi Attack
5106-WWW Ranson Johnson mailform.pl Access
5107-WWW Mandrake Linux /perl Access
5108-WWW Netegrity Site Minder Access
5109-WWW Sambar Beta search.dll Access
5110-WWW SuSE Installed Packages Access
5111-WWW Solaris Answerbook 2 Access
5112-WWW Solaris Answerbook 2 Attack
5113-WWW CommuniGate Pro Access
5114-WWW IIS Unicode Attack

■ **Release version 2.2.1.5**

1200-IP Fragmentation Buffer Full
1201-IP Fragment Overlap
1202-IP Fragment Overrun - Datagram Too Long
1203-IP Fragment Overwrite - Data is Overwritten
1204-IP Fragment Missing Initial Fragment
1205-IP Fragment Too Many Datagrams
1206-IP Fragment Too Small
1207-IP Fragment Too Many Frags
1208-IP Fragment Incomplete Datagram
3110-Suspicious Mail Attachment
3157-FTP PASV Port Spoof
3603-IOS Enable Bypass
5056-WWW Cisco IOS %% DoS
5057-WWW Sambar Samples
5058-WWW info2www Attack
5059-WWW Alibaba Attack

- 5060-WWW Excite AT-generate.cgi Access
- 5061-WWW catalog_type.asp Access
- 5062-WWW classifieds.cgi Attack
- 5063-WWW dmbparser.exe Access
- 5064-WWW imagemap.cgi Attack
- 5065-WWW IRIX infosrch.cgi Attack
- 5066-WWW man.sh Access
- 5067-WWW plusmail Attack
- 5068-WWW formmail.pl Access
- 5069-WWW whois_raw.cgi Attack
- 5070-WWW msadcs.dll Access
- 5071-WWW msacds.dll Attack
- 5072-WWW bizdb1-search.cgi Attack
- 5073-WWW EZshopper loadpage.cgi Attack
- 5074-WWW EZshopper search.cgi Attack
- 5075-WWW IIS Virtualized UNC Bug
- 5076-WWW webplus bug
- 5077-WWW Excite AT-admin.cgi Access
- 5078-WWW Piranha passwd attack
- 6054-DNS Version Request
- 6507-TFN2K Control Traffic
- 6508-Mstream Control Traffic
- **Release version 2.2.1.4**
 - 6056-DNS NXT Buffer Overflow
 - 6057-DNS SIG Buffer Overflow
 - 6195-RPC amd Buffer Overflow

- **Release version 2.2.1.3**

- 3650-SSH RSAREF2 Buffer Overflow
- 3990-BackOrifice BO2K TCP Non Stealth
- 3991-BackOrifice BO2K TCP Stealth 1
- 3992-BackOrifice BO2K TCP Stealth 2
- 4055-BackOrifice BO2K UDP
- 5055-HTTP Basic Authentication Overflow
- 6194-sadmind RPC Buffer Overflow
- 6500-RingZero Trojan
- 6501-TFN Client Request
- 6502-TFN Server Reply
- 6503-Stacheldraht Client Request
- 6504-Stacheldraht Server Reply
- 6505-Trinoo Client Request
- 6506-Trinoo Server Reply

- **Release version 2.2.1.2**

- 3155-FTP RETR Pipe Filename Command Execution
- 3156-FTP STOR Pipe Filename Command Execution
- 3308-Windows LSARPC Access
- 3309-Windows SRVSVC Access
- 5051-IIS Double Byte Code Page
- 5052-FrontPage Extensions PWD Open Attempt
- 5053-FrontPage _vti_bin Directory List Attempt
- 5054-WWWBoard Password
- 6193-RPC CMSD Buffer Overflow

- **Release version 2.2.1.1**

- 1104-IP Localhost Source Spoof

3038-Fragmented NULL TCP Packet
3039-Fragmented Orphaned FIN packet
3040-NULL TCP Packet
3041-SYN/FIN Packet
3042-Orphaned Fin Packet
3043-Fragmented SYN/FIN Packet
3201-Unix Password File Access Attempt
4054-RIP Trace
5034-WWW IIS newdsn attack
5035-HTTP cgi HylaFAX Faxsurvey
5036-WWW Windows Password File Access Attempt
5037-WWW SGI MachineInfo Attack
5038-WWW wwwsql file read Bug
5039-WWW finger attempt
5040-WWW Perl Interpreter Attack
5041-WWW anyform attack
5042-WWW CGI Valid Shell Access
5043-WWW Cold Fusion Attack
5044-WWW Webcom.se Guestbook attack
5045-WWW xterm display attack
5046-WWW dumpenv.pl recon
5047-WWW Server Side Include POST attack
5048-WWW IIS BAT EXE attack
5049-WWW IIS showcode.asp access
5050-WWW IIS .htr Overflow Attack
6055-DNS Inverse Query Buffer Overflow
6104-RPC Set Spoof
6105-RPC Unset Spoof

- **Release version 2.2.0.3**
 - 4053-Back Orifice
- **Release version 2.2**
 - 4002-UDP Flood
- **Release version 2.1.1.6**
 - 3109-Long SMTP Command
 - 3229-Website Win-C-Sample Buffer Overflow
 - 3230-Website Uploader
 - 3231-Novell convert
 - 3232-WWW finger attempt
 - 3233-WWW count-cgi Overflow
 - 3525-IMAP Authenticate Buffer Overflow
 - 3526-Imap Login Buffer Overflow
 - 3550-POP Buffer Overflow
 - 3575-INN Buffer Overflow
 - 3576-INN Control Message Exploit
 - 3600-IOS Telnet Buffer Overflow
 - 3601-IOS Command History Exploit
 - 4051-Snork
 - 4052-Chargen DoS
 - 4150-Ascend Denial of Service
 - 6118-RPC ttdb Sweep
 - 6191-RPC.tooltalk buffer overflow
 - 6192-RPC mountd Buffer Overflow
- **Release version 2.1.1.5**
 - 3030-TCP SYN Host Sweep
 - 3031-TCP FRAG SYN Host Sweep

3032-TCP FIN Host Sweep
3033-TCP FRAG FIN Host Sweep
3034-TCP NULL Host Sweep
3035-TCP FRAG NULL Host Sweep
3036-TCP SYN FIN Host Sweep
3037-TCP FRAG SYN FIN Host Sweep
3108-MIME Overflow Bug
6110-RPC RSTATD Sweep
6111-RPC RUSERSD Sweep
6112-RPC NFS Sweep
6113-RPC MOUNTD Sweep
6114-RPC YPPASSWDD Sweep
6115-RPC SELECTION_SVC Sweep
6116-RPC REXD Sweep
6117-RPC STATUS Sweep

■ **Release version 2.1.1.3**

3002-TCP SYN Port Sweep
3003-TCP Frag SYN Port Sweep
3005-TCP FIN Port Sweep
3006-TCP Frag FIN Port Sweep
3010-TCP High Port Sweep
3011-TCP FIN High Port Sweep
3012-TCP Frag FIN High Port Sweep
3015-TCP Null Port Sweep
3016-TCP Frag Null Port Sweep
3020-TCP SYN FIN Port Sweep
3021-TCP Frag SYN FIN Port Sweep

3106-Mail Spam
3107-Majordomo Execute Attack
3221-WWW cgi-viewsources Attack
3222-WWW PHP Log Scripts Read Attack
3223-WWW IRIX cgi-handler Attack
3224-HTTP WebGais
3225-WWW webserversendmail File Access
3226-WWW Webdist Bug
3227-WWW Htmlscript Bug
3228-WWW Performer Bug
3251-TCP Hijacking Simplex Mode
3400-Sunkill
6180-rexd Attempt
6190-statd Buffer Overflow

■ **Release version 2.1.1**

1001-IP options-Record Packet Route
1002-IP options-Timestamp
1004-IP options-Loose Source Route
1006-IP options-Strict Source Route
1102-Impossible IP Packet
1103-IP Fragments Overlap
2100-ICMP Network Sweep w/Echo
2101-ICMP Network Sweep w/Timestamp
2102-ICMP Network Sweep w/Address Mask
2150-Fragmented ICMP Traffic
2153-Smurf
3001-TCP Port Sweep

3100-Smail Attack
3101-Sendmail Invalid Recipient
3102-Sendmail Invalid Sender
3103-Sendmail Reconnaissance
3104-Archaic Sendmail Attacks
3105-Sendmail Decode Alias
3150-FTP Remote Command Execution
3151-FTP SYST Command Attempt
3152-FTP CWD ~root
3153-FTP Improper Address Specified
3154-FTP Improper Port Specified
3200-WWW Phf Attack
3202-WWW .url File Requested
3203-WWW .lnk File Requested
3204-WWW .bat File Requested
3205-HTML File Has .url Link
3206-HTML File Has .lnk Link
3207-HTML File Has .bat Link
3208-WWW campas Attack
3209-WWW Glimpse Server Attack
3210-WWW IIS View Source Attack
3211-WWW IIS Hex View Source Attack
3212-WWW NPH-TEST-CGI Attack
3213-WWW TEST-CGI Attack
3214-IIS DOT DOT VIEW Attack
3215-IIS DOT DOT EXECUTE Attack
3216-WWW Directory Traversal ../..

3217-WWW php View File Attack
3218-WWW SGI Wrap Attack
3219-WWW PHP Buffer Overflow
3220-IIS Long URL Crash Bug
3250-TCP Hijack
3300-NetBIOS OOB Data
3303-Windows Guest Login
3305-Windows Password File Access
3306-Windows Registry Access
3307-Windows Redbutton Attack
3401-Telnet-IFS Match
3500-Rlogin -froot Attack
4001-UDP Port Sweep
4100-Tftp Passwd File
6001-Normal SATAN Probe
6002-Heavy SATAN Probe
6050-DNS HINFO Request
6051-DNS Zone Transfer
6052-DNS Zone Transfer from High Port
6053-DNS Request for All Records
6102-RPC Dump
6150-ypserv Portmap Request
6151-ypbind Portmap Request
6152-yppasswdd Portmap Request
6153-ypupdated Portmap Request
6154-ypxfrd Portmap Request
6155-mountd Portmap Request

6175-rexd Portmap Request
6200-Ident Buffer Overflow
6201-Ident Newline
6250-FTP Authorization Failure
6251-Telnet Authorization Failure
6252-Rlogin Authorization Failure
6253-POP3 Authorization Failure
6255-SMB Authorization Failure
6300-Loki ICMP Tunneling
6302-General Loki ICMP Tunneling
8000:2101-FTP Retrieve Password File
8000:2302-Telnet-/etc/shadow Match
8000:2303-Telnet-+ +
8000:51301-Rlogin-IFS Match
8000:51302-Rlogin-/etc/shadow Match
8000:51303-Rlogin-+ +
10000:1000-IP-Spoof Interface 1
10000:1001-IP-Spoof Interface 2

■ **Release version 1.0**

1100-IP Fragment Attack
1101-Unknown IP Protocol
2000-ICMP Echo Reply
2001-ICMP Host Unreachable
2002-ICMP Source Quench
2003-ICMP Redirect
2004-ICMP Echo Request
2007-ICMP Timestamp Request

2008-ICMP Timestamp Reply
2011-ICMP Address Mask Request
2012-ICMP Address Mask Reply
2151-Large ICMP Traffic
2152-ICMP Flood
2154-Ping of Death Attack
3045-Queso Sweep
3050-Half-open SYN Attack
3160-Cesar FTP Buffer Overflow
3450-Finger Bomb
3602-Cisco IOS Identity
050-UDP Bomb
4600-IOS UDP Bomb
5290-Apache Tomcat DefaultServlet File Disclosure
5315-changedisplay.pl WWWthreads Privilege Elevation
5329-Apache/mod_ssl Worm Probe
5332-Wordtrans-web Command Exec
5381-VPASP SQL injection
6100-RPC Port Registration
6101-RPC Port Unregistration
6103-Proxied RPC Request
11013-Mutella File Request
11202-AOL / ICQ Activity
11203- IRC Channel Join

The following signatures are not associated with any particular release.

- 1105-Broadcast Source Address
- 1106-Multicast Ip Source Address

Numbers

10.1.9.201 default IP address, 84
1000 series signatures, 514–516
10000 series signatures, 595
2000 series signatures, 516–518
2900/3500 series switches
 vs. 4000/6000 switches, 393
 configuring for SPAN, 389–393
3000 series signatures, 518–540
4000 series signatures, 540–545
4000/6000 series switches
 vs. 2900/3500 switches, 393
 configuring for SPAN, 393–395
4210 IDS sensor, 45
 modifying BIOS for, 203
4215 IDS sensor, 45
 illustrated in case study, 61
4220 IDS sensor, modifying BIOS for, 204
4230 IDS sensor, 45
 Cisco IDS certification test and, 77
 modifying BIOS for, 204
4235 IDS sensor, 46
 illustrated in case study, 67
4250 IDS sensor, 46
4250 XL IDS sensor, 46
 illustrated in case study, 64
5000 series signatures, 546–582
6000 series signatures, 582–588
6500 series switch, module for, 47–49
7000 series signatures, 588
8000 series signatures, 589
900 series status alarms, 335–337
9000 series signatures, 590–594

A

access attacks, 23
access control, 17
Access-Control-List signatures, 277
Access Control Lists (ACLs), 351–357

blocking performed via, 360–366
IDS sensors and, 86
signatures for, 277
 where to apply, 365
Access signatures, 275
ACLs. *See* Access Control Lists
actions (performed by signatures), 326–334
activating signatures, 333
Active Defense detection, 40
active partition, 247
Additional Destinations (PostOffice Protocol), 147
Administration tab (IDM), 175–178
Advanced Filtering subtab (CSPM), 144
Advanced tab (CSPM), 146–148
Alarm level 3, 335
Alarm level 4, 335
Alarm level 5, 334
alarm notification type, configuring for IOS-IDS, 486
alarms, 334–337
 configuring, 457
 displayed
 in CSID Director for Unix, 159
 in CSPM Event Viewer, 152–155
 tuning, 458–460
 importance of, 334
all keyword, 398
anomaly-based IDS, 26, 31
antispoofing, 349
 IDS implementation and, 362
Application Intelligence Layer (AVVID architecture), 5
application partition, 247
apply command, 258
Architecture for Voice, Video, and Integrated Data. *See* AVVID Architecture
ARP signature series 7000 (list), 588
ASN (Autonomous System Number), 24
ATOMIC micro-engines, 281–286

atomic signatures, 275, 484
 attack (default password), 83, 100
 IDS sensors and, 239
 attacks, 22–25
 example of how they occur, 25
 phases of activity in, 430
 Audit Log Report, 466
 audit reports, 464–471
 sample report and, 468
 audit rules, creating/applying for IOS-IDS, 491
 authentication, 18
 authorization roles, for
 CiscoWorks2000, 443
 automatic updates, configuring in IDS Device Manager, 176–178
 Autonomous System Number (ASN), 24
 autostart utility, for CSPM, 124
 AVVID architecture, 3–6
 benefits of, 5

B

back door signature series 9000 (list), 590–594
 Back Orifice program, 541
 baselining networks, 272
 BGP (Border Gateway Protocol), 24
 BIOS
 modifying for older sensors, 203
 upgrading for IDS sensors, 108
 blocked addresses, determining status of, 373–375
 blocking, 120, 272, 347–381
 via ACLs, 360–366
 caution with, 139
 configuring
 via IDS Device Manager, 171
 sensors for, 366–373
 duration of, 351, 357
 IDS implementation and, 362
 importance of planning for, 349

 manual, 372
 Never Block Addresses option and, 368
 removing, 372
 vs. shunning, 351
 Blocking Devices tab (CSPM), 368–370
 blocking duration, 351, 357
 IDS implementation and, 362
 blocking forwarding sensor, 358
 Blocking tab (CSPM), 139–142, 368, 371
 bootstrapping IDS sensors. *See* IDS sensors, initializing
 Border Gateway Protocol (BGP), 24
 both command, 244
 bridging loops, 398, 399
 vs. RSPAN, 401
 broadcast frames, 386
 Building Distribution Module (SAFE blueprint), 9
 Building Module (SAFE Enterprise Campus Area), 8

C

CA certificates, 162
 cables, 79, 80
 Campus Area (SAFE blueprint), 7–10
 capturing traffic. *See* traffic, capturing
 case sensitivity
 IDS Director/sensor host names and, 241
 IP host names and, 86
 organization names and, 88
 passwords and, 83
 case studies of IDS deployment, 60–68
 CatOS switches. *See* SET-based switches
 CD for recovery/upgrade, using to reinitialize IDS sensors, 102
 central management of IDS sensors, 120–122
 Cerberus FTP Server, 247
 Certificate Authority (CA) certificates, 162
 certificates. *See* security certificates

- certification test for IDS, 3
 - 4230 IDS sensor and, 77
 - IDS/SM sensors and, 237, 240
 - IOS-IDS signatures and, 484
 - sensor status alarms and, 335–337
- cidServer command, 95
- cipher systems, physical security and, 18
- Cisco 4200 series IDS sensors, 43–46
- Cisco Call Manager, 5
- Cisco Catalyst 6000 IDS Module, 387
- Cisco Host Sensor software, 50
- Cisco IDS Active Update Notification, 474
- Cisco IDS certification test. *See* certification test for IDS
- Cisco IDS MC. *See* IDS Management Center
- Cisco IDS module for Cisco routers, 46
- Cisco IDS software
 - IDS Device Manager and, 161
 - updates for, 474
 - verifying version of via idsvers command, 97
 - versions 3.0 and 3.1, 190–192
 - updating, 216–218
 - versions 4.0 and later, 192–197, 205
 - updating, 218–222
- Cisco Intrusion Detection System Device Manager (IDM). *See* IDS Device Manager
- Cisco IOS IDS. *See* IOS-IDS
- Cisco Monitoring Center for Security. *See* Security Monitor.
- Cisco Network Security Database (NSDB), 121, 178
- Cisco PostOffice Protocol. *See* PostOffice Protocol
- Cisco Secure Intrusion Detection (CSID) Director for Unix. *See* CSID Director for Unix
- Cisco Secure Intrusion Detection Systems Exam (CSIDS 9E0-100), 3
- Cisco Secure Policy Manager (CSPM), 51, 122, 123–155
 - for accessing IDS sensors, 79
 - configuring, 129–155
 - hosts, defining via, 132–135
 - IDS sensors, defining via, 135–137
 - installing, 123–128
 - logging on to, 128
 - managed devices/blocked addresses, displaying status via, 374
 - master blocking sensors, using via, 371
 - networks, defining via, 130–132
 - sensors, configuring via, 368
 - signature files, displaying via, 273
 - signatures
 - custom, creating via, 326
 - excluding/including via, 321
 - software requirements for, 123
- Cisco Security Wheel, 15–20, 433
- ciscoids default user ID, 239
- CiscoWorks VPN/Security Management Solution. *See* VPN/Security Management Solution
- CiscoWorks2000, 431, 435–446
 - adding users to, 445
 - authorization roles for, 443
 - verifying installation for, 444
- clear config command, 243, 264
- CLI. *See* command line interface
- Client Layer (AVVID architecture), 4
- Code Red worm, 2
- COM ports, initializing IDS sensors and, 79
- Command and Control Network, 77
- command line interface (CLI), 51
 - initializing IDS sensors and, 76
- Command tab (CSPM), 148
- commit security acl command, 245, 409
- Communications Infrastructure settings, 87
- composite signatures, 275
- computer crime, 2
- Configuration mode (IDS/SM sensors), 239

- Configuration tab (IDM), 168–172
 - configuring
 - alarms, 457
 - blocking, 366–373
 - CSID Director for Unix, 157–160
 - CSPM, 129–155
 - IDS Device Manager, 161–178
 - IDS sensors. *See* IDS sensors, initializing
 - IDS sensors, 236–247
 - verifying the configuration, 246
 - internal networks, 319
 - IOS-based switches, for SPAN, 388–395
 - IOS-IDS, 485–492
 - verifying/testing, 498–507
 - IP fragment reassembly, 317
 - logging, 208–216
 - Remote Access, 201–204
 - routers, for Telnet sessions, 366
 - sensing properties, 320
 - signatures, 455–457
 - for IOS-IDS, 492–495
 - signatures listed by software version, 598–629
 - SPAN, 98–100, 244
 - SSH, 185–201
 - via IDS Device Manager, 198–200
 - switches
 - for SPAN, 387, 388–401
 - for RSPAN, 403–406
 - TCP session reassembly, 316
 - VACLs, 406–411
 - Connection signatures, 276
 - Console Agent (Host IDS), 50
 - Console Notification Report, 465
 - console port, using, 80
 - content-based signatures, 274
 - control port, placement of, 77
 - Control tab (CSPM), 149
 - Core Module (SAFE blueprint), 9, 66
 - Corporate Internet Module (SAFE blueprint), 12, 13
 - CPU Monitor tool (SolarWinds.Net), 484
 - crackers, vs. hackers, 361
 - create keyword, 397
 - critical hosts, IDS implementation
 - considerations and, 364
 - critical services, identifying, 58
 - Cross Protocol signature series 6000 (list), 582–588
 - CSI/FBI Computer Crime and Security Survey, 2
 - CSID Director for Unix, 122, 155–160
 - for accessing IDS sensors, 79
 - CSIDS 9E0-100 (Cisco Secure Intrusion Detection Systems Exam), 3
 - CSPM Event Viewer, displaying alarms in, 152–155
 - CSPM. *See* Cisco Secure Policy Manager
 - custom signatures, creating, 323–326
- ## D
- data retrieval access attacks, 23
 - database key, exporting, 127
 - database rules, for IDS Management Center server, 471–473
 - datagrams, guidelines for, 318
 - date and time, setting, 90
 - DDos attacks, 24
 - debug commands, 504–507
 - caution with, 507
 - newer commands and, 490
 - debug ip audit detailed command, caution with, 497
 - Deception Toolkit, 28
 - defaults
 - ciscoids user ID, 239
 - IP address, 84
 - IP host name, 85
 - password, 83, 100
 - IDS sensors and, 239
 - route for network, 86
 - sensor configuration, restoring, 226

- signature actions, setting for IOS-IDS, 491
- signature file (CSPM), 273
- denial of service attacks, 24
- Denial of Service signatures, 275
- deny all statement, 354
- destination switches, configuring
 - for IOS-based switches, 403
 - for SET-based switches, 405
- device management, 357
- Device tab (IDM), 165–168
- diag resetcount command, 264
- Diagnostic mode (IDSM sensors), 239
- disabling signatures, for IOS-IDS, 492
- Distributed Denial of Service attacks, 24
- Distribution Modules (SAFE blueprint), 9
- DoS attacks, 24
- downloads
 - IDS sensor image, 102
 - signature file updates, 150
- dropped packets, 311, 335–337
- dsniff tool, 285
- Dynamic ACLs, 360

E

- E-Commerce and VPN/RAS
 - module, 64–66
- E-Commerce Module (SAFE blueprint), 12
- Edge Distribution Module (SAFE blueprint), 9
- egress points, 56, 59
- electronic terrorism, 2
- e-mail messages, spam signature and, 494
- enable config command, IDSM
 - troubleshooting and, 263
- enable password, 367
- enable-privileged passwords, device
 - management and, 358
- encrypted traffic, 384, 419–422
 - defining configuring settings for, 91–93
- encryption, 18, 384
 - wireless access points and, 59

- engine-specific parameters (signature parameters), 278
- Enterprise Campus Area (SAFE blueprint), 8
- Enterprise Network Edge Area (SAFE blueprint), 12
- enterprise-wide IDS management, 429–479
- entry points, 56, 59
 - IDS implementation considerations and, 361
 - multiple
 - blocking at, 350
 - master blocking for, 358–360
- ettercap tool, 285
- event logging, configuring, 208
- event logs
 - displaying
 - in IDM browser, 209
 - via show eventfile current command, 264
 - exporting, 209–211
- event viewers
 - CSPM, 152–155
 - IDS, 51, 160, 173–175
- excluding signatures, for IOS-IDS, 493
- Exec mode (IDSM sensors), 239
- exploitation (phase of attack), 430
- Extended Access Control Lists, 354–357, 360
 - vs. Standard ACLs, 355
- Extensible Markup Language (XML),
 - RDEP protocol and, 55
- external networks (OUT), 319
- external threats, 22
- Extranet Module (SAFE blueprint), 13

F

- file integrity, 27
- filtering signatures, 142–145
- Filtering tab (CSPM), 142–145
- fingerprinting, 42

Firewall IOS-IDS. *See* IOS-IDS
 FLOOD micro-engines, 289–293
 Frame/ATM Module (SAFE blueprint), 14
 frames, 386

G

General signatures, 276
 Getting Started window (CSPM), 129
 Global engine parameters (signature parameters), 278–280

H

hackers, vs. crackers, 361
 Hidden attribute (signature parameters), 278
 HIDS. *See* Host IDS
 high severity level, 334
 Honeyd software, 28
 Honeynets software, 28
 honeypots, 28
 Host IDS, 3, 26, 27
 Host IDS sensors, 49–51
 Host Sensor Console software, 49
 host-based intrusion detection systems. *See* Host IDS
 hosts
 critical, IDS implementation considerations and, 364
 defining via CSPM, 132–135
 HP OpenView, 122
 HTTPS Web servers, 420
 hubs, 384
 vs. switches, 385
 Hybrid IDS, 28

I

ICMP signature series 2000 (list), 516–518
 identity cards, physical security and, 18
 IDM. *See* IDS Device Manager
 IDS. *See* entries at intrusion detection
 IDS blocking. *See* blocking

ids commands, 95–98
 IDS configuration
 clear config command for, 264
 troubleshooting, 503–506
 IDS Device Manager (IDM), 51, 122, 160–178
 accessing, 162
 Communications Infrastructure settings and, 87
 IDS sensors accessed via, 79
 Secure Shell, configuring via, 198–200
 signature files, displaying via, 273
 signatures
 custom, creating via, 324–326
 excluding/including via, 322
 IDS Director for Unix. *See* CSID Director for Unix
 IDS Event Viewer (IEV), 51, 160
 downloading, 173–175
 ids-installer command, 247, 250–253
 IDS Management Center, 431–447
 accessing, 442
 administering server for, 471–473
 defining email server settings for, 474
 IDS sensor hierarchy and, 448
 installing, 435–447
 installation steps for, 441
 verifying the installation, 444
 reports for, 464–471
 sample report and, 468
 IDS MC. *See* IDS Management Center
 IDS router module, 46
 IDS Sensor software. *See* Cisco IDS software
 IDS sensors, 40–68
 advanced capturing methods and, 415–419
 BIOS for
 modifying for older sensors, 203
 upgrading, 108
 Cisco devices managed by, 348
 configuration files for, 460–464
 updating, 148

- configuring for blocking, 366–373
- CSPM software versions and, 130
- default configuration for, restoring, 226
- defining
 - via CSID Director for Unix, 157–159
 - via CSPM, 135–137
- deploying, 56–68
- groups/subgroups and, 447–454
- identifying, 76–78
- IDS Management Center and, 432
- vs. IDSM sensors, 236
- image for, downloading, 102
- implementing, general considerations for, 361–365
- initializing (configuring), 75–118, 185–232
 - applying the configuration, 204–207
 - via CSPM, 137–151
 - via IDS Device Manager, 164–178
 - sysconfig-sensor command for, 83
 - writing to files, 89
- IP address for, changing, 159
- log files for, configuring
 - in CSPM, 145
 - in IDS Device Manager, 171
- managing, 51–56, 119–184
 - enterprise-wide, 429–479
 - ids commands for, 95–98
- password for, recovering, 100–102
- placement of, 59–68
- protection and, 447
- reinitializing, 102–107
- reports for, 464–471
 - sample report and, 468
- status of, checking via Poll button, 149
- updating, 216–226
- upgrading from 1.3 to 4.0, 107–112
- virtual, 290
 - ways to access, 79
- IDS Services Module (IDSM), 47–49, 68
- IDS signatures. *See* signatures
- idsconns command, 96
- IDSM filename structure, 262
- IDSM sensors, 233–269
 - architecture of, 235
 - booting from maintenance partition, 247–250
 - configuring, 236–247
 - verifying the configuration, 246
 - diagram of, 260
 - filename structure and, 262
 - vs. IDS sensors, 236
 - requirements for, 236
 - shutting down, 256–258
 - troubleshooting, 259–264
 - updating, 247–259
 - verifying success of, 254–256
- IDSM. *See* IDS Services Module
- IDSM-1 and IDSM-2, compared, 47–49
- idsstart command, 98
- idsstatus command, 95
- idsstop command, 97
- idsupdate command, 223
- idsvers command, 97
- IEV. *See* IDS Event Viewer
- image files, 103
- image. *See* software image
- informational severity level, 334, 335
- Informational signatures, 275
- ingress points. *See* entry points
- inpkts enable option, caution with, 398
- interface groups, 207
- internal networks (IN), defining/ configuring, 319
- internal threats, 22
- Internet ingress/egress points, 56, 59
- Internet Service Provider Area (SAFE blueprint), 13
- intrusion detection, 41
 - Active Defense and, 40
 - implementing, general considerations for, 361–365

- Intrusion Detection System Device Manager (IDM). *See* IDS Device Manager
 - intrusion detection systems (IDS), 25–38
 - defeating, attempts at, 32
 - how they work, 28–31
 - implementing, general considerations for, 361–365
 - IOS-based switches, configuring
 - for RSPAN, 403
 - for SPAN, 388–395
 - for VACLs, 410
 - IOS-IDS, 481–511
 - commands, changes in, 490
 - configuring, 485–492
 - six steps in, 485
 - verifying/testing the configuration, 498–507
 - limitations of, 482
 - responses to intrusions, 495–498
 - signatures, configuring for, 492–495
 - IOS routers, CSPM and, 123
 - IP addresses
 - best practices and, 87
 - changing for IDS sensors, 159
 - configuring for IDS sensors, 84
 - ip audit name command, 491
 - ip audit notify log command, 486
 - ip audit po local command, 487–490
 - ip audit po protected command, 489, 490
 - ip audit protected command, 490
 - ip audit signature command, 493, 495
 - ip audit smtp spam command, 494
 - IP blocking. *See* blocking
 - IP fragment reassembly, 317
 - IP host name, configuring for IDS sensors, 85
 - IP logging, configuring, 211–214
 - IP logs, generating, 214–216
 - IP-named ACLs, 360
 - IP Netmask, configuring for IDS sensors, 84
 - IP phones, 5
 - IP Security (IPSec), 151
 - caution with, 93
 - configuring secure communications and, 91
 - IP signatures series 1000 (list), 514–516
 - IP Telephony module (SAFE), 7
 - iplog interface group number command, 215
 - IPSec tunnels, configuring, 151
 - IPSec. *See* IP Security
 - IPv6, 422
 - ISP Module (SAFE blueprint), 14
- ## L
- learning disable option, 398
 - least privilege, 167
 - Lock and Key ACLs, 360
 - log file checkers, 27
 - log files, configuring
 - in CSPM, 145
 - in IDS Device Manager, 171
 - log notification type, configuring for IOS-IDS, 486
 - logging host command, 486
 - logging on
 - to CSPM, 128
 - to IDS Device Manager, 162–164
 - Logging tab (CSPM), 145
 - logging, configuring, 208–216
 - login password, for Telnet access, 366
 - loose reassembly option (TCP session reassembly), 316
 - low severity level, 335
- ## M
- MAC (Multicast Media Access Control), 112
 - maintenance partition, 247
 - booting IDSM sensor from, 247–250

- managed devices, 348
 - determining status of, 373–375
- Management Module (SAFE blueprint), 9
- managing IDS sensors, 51–56, 119–184
- Mantrap software (Symantec), 28
- master blocking, 358–360
- Master Blocking sensor, 141, 142
- Master Blocking Sensor subtab (CSPM), 371
- master blocking sensors, 358, 371
- Master engine parameters (signature parameters), 278–280
- Medium Campus Module (SAFE blueprint), 8
- Medium Network Edge (SAFE blueprint), 12
- medium severity level, 335
- micro-engines, 277–314
 - for 4200 series sensors (list), 280
- Minimum Event Level menu (CSPM), 142
- mls ip ids command, 411
- module enable command, 243
- monitor session command, 393–395
- monitoring interfaces
 - enabling/disabling, 205–207
 - grouping, 207
- monitoring port. *See* sniffing port
- Monitoring tab (IDM), 172–175
- monitoring techniques, 19
- MRTG tool, using for Cisco router, 484
- MSFC (Multilayer Switch Feature Card), 236, 407
- Multicast Media Access Control (MAC), 112
- multicast traffic, monitoring, 112
- Multilayer Switch Feature Card (MSFC), 236, 407
- multiport taps, 413

N

- NetRanger Configuration File Management Utility, 157
- netrangr account, 79, 83
 - administering sensors and, 94
 - vs. root account, 82
- Network and Host IDS software (Cisco), 41
- network attacks, 22–25
 - example of how they occur, 25
- network-based intrusion detection systems (NIDS). *See* Network IDS
- Network Campus Area (SAFE blueprint), 7–10
- Network Edge Area (SAFE blueprint), 10–13
- Network IDS, 3, 26
 - blocking and, 349
 - how it works, 29
- Network IDS sensors, 42–49
- Network Infrastructure Layer (AVVID architecture), 4
- network taps, 387, 411–415
 - vs. SPAN ports (table), 414
- network traffic, capturing. *See* traffic, capturing
- networks
 - analyzing/understanding, 57–59
 - baselining, 272
 - default route for, 86
 - defining via CSPM, 130–132
 - devices on, managing, 357
 - protected, configuring for IOS-IDS, 489
- Never Block Addresses option, 368
- NIDS. *See* Network IDS
- Nimda worm, 2
- no reassembly option (TCP session reassembly), 316
- none severity level, 334, 335

notification queue, changing size of for IOS-IDS, 490

notification type, configuring for IOS-IDS, 486

nr command, 94

nrConfigure utility, 157, 334

nr-director notification type, configuring for IOS-IDS, 486

nrstatus command, 156

NSDB. *See* Cisco Network Security Database

null-modem cable, 79, 80

O

OpenView (Hewlett-Packard), 122

OTHER engine, 311–314

OUT (external networks), 319

oversubscription of traffic, 337

P

Packet Capture device (interface), 138

packets, dropped, 311, 335–337

parameters

- sensing, 315–321
- signature, 277–280

passwords

- attack (default password), 83, 100
- IDS sensors and, 239
- enable, 367
- enable-privileged, device management and, 358
- IDS Device Manager and, 164
- login, for Telnet access, 366
- recovering, 100–102
- for root and netrangr accounts
- setting/changing, 83, 90

patches, importance of, 18

permit any statement, 494

PFC (Policy Feature Card), 236, 406

ping command, verifying service pack update and, 257

PIX firewalls, CSPM and, 123

Policy Feature Card (PFC), 236, 406

policy violation signature series 10000 (list), 595

Poll button (CSPM), 149

POP3 protocol, vs. Cisco PostOffice Protocol, 53

port mirroring. *See* Switched Port Analyzer (SPAN)

port monitor command, 389–392

Port-based SPAN (PSPAN), 395

Post Office Protocol POP3, vs. Cisco PostOffice Protocol, 53

PostOffice Protocol (Cisco), 53–55

- configuring on CSPM host, 133–135, 146–148
- local/remote parameters, configuring for IOS-IDS, 487–489

postshun/preshun ACLs, 361

prevention, Active Defense and, 40

Prince Partners Inc., case study illustrating IOS-IDS, 486

privilege escalation attacks, 24

probing (phase of attack), 430

Properties tab (CSPM), 137

Protected attribute (signature parameters), 277

protected networks, configuring for IOS-IDS, 489

PSPAN (Port-based SPAN), 395

PSTN Module (SAFE blueprint), 14

R

RDEP. *See* Remote Data Exchange Protocol

re-imaging IDS sensors, 102–107

reaction, Active Defense and, 40

rebooting, configuring IDS sensors and, 94

receive sessions (Rx), 244

reconnaissance (phase of attack), 430, 22

Reconnaissance signatures, 275

recovering IDSM sensors. *See* IDSM sensors, updating

recovery partitions, 103–107

recovery/upgrade CD, 102

Remote Access Networks, 56, 59

Remote Access, configuring, 201–204

Remote Data Exchange Protocol (RDEP), 53, 55

Remote SPAN (RSPAN)

- configuring, switches for, 401–406
- using with VACL, 410

remote-span command, 403

Remote User Network Edge Module (SAFE blueprint), 10

remove command, 264

reports, 464–471

- sample report and, 468

Required attribute (signature parameters), 278

reset command, 248–250

- IDSM troubleshooting and, 260

resources for further information

- AVVID architecture, 6
- IPSec tunnels, 151
- SAFE blueprint, 15
- security policies, 16, 435
- wildcard masks, 352

response techniques, 19

RFC 2827 filtering, 9

root account, 79

- vs. netranr account, 82
- tasks performed by, 81

route for network, default for, 86

routers

- Cisco IDS module for, 46
- configuring for Telnet sessions, 366
- IOS-IDS and, 483, 485

run Diagnostics command, 108

Rx (receive sessions), 244

S

SAFE axioms, 14, 58

SAFE blueprint, 3, 6–20

- benefits of, 6

Secure Blueprint for Enterprise Networks. *See* SAFE Blueprint

secure communications, configuring, 91–93

Secure Policy Manager. *See* Cisco Secure Policy Manager (CSPM)

Secure Shell (SSH), 420

- configuring, 185–201
- via IDS Device Manager, 198–200
- secure communications and, 92

security

- Cisco Security Wheel for, 15–20
- excluding/including signatures and, 321
- physical, importance of, 18

security certificates, verifying, 162

security models, 3

Security Monitor, 431

Security Monitor reports, 470

security patch updates, importance of, 18

security policies

- as part of Cisco Security Wheel, 16
- IDS Management Center and, 433–435

sensing interfaces, enabling/
disabling, 205–207

sensing parameters, 315–321

sensing properties, 320

Sensing tab (CSPM), 138, 320

sensor (default host name), 85

Sensor Configuration Deployment Report, 465

Sensor Configuration Import Report, 465

sensor groups, 447–454

- adding/deleting sensors and, 450–454

sensor status alarms, 335–337, 596

Sensor Version Import Report, 465

sensors. *See* IDS sensors; IDSM sensors

- server farm ingress/egress points, 56, 59
- Server Module (SAFE blueprint), 9
- SERVICE micro-engines, 286–289
- service packs, 474
 - remove command for, 264
 - updating on IDSM sensors, 258
- Services Control Layer (AVVID architecture), 4
- session command, 237, 239, 250
- session slicing, 32
- SET-based switches, configuring
 - for RSPAN, 404–406
 - for SPAN, 395–401
 - for VACLs, 408
- set boot device command, 247
- set command, 244, 395
- set module power command, 243, 257
 - IDSM troubleshooting and, 261
- set power up command, 238
- set rspan command, 404–406
- set security acl command, 245, 408
- set span command, 395–401
- set trunk command, 404
- set vlan command, 243, 404
- set vtp domain command, 404
- setup command, 239
 - verifying successful IDSM upgrade and, 254
- severity levels, 326–334
- show config command, 246, 247
 - IDSM troubleshooting and, 263
 - verifying service pack update and, 259
- show eventfile current command, 247
 - IDSM troubleshooting and, 264
- show interface command, 206
- show ip audit configuration command, 490, 495, 500–503
- show ip audit debug command, 503
- show ip audit interfaces command, 499
- show ip audit session command, 503
- show ip audit statistics command, 497, 503
- show module command, 237, 238, 256
 - IDSM troubleshooting and, 260
- show port monitor command, 392
- show run command, 353
- show running command, 392
- show running-config command, 367
- show security acl command, 246
 - IDSM troubleshooting and, 263
- show span command, 246
 - IDSM troubleshooting and, 263
- show vlan command, 261, 404
- shun command, 348
- shunning. *See* blocking
- Shutdown button (IDSM sensor), 260, 261
- shutdown command, 256
 - ensuring clean shutdowns and, 260
- signature actions, setting defaults for IOS-IDS, 491
- signature-based IDS, 26, 30
- signature classes, 275
- signature files
 - displaying, 273
 - Snort archive of, 323
- signature parameters, 277–280
 - configurable, available online, 328
 - tuning, 328
- signature types, 276
- Signature update files, 474
- signature wizard, 326–334
- signatures, 271–346
 - activating, 333
 - alarms and, 334–337
 - blocking and, 351
 - configuration files for, 460–464
 - configuring, 455–457
 - for IOS-IDS, 492–495
 - context-based, 274
 - custom, creating, 323–326
 - enabling/disabling via IDS Device Manager, 168–170
 - excluding/including, 321–323

- filtering, 142–145
- IDS implementation considerations
 - and, 363
- IDS Management Center and, 433
- on IOS-IDS, 484
- list of, 513–629
 - by series, 314, 514–598
 - by software release version, 598–629
- micro-engines for, 277–314
- remove command for, 264
- updating, 150, 222–226, 474
 - on IDSM sensors, 258
- SigWizMenu, 326–334
- Simple Filtering subtab (CSPM), 143
- Slammer Worm, 484
- Small Campus Module (SAFE blueprint), 8
- Small Network Corporate Internet Module (SAFE blueprint), 11
- Small Network Edge Area (SAFE blueprint), 11
- sniffing port, 77
- snoop command, 81, 384
- Snort signature file archive, 323
- software
 - Cisco IDS, 97, 190–197
 - for honeypots, 28
 - importance of upgrading, 18
- software image
 - downloading, 102
 - replacing, 247–253
 - uninstalling, 107
- Solaris snoop command, 384
- SolarWinds.Net, 484
- source switches, configuring
 - for IOS-based switches, 403
 - for SET-based switches, 404
- spam signature, 494
- SPAN ports, vs. network taps (table), 414
- SPAN. *See* Switched Port Analyzer
- spanning, bridging loops and, 398, 399
- spanning ports, 99, 387
- Spanning Tree Protocol (STP), 390
- Specter software, 28
- SQL Slammer Worm, 484
- SSH. *See* Secure Shell
- Standard Access Control Lists, 351–354, 360
- Standard Agent (Host IDS), 49
- STATE micro-engine, 293–296
- status alarms, 335–337
- Status LED (IDSM sensor), 260
- STP (Spanning Tree Protocol), 390
- strict reassembly option (TCP session reassembly), 316
- string-matching signature series 8000 (list), 589
- STRING micro-engine, 296–302
- structured threats, 21
- su command, 420
- subnet mask, configuring for IDS sensors, 84
- Subsystem Report, 465
- SWEEP micro-engines, 302–311
- switch backplane, capturing traffic directly from, 387
- Switched Port Analyzer (SPAN), 27
 - configuring, 98–100, 244
 - switches for, 387, 388–401
 - TCP resets and, 121
- switches
 - configuring
 - for RSPAN, 401–406
 - for SPAN, 387, 388–401
 - vs. hubs, 385
- switching, 384, 385–388
- switchport capture command, 411
- Symantec Mantrap software, 28
- sysconfig-sensor command, 76, 83
 - IDS Device Manager and, 161
- system access attacks, 24

T

taps. *See* network taps
 TCP3WayHandshake, 337
 TCP Embryonic Timeout, 317
 TCP Open Establish Timeout, 317
 TCPReassemblyMode, 337
 TCP resets, 26, 120, 121
 TCP session reassembly, 315–317
 TCP signature series 3000 (list), 518–540
 Terminal mode, 93, 94
 terminal server, setting up, 202
 terrorism against computer systems, 2
 testing, as part of Cisco Security Wheel, 19
 threats, 20–22
 time/time zone, setting, 90
 topology map, 130, 132–137
 traffic
 blocking. *See* blocking
 capturing, 383–428
 advanced methods for, 415–419
 directly from switch backplane, 387
 options for resolving problems with, 387
 VACL-based vs. SPAN ports, 408
 oversubscription of, 337
 traffic types, 276
 transmit sessions (Tx), 244
 transparent bridges, 386
 troubleshooting
 IDS configuration, 503–506
 IDS sensors, 259–264
 trunks, managing, 246
 Tx (transmit sessions), 244

U

UDP signature series 4000 (list), 540–545
 Unique parameter (signatures), 302
 Unix Director. *See* CSID Director for Unix
 unstructured threats, 21
 upgrade command, 218, 223

user ID, default for, 239
 utilities
 Back Orifice, 541
 CPU Monitor, 484
 dsniff, 285
 ettercap, 285
 MRTG, 484
 nrConfigure, 157

V

VACL. *See* VLAN Access Control Lists
 verification systems, physical security
 and, 18
 VGA mode, 93, 94
 virtual LANs (VLANs)
 advanced capturing methods and, 415–419
 spanning, 99
 on trunks, managing, 246
 troubleshooting IDS sensors and, 261
 virtual private networks (VPNs), 419
 virtual sensors, 290
 VLAN Access Control Lists (VACLs)
 advanced capturing methods and, 415–419
 configuring, 242, 244–246, 406–411
 using with RSPAN, 410
 VLAN-based SPAN, 99
 VLAN separation, 9
 VLAN SPAN (VSPAN), 395
 VLANs. *See* virtual LANs
 VMS. *See* VPN/Security Management Solution
 VNP Client, configuring IPsec tunnels
 and, 151
 VPN/Remote Access Module (SAFE
 blueprint), 13
 VPN/Security Management Solution
 (VMS), 51–53
 component compatibility and, 439
 VPN software, CSPM and, 123
 VPNs (virtual private networks), 419
 VSPAN (VLAN SPAN), 395

- vulnerabilities, IDS implementation
 - considerations and, 363
- vulnerability patching, 18

W

- WAN Edge Module (SAFE blueprint), 12
- WAN Module (SAFE blueprint), 13
- Watchdog Properties (PostOffice Protocol), 147
- Web Edition Agent (Host IDS), 49, 51
- web sites
 - AVVID architecture, 6
 - Cerberus FTP Server, 247
 - Cisco
 - configurable signature parameters, 328
 - Dynamic ACLs/IP-named ACLs, 360
 - IDS sensor image, downloading, 102
 - IPSec tunnels, 151
 - SAFE blueprint, 15

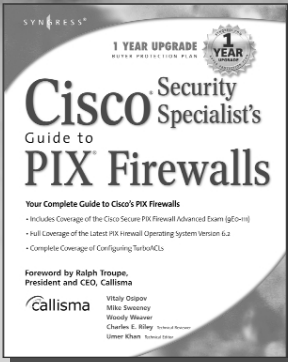
- signature file updates, downloading, 150
 - wildcard masks, 352
- Finisair, 415
- MRTG tool, 484
- Netoptics, 415
- Web/HTTP signature series 5000
 - (list), 546–582
- wildcard masks, 352
- wireless access points, 59
- Wireless LAN module (SAFE), 7
- worms, 2

X

- XML (Extensible Markup Language),
 - RDEP protocol and, 55

Syngress: The Definition of a Serious Security Library™

Syn-gress (sin-gres): *noun, sing.* Freedom from risk or danger; safety. See *security*.



AVAILABLE NOW
order @
www.syngress.com

Cisco Security Specialist's Guide to PIX Firewall

Demystifying the task of implementing, configuring, and administering Cisco's PIX firewall appliances, this book delivers a total solution both for managing these widely used devices and for passing the challenging Cisco Secure PIX Firewall Advanced Exam (9EO-571), a prerequisite for gaining prestigious Cisco Security Specialist 1 certification. Packed with insider tips and techniques on protocols, hardware and software components, troubleshooting and more, this powerful advisor illustrates attack concepts, explains must-know networking principles for optimizing and integrating PIX firewalls, sets forth real-world configuration and administration examples, and helps users master Cisco's infamous command line interface.

ISBN: 1-931836-63-9

Price: \$59.95 USA \$92.95 CAN

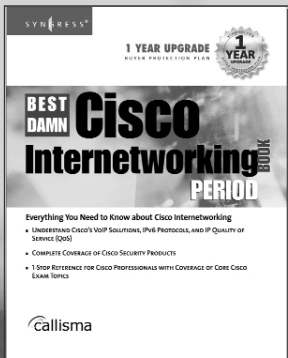
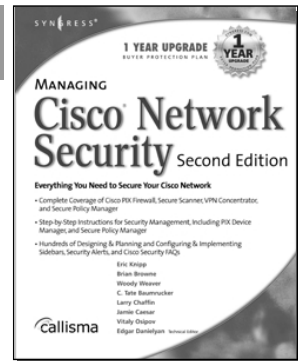
Managing Cisco Network Security, Second Edition

Offers updated and revised information covering many of Cisco's security products that provide protection from threats, detection of network security incidents, measurement of vulnerability and policy compliance, and management of security policy across an extended organization. These are the tools that you have to mount defenses against threats. Chapters also cover the improved functionality and ease of the Cisco Secure Policy Manger software used by thousands of small-to-midsized businesses, and a special section on Cisco wireless solutions.

ISBN: 1-931836-56-6

Price: \$69.95 USA \$108.95 CAN

AVAILABLE NOW
order @
www.syngress.com



AVAILABLE NOW
order @
www.syngress.com

The Best Damn Cisco Internetworking Book Period

Sure to become a dog eared reference for all Cisco engineers and administrators, this book shows readers everything they need to know about all Cisco internetworking topics. The book provides an understanding of Cisco's current VoIP solutions and the means to put them to work, showing how to configure all of Cisco's core VoIP products, among them Cisco CallManager software, Cisco 7910 series phones, and server-based IP PBXs. It discusses IPv6 Protocols, as well as IP Quality of Service (QoS) and how it applies to Enterprise and Internet Service Provider (ISP) environments, and much more!

ISBN: 1-931836-91-4

Price: \$59.95 US \$92.95 CAN