UNIVERSITÀ DEGLI STUDI DI CAMERINO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica

Dipartimento di Matematica e Informatica

REALIZZAZIONE DI UN SISTEMA DI NETWORKING SICURO PER UNA PMI

Tesi di Laurea compilativa
In Reti di Elaboratori

Laureando

Andrea Amatucci

Relatore

Dott. Fausto Marcantoni

ANNO ACCADEMICO 2007 / 2008

“Cento volte al giorno mi viene in mente che la vita interiore ed esteriore dipende dalle fatiche dei contemporanei e da quelle dei loro predecessori; io devo sforzarmi di ricambiare, in ugual misura, ciò che ho ricevuto e ancora ricevo.”

Albert Einstein

Ringraziamenti

Vorrei ringraziare tutti coloro che in questi anni, ed in particolare in questi ultimi difficili mesi, mi sono stati vicini. Per merito loro ho avuto la forza di non mollare e di andare avanti.

Ringrazio il Relatore Dott. Fausto Marcantoni che, nonostante le difficoltà che ho incontrato, mi ha permesso di affrontare questo periodo con il sorriso.

Guida e punto di riferimento eccellente.

INDICE

Introduzione

In ambito aziendale, una corretta gestione di una rete di elaboratori è fondamentale.

I computer, sono sempre più parte integrante della vita delle persone, a casa e a lavoro. Numerosi compiti vengono svolti interamente grazie ad essi. Le imprese, in particolar modo, si trovano a dover organizzare grandi quantità di dati, che spesso si rivelano indispensabili per poter portare avanti l'attività. Tramite una rete di computer, il compito di amministrare le informazioni diventa meno faticoso, e questo ha reso l'utilizzo delle macchine insostituibile.

Un altro strumento fondamentale per l'attività aziendale è, ovviamente, internet. Al giorno d'oggi, molte imprese intraprendono attività unicamente o parzialmente concentrate sulla compravendita online, o comunque utilizzano il web per allacciare contatti con potenziali acquirenti.

L'informazione è un bene che aggiunge valore all'impresa. Dal momento in cui, tali informazioni sono ormai quasi completamente immagazzinate su supporti informatici, le aziende hanno l'obbligo di procurarsi un'adeguata struttura di difesa dei propri dati. Questo è valido in particolar modo in un contesto in cui violazioni alla sicurezza sono in costante aumento.

Le aziende hanno un notevole interesse verso l'adozione di adeguati strumenti di sicurezza, ma non è soltanto questo che le spinge a concentrarsi in tale direzione. In realtà esistono degli obblighi ben precisi a loro carico. Ad esempio, secondo la legge, le imprese devono redigere annualmente il cosiddetto "documento programmatico sulla sicurezza" o DPS. La sua obbligatorietà scaturisce dalla normativa sulla protezione dei dati personali ed esiste per tutte le aziende, liberi professionisti, enti o associazioni che trattano tali dati con strumenti elettronici. Il fine è quello di attestare l'adozione delle procedure previste per il trattamento dei dati personali, tra cui gli strumenti finalizzati al mantenimento di integrità, disponibilità e segretezza degli stessi.

Tutto questo provoca, naturalmente, un bisogno di sicurezza, di proteggere la propria rete interna e i propri dati, dal momento in cui la rete fornisce sia vantaggi che insidie. Dal punto di vista aziendale è molto importante garantire l'integrità e la segretezza delle proprie informazioni, e non è raro che queste spesso vengano violate da malintenzionati utenti della rete.

Da qui nascono gli sforzi per rendere la propria rete sicura, attraverso la realizzazione di sistemi adibiti a tale compito.

La sicurezza informatica è quella branca dell'informatica che si occupa della salvaguardia dei sistemi informatici da potenziali violazioni dei dati.

La protezione dei dati viene impostata su più livelli. In primo luogo si deve negare l'accesso fisico ai sistemi a chi non ne è autorizzato. In realtà, in questo caso si tratterebbe di normale sicurezza e non di sicurezza informatica. Tuttavia non è un aspetto da sottovalutare. Può capitare di concentrarsi sull'adozione di complessi sistemi di protezione trascurando quelli basilari. Il secondo livello provvede a fornire un sistema di autenticazione e permessi. In questo modo vengono create le entità utente che hanno lo scopo di tenere sotto controllo le azioni intraprese dai soggetti che rappresentano. Questo aspetto è fondamentale al fine di monitorare modifiche apportate ai sistemi, sia da parte di intrusi esterni, che da utenza interna alla rete. Infine, il terzo livello di sicurezza riguarda il tracciamento delle attività. Vengono utilizzati file di log al fine di registrare ogni azione compiuta.

Lo scopo di questa tesi è quello di mostrare come è possibile realizzare una propria rete mantenendo un adeguato livello di sicurezza, sia da minacce provenienti dall'esterno che dall'interno, il tutto utilizzando software *open source*.

L'elaborato è stato organizzato in sei capitoli. Nel primo viene illustrato, nella maniera più ampia possibile, il tema della sicurezza e i problemi che ne derivano. Il tutto viene adattato ad un ipotetico ambiente aziendale.

Vengono mostrate le tipiche fasi di un attacco informatico. In questo modo è possibile avere un'idea delle minacce dalle quali è necessario proteggersi. Conseguentemente, viene fatta una panoramica sulle possibili contromisure da prendere al fine di evitare violazioni alla sicurezza.

Per l'argomentazione del tema trattato, sono state effettuate svariate prove in laboratorio. E' stata appositamente realizzata una rete di test, la quale verrà illustrata dettagliatamente in seguito.

Il secondo capitolo costituisce un'introduzione agli strumenti utilizzati per la realizzazione della suddetta rete. Come anticipato, sono stati scelti esclusivamente software open source. A partire dal sistema operativo, vengono mostrati i programmi adottati. Per ognuno di essi viene fatta una breve introduzione, in modo che il lettore possa avere un'immagine complessiva del lavoro stolto.

I capitoli successivi sono dedicati ad un argomento specifico. Il terzo, ad esempio, si riferisce ad uno degli strumenti più importanti per quanto riguarda la sicurezza in rete: il Firewall. Ne vengono illustrati dettagliatamente la struttura e il principio di funzionamento. In maniera particolare, si parla della soluzione open source iptables/netfilter, essendo tra le più valide e flessibili in circolazione.

Il quarto capitolo tratta degli Intrusion Detection System o IDS. Si tratta di strumenti realizzati allo scopo di informare gli amministratori di un'eventuale intrusione o attacco informatico. Esistono diverse soluzioni di questo tipo. Quella trattata con maggior attenzione è Snort.

Il capitolo cinque è dedicato ai Proxy server. Si tratta di apparati molto utili e importanti, i quali garantiscono un'elevata capacità di controllo sulla propria rete. Inoltre, sono utili al fine di migliorare le prestazioni della rete, gestendo in maniera intelligente le richieste multiple alle stesse risorse.

L'ultima parte dell'elaborato tratta un aspetto fondamentale nell'amministrazione di una rete di computer: il monitoraggio. Vengono illustrati strumenti che aiutano a tenere sotto controllo le attività svolte sulle macchine che compongono la rete Lan. Per motivi di diagnostica, i sistemi generano i cosiddetti messaggi di log. Spesso tali messaggi sono immagazzinati in maniera piuttosto confusionale, e gli strumenti mostrati in questo ultimo capitolo hanno lo scopo di organizzare opportunamente le informazioni migliorandone consultazione e leggibilità.

Capitolo 1

La Sicurezza

Panoramica sui problemi di sicurezza

Per garantire un sufficiente livello di sicurezza, è necessario un attento monitoraggio sul traffico proveniente dall'esterno e sulle attività svolte all'interno della rete.

Non è da escludere un comportamento illecito dagli utenti che lavorano sulle macchine che compongono la rete aziendale. Di conseguenza, è necessario porre attenzione al traffico entrante e uscente.

Comunque sia, le minacce di più grave entità restano quelle provenienti dalla rete esterna, internet. Esistono molti modi di condurre attacchi informatici e, quindi, violazioni alla sicurezza aziendale. Comunemente, ad un attacco vero e proprio, precede un *portscan*. Si tratta una tecnica utilizzata per raccogliere informazioni su un sistema connesso ad una rete, consentendo di rilevare eventuali porte in ascolto relativamente ai servizi a cui si riferiscono. Un buon punto da cui iniziare potrebbe, quindi, essere quello di rendere inutile qualunque azione di *portscanning*. Questo bloccherebbe sul nascere diverse tipologie di attacchi informatici. E' necessario fare molta attenzione ai programmi e ai servizi utilizzati, in quanto spesso sono questi a fornire vie di accesso ad eventuali intrusioni non gradite. Un servizio di rete, sicuramente si metterà in ascolto su delle porte ed aprirà delle connessioni con l'esterno. Una eventuale vulnerabilità di tale software, potrebbe essere sfruttata per compiere violazioni alla sicurezza.

Un altro modo per accedere a dati sensibili, senza averne l'autorizzazione, è l'utilizzo della tecnica detta *packet sniffing*. Essa consiste nell'attività di intercettazione dei dati che transitano in una rete telematica. Esistono diversi strumenti che permettono di utilizzare questo meccanismo. L'intercettazione dei pacchetti può essere svolta per scopi leciti, controllo e monitoraggio del traffico per l'individuazione di eventuali problemi di rete, o illeciti, appropriazione di informazioni sensibili quali dati personali e password.

In un ambiente aziendale che fa un largo uso della rete, è necessario prevenire attacchi di tipo DoS (Denial of Service). Tali attacchi mirano a rendere inservibili servizi di rete e connettività. Essi operano in modo tale da saturare un servizio fino a rendere impossibile un suo utilizzo legittimo. Largamente usati in passato da cracker in lotta con istituzioni commerciali e non. Attacchi di questo tipo sono stati compiuti soprattutto verso siti web in particolare. Venivano inoltrate numerosissime richieste alla stessa risorsa, fino a quando il sistema che la ospitava non era più in grado di soddisfarle tutte. In questo modo si negava a richieste legittime di usufruire di tali risorse.

Più avanti verranno descritte in maniera più approfondita le fasi di un tipico attacco informatico.

Proteggere la propria rete

Esistono numerosi modi per farsi strada verso un ambiente protetto.

Lo sfruttamento di falle nel software è una tecnica largamente utilizzata. Per questo *è bene tenere sempre aggiornati tutti i propri sistemi*. Eseguire gli aggiornamenti, rilasciati dalle software house di riferimento, è utile a risolvere svariate vulnerabilità che potrebbero rivelarsi un problema per la sicurezza della propria rete. Non bisogna stupirsi del fatto che ogni settimana escano nuovi bollettini di sicurezza. Lo sviluppo si software complessi come, ad esempio, i sistemi operativi, non è cosa da poco, e non capita raramente che vengano commessi errori, i quali porteranno a bug di varia natura ed entità.

Un'altra grave minaccia è l'esistenza di software appositamente sviluppato al fine di nuocere all'integrità dei sistemi. Tale software è conosciuto con il nome di *virus informatico*. I sistemi affetti da tale problema, sono costretti ad applicare un livello di protezione in più rispetto a quelli esenti. Si tratta di programmi detti *antivirus*. Il loro scopo è quello di

proteggere e liberare il sistema da eventuali infezioni. Non è un aspetto da sottovalutare. Esistono milioni di virus nel mondo. Non molti sono quelli veramente pericolosi, ma risulta necessario procurarsi strumenti che possano garantire un livello di protezione ampio. Attualmente, i sistemi maggiormente soggetti a tale insidie sono le piattaforme Windows, e non è difficile immaginarne il motivo data la base installata mondiale di tale sistema.

Uno dei principali, nonché più utilizzati, strumenti di difesa è il Firewall. Si tratta di un sistema che non può assolutamente mancare nell'ambito della sicurezza di una rete aziendale. Ha la funzione di filtrare tutto il traffico passante e di applicarvi determinate regole prestabilite, bloccando o accettando i pacchetti di rete.

Un altro strumento di difesa è il cosiddetto *Intrusion Detection System* o semplicemente IDS. Si tratta di un meccanismo di analisi del traffico volto a rilevare attacchi informatici di vario tipo. Può essere configurato per lavorare come IPS o Intrusion Prevention System che, a differenza del precedente, oltre a rilevare, blocca gli attacchi.

L'adozione dei suddetti strumenti, è un ottimo inizio per la messa in sicurezza di una rete aziendale. Ovviamente, il loro utilizzo combinato rende l'ambiente ancora più sicuro piuttosto che presi singolarmente.

Come precedentemente detto, però, le minacce non arrivano solamente dal mondo esterno. Esse possono generarsi anche all'interno della rete che si intende proteggere. E' bene, quindi, preoccuparsi anche del comportamento del personale aziendale. Di conseguenza, si rende necessario un monitoraggio delle attività interne ed, eventualmente, una selezione delle attività consentite e non.

A tal fine, è possibile avvalersi di ulteriori strumenti di analisi e filtraggio. Ad esempio il Proxy server. Si tratta, sostanzialmente, di un programma che si interpone tra un

client e un server e che si preoccupa di gestire la comunicazione tra i due. In questo modo è possibile agire in diversi modi sulle richieste. E' possibile selezionare quali sono permesse e quali no. Questo permette, oltre ad evitare le richieste a risorse considerate pericolose, di controllare, moderare e gestire intelligentemente la mole di dati passante in rete.

Tutti gli accorgimenti brevemente descritti sono stati testati al fine di realizzare questa tesi. In sostanza, il lavoro svolto è stato quello di creare una rete vera e propria, con un server e degli host. Successivamente ci si è impegnati per la messa in sicurezza di tale rete, attraverso l'adozione degli strumenti sopra citati. Nel capitolo successivo, ogni componente utilizzato verrà illustrato in maniera dettagliata.

Le fasi di un tipico attacco

In linea generale, le fasi di un attacco informatico sono sempre le stesse.

- *Hiding o Mascheramento*

Questa è la prima fase di un attacco. In realtà precede di molto l'intrusione vera e propria. Un qualunque cracker che voglia violare la sicurezza di un sistema, più o meno rilevante, non desidera certo essere scoperto. La prima cosa che provvederà a fare sarà compiere delle operazioni volte a mascherare la propria attività illecita. Lo scopo è quello di camuffarsi, nascondere la propria reale ubicazione al fine di evitare di essere tracciato e successivamente identificato. Questo è un aspetto da non sottovalutare. Richiede una vasta conoscenza della rete e, a volte, si tratta di operazioni anche molto complesse.

- *Information Gathering o Raccolta di informazioni*

Il secondo passo è quello di raccogliere informazioni sulla vittima. Affinché l'attacco vada a buon fine, il cracker deve possedere quante più informazioni possibili. In questa fase non viene eseguita nessuna reale intrusione. I dati in questione sono soprattutto informazioni di pubblico dominio. Si tratta infatti di raccogliere indirizzi IP, numeri telefonici, indirizzi e-mail e così via. Questo attraverso comuni strumenti come ping, whois, traceroute, DNS, ecc. Le tecniche comunemente utilizzate a tal fine sono le seguenti:

- ***Network Surveying*** – Raccolta di informazioni mediante canali pubblici. Attraverso questa tecnica il cracker entrerebbe in possesso di indirizzi IP, nomi di dominio, informazioni riguardanti l'Internet Service Provider che fornisce i servizi internet alla vittima ecc.
- ***Port Scanning*** – Attività di probing invasivo allo scopo di scoprire i sistemi attivi sulla rete vittima e quali porte sono in ascolto. Ogni sistema ha a 65536 possibili porte TCP, e altrettante UDP. Tramite questa tecnica è possibile definire una mappa approssimativa della rete da attaccare.
- ***Service Identification*** – Una fase molto importante di un attacco. Consiste nel tentare di scoprire quali servizi sono attivi sulle macchine da attaccare. E' ben noto che il software può contenere vulnerabilità a livello di sicurezza e permettere a malintenzionati di approfittarne per compiere violazioni.
- ***System Identification*** – Tecnica di probing attivo di un sistema. Consiste nel ricercare risposte che permettano di identificare il sistema operativo utilizzato sulla macchina bersaglio. Anche questa fase è molto importante. Come già detto, il software può contenere vulnerabilità e, ovviamente, i sistemi operativi non ne sono esenti.

Molte di queste tecniche possono essere eseguite tramite un noto software open source conosciuto con il nome di *nmap*.

- *System Penetration o Intrusione*

Dopo aver raccolto tutte le informazioni necessarie, è possibile procedere con l'intrusione vera e propria. Il tipo di attacco varia a seconda delle caratteristiche della rete e dei sistemi vittima. Possono essere sfruttate diverse tipologie di vulnerabilità, le quali vengono raggruppate nelle seguenti categorie:

- ***System Security*** – Si tratta di vulnerabilità a livello di sistema operativo e software di base. Tipici esempi di attacchi che fanno riferimento a questa categoria sono quelli che sfruttano l'esecuzione di codice arbitrario da remoto, tramite exploit specifici, per ottenere privilegi di amministratore, o anche attacchi di tipo password guessing o brute force per l'ottenimento delle credenziali di accesso al sistema.

- ***Network Security*** – In questa categoria sono compresi attacchi che mirano a compromettere il livello di sicurezza della rete bersaglio. I fattori che concorrono a rendere concreta questo tipo di minaccia sono diversi, ad esempio la presenza di punti di accesso non adeguatamente protetti, la tipologia della rete e dei sistemi di routing e firewalling, in particolar modo la possibilità di effettuare un monitoraggio passivo del traffico di rete tramite, ad esempio, *network sniffing*.
- ***Application Security*** – Riguarda vulnerabilità presenti nel software applicativo. In particolar modo nelle applicazioni che utilizzano la connettività. Questo è un aspetto che una ipotetica azienda di e-commerce deve tenere in seria considerazione. Le tecnologie che l'azienda utilizza per la creazione del proprio servizio online potrebbero essere sfruttate per compromettere la sicurezza del sistema o della rete. Sono esempi le web-application, ossia software solitamente utilizzato per creare pagine web (come java, PHP, Perl ecc.).
- ***Procedural Security*** – Categoria molto interessante. Riguarda lo sfruttamento di vulnerabilità derivanti dalle stesse tecniche di protezione della rete. Ad esempio lo sfruttamento di relazioni di fiducia che vengono instaurate tra due macchine che compongono la rete bersaglio. Attaccare con successo una macchina implicherebbe l'accesso all'altra.

- *Cleaning o Pulizia*

Dopo aver effettuato l'attacco con successo, l'intruso deve provvedere a cancellare le tracce della propria attività. In particolar modo eliminare i messaggi di log che, con l'intrusione, ha generato. In questo modo, rende impossibile all'amministratore della rete, o proprietario del sistema, di rendersi conto dell'avvenuta intrusione.

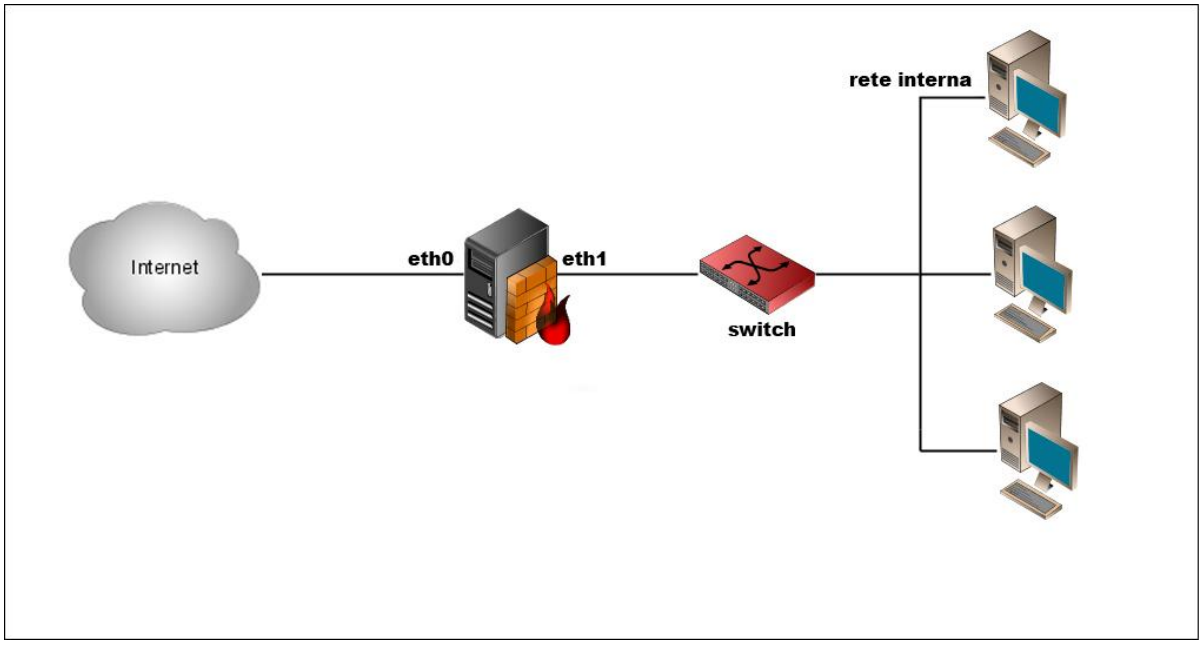
Capitolo 2

Realizzazione della rete

Struttura della rete

Nel seguente capitolo si andrà ad illustrare come è stata realizzata la rete di test. Come sistema di prova è stato preso un semplice personal computer. L'unico requisito richiesto era la presenza di due schede di rete, necessarie per la connessione contemporanea alla rete esterna e interna. L'idea era quella di porre tale macchina tra la rete internet e gli host della rete interna, in modo da poter canalizzare tutto il traffico attraverso di essa. Così facendo ci si riservava la possibilità di analizzare ed eventualmente filtrare i pacchetti transitanti.

Nella seguente immagine viene rappresentata in maniera semplice la rete creata.



La macchina rappresentata in nero, è il sistema di prova. Come è possibile notare è stata fornita di due schede di rete rispettivamente nominate eth0 e eth1. La simbologia appare abbastanza chiara. La macchina in questione è adibita principalmente al ruolo di gateway e firewall. E' posta tra la rete esterna e quella interna. Quest'ultima è composta da un numero arbitrario di host e il traffico è smistato da un semplice switch.

I computer della LAN interna non hanno subito particolari configurazioni. Sono macchine generiche sulle quali è installato un sistema operativo qualsiasi.

I due adattatori di rete sono stati configurati nel seguente modo:

L'interfaccia eth0 con:

- Indirizzo IP 213.82.177.140
- Subnet Mask 255.255.255.224
- Gateway 213.82.177.129
- DNS1 151.99.125.2
- DNS2 151.99.250.2

L'interfaccia eth1 con:

- Indirizzo IP 192.168.69.1
- Subnet Mask 255.255.255.0

La connessione ad internet avviene tramite una rete ad indirizzo IP statico. Per quanto riguarda la rete interna, solo il server ha un indirizzo fisso, gli host, invece, hanno l'indirizzo IP assegnato dal server **DHCP** installato.

La macchina si occupa anche dell'associazione indirizzi-nomi. Funziona, quindi, come server **DNS**. Sorge questa necessità dal momento in cui è stato definito un dominio. Nel caso di studio in questione, il dominio scelto come esempio è "naso.eu". In seguito verrà chiarito il perché di questo nome. E' stata, inoltre, aggiunta la funzionalità di **DynamicDNS** per l'aggiornamento dinamico delle associazioni indirizzi-nomi.

Come già anticipato, è stato utilizzato anche un **Proxy** server, cosa che si è rivelata molto utile per il controllo del traffico uscente.

Quasi scontato è l'utilizzo del **firewall** per il filtraggio dei pacchetti e natting. In questo caso è stato utilizzato il firewall fornito con il sistema operativo, iptables/netfilter, di cui si parlerà più in dettaglio nei prossimi capitoli.

Sistema di prova

Come già anticipato, il sistema preso per testare la realizzazione della rete, è un semplice personal computer. Le caratteristiche hardware non richiedono particolari configurazioni se non la presenza di due schede di rete. Nel caso in esame ci si è avvalsi di una macchina anche piuttosto datata, ma perfettamente adatta al ruolo da ricoprire. Tale sistema godeva delle seguenti caratteristiche:

- Processore x86
- 512 MB di RAM
- 30 GB di spazio su Hard Disk
- Scheda di rete eth0 per il collegamento alla rete esterna
- Scheda di rete eth1 per il collegamento alla rete interna

Durante le prove, è stato riscontrato un uso intensivo della cpu e della memoria ram, di conseguenza sarebbe saggio non risparmiare su queste due componenti, al fine di non sacrificare le prestazioni dell'intera rete. I requisiti descritti sono stati sufficienti per fini di test, ma nella realtà è ben diverso.

Installazione del sistema operativo

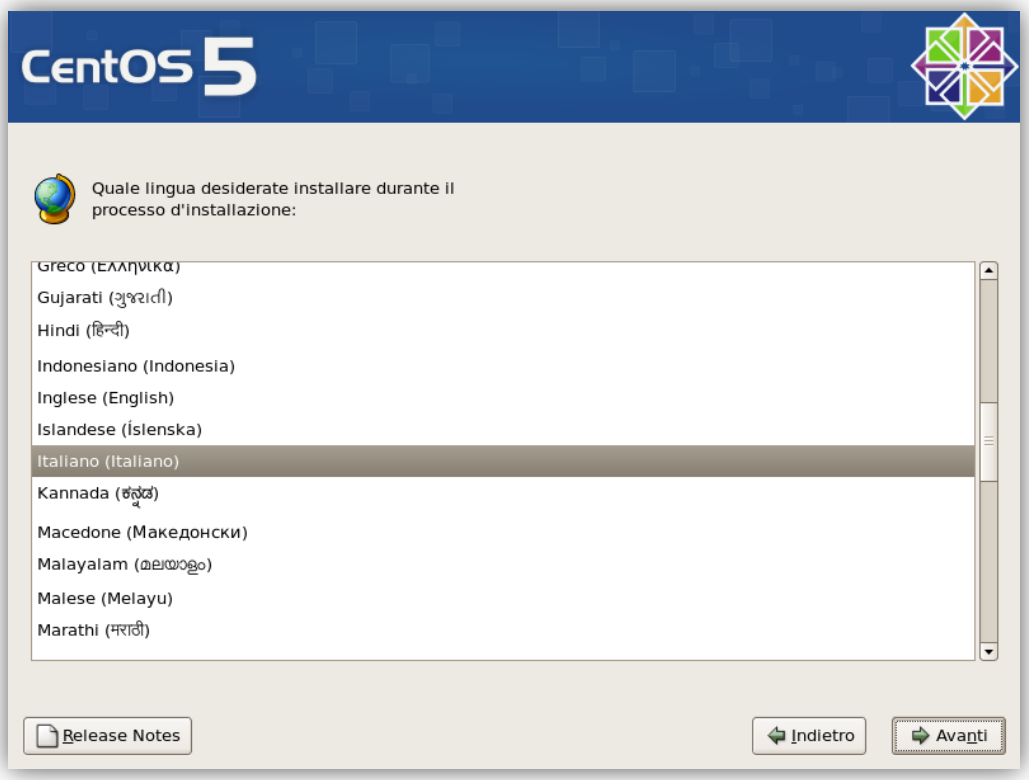
Come già detto nell'introduzione, questa tesi si prefigge l'obiettivo di realizzare un sistema di networking sicuro, utilizzando software open source.

La scelta del sistema operativo da usare è caduta sulla nota distribuzione Linux CentOS 5. Si tratta di un sistema basato sulla ben più famosa Red Hat Enterprise Linux, ed è pensata per un utilizzo server. Differisce dalla versione da cui deriva quasi solamente per l'assenza di assistenza, il motivo per cui la Red Hat è a pagamento.

Sul sito ufficiale del progetto (www.centos.org) sono disponibili immagini CD e DVD avviabili e che forniscono un pratico installer grafico che aiuta di molto il processo di installazione del sistema operativo.

Di seguito vengono riportati i passi seguiti per la corretta installazione del sistema sulla macchina di prova.

Le prime informazioni che vengono richieste all'utente sono la lingua e il layout della tastiera da utilizzare:

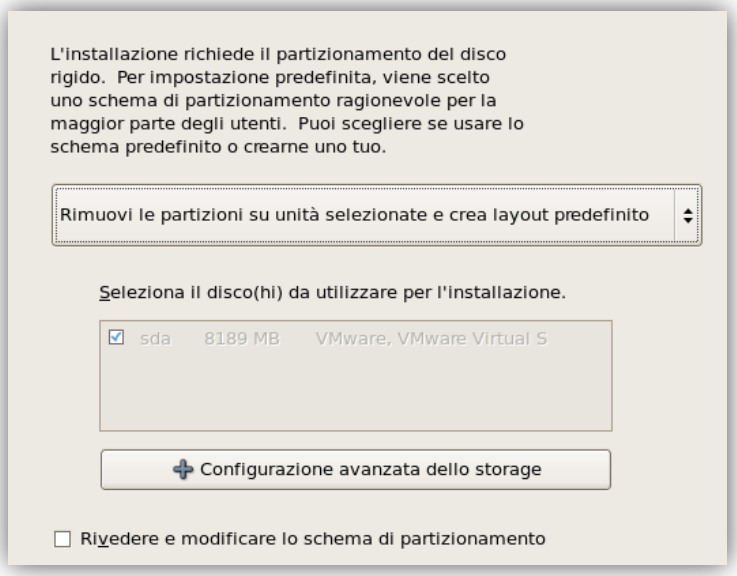


Successivamente viene richiesto di provvedere al corretto partizionamento del disco rigido. E' possibile scegliere una delle impostazioni predefinite oppure agire manualmente. Nel caso di un disco rigido formattato o da formattare, non occorre creare a mano le partizioni, in quanto non si corrono rischi di perdere dati.

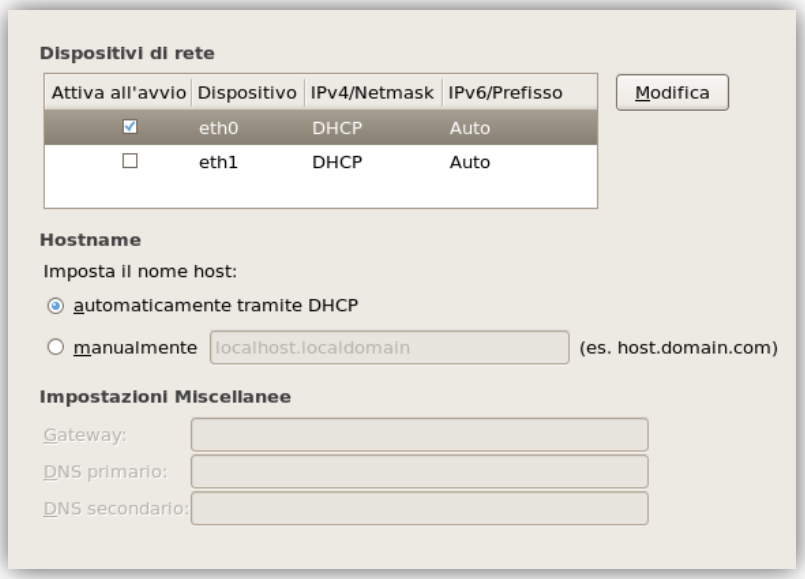
Le opzioni disponibili sono:

- Rimuovere le partizioni Linux sulle unità selezionate creando un layout predefinito
- Rimuovere le partizioni sulle unità selezionate creando un layout predefinito
- Creare un layout manualmente
- Utilizzare lo spazio disponibile per creare una struttura di default

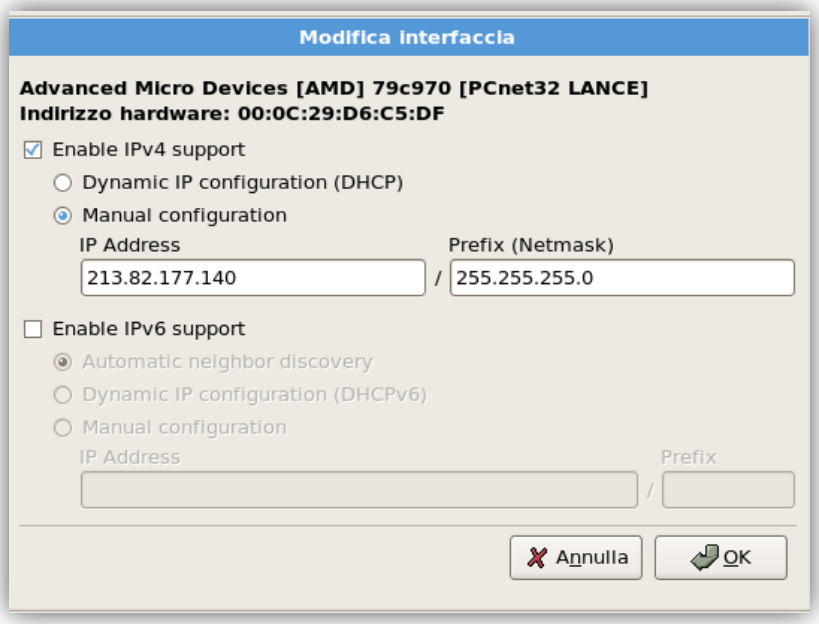
Per la macchina di prova è stata la seconda opzione, come da figura.

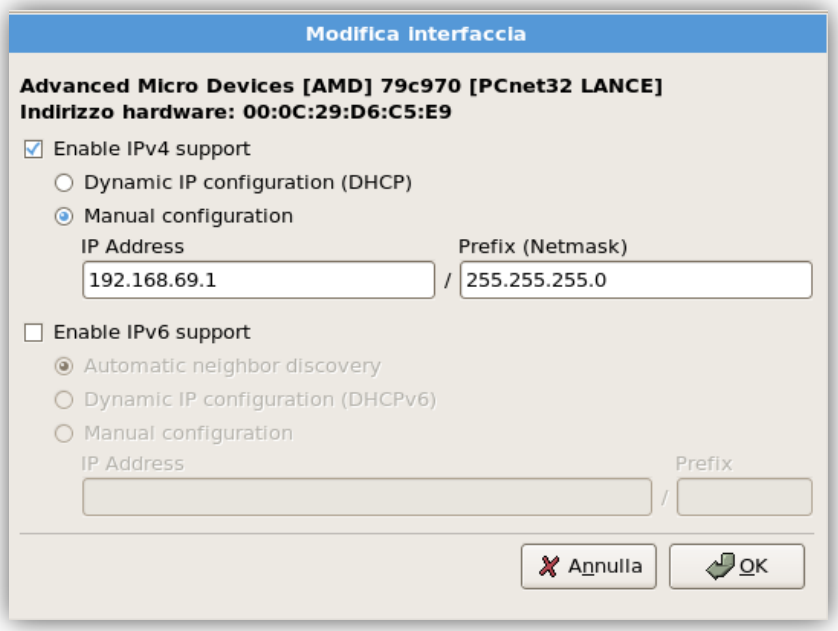


Adesso viene la parte più importante e delicata, ovvero la configurazione delle schede di rete. E' bene provvedere immediatamente alla corretta impostazione delle due interfacce per evitare problemi successivamente.



Le schede di rete vengono rilevate automaticamente e viene data la possibilità di configurarle prima dell'installazione del sistema. Selezionando una interfaccia e facendo click sul tasto modifica, si accede alle seguenti due finestre, rispettivamente per eth0 ed eth1.





Fatto questo è possibile impostare Gateway, DNS primario e secondario. Come è possibile notare è stato disabilitato il supporto all'ipv6, in questo caso inutile. La configurazione di rete è conclusa.

Il passo successivo riguarda la specifica dei pacchetti da installare definendo la destinazione del sistema. E' possibile scegliere tra un utilizzo Desktop e un utilizzo Server. E' consigliabile selezionare manualmente i pacchetti e le applicazioni da includere durante l'installazione.

Infine viene richiesto l'inserimento di una password di *root* tramite la seguente finestra.

A questo punto inizia la procedura di copia dei file sul disco rigido. Al termine, il sistema si riavvierà ed effettuerà il boot direttamente dall'hard disk e non più dal disco ottico. Per questioni di sicurezza, verrà richiesta la creazione di un utente con minori privilegi rispetto a root. Tale operazione non è obbligatoria, ma è comunque consigliata nella maggior parte dei casi.

Ad operazione conclusa si disporrà di un ottimo sistema per un utilizzo server. La distribuzione CentOS nasce proprio a questo scopo e dispone di un repository con una vasta disponibilità di software per la fornitura di servizi di rete. Risulta sicuramente un'ottima soluzione per un'azienda che necessiti di lavorare con tali strumenti. Offre, inoltre, il vantaggio di essere costituita completamente da software libero e questo comporta la totale gratuità nel suo utilizzo.

Al fine di rendere più sicuro e, sicuramente, più performante il sistema, sono stati disattivati alcuni servizi inutili. Ad esempio,

- tutti i servizi relativi al protocollo IPv6, ancora inutilizzato.
- SELinux, servizio di sicurezza non necessario ai fini di studio e test.
- Bluetooth – servizio per la gestione dello scambio di dati senza fili.
- Cups – Un servizio di stampa su macchine Unix.
- Ip6tables – Versione ipv6 del noto tool di configurazione del firewall.

Oltre a rendere più veloce l'esecuzione del sistema, rende più snella anche la fase di boot.

DHCP

Il **Dynamic Host Configuration Protocol (**tradotto: *protocollo di configurazione dinamica degli indirizzi*) o **DHCP** è un protocollo che permette, ai dispositivi di rete, di ricevere la configurazione IP necessaria per poter operare su una rete basata su Internet Protocol.

Per far parte di una rete, un elaboratore deve possedere un indirizzo univoco. Questo significa che due calcolatori non possono avere lo stesso indirizzo. In reti di grandi dimensioni, l'assegnazione degli indirizzi IP diventa un problema. Assegnare a tutte le macchine, che si collegano alla rete, un indirizzo diverso, può diventare un compito difficile per gli amministratori della rete, senza considerare che non tutte le macchine si collegano allo stesso momento.

Per rendere più semplice tale compito, viene usato un server che svolge appositamente la funzione di assegnare gli indirizzi agli host. Attualmente questi server possono trovarsi su macchine vere e proprie o su più semplici apparati come router e access point.

Il protocollo DHCP viene normalmente usato su reti locali. E' utile per rendere una macchina immediatamente pronta all'utilizzo della rete. Oltre all'assegnazione dell'indirizzo IP, viene usato per comunicare altre informazioni necessarie agli host per il loro corretto funzionamento in rete.

Ad esempio:

- Maschera di sottorete
- Default Gateway
- Indirizzo del server DNS primario
- Indirizzo del server DNS secondario
- Nome del dominio DNS di default
- Indirizzo del server WINS

- Indirizzo del server NTP
- Server Proxy

Componenti del protocollo

Il protocollo DHCP ha bisogno di due componenti fondamentali per funzionare:

- Un **server** DHCP: il calcolatore che assegna gli indirizzi IP (spesso un router), e anche il processo che svolge questa funzione.
- Un **client** DHCP: il calcolatore che ha bisogno di ottenere un indirizzo IP valido per la sottorete a cui è collegato, e anche il programma che si occupa di richiedere l'indirizzo IP e configurarlo.

E' prevista la presenza di un terzo componente:

- Un DHCP **Relay**: il calcolatore che si occupa di inoltrare le richieste DHCP ad un server, qualora questo non sia sulla stessa sottorete. Questo componente è necessario solo se un server DHCP deve servire molteplici sottoreti. Deve esistere almeno un DHCP relay per ciascuna sottorete servita. Ogni relay deve essere esplicitamente configurato per inoltrare le richieste a uno o più server.

Funzionamento

DHCP utilizza il protocollo UDP. Le porte registrate sono la 67 per il server e la 68 per il client.

Prima di ottenere l'indirizzo dal server DHCP, un client ha convenzionalmente un indirizzo impostato a 0.0.0.0. Per richiedere un indirizzo valido (e le relative informazioni annesse), il client invia in broadcast un pacchetto denominato DHCPDISCOVER. Tale pacchetto viene ricevuto da tutti gli host che compongono la rete, ma a rispondere, eventualmente, saranno solamente i server DHCP. La risposta avviene mediante un pacchetto chiamato DHCPOFFER che, appunto, fornisce tutti i dati necessari al client per prendere parte alla rete. Il client attende un certo tempo prima di rispondere alle offerte, questo per poter permettere a tutte le offerte di giungergli. Successivamente effettua una

scelta tra tutte le offerte ricevute, e comunica al relativo server DHCP di aver acquisito l'indirizzo comunicato. Questo viene fatto tramite la spedizione di un pacchetto conosciuto come DHCPREQUEST. Questo pacchetto viene mandato in broadcast, in modo tale da far sapere a tutti i server DHCP, quale indirizzo è stato scelto. Infine, il server conferma l'avvenuta assegnazione con un pacchetto chiamato DHCPACK. A questo punto, il client, avendo acquisito un indirizzo di rete valido, fa parte della rete a tutti gli effetti e può utilizzare la connettività a disposizione.

L'indirizzo assegnato, rimarrà tale per un limitato intervallo di tempo, denominato *tempo di lease* o *lease time*. Per rinnovare l'indirizzo prima della scadenza, il client deve rimandare un pacchetto DHCPREQUEST al server, il quale ritornerà un DHCPACK.

Realizzazione di un server DHCP

L'installazione del server Dhcp, per quanto riguarda il sistema operativo utilizzato, si è rivelata molto semplice. Il pacchetto necessario è presente nel repository ufficiale della distribuzione in uso, di conseguenza è stato sufficiente usare il comando:

```
# yum install dhcpd
```

Una volta installato il pacchetto è necessario configurarlo. E' possibile fare questo da Webmin (configuratore grafico che verrà illustrato in seguito), oppure manualmente, tramite un semplice editor di testo. Tutte le impostazioni vanno modificate nel file /etc/dhcpd.conf.

Per quanto riguarda la Lan, le impostazioni necessarie sono:

```
# LAN
subnet 192.168.69.0 netmask 255.255.255.0 {
      option subnet-mask 255.255.255.0;
      option domain-name "naso.eu";
      option routers 192.168.69.1;
      option domain-name-servers 192.168.69.1;
      range dynamic-bootp 192.168.69.100 192.168.69.200;
      default-lease-time 31200;
      max-lease-time 62400;
      }
```

Questo segmento definisce:

- La rete, indirizzi e subnet mask
- Il nome del dominio
- L'indirizzo del server
- Il range di indirizzi da assegnare
- i tempi di lease

A questo punto il server è già pronto a divenire operativo. E' sufficiente eseguire il seguente comando per iniziare ad assegnare subito indirizzi IP.

```
# service dhcpd start
```

Sono disponibili, inoltre, i parametri `stop` e `restart` per fermare e riavviare il servizio.

DNS

Domain Name System (spesso indicato con **DNS**) è un servizio utilizzato per la risoluzione di nomi in indirizzi IP e viceversa. Il servizio è realizzato tramite un database distribuito, costituito dai server DNS. Il nome DNS denota il protocollo che regola il funzionamento del servizio, i programmi che lo implementano, i server su cui questi girano, l'insieme di questi server che cooperano per fornire il servizio.

L'operazione di conversione di un nome in un indirizzo è detta **risoluzione** DNS, mentre la conversione di un indirizzo IP in nome è detta **risoluzione inversa**.

Fondamentalmente, il motivo dell'esistenza di un tale servizio è dato dalla più facile memorizzazione di una stringa di caratteri, piuttosto che di una sequenza di numeri, da parte di un essere umano. I siti web, fisicamente si trovano negli hard disk dei server sparsi per il globo, i quali insieme formano la rete internet. Per visitare un determinato sito web, occorre, quindi, identificare il server che lo ospita. Per identificare tale macchina occorre il suo indirizzo IP. In sostanza, per visitare il sito, dovremmo conoscere e digitare l'indirizzo IP in un browser. Se internet fosse composta da un unico sito web, questo andrebbe bene. Non dovrebbe risultare troppo complicato tenere a mente una sequenza di, al più, 12 cifre. Il problema nasce vista la molteplicità di siti e risorse disponibili online. Un essere umano non sarà mai in grado di memorizzare gli indirizzi di tutti i siti di suo interesse. Per questa

ragione è stato sviluppato il sistema DNS. L'uomo è capace di memorizzare nomi e parole che identificano le risorse online. Il sistema DNS, poi, provvede a tradurre queste parole nei corretti indirizzi e, quindi, permettere all'utente la fruizione dei contenuti senza la minima fatica.

Lo spazio dei nomi viene implementato gerarchicamente. I server DNS vengono disposti in una gerarchia ad albero, in cui in cima si trovano i root server. In questo modo è possibile suddividere l'intero spazio dei nomi in aree denominate *zone*. La gestione di una zona è **delegata** dalla zona superiore tramite dei record di tipo NS (Name Server). Ad esempio, nella zona *.org* ci sarà una delega per la zona *esempio.org* (ammesso che esista) ai server DNS che la gestiscono. All'interno di una zona possono essere delegate delle zone di livello inferiore, ad esempio in *esempio.org* potrebbero esistere deleghe per *livello3.esempio.org* o per *livello4.livello3.esempio.org*. Per ragioni di ridondanza, ciascuna zona è **replicata** su più server, e di conseguenza la delega è costituita da più record NS, che indicano che ciascuno dei server indicati contiene le informazioni per quella zona. Il server che contiene le informazioni per una zona, viene detto **autoritativo** per quella zona.

Come già detto, una delega può essere presente su diversi server. Ovviamente, tutti i server dovrebbero contenere le stesse informazioni relative alla zona in questione. Per far si che le informazioni siano distribuite nella maniera corretta, sono stati definiti server di tipo *master* e server di tipo *slave*. Gli aggiornamenti vengono effettuati sui server master, successivamente gli slave copieranno i dati dai master. Per tenere sotto controllo le diverse versioni relative alle varie zone, i record vengono accompagnati da un numero di serie. In questo modo è possibile identificare le differenti versioni per le stesse zone. Il numero di serie deve essere aumentato ogni volta che vengono apportate delle modifiche.

L'operazione di copia di tutti i record di una zona, dal master ad uno slave è detta *zone transfer*, e può essere *completa* (tutto il contenuto della zona viene copiato) o *incrementale* (vengono copiati solo i record modificati rispetto alla versione già presente.

Interrogazione

Per ottenere la risoluzione di un nome, si parte da uno dei root server. Questi reindirizzeranno al server che gestisce il dominio di secondo livello in questione. Si procede fino a giungere al server autoritativo per il nome desiderato. Questo meccanismo prende il nome di **ricorsione**.

Funzioni dei Server DNS

Un server DNS può essere configurato per assolvere ad una o più delle seguenti funzioni:

- **server autoritativo** per una o più zone, ovvero il server su cui sono configurati i dati di una zona, e che è delegato a gestirla tramite record NS inseriti nella zona superiore. Normalmente sono presenti più server autoritativi per una zona. Molte implementazioni permettono di modificare i dati di una zona solo su un server. Il server autoritativo può a sua volta essere:
    - **primario** - server autoritativo su cui vengono modificati i dati di una zona
    - **secondario** - server autoritativo che copia i dati di zona da un primario
- **server ricorsivo** - il server che viene configurato in una popolazione di client, che si occupa di risolvere le query che riceve interrogando i server originali, e mantenendo una cache delle risposte ricevute
    - **query forwarder** - un server che viene configurato in una popolazione di client, che risolve le loro query non direttamente ma interrogando un server ricorsivo.

Realizzazione di un server DNS

Esistono diversi pacchetti software che servono a realizzare un server DNS. Di seguito alcuni dei più diffusi:

- **BIND** (Berkeley Internet Name Domain) – Utilizzato su sistemi UNIX. Uno dei più diffusi.
- **DJBDNS** (Dan J Bernstein's DNS implementation)
- **PowerDNS**
- **MaraDNS**
- **DDNS** (Dynamic Domain Name System) Il servizio DNS Microsoft.

Per la rete di test, è stato scelto il primo, essendo il sistema operativo utilizzato, basato su Unix.

L’installazione è piuttosto semplice. Per quanto riguarda la distribuzione in uso, è possibile usufruire dei pacchetti precompilati presenti nel repository ufficiale. I pacchetti da installare sono i seguenti:

- bind
- bind-chroot

Il primo è il programma vero e proprio, il secondo permette di rendere più sicuro il server DNS, nascondendolo dietro un falso percorso. Questo permette di proteggere il proprio sistema in due direzioni, dall'interno e dall'esterno. Chroot fa si che il programma installato non veda altro che i file relativi a se stesso. In questo modo potrà modificare solo i file ad esso appartenenti.

Se si sfruttasse una vulnerabilità del programma per entrare nel sistema, non si potrebbe che vedere e modificare i file del programma sfruttato, in quanto chroot non permetterebbe di uscire al di fuori del perimetro creato per l'installazione.

Per l'installazione è sufficiente eseguire il seguente comando:

```
# yum install bind bind-chroot
```

Alternativamente, è sempre possibile installare il programma compilando i sorgenti. La procedura è approssimativamente sempre la stessa:

```
# ./configure
# make
# make install
```

E' bene fare attenzione alle variabili che è possibile impostare manualmente. Le principali sono:

- --*with-libtool*: per abilitare l'uso delle shared library.
- --*with-openssl*: per abilitare l'uso della crittografia per l'autenticazione sicura.
- --*with-openssl=/prefix*: per specificare il percorso di installazione di openssl, nel caso non fosse stato installato nel percorso di default
- --*enable-libbind*: per abilitare l'uso delle librerie libbind di BIND versione 8, visto che attualmente le librerie di BIND 9 sono ancora considerate in sviluppo, può essere interessante l'uso di quelle meno recenti ma più stabili.
- --*enable-threads* o --*disable-threads*: utile per macchine multiprocessore.
- --*prefix=/...:* per specificare una directory di installazione alternativa per il servizio named.

- --*sysconfdir*: per specificare un percorso alternativo per il file di configurazione named.conf.

Ad ogni modo se si vuole avere un prospetto completo di tutte le opzioni del comando *configure* eseguire:

```
# ./configure --help
```

BIND è un pacchetto software composto principalmente da:

- Il **servizio** vero e proprio detto Domain Name Service. Tale servizio è chiamato named.
- Una **libreria** per la risoluzione dei nomi.
- Una serie di **tool** e **utility** per la gestione e la configurazione del servizio. (rndc, named-checkconf).

Le modifiche alla configurazione vanno specificate nel file:

```
/etc/named/chroot/etc/named.conf
```

Si tratta di un semplice file di testo che contiene le direttive. Specifica, inoltre, le zone create e la loro ubicazione all'interno del file system.

Le principali direttive sono:

- **Acl** (Access Control List) Questa direttiva permette di assegnare un nome arbitrario ad un determinato range di indirizzi per facilitarne poi la loro gestione in altre specifiche del named.conf.
  La sua sintassi è la seguente:

```
acl nome-acl {
    lista_di_indirizzi
};
```

- **controls** - Permette la definizione di un canale usato dall'amministratore per gestire il servizio. Questi canali sono principalmente usati dall'utility **rndc** per inviare comandi al server.

```
controls {
   inet (indirizzo_ip | * ) [ port numero_porta ] allow {  acl  }
         keys {  lista_chiavi  };
};
```

- **key**: Definisce una chiave segreta per l'autenticazione mediante signature (TSIG, DNSSEC) viene detta *shared* perchè è condivisa dal server e delle applicazioni che ne fanno uso.

```
key nome_chiave {
    algorithm stringa;
    secret stringa;
};
```

Il nome della chiave è arbitrario, va poi ovviamente ripetuto uguale in tutte le direttive che lo richiedono. La direttiva *algorithm* in teoria potrebbe essere di tipo qualunque ma attualmente (BIND 9.2.2, marzo 2003) named supporta esclusivamente hmac-md5.

```
zone "db.esempio" {
        type master;
        file "/var/named/db.esempio";
};
```

- **include**: Permette di definire un file esterno contenente ulteriori direttive. Può essere utile nel caso in cui si abbia necessità di dare l'accesso in lettura al **named.conf** a un gruppo di utenti, ma non si vuole che la secret key sia accessibile. Si specifica così il parametro key in un file esterno non leggibile e lo si *"include"* nel named.conf.

include <nome_file>

- **zone**: usata per definire le zone e i rispettivi file.

Dichiarazione per la zona radice:

```
zone "." IN {
    type master;
    file "/var/named/root.hint";
};
```

Dichiarazione per localhost:

```
zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "/var/named/127.0.0.db";
};
```

Quest'ultima è una delle più importanti. Oltre a definire le zone per la propria rete, occorre la specifica per la zona radice o zona '.':

Per quanto riguarda la versione installata sul sistema di prova, il file di configurazione, inizialmente, contiene ben poco:

```
options {
      directory "/etc";
      pid-file "/var/run/named/named.pid";
      forwarders {
            151.99.125.2;
            151.99.250.2;
            };
      };
```

Vengono specificati la directory contenente il file in questione, il pid-file e i server DNS primario e secondario, usati per internet.

Le impostazioni si arricchiscono nel momento in cui vengono create le zone, che vengono così definite:

```
zone "naso.eu" {
      type master;
      file "/var/named/naso.eu.hosts";
      };

zone "69.168.192.in-addr.arpa" {
      type master;
      file "/var/named/192.168.69.rev";
      };
```

Le zone create sono due:

- naso.eu – la zona "*diretta*".
- `69.168.192.in-addr.arpa` – la zona "*inversa*".

Entrambe sono zone di tipo master e vengono specificati i file di zona creati con relativo percorso all'interno del file system.

A questo punto è possibile avviare il servizio DNS tramite il seguente comando:

```
# service named start
```

Come per il dhcp, named dispone dei parametri `stop` e `restart`.

Una volta configurati correttamente DHCP e DNS, si vorrà, probabilmente, renderli avviabili al boot di sistema. E' possibile fare questo tramite i comandi:

```
# chkconfig dhcpd on

# chkconfig named on
```

DynamicDNS

Il *DNS dinamico*, o più brevemente *DDNS*, è un meccanismo che permette di inserire automaticamente in una zona DNS, una macchina che ottiene un indirizzo non predefinito. Si utilizza per mantenere l'associazione nome-macchina anche se l'indirizzo cambia nel tempo. Risulta una tecnologia fondamentale nei casi di server con indirizzo IP non statico. Senza questo accorgimento, un server non risulterebbe sempre raggiungibile tramite il suo nome DNS.

I nomi DNS sono normalmente associati stabilmente ad indirizzi IP, i quali a loro volta sono stabilmente assegnati ad host che hanno funzioni di server. Molti host, in particolare quelli che si collegano ad internet utilizzando i servizi di un ISP, ricevono invece un indirizzo diverso ad ogni connessione. Pertanto è impossibile raggiungerli da internet, in quanto non si conosce il loro indirizzo IP. Il DNS dinamico permette a questi host di essere sempre raggiungibili attraverso il loro nome DNS.

Nel caso in esame, è possibile configurare il dhcp server affinché aggiorni i file di zona nel momento in cui assegna un indirizzo IP ad un host. Per la realizzazione del DNS dinamico è necessario agire sui file di configurazione *named.conf* e *dhcpd.conf*.

Dal lato del DNS (named.conf) è, per prima cosa, necessario creare una chiave per la definizione dei permessi di aggiornamento dei file di zona, tramite la direttiva **Key** precedentemente descritta. Ad esempio:

```
key DHCP_UPDATER {
    algorithm HMAC-MD5;
    secret "Yj95beDnn=34fghSN";
};
```

In questo caso, la chiave è stata chiamata DHCP_UPDATER ed è stata creata tramite l'algoritmo hmac-md5. La direttiva secret specifica la stringa utilizzata per il *signing* (firma) delle connessioni.

In realtà, l'utilizzo di una chiave non è obbligatorio. E' possibile specificare direttamente nella direttiva allow-update, una lista di indirizzi ip o una acl per il controllo degli update automatici. Tuttavia, per ragioni di sicurezza, è preferibile l'utilizzo di tali chiavi.

Bisogna stabilire, poi, il canale di controllo che verrà utilizzato dal server DHCP per l'invio dei dati da aggiornare. Essendo DNS e DHCP entrambi sulla stessa macchina, saranno concesse solamente connessioni provenienti dalla macchina stessa (localhost).

```
controls {
    inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "DHCP_UPDATER"; };
};
```

Per finire, per ogni file di zona va specificata la direttiva *allow-update*, la quale garantisce la possibilità di effettuare aggiornamenti solamente ai possessori della chiave.

```
zone "esempio.com" IN {
    type master;
    file "esempio.com.db";
    allow-update { key DHCP_UPDATER; };
};
```

In questo esempio, i possessori della chiave DHCP_UPDATER avranno il permesso di eseguire aggiornamenti del file di zona relativo alla zona esempio.com.

Dal lato del server DHCP invece, è necessario apportare le seguenti modifiche al file di configurazione. E' indispensabile far conoscere al server DHCP la chiave da usare per gli updates e, ovviamente, specificarla in ogni zona definita. Nel seguente modo:

```
key DHCP_UPDATER {
      algorithm hmac-md5;
      secret
"z+q3DukgZqQ95mzJe0gx7Q==";
}

zone naso.eu. {
      primary 127.0.0.1;
      key DHCP_UPDATER;
}

zone 69.168.192.in-addr.arpa. {
      primary 127.0.0.1;
      key DHCP_UPDATER;}
```

La direttiva primary specifica la posizione del name server. In questo caso la macchina su cui si sta lavorando, fa sia da dhcp che da dns, di conseguenza viene specificato l'indirizzo localhost.

Inoltre, è fondamentale impostare la direttiva *ddns-update-style* su "*interim*". Tale valore sostituisce l'ormai deprecato "*ah-hoc*".

Firewall

Il firewall, letteralmente "muro tagliafuoco", è un componente passivo di difesa che viene solitamente posto tra due reti: una esterna e una interna.

Si usa principalmente per proteggere la propria rete interna da quella esterna. La rete esterna solitamente è rappresentata da internet mentre quella interna è una semplice rete locale.

Durante la progettazione di una rete è importante definire delle regole per stabilire come si può accedere ad alcuni servizi di rete importanti. Un firewall permette di specificare tali regole in modo da creare relazioni sicure tra gli host della rete e l'ambiente esterno.

Può consistere in un apparato specializzato o un semplice computer adibito a tale compito. Ad essere interessante, nonché rilevante ai fini di questa tesi, è il secondo caso. Il computer in questione deve avere almeno due schede di rete e ovviamente software apposito alla realizzazione del firewall. Una scheda è utilizzata per il collegamento alla rete esterna, mentre l'altra per la rete interna. In questo modo tutto il traffico passa inevitabilmente per la macchina che fa da firewall e questo consente di analizzare tutti i pacchetti che transitano.

Il compito principale di tale apparato è quello di filtrare il traffico entrante e uscente secondo determinate regole stabilite a priori.

Oggi la quasi totalità dei sistemi operativi include funzionalità di firewall. Questo è dovuto alla sempre crescente necessità di sicurezza. Sono inoltre disponibili sul mercato diverse soluzioni software con funzionalità di firewall, in particolare per la piattaforma Windows.

I sistemi Linux hanno sempre avuto strumenti capaci di filtrare i pacchetti. Nel tempo sono stati apportati notevoli miglioramenti a tali applicazioni. Quella che si può, in una qualche maniera, definire la prima generazione, era denominata *ipfw* (IP firewall) e forniva capacità di filtraggio di base. Attualmente, tale soluzione è caduta quasi completamente in disuso. La seconda generazione è costituita da *IP chains*. Si tratta di una evoluzione di ipfw ed è ancora comunemente usato. L'ultima generazione prende il nome di *Netfilter*. Quest'ultima è la soluzione attualmente maggiormente diffusa e utilizzata. Spesso confusa con il ben più noto *Iptables*. Quest'ultimo, in realtà, non è altro che un tool creato per

impartire comandi a Netfilter. Venne adottato a partire dal kernel 2.4, in quanto Netfilter è un componente vero e proprio del kernel Linux.

Tipologie di firewall

Esistono diverse tipologie di firewall classificabili sulla base della loro complessità.

- **Packet Filter Firewall** o **Stateless Firewall**. Il più semplice. La sua funzione è quella di analizzare gli header di ogni pacchetto e decidere che azione intraprendere in base alle regole preconfigurate. Non tiene traccia delle connessioni, quindi non è sempre in grado di riconoscere a quale connessione i pacchetti fanno parte. Soluzioni di questo tipo sono spesso in grado di capire se i pacchetti che transitano fanno parte di una connessione stabilita oppure nuova, ma non sono in grado di riconoscere quei pacchetti, generati da codice malevolo, che fingono di far parte di connessioni già stabilite.

- **Stateful Inspection**. Questi firewall risolvono i problemi inerenti alla tipologia appena vista. Essi sono in grado di tenere traccia dei pacchetti che transitano e di ricostruire lo stato delle connessioni . Sono capaci di riconoscere a quali connessioni i pacchetti fanno riferimento. In questo caso è possibile identificare pacchetti malevoli che non fanno parte di alcuna connessione.
  Si rivelano molto utili per quei protocolli che aprono più connessioni. E', ad esempio, il caso del protocollo FTP.

- **Deep Inspection Firewall**. Effettuano controlli fino al livello 7 della catena ISO/OSI. Controllano il contenuto applicativo dei pacchetti.

- **Application Layer Firewall**. Intercettano connessioni a livello applicativo. E' il caso dei server proxy. Una macchina della rete non dispone di connessione diretta alla rete esterna, ma soltanto di una connessione al server proxy. Sarà quest'ultimo a decidere se far passare la connessione all'esterno oppure no. Può essere utilizzato per bloccare il traffico non desiderato.

Per la realizzazione di un firewall tramite l'utilizzo di software open source è possibile fare affidamento sull'ottima combinazione iptables/netfilter. Si tratta di una soluzione ampiamente utilizzata su sistemi Linux e decisamente valida in ambienti aziendali.

Proxy

Un proxy è generalmente una macchina che si interpone tra un server e un client. Il client non si connette direttamente al server, ma al proxy. Sarà quest'ultimo a gestire la connessione tra i due inoltrando le richieste al server e le relative risposte al client.

Il motivo principale dell'utilizzo di un server proxy è quello di aumentare le prestazioni della rete. Ciò è possibile grazie alla funzionalità di cache fornita dal server. Vengono, infatti, mantenute informazioni che potrebbero essere utili a soddisfare richieste multiple alla stessa risorsa. Quando viene, ad esempio, richiesta una pagina web, il proxy controlla se questa è presente in locale. Nel caso non lo fosse, viene richiesta al server web in questione e successivamente fornita all'host che ne ha fatto richiesta. La pagina viene poi salvata e conservata sul disco rigido del server proxy. In questo modo, ad una successiva richiesta, essa può essere immediatamente fornita, previa verifica di aggiornamenti, senza doverne scaricare nuovamente il contenuto.

In aggiunta a quanto appena detto, un proxy consente di controllare ed eventualmente filtrare il traffico. Un amministratore, potrebbe voler bloccare le richieste a contenuti che ritiene dannosi per gli utenti e gli host che gestisce.

L'utilizzo di tali sistemi offre un notevole vantaggio in ambienti costituiti da un elevato numero di macchine interconnesse. E' sicuramente il caso di una qualunque azienda di medie dimensioni. Una ipotetica rete aziendale è formata da un discreto numero di computer, i quali necessitano di connettività. Una rete poco performante potrebbe significare una perdita di produttività. Allo stesso modo, o probabilmente peggio, una rete non adeguatamente protetta e monitorata potrebbe scaturire in problemi di varia natura, sicurezza in primo luogo.

Tipologie

I server proxy possono essere distinti in base al lavoro che svolgono. Le necessità degli amministratori di rete possono essere di diversa natura. Essendo questi server degli strumenti particolarmente versatili, è possibile trovare dei proxy destinati a compiti differenti.

- **Caching proxy server –** Si tratta di una tipologia di proxy adibita alla funzione di cache. Viene largamente utilizzata dagli Internet Service Provider e da aziende di

grandi dimensioni. Questo tipo di server conserva risorse richieste frequentemente in modo che possano essere immediatamente disponibili al momento di richieste successive. Fisicamente sono composti da macchine molto veloci, in particolare per quanto riguarda l'immagazzinamento dati. Non è raro infatti che vi vengano implementati sistemi RAID per l'aumento delle prestazioni delle operazioni sui dischi rigidi.

- **Web Proxy** – Si tratta di *Caching proxy server* concentrati sul caching delle pagine web. Il loro scopo è quello di rendere la navigazione web più veloce.

- **Proxy per il filtraggio dei contenuti** – Sono proxy che hanno lo scopo di bloccare le richieste a contenuti non consentiti. Tali contenuti possono avere diversa natura, ad esempio URL, file o semplici parole. Vengono largamente utilizzati, in particolare in ambienti in cui determinati contenuti potrebbero risultare offensivi, se non dannosi per gli utenti e per gli host della rete.

Intrusion Detection and Prevention System (IDS/IPS)

Un IDS è un sistema utilizzato per identificare attività non autorizzate sulla rete. In particolare si prefigge l'obiettivo di analizzare il traffico entrante e uscente al fine di rilevare attacchi informatici di vario genere, ad esempio accessi non consentiti, Denial of Service ecc…

Un sistema di questo genere può essere software o hardware, ma più in genere è una combinazione di entrambi. L'IDS costituisce un apparato molto importante nella sicurezza di una rete. Consente di scandire tutto il traffico e di rilevare molteplici tipologie di attacchi.

In linea generale, un intrusion detection system è formato da diverse componenti.

- I *sensori* – Possono essere uno o più. La loro utilità è quella di ricevere il traffico dalla rete. Tale traffico verrà poi analizzato in cerca di attività non consentite.

- Una *console* – utile a monitorare il traffico e gli allarmi generati dai sensori.

- Un *motore* centrale – che analizza il traffico raccolto dai sensori ed eventualmente genera allarmi o alert.

E' importante ricordare che il compito di un IDS è esclusivamente quello di rilevare attacchi informatici. Al verificarsi di una intrusione, tutto ciò che l'IDS fa è quello di comunicare all'amministratore il verificarsi di tale attività non consentita. Può fare questo tramite la console o, come accade frequentemente, mandando una e-mail.

Al fine di offrire un servizio più completo, e un livello di sicurezza più elevato, si utilizza il cosiddetto Intrusion Prevention System o IPS. Si tratta di una soluzione del tutto simile alla precedente. L'unica differenza sta nel fatto che, in questo caso, gli attacchi vengono fermati prima che abbiano la possibilità di andare a buon fine.

Esistono, inoltre, diverse estensioni a questi sistemi. Si tratta di software che aggiungono funzionalità di vario genere. Ad esempio programmi che consentono al sistema IDS/IPS di collaborare con il firewall, aggiornando automaticamente le regole di quest'ultimo.

IDS e IPS sono soluzioni che difficilmente mancano all'interno di una rete aziendale. Al mondo esistono innumerevoli tipologie di aziende diverse, ma tutte hanno la necessità di proteggere i propri dati. Una perdita di dati per opera di una intrusione nella rete o, comunque, per un attacco in generale, può portare a seri problemi. Problemi che possono essere legati alla privacy di dipendenti e clienti, problemi legati ai capitali con cui l'azienda lavora, e così via.

Antivirus

Come già anticipato, i virus informatici costituiscono una grave minaccia per quanto riguarda la sicurezza di una rete, specie se si tratta di reti aziendali in cui la perdita di dati può costituire un grave problema. Esistono software che tentano di porre rimedio a questa pericolosa insidia. Si tratta dei cosiddetti Antivirus. Ne esistono molti e in diverse versioni. Essendo soprattutto diffusi su piattaforme Windows, per la maggior parte sono programmi commerciali e, quindi, a pagamento. Tuttavia esiste qualche rara soluzione Open Source. E' il caso di ClamAV. Si tratta di un famoso software Antivirus per sistemi Linux.

Senza un'attenta analisi potrebbe sembrare inutile avere un software antivirus installato su una distribuzione Linux. Non è così. Naturalmente tale software non mira a proteggere la macchina su cui è installato, essendo il numero di virus scritti per tale sistema trascurabile. Tuttavia, molto spesso capita che macchine Linux si trovino a difesa di reti composte di host Windows. E' facile capire che tale sistema ha lo scopo di proteggere le macchine della rete interna dalle infezioni indesiderate.

Gli Antivirus costituiscono uno strumento di difesa essenziale. Bisogna, però, tenere a mente che non è sufficiente utilizzare una macchina per proteggere l'intera sottorete. Sarebbe uno sbaglio non includere un antivirus per ogni macchina che compone la rete. Esistono molti tipi di connessioni e file differenti. Un virus può intrufolarsi all'interno della rete molto facilmente e non è sufficiente controllare solo l'ingresso. Un virus potrebbe essere contenuto in un archivio compresso scaricato da un sito, dalla posta elettronica, ma, caso ancor più frequente e pericoloso, potrebbe essere introdotto nella rete tramite un supporto rimovibile come, ad esempio, una pen drive USB.

Una volta all'interno, un virus può diffondersi anche molto velocemente. E' bene quindi che ogni macchina abbia a disposizione una propria protezione contro tale minaccia.

Configurazione con Webmin

Webmin è un software open source che mette a disposizione dell'utente strumenti per la configurazione e gestione, in locale o remota, di sistemi unix based.

Racchiude al suo interno un server web. Questo permette il suo funzionamento senza che sia necessario installare e configurare un server web vero e proprio, come ad esempio Apache. Di conseguenza, per poter usare webmin è sufficiente un qualunque browser.

Di default permette di interagire ampiamente con il sistema e con le applicazioni installate. Si rivela molto utile per la configurazione di software che non prevedono una propria interfaccia grafica, quindi utilizzabili sono da linea di comando.

Si tratta di un software il cui scopo è quello di rendere il sistema più *user friendly* possibile, e questo risulta evidente sin dal primo utilizzo. L'installazione si rivela molto semplice. Sul sito web sono disponibili i pacchetti precompilati per diversi sistemi. Sono

liberamente scaricabili i pacchetti per sistemi Red Hat, Debian e Solaris. Trattandosi di software libero, vengono messi a disposizione anche degli archivi con il codice sorgente.
Per la rete di test è stata usata la versione RPM, la cui installazione è stata effettuata con il seguente comando:

```
# rpm –iv webmin-VERSIONE.noarch.rpm
```

L'installazione da sorgenti non è molto differente. In questo caso è necessario scaricare l'archivio tar.gz e scompattarlo tramite:

```
# tar xvzf webmin-VERSIONE.tar.gz
```

Si verrà a creare una cartella contenente diversi file. Ciò che interessa è lo script di installazione che è possibile lanciare con i comandi:

`# cd webmin-VERSIONE` (con questo comando ci si sposta nella cartella appena creata)
`# ./setup.sh`

La procedura di installazione è interattiva, e all'utente verranno poste solamente alcune domande relative alla prima configurazione di Webmin.

Config file directory [/etc/webmin]:
(Directory contenente i file di configurazione)
Log file directory [/var/webmin]:
(Directory dove saranno posti i file di LOG)
Full path to perl (default /usr/bin/perl):
(Percorso dell'interprete Perl)
Web server port (default 10000):
(Porta sulla quale Webmin rimarrà in ascolto)
Login name (default admin):
(Premiamo INVIO oppure scriviamo un diverso nome utente)
Admin Password:
(Inseriamo la password desiderata)
Use SSL(y/n):
(y per usare il protocollo SSL, n altrimenti)
Start Webmin at boot time (y/n):
(se si vuol far partire webmin al boot di sistema, n altrimenti)

Dopo aver seguito le istruzioni a schermo, webmin sarà subito pronto all'uso. E' possibile accedere alla pagina principale semplicemente digitando nel browser l'indirizzo:

http://localhost:10000

10000 è la porta di default. In fase di installazione viene data la possibilità di cambiare la porta in ascolto, nel qual caso bisogna indicare al browser la porta scelta.

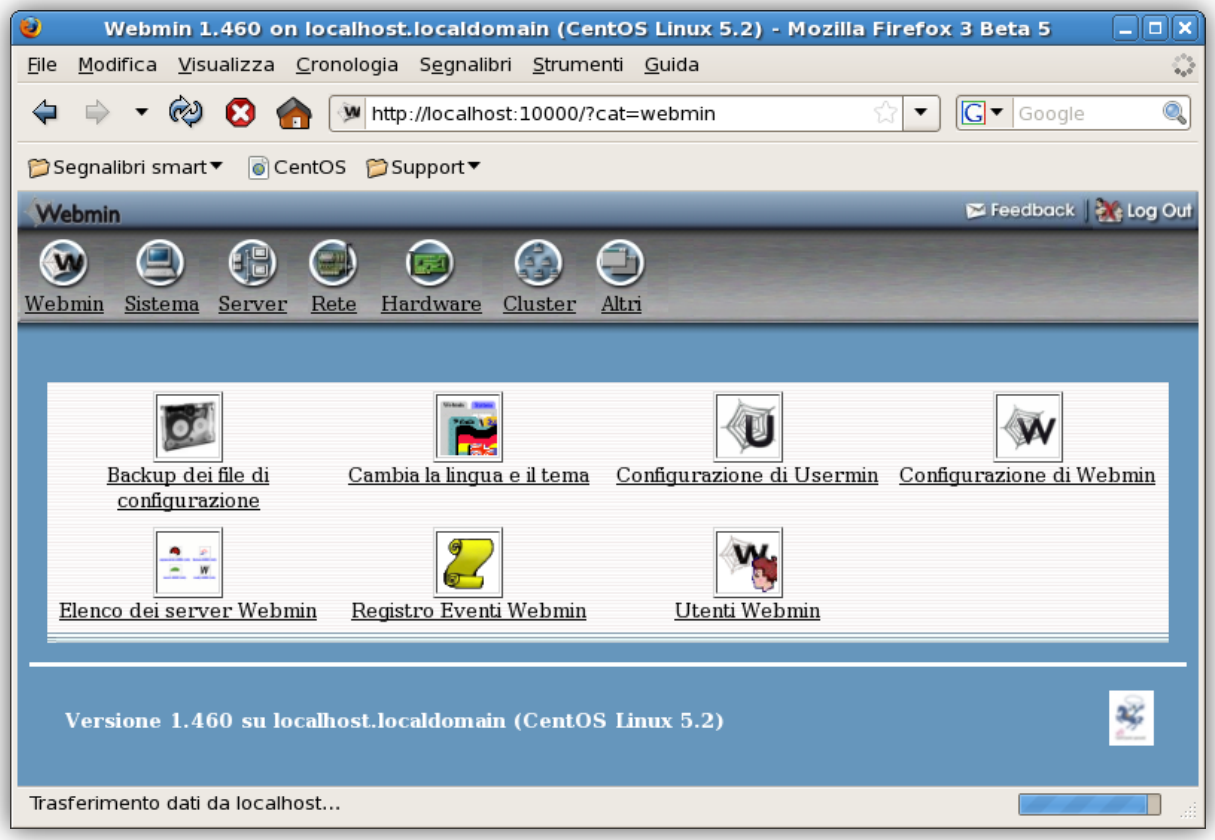
L'interfaccia di Webmin è organizzata nel seguente modo. Nella parte superiore si possono trovare sette categorie, ognuna delle quali contiene un certo numero moduli.

La versione di base di webmin comprende un vasto numero di moduli per la configurazione di diversi aspetti del sistema. Tuttavia è possibile aggiungere successivamente ulteriori moduli a seconda delle proprie necessità. Un esempio verrà riportato in seguito in proposito alla configurazione dell'IDS Snort, per il quale, appunto, esiste un modulo apposito.

Di seguito sono riportati i moduli utilizzati per il sistema di test:

- Categoria Server
  - BIND DNS server
  - DHCP Server
  - Snort IDS Admin
  - Server di Database Mysql
  - Squid Proxy Server
- Categoria Rete
  - Linux Firewall
- Categoria Sistema
  - Pianificazione Cron Job

La home page di Webmin dovrebbe presentarsi simile alla seguente:



La parte grafica è stata curata notevolmente. E' possibile personalizzare l'aspetto dell'interfaccia grazie ai numerosi temi disponibili.

Il sito web di riferimento è http://www.webmin.com.

Capitolo 3

Firewall

Iptables/Netfilter

Iptables è un software open source che permette all'utente di stabilire determinate regole di filtraggio del traffico di rete e di reindirizzamento NAT (Network Address Translation).

Si è soliti riferirsi a tale struttura con il termine Iptables, ma in realtà gran parte del merito va a Netfilter. Quest'ultimo è un componente del kernel Linux che si occupa dell'intercettazione e manipolazione dei pacchetti che attraversano la macchina. Iptables costituisce solo lo strumento con cui l'utente sarà in grado di gestire il comportamento di Netfilter.

La combinazione Iptables/Netfilter fu adottata a partire dalla versione 2.4 del kernel Linux. Precedentemente venivano usati altri sistemi, quali *ipfwadm* (usato fino al kernel 2.2) e *ipchain* (fino al kernel 2.4).

Con quest'ultimo, in particolare, si introduce il concetto di *catena* (*chain*). Una catena è costituita da un insieme di regole a cui un pacchetto viene indirizzato.

Funzionamento

I dati che transitano in rete sono suddivisi in pacchetti di dimensione prefissata. Ogni pacchetto può essere analizzato in modo tale da decidere la giusta azione da compiere in relazione ad esso.

Il framework netfilter/iptables si compone di due parti distinte ma cooperanti. La prima, Netfilter, lavora in kernel space ed è presente in quasi tutte le distribuzioni Linux. Offre la possibilità di essere integrata con moduli e plug-in aggiuntivi per l'inserimento di nuove funzionalità. La seconda, Iptables, lavora in User Space. Essa permette all'utente

stesso di interagire con Netfilter. Non è necessariamente installata di default in quanto non sempre si ha bisogno di lavorare con un firewall. Risulta comunque molto semplice munirsi di tale applicazione, in quanto presente in tutti i repository online. Non è necessario provvedere personalmente all'installazione di iptables se si lavora su distribuzioni destinate ad un utilizzo server o enterprise, come ad esempio Red Hat, CentOS o Debian.

Iptables risulta fondamentale nel caso in cui si abbia bisogno di far funzionare il proprio sistema come firewall. Costituisce il mezzo con cui l'utente è in grado di definire le regole per i pacchetti e tutta una serie di opzioni per un'opportuna configurazione del firewall.

Struttura

Il sistema iptables/netfilter è organizzato in tabelle (tables) e catene (chains).

Le tabelle standard sono tre:

- Filter
- Nat
- Mangle

La tabella ***Filter*** è quella dedicata al filtraggio del traffico passante per la macchina. L'obiettivo è quello di bloccare o far passare i pacchetti in transito. Essa contiene a sua volta tre catene predefinite:

- *INPUT* - alla quale sono indirizzati i pacchetti destinati al sistema.
- *OUTPUT* - alla quale sono indirizzati i pacchetti generati dal sistema.
- *FORWARD* - alla quale sono indirizzati i pacchetti che transitano per il sistema, ovvero quelli né generati dal sistema e né destinati ad esso.

La tabella ***Nat*** regola le attività di natting, ovvero ha l'obiettivo di modificare opportunamente indirizzi e porte dei pacchetti. Tale funzionalità si rivela molto utile nei sistemi muniti di più interfacce di rete e che fungono da router. Di conseguenza, risulta fondamentale nei casi di reti con un determinato numero di Host tutti contemporaneamente connessi ad internet, ma con un unico indirizzo IP esterno.

Questa tabella presenta tre catene prestabilite:

- ***PREROUTING*** - catena in cui vengono processati i pacchetti in ingresso prima che il sistema ne decida l'instradamento tramite le tabelle di routing. Usata per il Nat sulla destinazione o DNAT.
- ***POSTROUTING*** - catena in cui vengono processati i pacchetti in uscita dopo che il sistema ne abbia deciso l'instradamento tramite le tabelle di routing. Usata per il Nat sulla sorgente o SNAT.
- ***OUTPUT*** - catena attraverso la quale passano i pacchetti generati e uscenti dal sistema.

Come appena accennato, Iptables suddivide il NAT in due tipi:

- **Source NAT** o **SNAT** – si ha quando il NAT riscrive l'indirizzo sorgente di un pacchetto. Opera in fase di PostRouting, ovvero subito dopo che il pacchetto è stato elaborato dal processo di routing e pronto per essere immesso in rete. Il mascheramento o masquerading è una tecnica SNAT che serve a mascherare una rete privata che si trova dietro un unico indirizzo IP esterno.
- **Destination NAT** o **DNAT** – si ha quando si riscrive l'indirizzo IP di destinazione del pacchetto IP. Opera subito prima che il pacchetto venga elaborato dal processo di routing, quindi in PreRouting. Port-Forwarding e Trasparent-Proxy sono tecniche DNAT.

Per esempio, per modificare l'indirizzo IP sorgente dei pacchetto uscenti dalla rete LAN di prova, sarebbe necessario inserire la regola seguente:

```
# iptables –t nat –A POSTROUTING –o eth0 –j SNAT
                                --to-source 213.82.177.140
```

Quanto appena mostrato è un esempio di SNAT. La catena usata è, infatti, POSTROUTING. I pacchetti uscenti prendono l'indirizzo 213.82.177.140 come indirizzo sorgente.

Nel caso di una connessione a internet con IP dinamico e non statico, è meglio utilizzare la seguente regola:

iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

Come è facile notare, non viene specificato un indirizzo IP con cui cambiare l'effettivo indirizzo sorgente. Il sistema prende automaticamente l'indirizzo fornito di volta in volta dal provider dei servizi internet, e lo utilizza per lo SNAT.

Per fare un esempio di DNAT, si ipotizzi di voler fare in modo che ogni pacchetto entrante sull'interfaccia eth0 acquisisca l'indirizzo di destinazione 192.168.69.100. Occorrerebbe la seguente regola:

```
# iptables –t nat –A PREROUTING –i eth0 –j DNAT
                              --to-destination 192.168.69.100
```

Altro esempio, redirect dei pacchetti destinati all'indirizzo 192.168.69.25 verso l'indirizzo 192.168.69.37.

```
# iptables –t nat –A PREROUTING –d 192.168.69.25 –j DNAT
                              –to-destination 192.168.69.37
```

Anche prese a stomaco vuoto, queste regole risultano già abbastanza chiare. Comunque sia nelle seguenti pagine viene spiegato dettagliatamente il formato delle regole e le relative opzioni che si possono utilizzare per la loro scrittura.

La tabella ***Mangle*** ha il compito di permettere la modifica di varie opzioni relative ai pacchetti. Non viene usata spesso. Essa contiene tutte le catene prestabilite sopra citate.

Appena installato il pacchetto iptables è subito possibile definire delle regole attraverso l'utilizzo del terminale (shell).

Le regole

Il formato di una tipica regola per iptables è il seguente:

```
iptables [ -t <nome tabella> ] <comando> <nome catena> <parametro1>
\ <opzione1> … <parametro-n> <opzione-n>
```

Come è possibile notare, il comando per impostare le regole è iptables.

- L'opzione –t permette di specificare la tabella a cui la regola va applicata. Se omessa, viene usata di default la tabella FILTER.

- <comando> permette di stabilire cosa iptables deve fare. Esistono diversi comandi a disposizione dell'utente, i più importanti sono:
    - -A per aggiungere una regola alla tabella/catena in questione.
    - -R per rimpiazzare una regola esistente
    - -D per eliminare una regola esistente
    - -L per listare le regole esistenti
    - -F per eliminare tutte le regole contenute nella catena specificata. Se non viene indicata una catena in particolare allora vengono eliminate tutte le regole in tutte le catene.
- <nome catena> permette di delineare la catena alla quale la regola va applicata.

Seguono poi coppie di parametri e opzioni. Ad esempio:

- -p <protocollo> permette di definire il protocollo dei pacchetti. La regola verrà applicata a tutti i pacchetti di quel protocollo.
- -i <interfaccia di rete> permette di definire l'interfaccia di rete per i pacchetti in ingresso.
- -o <interfaccia di rete> permette di definire l'interfaccia di rete per i pacchetti in uscita.
- -s <sorgente> specifica i pacchetti provenienti da una determinata sorgente. Tale sorgente può essere specificata tramite indirizzo IP, interfaccia di rete o nome DNS.
- -d <destinazione> specifica i pacchetti diretti verso una determinata destinazione. La specifica della destinazione è analoga a quella per la sorgente.

Molto importante è l'ultima coppia. Si tratta dell'**obiettivo (**o **target)** della regola. Esso definisce l'azione da compiere nel caso in cui un pacchetto risponda alle caratteristiche definite nella regola.

L'obiettivo viene specificato con l'opzione –j e prevede uno tra i seguenti parametri:

- **ACCEPT** – Il pacchetto viene accettato e può procedere verso la sua destinazione.
- **DROP** – il pacchetto viene scartato e non potrà raggiungere la sua destinazione. Non viene mandata alcuna notifica al mittente.

- **QUEUE** – il pacchetto viene messo in una cosa. Questo parametro è molto importante. Fa in modo che i pacchetti messi in coda possano essere processati e analizzati da una applicazione. E' il caso, ad esempio, di Snort avviato in modalità *inline*.
- **REJECT** – il pacchetto viene scartato e viene mandata una notifica al mittente. E' il caso delle notifiche ICMP port-unreachable.
- **DNAT** – viene modificato l'ip di destinazione del pacchetto. Disponibile solo nella tabella nat e nelle catene PREROUTING e OUTPUT.
- **SNAT** – viene modificato l'ip sorgente del pacchetto. Disponibili solo nella tabella nat e nella catena POSTROUTING.
- **MASQUERADE** – simile a SNAT, si usa quando i pacchetti escono da un'interfaccia con ip dinamico ad esempio dhcp, dialup, dsl. Si usa solo nella catena POSTROUTING della tabella nat.
- **REDIRECT** – redirige il pacchetto a una porta locale. Usabile nelle catene PREROUTING e OUTPUT della tabella nat. Usato solitamente per realizzare un proxy trasparente, nel caso in cui la macchina con iptables stia eseguendo un server proxy.
- **RETURN** – Interrompe l'attraversamento della catena e segue la policy di default.
- **TOS** – Previsto solo nella tabella mangle. Permette di cambiare il TOS (Type Of Service) del pacchetto.
- **MIRROR** – Ha l'effetto di produrre una copia dei pacchetti da rimandare al mittente.
- **LOG** - usato per loggare il pacchetto via syslog.

Monitoraggio delle connessioni

Iptables offre, tra le altre, una importante funzionalità: il monitoraggio delle connessioni. E', dunque, in grado di riconoscere i pacchetti di una stessa connessione (stateful firewall). Questo permette di creare delle regole per i pacchetti, in relazione alla connessione a cui appartengono. Il Nat usa questo meccanismo per tradurre correttamente gli indirizzi dei pacchetti che appartengono alla stessa connessione.

Questa funzionalità viene offerta e gestita dal modulo *ip_conntrack* (ip connection tracking). Tale modulo è capace di tenere traccia delle connessioni tra gli host di una rete. Ha la capacità di eseguire il tracking dei protocolli TCP, UDP e ICMP.

Ad ogni pacchetto viene assegnato uno tra i seguenti stati:

- **NEW** – per il primo pacchetto che inizia una nuova connessione.
- **ESTABLISHED** – per i pacchetti che fanno parte di una connessione già stabilita.
- **RELATED** – per i pacchetti che in qualche modo sono correlati a connessioni già stabilite. Tipico delle connessioni FTP.
- **INVALID** – per i pacchetti che non rientrano in nessuna delle categorie appena dette e che solitamente vengono scartati.

Netfilter lavora in spazio kernel. Questo gli consente di monitorare in tempo reale lo stato delle connessioni. E' possibile esaminare in tempo reale il lavoro svolto dal modulo ip_conntrack tramite il seguente comando:

```
# tail –f /proc/net/ip_conntrack
```

E', inoltre, possibile decidere quante connessioni devono essere gestite da netfilter, modificando il valore contenuto all'interno del file:

```
/proc/sys/net/ipv4/ip_conntrack-max
```

Il programma

Una volta installato, iptables deve essere eseguito in modalità demone. Viene creato un apposito servizio che è possibile avviare, fermare e riavviare tramite il seguente comando:

```
# service iptables start | stop | restart
```

Si ottiene il medesimo effetto con:

```
# /etc/init.d/iptables start | stop | restart
```

E' importante tenere a mente questi comandi perché ogni volta che viene effettuata una modifica alle regole, occorre riavviare il servizio per rendere le modifiche effettive.

Ogni volta che iptables viene avviato, carica le regole dal file posizionato in `/etc/sysconfig/iptables`. Quest'ultimo è il file di configurazione di iptables. In esso sono contenute le regole.

Tutto questo si riferisce al modo in cui è organizzato il programma su una distribuzione CentOS 5. Bisogna fare spesso attenzione in quanto ci possono essere differenze tra una distribuzione e l'altra.

Configurazione

La configurazione del firewall può risultare ostica se effettuata da linea di comando. Per facilitare l'utente sono stati realizzati diversi tool di configurazione che dovrebbero facilitare il compito. Per lo più si tratta di interfacce grafiche che aiutano l'utente nella creazione di regole, tramite appositi form da riempire. L'utilizzo di questi strumenti aiuta molto l'amministratore in quanto non richiedono una profonda conoscenza della sintassi dei comandi di iptables. Tra i più usati è possibile trovare:

- Webmin
- Firewall Builder

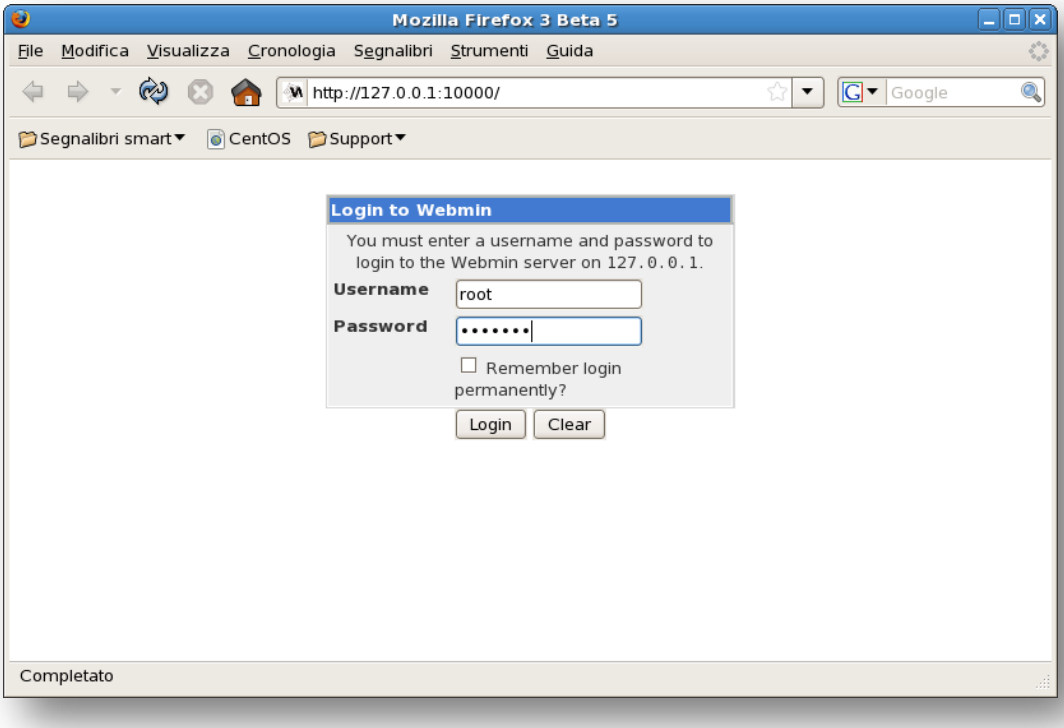
Webmin è un comodo centro di configurazione web-based, che aiuta a configurare diversi programmi, non solo il firewall. Questo tool racchiude un piccolo server web che consente all'utente di interagire tramite un comune browser. Si tratta di un configuratore molto famoso ed utilizzato e, ovviamente, open source.

Firewall Builder, invece, è specializzato nella configurazione di firewall. Oltre a supportare iptables/netfilter, consente di interagire anche con altre soluzioni come ipfilter, ipfw, OPENBSD PF, Cisco Pix. Questo lo rende uno strumento molto potente e molto utile a chi deve lavorare con firewall su diverse piattaforme.

Webmin

Essendo un tool capace di fornire supporto alla configurazione di moltissimi aspetti del sistema, webmin è stato scelto per la realizzazione della rete di test e dei relativi servizi attivati. Di seguito è possibile vedere un esempio di creazione di una regola per il firewall.

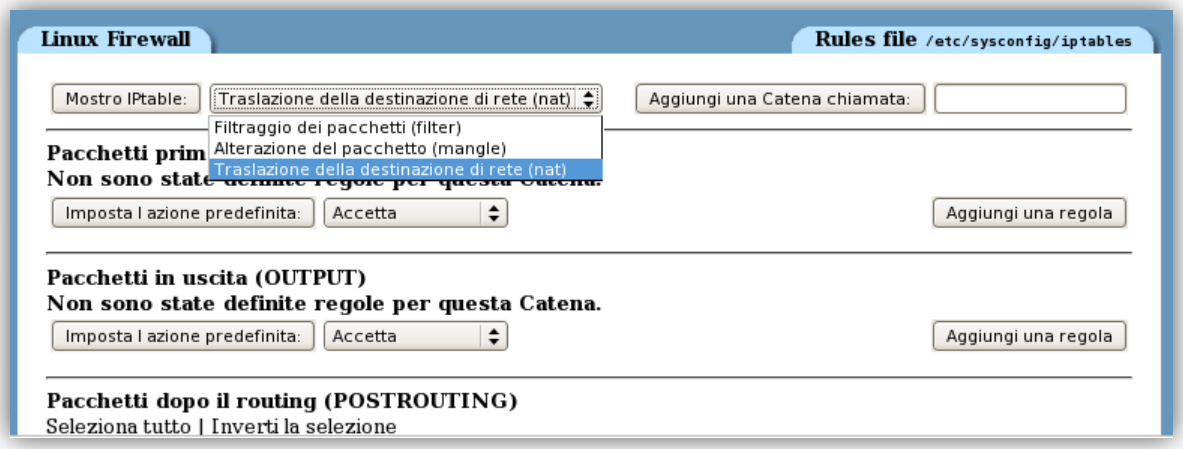
Innanzi tutto la schermata di login:



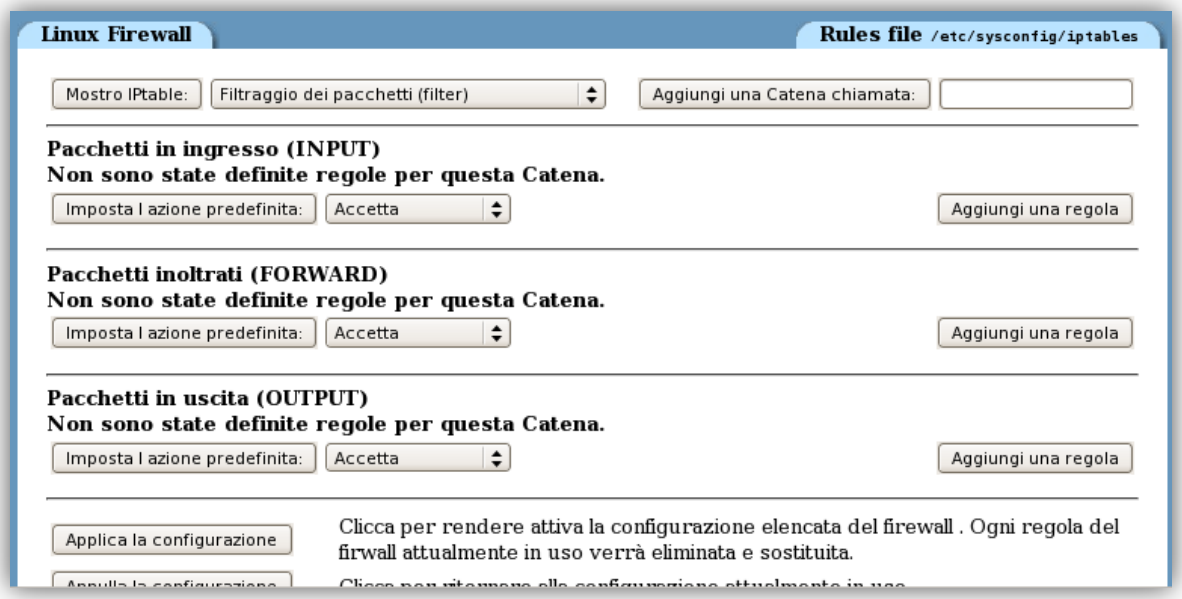
Viene richiesta l'autenticazione. Essendo un sistema di prova e per motivi di praticità, si è scelto di usare sempre l'utente root in modo da non avere problemi di permessi.

Nella sezione *Rete* è presente un collegamento che porta alla pagina di configurazione di iptables.

Dopo aver fatto click sul collegamento ci si troverà di fronte una pagina di questo genere:



Il primo pulsante permette di scegliere la tabella da visualizzare ed, eventualmente, modificare. Nell'esempio sopra riportato viene presa in esame la tabella filter, in quanto risulta tra le più utilizzate. Nella pagina relativa a tale tabella sarà possibile visualizzare le tre catene standard con le relative regole.



E' possibile aggiungere una regola ad una catena semplicemente facendo click sul pulsante a destra. Per fare un esempio, si immagini di voler inserire una regola nella catena INPUT.

Facendo click sul pulsante “Aggiungi una regola” si giunge alla seguente schermata.

Per aggiungere la regola desiderata è sufficiente riempire opportunamente i campi mostrati in figura. E’ facile notare una ampia flessibilità nella creazione delle regole, dato il numero di azioni e criteri che è possibile scegliere. Non ci sono campi obbligatori. E’ bene, però, scrivere un commento nel campo apposito per dare una maggior visibilità alla regola. Tutto questo va, ovviamente, a vantaggio dell’amministratore.

Ciò che viene richiesto immediatamente è il target della regola, ovvero l’azione da intraprendere nel caso in cui un pacchetto rispecchi i criteri stabiliti per quella regola.

Nella sezione sottostante è possibile definire le condizioni che devono essere verificate affinché l’azione scelta sia efficace. Nell’immagine successiva viene riportato un esempio. Si è scelto di respingere tutti i pacchetti provenienti dall’indirizzo IP 192.168.69.100.

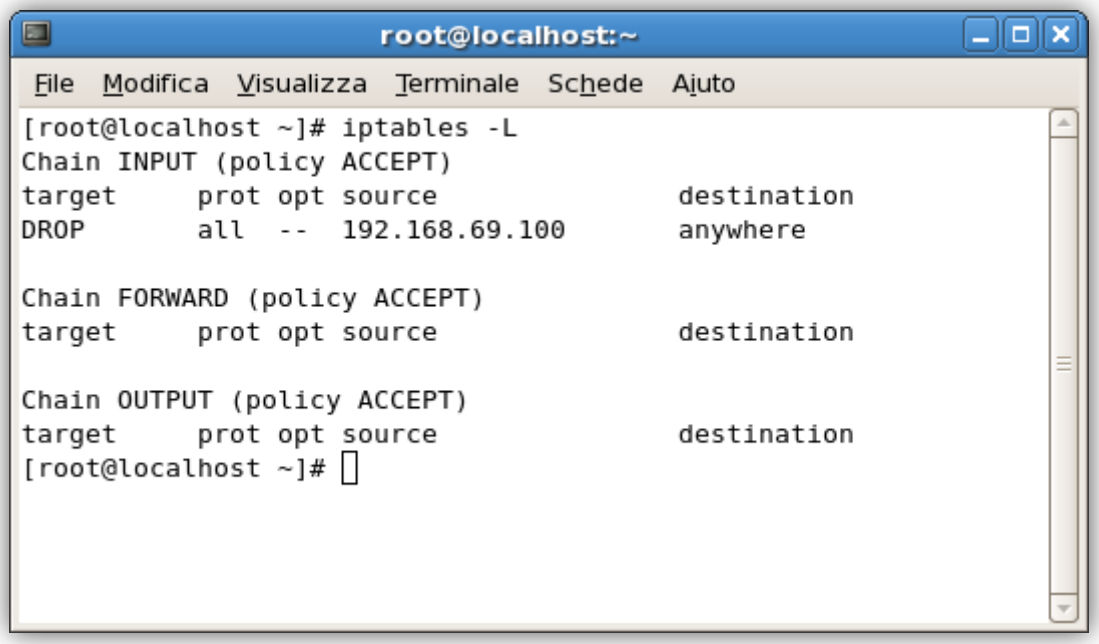
E’ sufficiente riempire i campi in questo modo.

Il risultato sarà visibile nella seguente schermata che racchiude tutte le regole per ogni catena.

Per una ulteriore verifica è possibile utilizzare il comando:

```
iptables -L
```

il quale produrrà il seguente output:



Da questo momento tutti i pacchetti provenienti da 192.168.69.100 saranno scartati.
Come precedentemente mostrato, è possibile utilizzare il menù a tendina per selezionare la tabella a cui apportare le dovute modifiche. La tabella NAT si presenta in maniera simile alla tabella FILTER. Vengono esposte le catene PREROUTING, OUTPUT e POSTROUTING.

Da notare la presenza della regola per il mascheramento degli indirizzi. Essa corrisponde a:

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

La tabella *mangle* si presenta nel seguente modo:

Nelle schermate mostrate è presente un apposito strumento per la creazione di catene custom. Creando la catena “esempio” nella tabella filter, si avrebbe il seguente risultato:

Firewall Builder

Firewall Builder è uno strumento di configurazione per firewall molto potente. Supporta le seguenti soluzioni firewall:

- Iptables/netfilter
- Ipfilter
- Pf
- Ipfw
- Cisco PIX
- Access List dei router Cisco

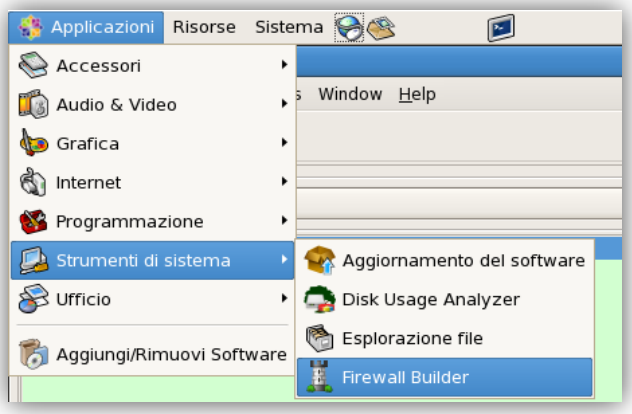
A differenza di Webmin, è possibile interagire con Firewall Builder senza l'utilizzo di un browser in quanto l'interfaccia grafica non si basa su server web. Inoltre è possibile configurare il firewall anche di altre macchine tramite una connessione ssh.

Si tratta, come ovvio, di software libero, di conseguenza l'installazione è possibile compilando il codice sorgente. Sono disponibili in rete i pacchetti precompilati. Per quanto riguarda le distribuzioni Red Hat è possibile usufruire dei pacchetti rpm che si possono trovare nel repository Dag Wieers (http://dag.wieers.com). In questo modo l'installazione del programma risulta semplice e immediata.

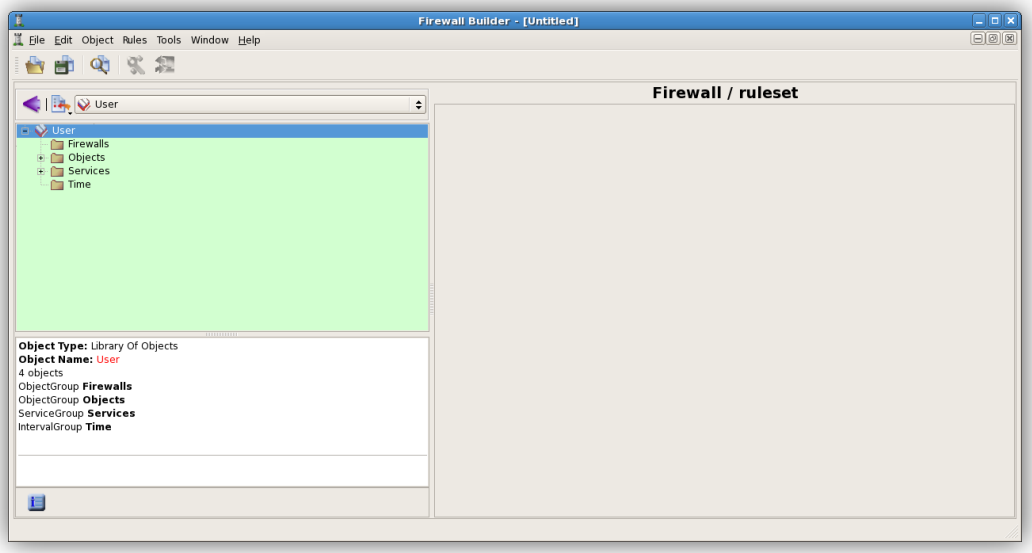
Il comando da utilizzare è:

```
#rpm -i fwbuilder-*VERSIONE*.rpm
```

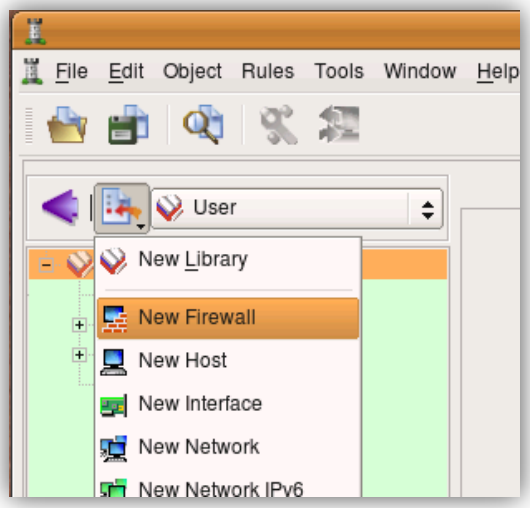
L'icona di avvio del programma verrà inserita nel menù "Applicazioni", sotto il gruppo "Strumenti di Sistema".



All'avvio dell'applicazione, ci si troverà di fronte a una schermata del tutto simile alla seguente:

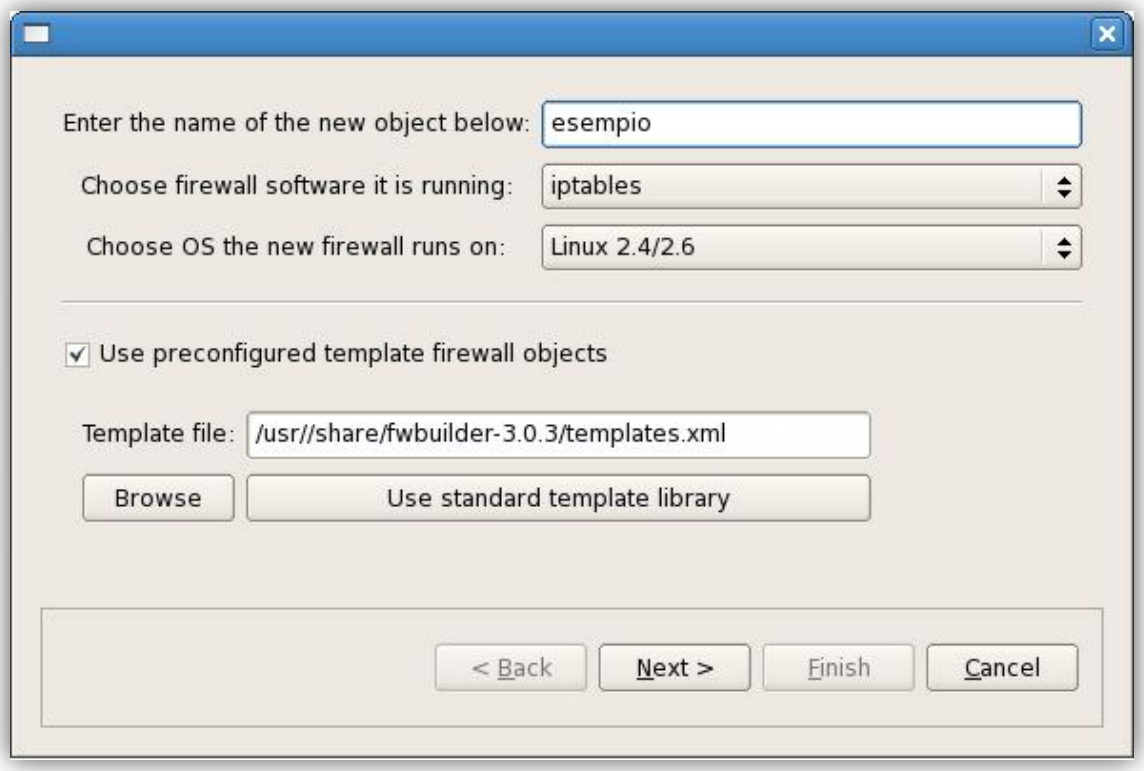


Sono ben visibile due colonne. Quella di sinistra serve per la visualizzazione ad albero di tutti gli oggetti (firewall, host, indirizzi, servizi ecc…). Quella di destra, invece, serve per la configurazione vera e propria del firewall. E' contemplato l'utilizzo del *Drag and Drop* per rendere l'utilizzo il più intuitivo possibile.



Il funzionamento del programma è grosso modo il seguente. Inizialmente è necessario creare una configurazione per un particolare firewall tra quelli supportati. Questa operazione è molto semplice in quanto guidata da un'apposita finestra. Tutto ciò che viene richiesto è il tipo di firewall da utilizzare e un nome qualsiasi con cui identificarlo.

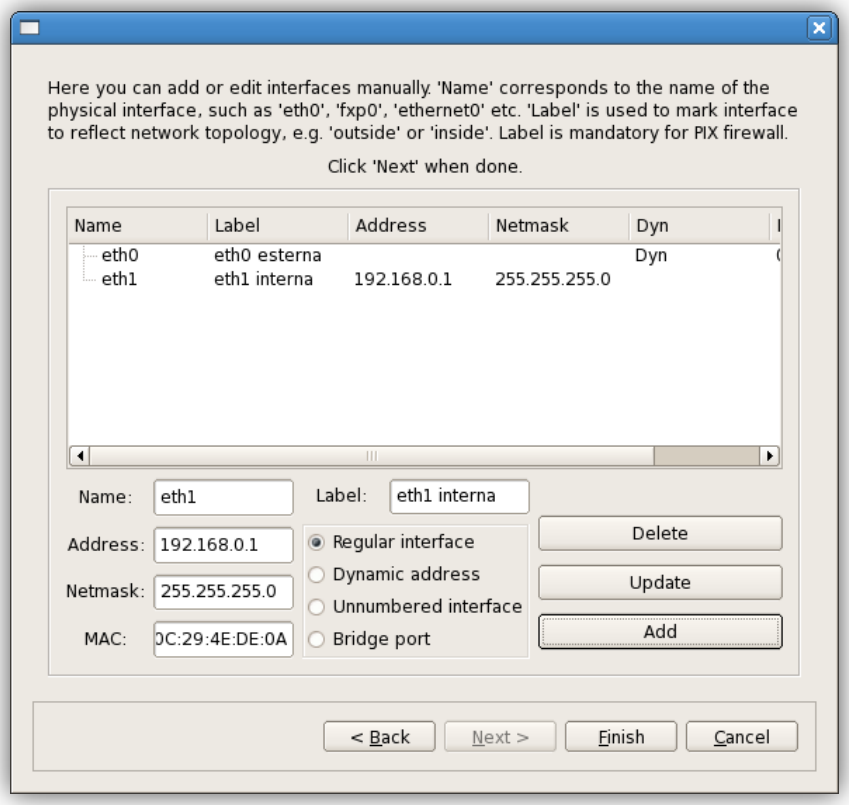
La finestra per l'inserimento delle uniche informazioni richieste è la seguente:



Al fine di aiutare l'utente, Firewall Builder offre diversi template standard. Selezionando la voce "Use preconfigured template firewall objects", viene concessa la possibilità di scegliere tra diversi modelli prestabiliti:

- Due schede di rete. Eth0 come interfaccia esterna e eth1 come interfaccia interna.
- Tre schede di rete. Eth0 come interfaccia esterna, eth1 come interfaccia interna ed eth2 per una sottorete DMZ.
- Una scheda di rete. Eth0 per la connessione alla rete esterna. E' il caso in cui l'unica macchina da proteggere è quella su cui girano Firewall Builder e il firewall.
- Una configurazione di esempio per router Cisco.
- Una configurazione di esempio per Web-server.
- Linksys Firewall.

Alternativamente è possibile scegliere di non utilizzare modelli prestabiliti e creare manualmente una configurazione da zero, come fatto per l'esempio seguente.



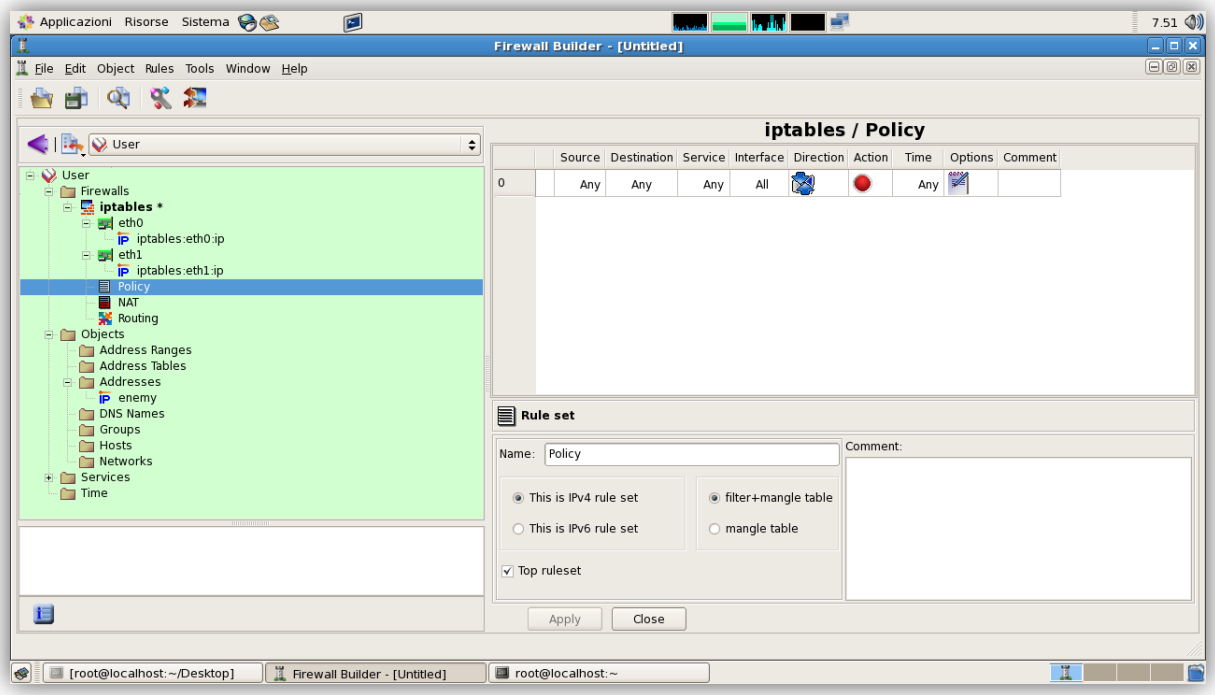
A questo punto è possibile cominciare a personalizzare la propria configurazione. E' bene ricordare che tutte le modifiche che vengono apportate non hanno immediatamente effetto. Esse verranno applicate solo dopo aver effettuato due operazioni fondamentali:

- Compilazione
- Installazione

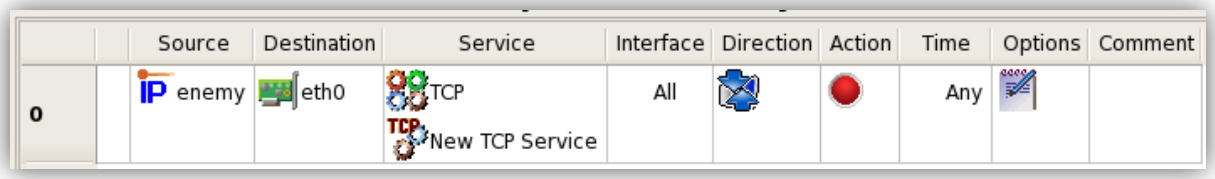
Con la prima operazione, Firewall Builder crea degli script che serviranno ad applicare le regole al firewall. La seconda, serve, appunto, per rendere le modifiche effettive. Questa può essere effettuata a mano oppure tramite l'interfaccia del programma. E' sicuramente preferibile, in quanto più rapido e semplice, scegliere quest'ultima soluzione.

Ad esempio, si immagini di voler creare una regola che blocchi il traffico TCP sulla porta 80 proveniente dall'indirizzo 151.1.245.1 e diretto all'interfaccia di rete eth0 con indirizzo 213.82.177.140. E' possibile procedere nel seguente modo.

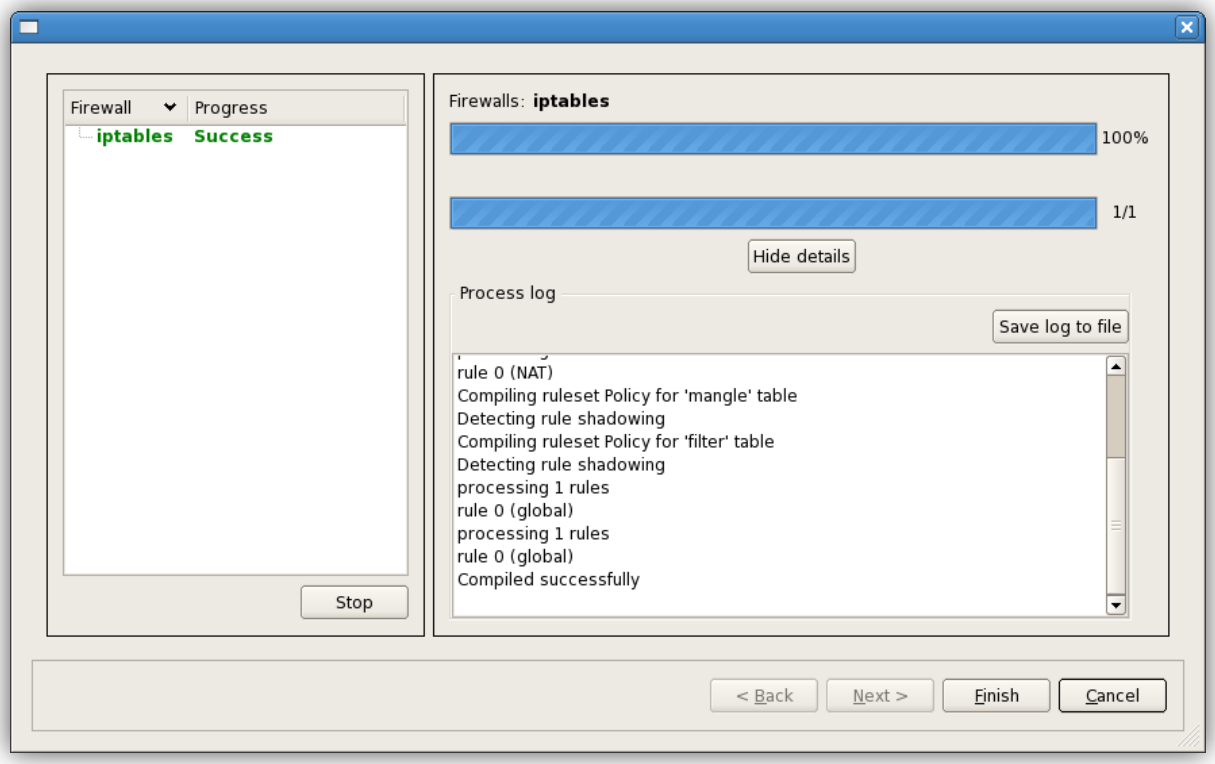
Inserire una nuova regola. Per fare questo, fare click sulla voce “insert rule” presente nel menù “Rules”.



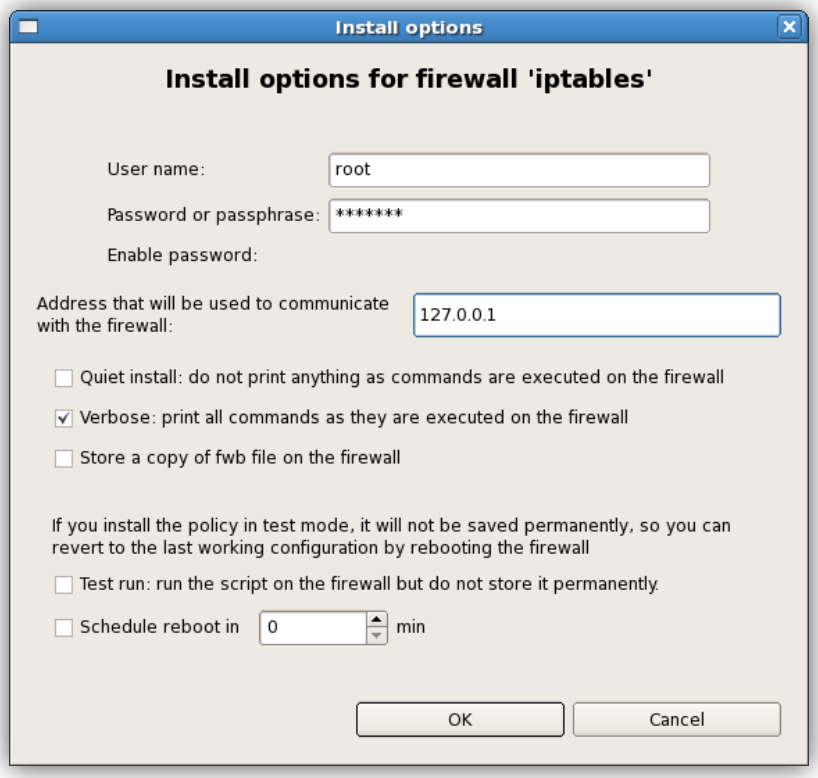
Creare l’indirizzo 151.1.245.1 nella sezione addresses e, tramite drag and drop, trascinarlo nel campo *source* della regola appena aggiunta. Nell’esempio, l’indirizzo creato viene identificato con l’etichetta “enemy”. Trascinare poi l’interfaccia eth0 nel campo *destination*. Per specificare il protocollo TCP è necessario creare un nuovo servizio TCP che specifica il traffico diretto alla porta 80. Anche in questo caso procedere con un semplice Drag and Drop. Si dovrebbe avere un risultato simile al seguente:



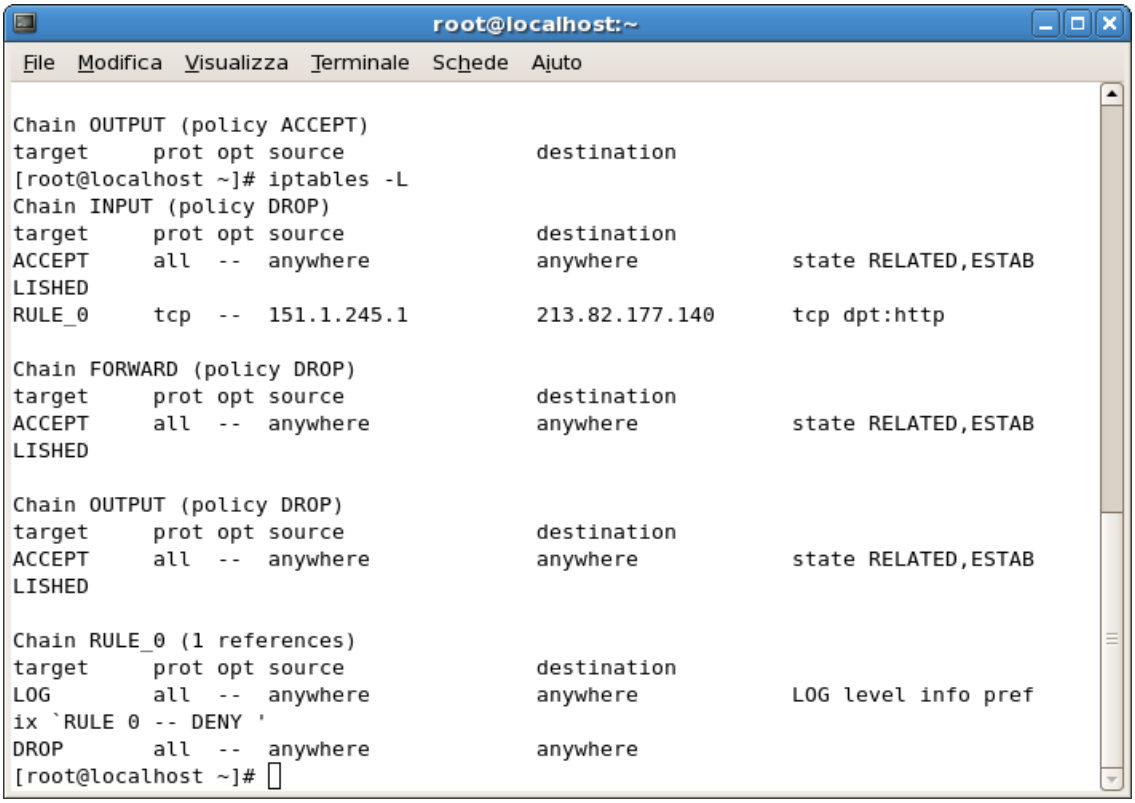
A questo punto è possibile compilare ed installare la regola definita.



Per rendere effettive le modifiche è necessario comunicare a Firewall Builder le credenziali con cui svolgere le operazioni e l'indirizzo della macchina alla quale effettuare la connessione ssh.



La finestra di installazione è del tutto simile a quella di compilazione e non verrà mostrata. Dopo aver applicato le modifiche, sarà subito possibile verificarle tramite l'apposito comando.



Regole applicate al sistema di prova

Come è possibile intuire dal titolo del paragrafo, verranno mostrate le principali regole applicate al sistema di prova. Si tratta di regole di utilizzo molto comune. Riguardano servizi particolarmente diffusi e utilizzati.

Per iniziare, si intende bloccare il ping sulla macchina. In particolare, non deve essere possibile effettuare ping dall'esterno. Questo potrebbe rendere impossibile qualche tipologia di scan, anche se ce ne sono altre, più raffinate, che non incontrano problemi anche con questo accorgimento.

Comunque, per fare questo è necessario bloccare i pacchetti di tipo **ICMP**. La più semplice delle regole che si possono utilizzare per raggiungere lo scopo è la seguente:

```
# iptables –A INPUT –p icmp –j DROP
```

In questo modo verranno scartati tutti i pacchetti ICMP ricevuti e il ping verrà bloccato a prescindere dalla provenienza. Risulta, infatti, impossibile verificare lo stato attivo o no della macchina dalla rete esterna, interna e perfino dalla macchina stessa (ping localhost). E' consigliabile aggiungere delle opzioni. Prima fra tutte, l'interfaccia di rete.

```
# iptables –A INPUT –i eth0 –p icmp –j DROP
```

Specificando questo parametro facciamo in modo che non possa essere effettuato ping solamente dall'esterno. Le macchine che compongono la rete interna potranno ancora verificare che il server sia attivo, in quanto i loro pacchetti icmp raggiungeranno l'interfaccia eth1.

Ping è un tool di rete molto utile. In molti vorrebbero non farne a meno. E' comunque possibile utilizzare delle regole personalizzate per poter evitare un uso sbagliato di questa risorsa. Ad esempio si potrebbe limitare la lunghezza dei pacchetti ICMP per evitare tentativi di *ping flood*.

Di default, la lunghezza dei pacchetti è impostata su 84 byte. Il traceroute di Windows è impostato su 92 byte. Di conseguenza si potrebbe limitare la lunghezza dei pacchetti a 93 byte.

```
#iptables –A INPUT –i eth0 -p ICMP --icmp-type 8 -m length --
length 93: -j DROP
```

In questo modo si fissa la lunghezza di 92 byte per i pacchetti permessi. I pacchetti con lunghezza superiore verranno scartati. Per rendere la regola di più semplice comprensione, qui di seguito ne vengono illustrate le opzioni utilizzate.

- --icmp-type - serve per specificare il tipo di pacchetto ICMP. Esistono diversi tipi e vengono distinti da un codice. In questo caso è stato utilizzato il codice 8 che si riferisce ai pacchetti di tipo echo request.
- -m lenght – l'opzione –m corrisponde a match. Serve per specificare un criterio. Nel caso in esame il criterio è la lunghezza del pacchetto. "lenght", infatti, si riferisce al match module dedicato al riconoscimento della lunghezza dei pacchetti analizzati.
- --lenght 93 – è l'opzione relativa al match module lenght. Permette di stabilire la lunghezza dei pacchetti. In questo caso la lunghezza è stata fissata a 93.

Da questo momento in poi, le macchine della rete interna potranno effettuare ping sulla macchina server senza restrizioni. Dall'esterno invece, sarà possibile solo con pacchetti di lunghezza minore a 93 byte.

Parlando di altri servizi, normalmente vengono lasciate aperte le seguenti porte:

- 80 per il traffico HTTP.
- 443 per il traffico HTTPS.
- 53 per i DNS.

Le corrispondenti regole sono:

```
# iptables –A OUTPUT –p tcp –o eth0 –dport 80 –j ACCEPT
# iptables –A OUTPUT –p tcp –o eth0 –dport 443 –j ACCEPT
# iptables –A OUTPUT –p tcp –o eth0 –dport 53 –j ACCEPT
# iptables –A OUTPUT –p udp –o eth0 –dport 80 –j ACCEPT
```

A questo punto, alla macchina sarà concesso di inoltrare richieste DNS e di navigare in internet con protocollo HTTP e HTTPS.

Un altro protocollo generalmente molto utilizzato è FTP. File Transfer Protocol è un protocollo per il trasferimento di dati. Qui le cose si fanno leggermente più complesse di quanto visto fino ad ora.

Il protocollo FTP inizialmente richiede che sia stabilita una connessione FTP tra server e client. Ciò avviene tramite una richiesta inoltrata dal client al server, sulla porta 21. Il server riceve la richiesta e accetta la connessione. La macchina che fa da firewall, quindi, dovrà accettare le connessioni provenienti dagli host della rete interna e inoltrarle al server FTP esterno.

```
# iptables –A FORWARD –s 192.168.69.0/24 –p tcp --dport ftp –j ACCEPT
```

Ora è necessario inserire una regola per fare in modo che il firewall accetti pacchetti FTP dal server esterno e diretti agli host della rete interna. In questo caso viene usata un'opzione particolare per evitare richieste di connessioni dall'esterno.

```
# iptables –A FORWARD –s 0/0 –p tcp ! --syn --sport ftp –j ACCEPT
```

Ora è arrivato il momento di gestire la connessione per il trasferimento dei dati. FTP supporta due modalità: attiva e passiva. La prima è quella che crea più problemi. Il client stabilisce indirizzo e porta sui quali deve avvenire la connessione e le comunica al server unitamente al comando PORT. Questo porta ad una situazione problematica. Il server riceve indirizzo e porta a cui connettersi per effettuare il trasferimento dati. Se il client è protetto da un firewall, spesso quest'ultimo rifiuta la connessione. Per regolare questa situazione si può agire nel seguente modo:

```
# iptables –A FORWARD –s 0/0 –p tcp --sport ftp-data –d 192.168.69.0/24
--dport 1024: -j ACCEPT

# iptables –A FORWARD –s 192.168.69.1 –p tcp ! --syn --sport 1024: -d 0/0
--dport ftp-data –j ACCEPT
```

La porta ftp-data corrisponde alla porta 20. Viene utilizzata per il trasferimento dei dati FTP in modalità attiva. E' una convenzione introdotta al fine di limitare i rischi derivanti da questa modalità. La porta di destinazione è solitamente una porta ad elevata numerazione. Per indicare questo viene utilizzata la sintassi 1024: che sta a significare una porta compresa tra 1024 e 65535.

La modalità passiva, invece, non crea nessun problema. In questo caso è il server a stabilire indirizzo e porta. Il client invia un pacchetto con il comando PASV. Il server risponde con le informazioni necessarie perché la connessione avvenga. Il client a questo punto richiede la connessione al server. In questo caso la richiesta di connessione è dall'interno verso l'esterno, e il firewall non rifiuta nulla.

```
# iptables –A FORWARD –s 192.168.69.0/24 –p tcp --sport 1024: -d 0/0
--dport 1024: -j ACCEPT

# iptables –A FORWARD –s 0/0 –p tcp ! --syn --sport 1024: -d
192.168.69.0/24 --dport 1024: -j ACCEPT
```

E' preferibile utilizzare la modalità passiva in quanto comporta meno rischi. E' bene ricordare che i browser internet usano questa modalità di default, mentre molti client FTP usano la modalità attiva.

Nel caso del client FTP Linux è possibile passare dalla modalità attiva a quella passiva molto semplicemente:

```
# ftp
ftp > passive
passive mode on.
ftp >
```

Il file di configurazione del firewall generato da iptables-save è presente nell'appendice.

Capitolo 4

Intrusion Detection and Prevention System

Come anticipato in precedenza, un Intrusion Detection System è un sistema che permette di analizzare il traffico di rete alla ricerca di eventuali intrusioni e attacchi informatici. E' un aspetto molto importante per la sicurezza di un'azienda, piccola o grande che sia.

I sistemi IDS forniscono un adeguato livello di sicurezza rispetto diversi tipi di attacchi. Sono in grado di riconoscere molte attività illecite, quali ad esempio:

- Portscan
- Malware
- Privilege escalation – exsploit
- Virus
- Trojan
- Dos
- Ddos
- E così via.

Più in generale, un attacco ha la finalità di accedere a dati protetti. Un accesso non autorizzato può compromettere tre aspetti importanti dei dati stessi:

- *Confidenzialità* – L'attacco mira a violare la segretezza dei dati.
- *Integrità* – l'attacco mira a modificare i dati rendendoli non veritieri o inservibili.
- *Disponibilità* – l'attacco mira a non rendere possibile l'accesso ai dati a chi è autorizzato.

Lo scopo di un IDS, e di qualsiasi altro sistema di sicurezza, è quello di impedire il verificarsi dei casi appena citati.

Normalmente un IDS ha il solo compito di notificare il verificarsi di intrusioni e/o attacchi all'amministratore della rete. Il suo scopo principale non è quello di prendere provvedimenti e contromisure. Tuttavia è possibile fare in modo che questo accada. Da qui sorge la distinzione tra IDS passivi e IDS attivi.

- Un IDS si dice *Passivo* se, al verificarsi di un attacco, produce solamente una voce di log o una notifica via mail.
- Un IDS si dice *Attivo* se, al verificarsi di un attacco, produce una notifica e prende delle contromisure nei confronti di tale attacco.

I sistemi IDS, facenti parte di quest'ultima categoria, provvedono ad aggiornare opportunamente le regole del firewall. In tal modo vengono scartati i pacchetti relativi alla connessione che trasporta l'attacco informatico. In realtà gli IDS attivi sono anche conosciuti con il nome di IPS o Intrusion Prevention System.

Normalmente, però, gli IDS funzionano solo come campanello di allarme. Al verificarsi si una intrusione o, più in generale, di un attacco, essi generano un cosiddetto *Alert*. Si tratta di un messaggio rivolto all'amministratore della rete volto a informare quest'ultimo dell'avvenuta violazione alla sicurezza. Esistono diversi modi in cui un IDS può comunicare questi allarmi e verranno mostrati successivamente più in dettaglio.

Il problema di ogni IDS è la generazione di Alert fasulli. Questi possono essere classificati in due categorie:

- *Falsi Positivi*: L'IDS genera un Alert anche se non si sono verificate attività anomale.
- *Falsi Negativi*: L'IDS non genera alcun Alert anche se in realtà è avvenuta una intrusione.

Sarebbe saggio configurare al meglio il proprio IDS in modo che non si verifichino né l'uno né l'altro. Tuttavia il caso peggiore è sicuramente quello dei Falsi Negativi, ovvero il verificarsi di intrusioni e rimanerne del tutto ignari. Non bisogna comunque sottovalutare la presenza di Falsi Positivi. Un loro numero troppo elevato potrebbe portare ad un cattivo uso di tutto il sistema IDS, se non proprio alla sua inutilità dovuta alla difficoltà di monitoraggio e di distinzione ciò che veramente è una minaccia. E' quindi molto importante provvedere al

cosiddetto *tuning* del sistema, ovvero configurarlo al meglio al fine di eliminare via via tutti i falsi positivi e negativi per arrivare ad avere un IDS attivo e funzionante. In questo caso l'IDS costituisce un grande vantaggio nei confronti di eventuali malintenzionati.

E' possibile classificare i sistemi IDS in tre categorie:

- Network IDS
- Host IDS
- Stack IDS

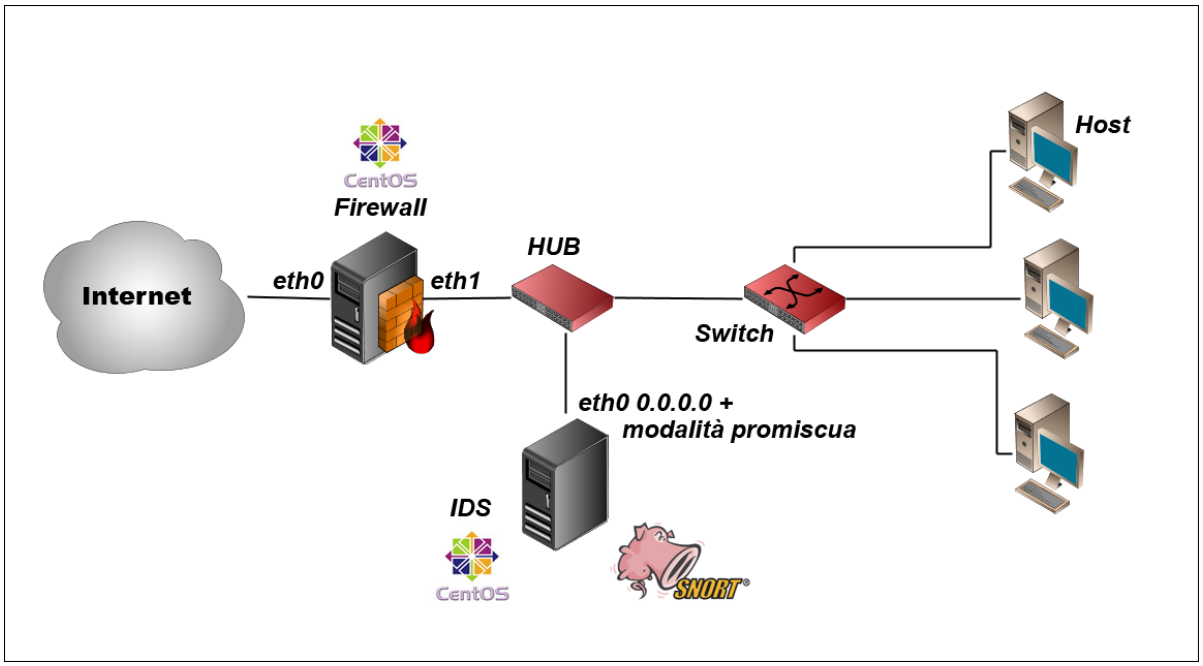
I Network IDS sono sistemi passivi di controllo. Il loro scopo è quello di intercettare il traffico di rete ed analizzarlo. Tale analisi avviene mediante regole o signatures.

Gli Host IDS, oltre a funzionare come Network IDS, controllano anche i dati presenti sulla macchina. In particolare si occupano di analizzare i file di log del sistema su cui sono installati in cerca di attività anomale.

La terza categoria, gli Stack IDS, controllano il traffico di rete a basso livello. Nello specifico agiscono a livello dello stack TCP/IP, decidendo quali pacchetti lasciar passare e quali no ancor prima che il sistema operativo e le applicazioni possano controllarne il contenuto.

Esistono diversi strumenti hardware e software che lavorano come un IDS. Tra i più noti e utilizzati è possibile trovare *Snort*. Esso rappresenta una validissima soluzione IDS open source che non ha nulla da invidiare a versioni commerciali.

Per quanto riguarda la rete di test, le prove sono state svolte tutte sulla stessa macchina, di conseguenza tutti i software, tra cui Snort, sono stati installati sullo stesso sistema. In questo modo, la macchina IDS, risulta del tutto visibile agli host interni e esterni. In realtà, un IDS non andrebbe proprio usato in questo modo. La maniera corretta di utilizzare tali sistemi sarebbe quella di adibire una macchina a tale compito e renderla invisibile tramite piccoli accorgimenti. Nella seguente immagine viene riportato un esempio di come dovrebbe essere strutturata la rete per un corretto utilizzo dell'Intrusion Detection System.



Come è possibile notare, vengono utilizzate due macchine oltre ai normali host interni. Una macchina che funge da gateway verso l'esterno, quindi con funzionalità di dhcp, dns, dynamic dns, firewall, e una macchina adibita ad IDS. Quest'ultima viene collegata alla rete tramite un semplice Hub. Come è noto, gli hub non smistano il traffico come invece fanno gli switch. Essi duplicano i pacchetti su tutti i canali a cui sono collegati. Spetta ai riceventi di tali pacchetti riconoscere se ne sono i destinatari o no. Questo consente alla macchina IDS di ricevere tutto il traffico di rete e sniffarlo.

Per rendere il sistema invisibile è necessario impostare l'interfaccia di rete della macchina in questione in modalità promiscua e senza IP.

Snort

Snort è un noto Intrusion Detection System. Si tratta di un software open source ormai affermato nel campo della sicurezza in rete. Gode di un'ampia portabilità in quanto è disponibile per piattaforme Unix e Windows. Esiste ormai da molto tempo ed è presente una vasta comunità che mantiene il progetto. Ora è ben chiaro il motivo del nome di dominio scelto per la rete di test.

Il software è disponibile sul sito di riferimento http://www.snort.org. Oltre ai normali pacchetti binari e sorgenti, sono disponibili molte estensioni per il programma. Si tratta di software aggiuntivo da abbinare a Snort per includere ulteriori funzionalità.

Snort è un IDS di tipo Network. E' in grado di sniffare e analizzare il traffico di rete. Per l'analisi, il software, sfrutta un flessibile linguaggio di regole. Tali regole servono a stabilire i criteri che definiscono un pacchetto malevolo.

In linea generale tutti i Network IDS pongono le interfacce utilizzate per la cattura dei pacchetti di rete in modalità promiscua. Non è necessario provvedere manualmente poiché è compito di Snort porre le interfacce di rete in promisc mode.

Il sistema utilizzato per far girare un Intrusion Detection System viene comunemente denominato *sensore*. In realtà, durante le prove effettuate in laboratorio, è stato riscontrato che su una stessa macchina possono essere presenti anche più sensori in base al numero di adattatori di rete di cui la macchina è munita. Nel caso in esame erano a disposizione due schede di rete, una per la rete esterna e una per la rete interna. Sono state avviate due istanze di Snort, una per interfaccia di rete. In questo modo il sistema presenta due sensori. Solitamente i sensori prendono il nome dall'indirizzo IP dell'interfaccia di rete di riferimento. E' stato facile verificare la presenza di due sensori grazie al supporto fornito dal database di log e alert di cui verrà parlato più avanti. E' possibile usare una base di dati per l'archiviazione degli allarmi generati dall'IDS. Tra le informazioni immagazzinate è possibile trovare anche i sensori.

Fondamentalmente, Snort viene fornito senza interfaccia grafica. Il suo utilizzo è principalmente previsto in modalità demone e trasparente all'utente. Per un miglior monitoraggio del suo funzionamento è consigliato l'utilizzo di uno dei noti front-end disponibili in rete.

Come accennato in precedenza, esiste anche un modulo per Webmin, che aiuta alla configurazione del noto IDS.

Funzionamento

Snort è un IDS basato su regole. Ha bisogno, infatti, di file contenenti direttive precise riguardo le caratteristiche dei pacchetti incriminati. Tali regole vengono fornite dai canali ufficiali che mantengono il programma. Ne esistono di diverse tipologie, più o meno aggiornate. Comunque sia sono in ogni caso necessarie per il corretto funzionamento del sistema di rilevamento. Sul sito ufficiale sono disponibili regole per ogni versione di Snort. Le release più aggiornate sono messe a disposizione solo degli utenti registrati, tuttavia tale registrazione non è obbligatoria, in quanto vi sono anche regole liberamente scaricabili da chiunque. Snort può essere paragonato ad un software antivirus. Si tratta di software che utilizzano regole e signatures per il controllo dei dati. Tali regole vanno aggiornate continuamente al fine di garantire un adeguato livello di protezione. Come ovvio, un software IDS o Antivirus non aggiornato non costituisce una buona barriera contro le insidie della rete.

Per i test in laboratorio, erano a disposizione poche macchine e non sono state effettuate prove particolari. Snort è stato installato su una sola macchina che controllava tutto il traffico entrante e uscente. Di norma sarebbe consigliabile utilizzare più macchine adibite al ruolo di IDS e disposte in punti diversi della rete. Questo in modo particolare in presenza di un numero elevato di host. Usare sistemi IDS diversi può portare un vantaggio anche dal punto di vista della raccolta degli alert. E' possibile, infatti, disporre questi sistemi in base alle necessità attivandovi diversi tipi di regole in base agli attacchi previsti sui rispettivi segmenti di rete.

Snort è pensato per funzionare in quattro diverse modalità:

- **Sniffer** mode: utile al fine di intercettare tutto il traffico di rete che successivamente sarà analizzato. Per lo sniffing dei pacchetti vengono utilizzate le librerie PCAP.
- **Packet Logger** mode: usato per loggare il traffico di rete.
- **NIDS** mode o **Netword Intrusion Detection** Mode: la funzionalità per cui Snort è nato. Effettua l'analisi del traffico sniffato in cerca di minacce alla sicurezza della rete.
- **Inline** mode: si può definire un'evoluzione della modalità precedente. Eseguito in inline mode, Snort provvederà a bloccare l'attacco oltre che a notificarlo. In questa

modalità il traffico non viene intercettato grazie alle librerie PCAP, ma grazie alla cooperazione con il firewall Iptables.

Per far funzionare Snort in inline mode è purtroppo necessario ricompilare il codice sorgente specificando l'abilitazione di questa modalità.

Tra il software opzionale, di cui è consigliabile fare uso, è bene citare il famoso programma *Oinkmaster*. Si tratta di uno script per l'aggiornamento automatico delle regole. Può sembrare banale data la sua semplicità e leggerezza, ma non è così. Come già anticipato è molto importante tenere le regole aggiornate e questo software aiuta di molto nel compito.

Esistono, inoltre, altri software che possono essere utilizzati insieme a Snort per offrire una migliore cooperazione con il Firewall. E' il caso, ad esempio, di soluzioni come *Guardian* o *FwSnort*.

Il primo ha il compito di tradurre le regole di snort in regole per il firewall iptables. In questo modo non appena giunge un pacchetto i cui requisiti coincidano con quelli specificati nelle regole, questo viene scartato dal firewall. Si tratta di un semplice script che, date in input le regole di Snort, produce un ulteriore script che, se eseguito, applica le relative modifiche alle catene del firewall. Il difetto di questo programma è che va eseguito manualmente o, al limite, tramite un'azione pianificata con cron. Probabilmente la soluzione migliore la offre Guardian.

Lo scopo di Guardian è lo stesso di FwSnort. La differenza sostanziale tra i due sta nella modalità di esecuzione dei programmi. Mentre FwSnort è uno script eseguito, al massimo, ad intervalli regolari, Guardian rimane attivo in modalità demone. Questo permette al programma di monitorare continuamente le attività di Snort. Non appena quest'ultimo genera un Alert, Guardian provvede a tradurre la regola relativa ed applicarne una opportuna al firewall. Da questo momento in poi, se si dovesse verificare un altro attacco dello stesso tipo, questo verrà immediatamente bloccato.

Non bisogna poi dimenticare il supporto al noto software antivirus ClamAV. Grazie alla cooperazione con quest'ultimo, Snort è in grado di segnalare anche la presenza, e quindi l'intrusione, di virus all'interno della rete che si intende proteggere. A tal fine è stato incluso, tra gli altri, un preprocessore apposito.

Logging

In generale, Snort è progettato per cooperare con diversi software. Questo al fine di rendere il suo utilizzo più intelligente e, ovviamente, migliore il monitoraggio delle attività sulla rete. In particolare è importante la possibilità di integrare il logging dei pacchetti e degli alert con un database.

Snort supporta diversi tipi di DBMS tra cui:

- MySQL
- Postgres
- MS SQL server

Questo permette di organizzare i dati e di renderli visibili in maniera migliore, soprattutto tramite l'utilizzo di software aggiuntivi, come quelli descritti nel prossimo paragrafo. Nel capitolo relativo al monitoraggio e al logging (Capitolo 6) viene citato il software Logsnorter. Si tratta di un piccolo programma per una migliore gestione del logging di Snort.

Interfacce grafiche

Come già anticipato, Snort non prevede un'interfaccia grafica. Esistono software appositi che forniscono supporto grafico a Snort. Tra i più utilizzati si possono trovare delle GUI web-based quali, ad esempio, i famosi Acid e Base. Queste sono interfacce che funzionano tramite un web server come Apache. Sono scritte in linguaggio PHP e garantiscono un miglior controllo sul traffico di rete, inoltre provvedono a creare statistiche e grafici sui dati raccolti.

Si tratta di interfacce molto diffuse, ma hanno il difetto di non fornire un aggiornamento in tempo reale degli alert e dei log. A coprire questa mancanza sono presenti altri tipi di software quali **Sguil**. Quest'ultima è un'interfaccia scritta in Tcl/Tk che non si appoggia ad un server web come le precedenti.

Capitolo 5

Proxy Server

Un Proxy server è uno strumento molto utile per gestire al meglio il traffico su una rete composta da un numero medio/alto di host. Offre la possibilità di migliorare le prestazioni della rete e di monitorare e controllare al meglio le attività. Quindi fornisce:

- Gestione intelligente del traffico con conseguente miglioramento delle prestazioni.
- Monitoraggio e Controllo delle attività.

Come già anticipato, su una rete aziendale circolano dati sensibili. Tali dati vanno protetti. Attività illecite possono minacciare le informazioni che si intende proteggere. Le minacce, nella maggior parte dei casi, arrivano dall'esterno. E' bene, però, controllare le attività degli host della rete interna.

Un uso non conforme a ciò che concerne le attività aziendali potrebbe portare al sorgere di vulnerabilità che potrebbero essere sfruttate per compiere violazioni alla sicurezza. Un proxy può fornire gli strumenti per definire cosa è concesso e cosa non lo è. Questo è importante ai fini della protezione, quindi, come già detto, per evitare l'insorgere di falle di sicurezza. Allo stesso tempo, consente di controllare l'operato dei dipendenti dell'azienda e impedire che utilizzino le strutture aziendali per fini non lavorativi.

Pur non disponendo di un elevato numero di macchine per la realizzazione della rete di test, è stato deciso di includere ugualmente un proxy server. Un'ottima soluzione open source è costituita dal diffusissimo **Squid** Server.

Squid

Squid è un popolare software libero che offre funzionalità di proxy e web cache. Molto popolare nel mondo Unix per il quale è stato pensato e progettato. E' principalmente un proxy HTTP e FTP , ma supporta numerosi altri protocolli. Inizialmente era disponibile per sistemi Unix. Allo stato attuale è possibile l'uso anche su piattaforme Windows.

Oltre ad aumentare le prestazioni della rete, permette di filtrare le richieste provenienti dai computer della rete interna. E', infatti, possibile decidere quali risorse sono consentite e quali no.

Installazione e Configurazione

Il software necessario è presente nel repository ufficiale della distribuzione in uso. E', quindi, possibile procurarsi il programma mediante il comando:

```
# yum install squid
```

La configurazione del server avviene mediante la modifica delle direttive presenti nel file di configurazione `/etc/squid/squid.conf`.

La prima cosa da fare è definire le reti con cui squid deve lavorare. Bisogna, quindi, dare l'accesso agli host della rete interna.

```
acl rete src 192.168.69.0/24
http_access allow rete
```

Un'altra cosa a cui fare attenzione è la porta di connessione al proxy, che di default è 3128. Normalmente non si ha bisogno di modificare questa impostazione, tuttavia viene lasciata la possibilità di scegliere su quale porta il server deve rimanere in ascolto.

Allo stato attuale il proxy è già pronto a funzionare. Con la configurazione corrente, però, è possibile usare il server solo come cache per la navigazione. Se l'intenzione fosse questa, allora non rimane altro che avviare il servizio con il seguente comando:

`# service squid start` o in alternativa:

`#/etc/init.d/squid start`

Sarebbe, inoltre, buona norma rendere il server avviabile al boot di sistema tramite il comando:

```
# chkconfig squid on
```

Sicurezza

Allo stato attuale, il proxy lavora solo come web cache. Non sono state definite regole per la sicurezza.

Squid garantisce la possibilità di intervenire in tal senso permettendo all'utente di impedire l'accesso a determinate risorse, che possono essere siti web e/o file. Questo consente di tutelarsi da utilizzi non consentiti della propria rete, in particolare da comportamenti illeciti provenienti dagli host rete interna.

Impedire l'accesso a determinati siti web

E' possibile bloccare l'accesso a siti web prestabiliti. Anche in questo caso bisogna intervenire nel file di configurazione squid.conf.

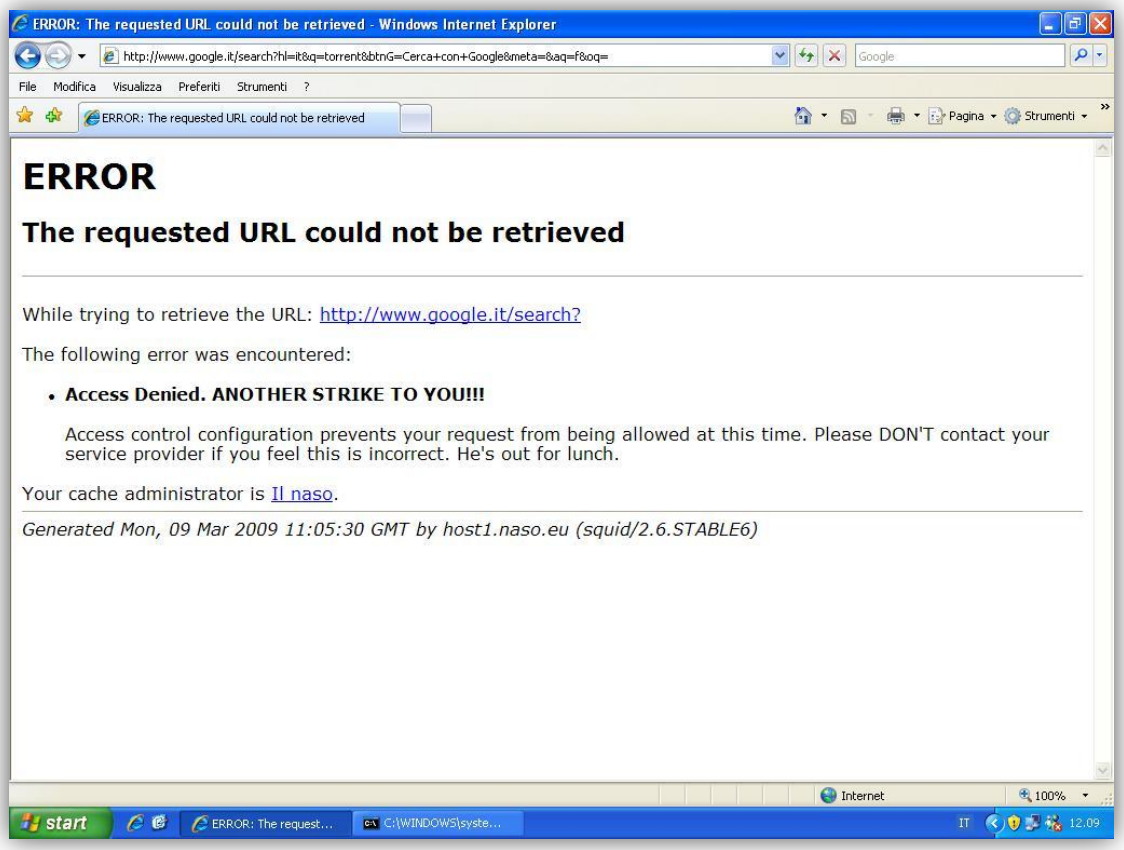
Si tratta di modificare la cosiddetta Access Control List, comunemente conosciuta come ACL. Per esempio, se volessimo inibire l'accesso al sito *www.esempio.com* dovremmo definire:

```
acl siti_bloccati dstdomain .esempio.it
http_access deny siti_bloccati
```

In maniera analoga è possibile bloccare le URL che contengano una determinata parola. Ad esempio, se si volesse evitare l'accesso ai siti che contengono la parola "esempio" bisognerebbe intervenire in questo modo:

```
acl url_bloccate url_regex -i esempio
http_access deny url_bloccate
```

Se da una macchina della rete interna si tentasse di accedere ad un sito bloccato, il server risponderebbe con la seguente pagina html di errore:



E' possibile personalizzare o aggiungere delle pagine di errore nel percorso `/etc/squid/errors/`. Nell'immagine appena mostrata vengono riportati i seguenti dati:

- Data e ora in cui è stato generato l'errore.
- Squid/numero versione
- URL che ha generato l'errore. Nell'esempio http://www.google.it/search?.
- Motivo per cui l'errore è stato generato. Nell'esempio ACCESS DENIED

Nell'esempio mostrato, era stato bloccato l'accesso ad URL contenenti la parola "torrent". E' bastato effettuare una ricerca sul motore di ricerca Google per generare questa pagina di errore. La scelta della parola "torrent" non è un caso. E' ben noto il problema che il p2p, soprattutto quello illegale, provoca alle reti pubbliche e private. Qualsiasi azienda vorrebbe tutelarsi dall'utilizzo della propria rete per effettuare scambi di file in rete.

Questo principalmente per due motivi:

- Evitare problemi di natura legale
- Evitare il sovraccarico della rete aziendale.

E' bene ricordare che ad ogni modifica apportata al file di configurazione è necessario riavviare il server affinché le modifiche abbiano effetto.

Bloccare il download di determinati tipi di file

Come appena detto, un amministratore potrebbe voler prendere dei provvedimenti contro il download di file non consentiti al fine di evitare alla compagnia problemi legali e sovraccarico della rete.

Squid consente di bloccare intere categorie di file semplicemente specificandone l'estensione. In particolar modo, sarebbe saggio impedire il download di file di rilevanti dimensioni, ad esempio i file multimediali.

Per fare questo è necessario, come prima cosa, creare un file in cui si definiscono le estensioni dei file da bloccare.

Al fine di testare questa funzionalità, sono stati seguiti i seguenti passi. Per prima cosa è necessario creare il file `block.files.acl`. E' sufficiente un qualunque editor di testo come emacs o vi.

```
# vi /etc/squid/block.files.acl
```

Questo file è organizzato in righe. Ogni riga deve contenere una estensione da bloccare. E' possibile usare la seguente sintassi.

```
\.[Ee][Xx][Ee]$
\.[Aa][Vv][Ii]$
\.[Mm][Pp][Gg]$
\.[Mm][Pp][Ee][Gg]$
\.[Mm][Pp]3$
```

In questo modo si definiscono le estensioni dei file di cui si vuole impedire il download.
E' il caso dei file exe, avi, mpg, mpeg e mp3.

A questo punto è necessario far conoscere a squid la posizione del nuovo file appena creato. Per fare questo bisogna modificare il file squid.conf. In particolare è necessario creare una nuova direttiva “acl” nell’omonima sezione.

```
acl file_bloccati urlpath_regex "/etc/squid/blocks.files.acl"
```

Ora, come fatto in precedenza:

```
deny_info ERR_BLOCKED_FILES blockfiles
http_access deny file_bloccati
```

In questo modo è stato negato l’accesso alle URL che puntano a file con estensione specificata in block.files.acl. L’errore visualizzato dipenderà dal contenuto del file ERR_BLOCKED_FILES. Tale contenuto, ovviamente, può essere modificato e personalizzato a proprio piacimento.

Dansguardian

Per un migliore filtraggio dei contenuti è possibile utilizzare DansGuardian. Si tratta di un software concepito per operare come servizio accessorio per il proxy Squid. Permette un’ampia configurabilità e ciò lo rende uno strumento molto versatile.

Il filtraggio è attuato usando molti metodi, quali l'analisi dell'URL e del dominio, del contenuto testuale, delle immagini, del contenuto MIME, dell'estensione del file ed è in grado di controllare liste anche molto grandi di domini.

Installazione

E’ possibile installare Dansguardian sia da sorgenti che da repository. In questo ultimo caso è necessario il repository DagWieers per distribuzioni Red Hat. In tal modo è sufficiente utilizzare il comando:

```
# yum install dansguardian
```

Da sorgenti invece, è necessario innanzi tutto procurarsi l'archivio compresso. Successivamente seguire i seguenti passi:

```
# tar xvzf <nome_pacchetto>
# cd dansguardian-*VERSIONE*
# ./configure
# make
# make install
```

Una volta installato, il servizio va avviato tramite il comando:

```
# /etc/init.d/dansguardian start
```

Ovviamente sono disponibili i parametri stop e restart, così come per tutti gli altri servizi.

Configurazione

I file di configurazione sono contenuti in /etc/dansguardian. Di default, dansguardian, è in ascolto sulla porta 8080. E' necessario, dunque, che gli host si connettano a questa porta per fare in modo di sfruttare i filtri di dansguardian.

E' possibile agire in due modi.

- Configurare i client per l'accesso alla porta 8080. Questo comporta l'impostazione della porta 8080 su ognuno dei client che compongono la rete interna.
- Configurare il server aggiungendo una regola a IPtables in modo da redirigere il traffico dalla porta 3128 alla porta 8080.

Nel primo caso, però, è bene evitare di lasciare la possibilità di connessione diretta al server proxy tramite la porta 3128. E' possibile prevenire questa eventualità semplicemente aggiungendo la seguente regola ad iptables:

```
# iptables -A INPUT -m tcp -p tcp -s !127.0.0.1 --dport 3128 -j DROP
```

In questo modo si provvede a scartare tutti i pacchetti diretti alla porta 3128 e che non provengano dalla macchina stessa.

Il file di configurazione principale è dansguardian.conf. Normalmente non occorre apportare modifiche alla configurazione di default.

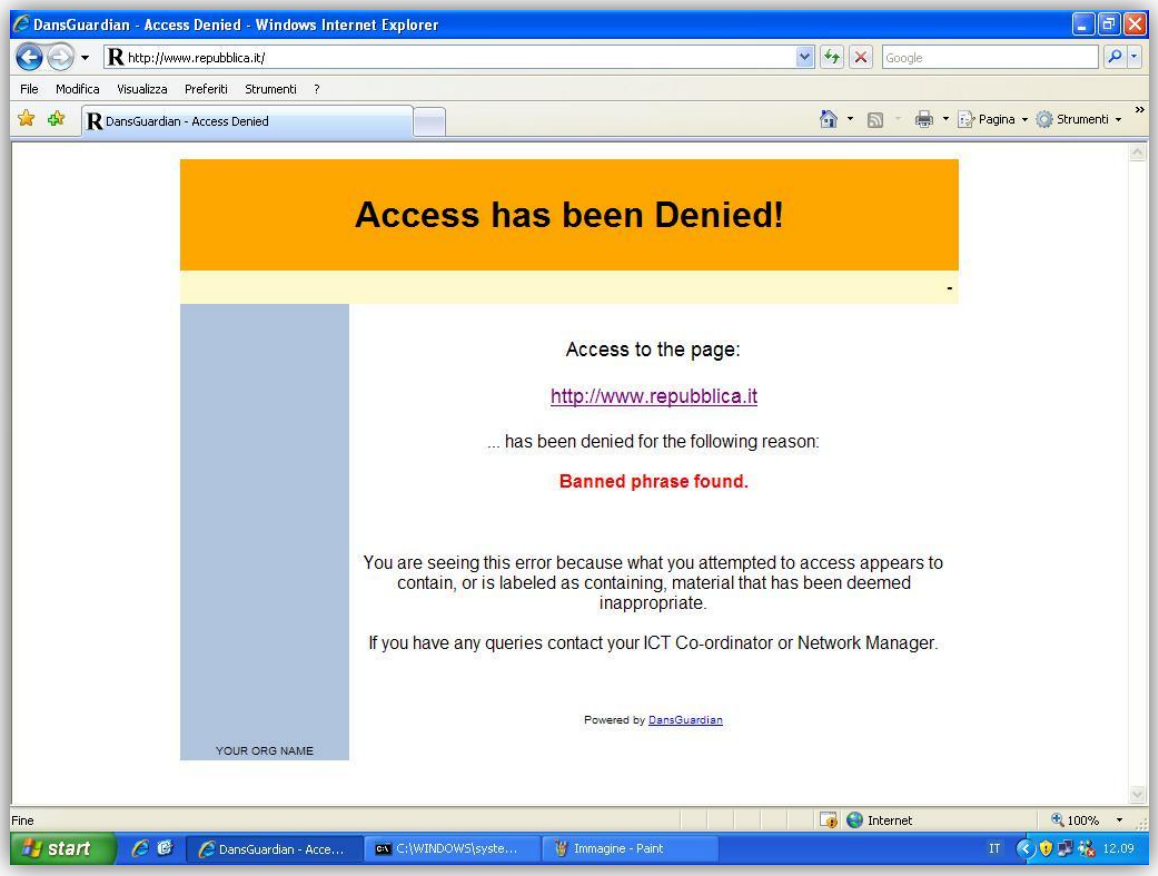
Per il filtraggio dei contenuti, dansguardian utilizza diversi file posizionati in /etc/dansguardian. Questi possono avere due finalità:

- Negare l'accesso ad un contenuto. Il nome inizia con *banned*.
- Consentire l'accesso ad un contenuto. Il nome inizia con *Exception*.

Ad esempio:

- **bannediplist** – indirizzi IP dei PC che non devono avere accesso al web.
- **banneduserlist** – nomi degli utenti che non devono avere accesso al web.
- **exceptioniplist** - contiene l'elenco degli indirizzi IP che devono saltare il controllo dei contenuti (ad.es. gli indirizzi dei PC degli amministratori).
- **exceptionurllist** – contiene l'indirizzo di pagine di siti che non sono filtrate.
- **Ecc…**

La seguente schermata è la risposta di dansguardian alla richiesta di un contenuto bloccato.



Nell'esempio riportato è stato impedito l'accesso al sito http://www.repubblica.it. Dansguardian riporta le seguenti informazioni:

- Motivo della schermata di errore: Accesso negato – banned phrase found.
- URL a cui è stato impedito l'accesso

Protocollo Wpad

Il **Web Proxy Autodiscovery Protocol**, o semplicemente WPAD, è un protocollo che consente di auto configurare le impostazioni proxy di una macchina. Grazie a questo meccanismo non è necessario impostare manualmente indirizzo del server proxy e relativa porta.

Tale tecnologia è stata sviluppata grazie ad una collaborazione di diverse software house, tra cui Microsoft. Non c'è da stupirsi del fatto che internet explorer sia uno dei browser web che supportano questo protocollo. Normalmente, viene usato un Javascript per l'autoconfigurazione delle impostazioni proxy. Tale script è comunemente indicato con il nome wpad.dat.

Per abilitare le funzionalità offerte dal protocollo WPAD occorre:

- Configurare opportunamente il proxy server Squid.
- Modificare il file /etc/mime.types per specificare il MIME TYPE dei file .dat
- Creare il file wpad.dat nella document root del web server Apache.
- Creare un alias di tipo "wpad.dominio.it" sul server web su cui abbiamo creato il file wpad.dat

Dunque, per prima cosa, tramite un qualsiasi editor di testo, modificare il file mime.types presente nella directory /etc ed aggiungervi la seguente riga:

```
application/x-ns-proxy-autoconfig pac dat
```

Ora occorre creare il javascript wpad.dat e posizionarlo nella document root del web-server Apache (di default /var/www/html). Il contenuto di tale script dovrebbe essere qualcosa di simile al seguente segmento di codice.

```
function FindProxyForURL( url, host )
{
  if( isPlainHostName( host ) ||
      dnsDomainIs( host, "naso.eu" ) ||
      shExpMatch( url, "https*" ) ||
      shExpMatch( url, "snews*" ) )

      return "DIRECT";
  else
      return "PROXY 192.168.69.1:3128; " + "DIRECT";
}
```

Come è facile notare, si tratta di uno script davvero breve. All'interno del costrutti if, vengono effettuati i seguenti controlli:

- isPlainHostName(host) - Controllo se viene specificato un dominio.
- dnsDomainIs (host, "naso.eu") - Controllo se viene richiesta una URL del dominio locale indicato.
- shExpMatch(url, "https*") – Controllo se è stato richiesto l'utilizzo di un protocollo sicuro.

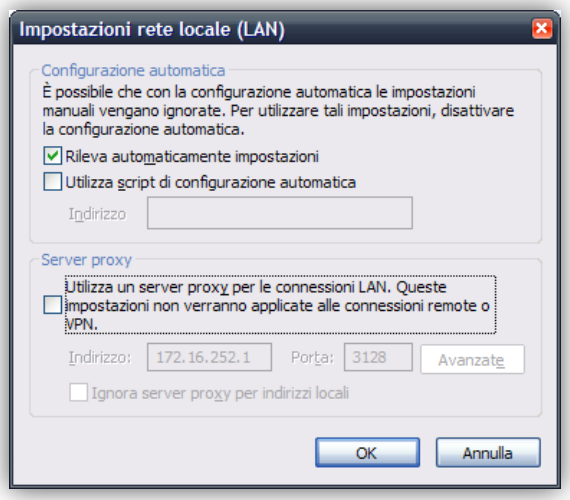
Dopo aver effettuato tali controlli viene deciso se occorre usare un proxy oppure no. L'istruzione return "DIRECT"; specifica che è possibile accedere direttamente alla risorsa richiesta. Al contrario, return "`PROXY 192.168.69.1:3128;`" indica di usare il server proxy specificato.

192.168.69.1 è l'indirizzo della macchina su cui gira il software Squid e che, di conseguenza, agisce da server proxy. 3128 è la porta di default su cui Squid rimane in ascolto. "Naso.eu" è il dominio locale per cui è possibile non utilizzare il server proxy.

A questo punto occorre creare un alias per accedere al javascript. Nella configurazione del server DNS, in particolare in quella del dominio creato, è necessario inserire una riga che permetterà di risolvere il nome wpad.dominio.it con l'indirizzo IP del server web Apache che mette a disposizione lo script. Ponendo per esempio, che il server web si trovi sulla macchina con indirizzo Ip 192.168.69.10, la riga da aggiungere sarebbe:

```
wpad    IN      A       192.168.69.10
```

Dopo aver riavviato i servizi coinvolti nelle modifiche sopra riportate, è necessario configurare il browser al fine del rilevamento automatico delle impostazioni proxy.



Aggiungere il Repository Dag-Wieers

Spesso accade che il software di cui si necessita non sia presente nel repository ufficiale della distribuzione. Per questo motivo, esistono repository non ufficiali, sviluppati e mantenuti da terzi.

Per quanto riguarda le distribuzioni Red Hat, il repository non ufficiale più famoso ed utilizzato è conosciuto con il nome Dag-Wieers. Qui di seguito verrà illustrato come utilizzare questa utile risorsa, alla quale in diverse occasioni si è ricorso durante il lavoro svolto per questa tesi. Il sito di riferimento è *http://dag.wieers.com/*. E' possibile infatti procurarsi i pacchetti da questo repository in due modi:

- Aggiungendo il repository in modo da poter installare i pacchetti e le dipendenze tramite yum.
- Scaricando i pacchetti direttamente dal sito. (http://dag.wieers.com/rpm/)

La via più semplice per aggiungere questo repository è quella di scaricare e installare il pacchetto rpm che automaticamente configura il sistema yum. Il pacchetto è presente al link http://apt.sw.be/redhat/el5/en/i386/rpmforge/RPMS/rpmforge-release-0.3.6-1.el5.rf.i386.rpm, di conseguenza è possibile eseguire il comando:

```
# rpm -Uhv http://apt.sw.be/redhat/el5/en/i386/rpmforge/RPMS/rpmforge-release-0.3.6-1.el5.rf.i386.rpm
```

Questo configurerà automaticamente il necessario per l'utilizzo dei repository, chiavi incluse. E' possibile verificarne l'effetto guardando il contenuto della directory /etc/yum.repos.d/.

```
[root@localhost yum.repos.d]# ll
totale 24
-rw-r--r-- 1 root root 2049 19 giu  2008 CentOS-Base.repo
-rw-r--r-- 1 root root  626 19 giu  2008 CentOS-Media.repo
-rw-r--r-- 1 root root  684  8 mar  2007 mirrors-rpmforge
-rw-r--r-- 1 root root  428  8 mar  2007 rpmforge.repo

[root@localhost yum.repos.d]# cat rpmforge.repo
# Name: RPMforge RPM Repository for Red Hat Enterprise 5 - dag
# URL: http://rpmforge.net/
[rpmforge]
name = Red Hat Enterprise $releasever - RPMforge.net - dag
#baseurl = http://apt.sw.be/redhat/el5/en/$basearch/dag
mirrorlist = http://apt.sw.be/redhat/el5/en/mirrors-rpmforge
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge
enabled = 1
protect = 0
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1

[root@localhost yum.repos.d]# cat mirrors-rpmforge
http://apt.sw.be/redhat/el5/en/$ARCH/dag
http://archive.cs.uu.nl/mirror/dag.wieers/redhat/el5/en/$ARCH/dag
http://ftp2.lcpe.uni-sofia.bg/freshrpms/pub/dag/redhat/el5/en/$ARCH/dag
#http://ftp.heanet.ie/pub/freshrpms/pub/dag/redhat/el5/en/$ARCH/dag
http://ftp-stud.fht-esslingen.de/dag/redhat/el5/en/$ARCH/dag
http://mirror.cpsc.ucalgary.ca/mirror/dag/redhat/el5/en/$ARCH/dag
http://mirrors.ircam.fr/pub/dag/redhat/el5/en/$ARCH/dag
http://rh-mirror.linux.iastate.edu/pub/dag/redhat/el5/en/$ARCH/dag
http://rpmfind.net/linux/dag/redhat/el5/en/$ARCH/dag
http://wftp.tu-chemnitz.de/pub/linux/dag/redhat/el5/en/$ARCH/dag
```

A questo punto sarà sufficiente aggiornare la lista di pacchetti disponibili tramite yum.

Capitolo 6

Monitoraggio e Logging

Nella gestione della sicurezza è molto importante tenere sotto controllo tutto ciò che accade. Per essere più precisi, è necessario tenere traccia delle attività svolte sulle macchine che compongono la rete. Queste attività possono avere varia natura, accessi utente, installazione di software, attività di programmi e servizi e così via. Nel caso della sicurezza della rete e degli host che la compongono, un amministratore avrà bisogno di controllare:

- Tentate intrusioni
- Intrusioni andate a buon fine
- Modifiche fatte al sistema violato da parte di un eventuale intruso.

Il tracciamento di tali attività avviene mediante l'analisi approfondita di ciò che viene normalmente chiamato log di sistema.

I log sono dei file che contengono informazioni sullo stato di una macchina. Essi raccolgono molti dati tutti insieme e spesso capita che per la maggior parte siano costituiti da messaggi di nessuna importanza. Tali messaggi possono essere definiti "rumore". Allo stesso tempo, però, i log contengono informazioni molto importanti, le quali vanno tenute in seria considerazione.

E' importante avere sempre dei log validi e, in alcuni casi, mantenerne delle copie. Questo è un bisogno che deriva dalle abitudini dei cracker. Questi ultimi, per cancellare le tracce delle loro attività illecite, pensano bene di mettere le mani nei file di log del sistema in cerca dei messaggi generati dalla loro intrusione, per poi, ovviamente, eliminarli.

I sistemi Linux dispongono di un sistema di logging che permette di raccogliere i messaggi generati dalla macchina, in particolare dai programmi e servizi che vi girano, e di renderli consultabili dall'utente. Questo sistema prende il nome di Syslog.

I messaggi vengono immagazzinati nel file /var/log/messages. Ogni sistemista che si rispetti tiene sempre aperta una console per la consultazione dei log. Per fare questo, solitamente, si usa il comando:

tail –f /var/log/messages

Il comando tail non fa altro che visualizzare le ultime dieci righe del file indicato. In questo modo però, viene visualizzata una quantità molto ridotta di informazioni. Come è possibile notare dalla sintassi del comando mostrato, è stata aggiunta l'opzione –f che produce un output continuo e, quasi, in tempo reale. In questo modo è possibile osservare un flusso continuo di messaggi. Per quanto riguarda le prove effettuate in fase di test, questo sistema si è rivelato molto utile al fine di verificare il corretto funzionamento dei servizi installati ed utilizzati.

Un esempio di output prodotto dal comando appena mostrato è la seguente:

```
Mar 29 16:46:26 localhost snort[4918]: Parsing Rules file /etc/snort/snort.conf
Mar 29 16:46:26 localhost snort[4918]: PortVar 'HTTP_PORTS' defined :
Mar 29 16:46:26 localhost snort[4918]:  [ 80 ]
Mar 29 16:46:26 localhost snort[4918]:
Mar 29 16:46:26 localhost snort[4918]: PortVar 'SHELLCODE_PORTS' defined :
Mar 29 16:46:26 localhost snort[4918]:  [ 0:79 81:65535 ]
Mar 29 16:46:26 localhost snort[4918]:
Mar 29 16:46:26 localhost snort[4918]: PortVar 'ORACLE_PORTS' defined :
Mar 29 16:46:26 localhost snort[4918]:  [ 1521 ]
Mar 29 16:46:26 localhost snort[4918]:
Mar 29 16:46:26 localhost snort[4918]: Frag3 global config:
...
Mar 29 16:46:28 localhost kernel: device eth0 entered promiscuous mode
Mar 29 16:46:28 localhost snort[4918]: Initializing daemon mode
...
Mar 29 16:46:30 localhost snort[4919]: Snort initialization completed successfully
(pid=4919)
```

L'esempio riportato mostra i messaggi di log generati dall'esecuzione del demone snort.

Come precedentemente anticipato, è molto importante avere la garanzia di disporre di un log valido. E' molto importante non ritrovarsi con file di log compromessi da eventuali

intrusi. E' possibile prendere diverse contromisure a livello di permessi, ma esistono comunque delle vie per aggirare questo tipo di protezioni.

Una via comune e molto efficace è quella di creare un log server. Il compito di questa macchina è quello di raccogliere i log di tutte le macchine che compongono la rete. In questa maniera è garantita la validità dei messaggi generati. Un intruso potrà cancellare le tracce che ha lasciato sulla macchina violata, ma i messaggi saranno già stati spediti al log server, il quale conserverà i log originali della macchina di cui è stata violata la sicurezza. E', però, ovvio che la macchina che funge da log server va accuratamente protetta. Se un intruso mettesse le mani anche su questa macchina allora verrebbero compromessi i log di tutta la rete.

Un log server, oltre a costituire un vantaggio in termini di sicurezza, è utile per una migliore organizzazione e leggibilità delle informazioni raccolte. Avendo un unico centro di raccolta, diventa più semplice effettuare una analisi in tempo reale dei messaggi generati da tutte le macchine. Senza questa funzionalità sarebbe necessario consultare in sequenza i log di tutti i sistemi, spostandosi tra le varie postazioni. Tramite il server centralizzato invece è possibile osservare tutte le attività da una sola postazione. Oltre al monitoraggio è, ovviamente, possibile archiviare e creare copie di backup dei log. Questo ultimo aspetto è molto importante. Recenti sviluppi legislativi hanno imposto un obbligo di monitorare e tenere traccia di tutto ciò che accade in una rete privata e pubblica.

Una corretta configurazione del sistema di logging centralizzato dovrebbe rispondere ai seguenti requisiti:

- Tutte le macchine monitorate devono inviare i messaggi di log a una macchina remota. Questo al fine di rendere inutile la cancellazione locale di tali messaggi da parte di un eventuale intruso. Inoltre, come già detto, permette una migliore gestione, archiviazione e consultazione dei log.
- I messaggi di log devono essere inviati in maniera crittografata. Questo al fine di evitare che le informazioni vengano intercettate tramite tecniche di sniffing dei pacchetti di rete.
- Il log server deve essere adeguatamente protetto. L'accesso deve essere possibile solo da console e in nessun caso deve avere servizi in esecuzione che non riguardano l'attività a cui è adibito. Questo è molto importante perché dei servizi accessori

potrebbero contenere vulnerabilità di sicurezza che potrebbero essere sfruttate da un attacker per accedere al server.

- Le informazioni devono essere opportunamente organizzate. Questo per facilitare la consultazione dei dati minimizzandone il tempo di controllo.

In ogni caso non deve verificarsi la perdita, parziale o totale, dei dati raccolti.

Per realizzare tutto questo è possibile utilizzare il software Syslog-NG.

Syslog-NG

Syslog-NG è uno strumento di tracciamento degli eventi di sistema. Fornisce una gestione centralizzata dei log dei sistemi che compongono una rete, indipendentemente dalle piattaforme presenti. Oltre a questo, mette a disposizione strumenti per la personalizzazione del servizio. In particolare sono presenti filtri da applicare al contenuto dei messaggi di log, per la personalizzazione della memorizzazione delle informazioni raccolte.

Syslog-NG nasce dall'esigenza di uno strumento avanzato di monitoraggio delle attività svolte sugli host di una rete. Il controllo costante dei messaggi di sistema è fondamentale per mantenere un buon livello di sicurezza del sistema e, trattandosi di log centralizzato, dell'intera rete.

Uno degli obiettivi che i progettisti di tale sistema si sono posti, è stato quello di rendere il filtraggio dei messaggi basato sui contenuti. In questo modo è possibile stabilire quali messaggi devono essere loggati e devono raggiungere il log server. Questo per permettere ad amministratori e sistemisti di personalizzare il sistema in base alle loro esigenze.

Installazione

Il primo passo da compiere per installare Syslog-NG è quello di procurarsi il pacchetto contenente il software. E' possibile effettuare il download direttamente dal sito del produttore (http://www.balabit.hu/en/downloads/Syslog-NG/downloads/).

E' disponibile un archivio compresso contenente il file sorgente. Occorre estrarlo mediante il seguente comando:

```
$ tar xvzf syslog-ng-*VERSIONE*.tar.gz
```

Verrà creata una directory contenente i file estratti.

Al fine di rendere il log server più anonimo, è consigliabile apportare delle piccole modifiche al file `src/syslog-ng.h`. Tramite un qualsiasi editor di testo, modificare le seguenti linee:

```
# define PATH_SYSLOG_NG_CONF "syslog-ng.conf"
# define PATH_SYSLOG_NG_CONF "/etc/syslog-ng/syslog-ng.conf
```

In qualcosa come, ad esempio:

```
# define PATH_SYSLOG_NG_CONF default.conf"
# define PATH_SYSLOG_NG_CONF "/etc/.conf/default.conf
```

A questo punto, previa installazione delle librerie libol, è possibile procedere alla compilazione del sorgente. I comandi da eseguire sono:

```
$ ./configure
$ make
# make install
```

Alternativamente, sono disponibili binari precompilati sul repository Dag-Wieers. In questo caso l'installazione è immediata e avviene mediante i seguenti comandi:

```
# yum install syslog-ng                    (se effettuato da yum)
# rpm –i syslog-ng-*VERSIONE*.rpm (se effettuato da pacchetto rpm)
```

Per quanto riguarda la compilazione da sorgente è bene fare attenzione ai possibili parametri da affiancare al comando ./configure. Tra i principali è possibile trovare

```
--prefix=PATH
```

il quale consente di specificare il prefisso di installazione. Questo è utile nel caso si volesse cambiare il percorso di default del programma. Normalmente, infatti, il software viene posizionato in /usr/local. Per visualizzare tutte le opzioni disponibili è necessario usare il comando:

```
./configure --help
```

Configurazione ed Utilizzo

In questo paragrafo verrà mostrato come personalizzare la configurazione delle componenti di Syslog-NG.

La struttura del sistema è composta dalle seguenti componenti:

- Destinazioni
- Sorgenti
- Filtri

In Syslog-NG il percorso di un messaggio, conosciuto anche come message path o message route, consiste in un insieme di sorgenti, filtri e destinazioni. Un messaggio viene inserito in una delle sorgenti definite, successivamente, se risponde ai requisiti definiti nelle regole di filtraggio, viene trasmesso fuori tramite una delle uscite.

Sorgenti

Una sorgente è un insieme di driver, che raccoglie i messaggi usando un metodo predefinito. Piattaforme diverse usano differenti metodi per mandare i messaggi al servizio che si occupa di gestire il logging, molti dei quali sono supportati da Syslog-NG. Quest'ultimo, infatti, gode di una elevata compatibilità con vari sistemi operativi. In particolare è noto il supporto a sistemi Linux, BSD e Solaris.

Le modifiche alla configurazione di Syslog-NG vanno apportate all'apposito file syslog-ng.conf (precedentemente rinominato default.conf). Di conseguenza, ogni dichiarazione di sorgente, destinazione e filtro va effettuata all'interno di tale file.

Di seguito la sintassi per la dichiarazione di sorgente:

```
source <identificatore> { source-driver(parametri);
                          source-driver(parametri);… };
```

L'identificatore è una parola che permette di distinguere la sorgente in questione dalle altre già definite. Ovviamente deve essere univoco.

Ogni possibile meccanismo di comunicazione ha il suo corrispondente driver in Syslog-NG. Ad esempio per aprire una socket Unix SOCK_DGRAM si utilizza il driver unix-dgram. In maniera simile, per aprire un socket Unix SOCK_STREAM si usa il driver unix-stream. Quest'ultimo in particolare è usato su sistemi Linux.

Nei sistemi operativi Linux solitamente si usa il socket SOCK_STREAM, il quale viene chiamato /dev/log. Nei sistemi BSD si utilizza il socket SOCK_DGRAM chiamato /var/run/log. La differenza tra i due è sottile, ma fondamentale. Il socket SOCK_DGRAM, usato su BSD, è affetto da possibile perdita di dati. Cosa che su SOCK_STREAM, sotto Linux, non accade.

Un esempio di dichiarazione di sorgente, utilizzando i driver appena descritti è il seguente:

```
source src { unix-stream("/dev/log"); internal();
                          udp(ip(0.0.0.0) port(514)); };
```

Prima di spiegare il significato di ogni parametro definito, è necessario dire che ogni driver ha bisogno dei propri parametri. Tali parametri sono detti posizionali. Questo significa che devono essere inclusi in un preciso ordine. Il driver unix-stream ha bisogno di un solo parametro obbligatorio, il nome del socket da ascoltare. Il resto degli argomenti passati al driver è costituito da parametri opzionali. Questi ultimi possono essere dichiarati in qualsiasi ordine e hanno la forma `nome_parametro(valore)`.

Nell'esempio sopra riportato sono stati utilizzati il driver unix-stream a cui è stato passato il socket /dev/log. In aggiunta sono stati definiti alcuni dei parametri sopra citati. Qui di seguito vengono elencati alcuni dei parametri opzionali che è possibile utilizzare.

- *Internal* – Si usa per i messaggi generati internamente da Syslog-NG.
- *File* – Si usa per leggere da file specificati.
- *Tcp* – ascolta sulla porta tcp specificata.
- *Udp* – ascolta sulla porta udp specificata.

I seguenti driver possono essere utilizzati nella dichiarazione di sorgente:

- *Internal*() – Se si desidera ricevere messaggi da Syslog-NG è necessario utilizzare questo driver in una source statement.
- *Unix-stream*(), *unix-dgram*() – si tratta di driver adibiti all'apertura di socket unix. Il primo è usato prevalentemente su sistemi Linux. E' di tipo connection oriented e utilizza SOCK_STREAM. Il secondo è usato su sistemi BSD e utiliza SOCK_DGRAM. Quest'ultimo non è connection oriented per cui è possibile perdita di messaggi. Entrambi hanno il nome del socket da creare come unico parametro obbligatorio. Questi driver necessitano di un nome da associare alla socket creata e di alcuni parametri opzionali che è possibile vedere qui di seguito.
  - *Owner*() – definisce l'uid della socket.
  - *Group*() – definisce il gid della socket.
  - *Perm*() – definisce i permessi sulla socket.
  - *keep-alive*() – usato per tenere le connessioni attive quando Syslog-NG viene riavviato. Può essere usato solo con unix-stream.
  - *max-connections*() – usato per limitare il numero di connessioni aperte simultaneamente. Come il precedente parametro, può essere usato solo con unix-stream.
- Tcp() e Udp() – Questi driver lasciano ricevere messaggi dal network. Come suggerito dal nome, vengono usati protocolli TCP e UDP. Il primo è un protocollo connection oriented. Offre, quindi, connessioni controllate e provvede alla ritrasmissione dei pacchetti persi. Al contrario, UDP è un protocollo più veloce, ma meno sicuro di TCP. Non viene effettuato controllo sull'effettiva ricezione dei pacchetti. Di conseguenza ci può essere perdita di messaggi. Questi driver non richiedono parametri particolari. Di default si mettono in ascolto su tutte le interfacce disponibili.
- file() – Solitamente i messaggi del kernel sono contenuti in file speciali. Sotto linux questo file è situato in /proc/kmsg, su BSD, invece, è /dev/kmsg. Per leggere questi file si utilizza il driver file().

Filtri

I filtri sono dei componenti di Syslog-NG che specificano come i messaggi devono essere instradati. Nella realtà pratica, un filtro è un'espressione booleana che, se verificata, permette ad un messaggio di essere trasmesso in uscita.

Di seguito viene riportata la sintassi per la dichiarazione di questo tipo di componente.

```
Filter <identificatore> { espressione booleana; };
```

Anche in questo caso l'identificatore deve essere univoco. L'espressione tra parentesi può utilizzare gli operatori booleani AND, OR e NOT.

Un esempio di dichiarazione di un filtro è il seguente.

```
Filter denyword { host("host1") and match("deny"); };
```

In questo caso il filtro instraderà i messaggi provenienti dall'host host1 e contenenti la parola "deny".

Nelle vecchie versioni di Syslog-NG era presente un filtro di default che veniva applicato a tutti i messaggi che non fossero stati associati ad altri filtri definiti. Tale peculiarità è stata, però, eliminata dalla versione 1.5.x.

Destinazioni

E' necessario definire verso dove mandare i messaggi. Per fare questo occorre dichiarare una destinazione. Un messaggio parte da una sorgente e attraversa delle regole di filtraggio che, se soddisfatte, permettono l'instradamento del messaggio verso una destinazione.

La sintassi per la dichiarazione di una destinazione è la seguente:

```
destination <identificatore> { destination-driver(parametri);
                               destination-driver(parametri); … };
```

I driver e i parametri per le destinazioni, sono gli stessi che si hanno per le sorgenti. Tali driver vengono utilizzati per inviare i messaggi verso destinazioni esterne al sistema Syslo-NG.

- File() – probabilmente è il driver di destinazione più importante. Serve a mandare i messaggi ad un file stabilito.
- Unix-stream(), unix-dgram() – usati per inviare i messaggi verso socket unix.
- Tcp(), udp() – usati per instradare i messaggi in rete verso determinati host.

Log

Per collegare sorgenti, filtri e destinazioni occorrono dei comandi di log. Qui di seguito viene riportato il formato standard per questi tipi di comandi.

```
log { source s1; source s2; … filter f1; filter f2; …
                          destination d1; destination d2; … };
```

Un comando di questo tipo fa in modo che i messaggi provenienti dalle sorgenti indicate arrivino alle destinazioni se soddisfano le regole di filtraggio.

Tuning

Come sarà ormai chiaro, esiste un'ampia possibilità di personalizzazione del servizio. Le configurazioni di default dovrebbero essere sufficienti a garantire buone funzionalità nel caso di logging su un singolo sistema, ma per un logging centralizzato in una rete con più host le cose cambiano. E' bene quindi prendere familiarità con la sintassi delle dichiarazioni ed effettuare prove e test fino a quando non si raggiunge il livello di logging qualitativo desiderato.

Logsnorter

Avendo trattato i sistemi IDS, è coerente parlare di Logsnorter. Si tratta di un piccolo programma, più specificatamente uno script Perl, che ha il compito di analizzare i messaggi Syslog generati da firewall Unix, nel caso in questione iptables/netfilter.

In particolare, Logsnorter va in cerca dei messaggi riguardanti pacchetti scartati. Tutto questo al fine di inserire tali messaggi nel database di log usato con Snort. L'utilità è quella di tenere traccia del traffico bloccato nella maniera migliore possibile.

Per l'utilizzo di questo programma, quindi, è necessario avere installati snort e un DBMS come Mysql con il quale creare il database di logging. Presupponendo di aver già

provveduto all'installazione e configurazione di entrambi, verranno illustrati i passi da seguire per avere un Logsnorter funzionante.

Si immagini di aver creato il Database "snortlog". Tale database verrà usato da Snort per l'archiviazione degli alert, e da Logsnorter per il recupero di messaggi di log generati da Iptables a seguito di pacchetti scartati.

Per prima cosa è bene impostare correttamente i permessi alle tabelle che compongono il database, in modo che Logsnorter possa interagirvi.

```
# mysql –u root –p
SET PASSWORD FOR logsnorter@localhost=PASSWORD('password');
grant insert,select on snortlog.* to logsnorter@localhost;
```

In questo modo Logsnorter non avrà problemi ad apportare modifiche alle tabelle del Database.

A questo punto è necessario creare l'utente logsnorter.

```
# useradd logsnorter
# passwd logsnorter
```

Per motivi di sicurezza è bene impedire a tale utente di loggarsi sul sistema. Per fare questo è necessario modificare il file /etc/passwd aggiungendo /sbin/nologin alla riga dedicata all'utente appena creato.

A questo punto è possibile procedere all'installazione del programma vero e proprio.

```
# mkdir /usr/local/logsnorter
# cd /usr/local/logsnorter
# wget http://www.snort.org/dl/contrib/other_logs/logsnorter-0.2-tar.gz
# tar xvzf logsnorter-0.2.tar.gz
# mv logsnorter-0.2.tar.gz logsnorter
# chmod 711 logsnorter
```

Logsnorter legge le opzioni di configurazione dal file logsnorter.conf. Tale file deve essere spostato in /etc. Modificare le seguenti righe per permettere al programma di accedere al database:

```
$db_server='localhost';
```

```
$db_usercode='logsnorter';
$db_database='snortlog';
$db_password='password';
```

A questo punto tutto è pronto affinché logsnorter funzioni.

```
# /usr/local/logsnorter –t –T /var/log/snortlog
```

Il precedente commando server per avviare logsnorter.

Conclusioni

In conclusione, durante la realizzazione di questo lavoro è stata mostrata una via per ottenere un discreto livello di sicurezza per una rete di piccole-medie dimensioni, come ad esempio una rete aziendale, utilizzando esclusivamente software open source. Di conseguenza, si può affermare che è possibile proteggere la propria rete con strumenti di ottima qualità senza andare incontro ad ingenti spese, le quali si avrebbero utilizzando soluzioni commerciali.

Naturalmente, tutto ciò che è possibile proteggere può essere comunque violato. Per quante precauzioni e contromisure si possano prendere, non si potrà mai ottenere un livello di sicurezza totale. Le soluzioni mostrate in questa tesi costituiscono sicuramente un ottimo punto di inizio per rendere la vita difficile ad eventuali intrusi. E' buona norma tenersi sempre aggiornati su nuovi strumenti, soprattutto su nuove versioni di quelli già esistenti ed usati. E' necessario porre la dovuta attenzione sulle patch di sicurezza del software di base, in particolar modo il sistema operativo.

Riepilogando, i principali servizi studiati per la stesura di questa tesi sono stati: DHCP, DNS, Proxy, Firewall, IDS, Syslog, Syslog-NG.

Il protocollo DHCP usato per l'assegnazione degli indirizzi IP alle macchine della rete LAN interna. Configurando gli host per l'acquisizione automatica di un indirizzo, si evitano conflitti di indirizzi sulla rete. Questo perché, come noto, ogni macchina deve possedere un indirizzo prettamente univoco per essere correttamente identificata e raggiunta.

Il servizio DNS è stato usato per l'associazione di indirizzi IP con nomi e domini. Nella rete installata per fini di test, è stato creato il dominio naso.eu. Grazie al server DNS, le macchine appartenenti al dominio, sono accessibili utilizzando il loro nome invece che il loro indirizzo. Questa funzionalità risulta particolarmente utile nei casi di servizi che richiedono la connessione a particolari macchine, ad esempio i server web. Parlando di aziende, accade spesso che queste aprano siti web destinati alla pubblicizzazione e vendita di prodotti. Normalmente le aziende non si appoggiano a servizi di hosting online, ma preferiscono mantenere personalmente i server che forniscono il servizio. Di conseguenza è

fondamentale possedere un dominio che sia facilmente comprensibile e memorizzabile da potenziali clienti.

Il server Proxy, è uno strumento dalle ampie potenzialità. Offre due servizi molto importanti per una rete di computer. Si tratta di Web-cache e filtraggio dei contenuti. Il primo rende la rete più performante, gestendo in maniera intelligente le richieste alle stesse risorse. Il secondo, invece, fornisce la possibilità di monitorare le attività online delle macchine che compongono la sottorete. E' un servizio fondamentale per evitare che la rete aziendale venga utilizzata per fini non conformi alle attività aziendali. Il software utilizzato per fini di test è il famoso Squid Proxy Server. Si tratta di una soluzione molto diffusa in ambianti Unix.

Il firewall illustrato è il famoso Iptables/Netfilter. Attualmente si tratta della soluzione firewall maggiormente utilizzata in ambienti open source. Inoltre si è rivelato uno strumento molto flessibile e semplice da configurare. Le regole hanno una sintassi particolarmente semplice e non richiedono una grande esperienza. Bastano poche ore per acquisire familiarità nella lettura e scrittura delle regole.

L'intrusion Detection System, è un sistema di monitoraggio del traffico di rete che ha lo scopo di avvertire l'utente del verificarsi di attività anomale, in particolare attacchi informatici e intrusioni.

La soluzione software presa in considerazione prende il nome di Snort. Si tratta di un sistema IDS molto potente ed è ritenuto tra i più validi in circolazione. Durante la fase di test, ha dimostrato la sua validità generando degli allarmi per ogni attacco ricevuto. Si è rivelato, inoltre, molto performante. Il computer adibito ad IDS è stato lasciato esposto in rete per molto tempo. Il sistema ha provveduto a notificare e loggare tutte le attività anomale monitorate.

In definitiva, si può dire che si tratta di uno strumento fondamentale per la messa in sicurezza di una rete di computer. Costituisce un aiuto non indifferente per gli amministratori, per quanto riguarda l'attività di monitoraggio. Tramite un piccolo software aggiuntivo è stato possibile fare in modo di inoltrare i report ad una casella di posta elettronica. Questa funzionalità, attiva ventiquattro ore su ventiquattro, costituisce una discreta possibilità di controllo anche al di fuori degli orari di lavoro. In questo modo non si

rischia di trovare brutte sorprese. Inoltre, se necessario, si ha la possibilità di intervenire tempestivamente al verificarsi di casi di particolare gravità.

Tutti questi software che lavorano insieme, generano dei messaggi di log. Tali messaggi vengono conservati e messi a disposizione per una eventuale consultazione da parte degli amministratori di rete. E' molto importante non trascurare questo aspetto. Molte volte si sono verificati malfunzionamenti sul sistema in prova. Un'attenta lettura dei file di log è stata fondamentale al fine di risolvere i problemi riscontrati.

I sistemi Linux, normalmente, utilizzando un semplice sistema di raccolta dei messaggi di log. Si tratta di Syslog. Tale sistema, generalmente, è sufficiente a soddisfare le esigenze dei più. Tuttavia, esistono software più avanzati che forniscono maggiori funzionalità. Ad esempio, si è parlato di Syslog-NG. Quest'ultimo consiste in un meccanismo di log migliorato rispetto al normale Syslog. Fornisce la possibilità di creare un Log Server per la raccolta dei messaggi di log di tutte le macchine della sottorete. Questo costituisce un altro aspetto importante per quanto riguarda la sicurezza generale. I malintenzionati avranno sempre l'accortezza di cancellare le tracce del loro passaggio sulla macchina che hanno violato. Utilizzando un log server, però, queste tracce vengono eliminate solo in superficie.

Non bisogna mai dimenticare che è necessario curare la sicurezza dell'intera rete. E' altamente sconsigliabile proteggere al massimo delle proprie potenzialità un sistema, e trascurarne un altro. Ciò comporterebbe un problema non indifferente. Un eventuale intruso, potrebbe sfruttare questa vulnerabilità per violare altre macchine. E' molto importante valutare attentamente ogni aspetto della rete che si intende proteggere. E' bene ricordarsi che non basta proteggere adeguatamente il gateway. Questo potrebbe essere comunque violato. A quel punto, se le macchine interne non hanno protezioni, l'intruso è libero di fare ciò che vuole.

Come già detto, i software trattati in questa tesi, costituiscono un ottimo strumento di amministrazione, gestione e sicurezza per la propria rete. Naturalmente, si tratta di programmi che richiedono un accurato monitoraggio. Gli amministratori dovranno costantemente controllare il funzionamento di tutti i servizi installati ed utilizzati. Inoltre è fondamentale imparare a riconoscere le anomalie e i malfunzionamenti che si possono verificare.

Occorre molta attenzione nell'analisi dei log generati dal sistema server e dalle macchine che compongono la rete interna. I messaggi di log costituiscono il più grande aiuto che un amministratore possa avere nell'attività di monitoraggio.

Concludendo, è evidente che non è sufficiente il solo utilizzo di strumenti di sicurezza. Trattandosi di software complessi è frequente che si verifichino errori e malfunzionamenti. Fondamentale, quindi, il lavoro di controllo svolto dagli amministratori. Allo stesso tempo, senza le adeguate misure di sicurezza, la componente umana può fare ben poco. Questo fa capire che, in ogni caso, è necessario combinare le proprie potenzialità.

Appendice

File di configurazione

Dhcpd.conf (etc/dhcpd.conf)

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#
ddns-update-style interim;
ignore client-updates;
ddns-updates on;
ddns-domainname "naso.eu";

deny bootp;
authoritstive;
# LAN
subnet 192.168.69.0 netmask 255.255.255.0 {
      option subnet-mask 255.255.255.0;
      option domain-name "naso.eu";
      option routers 192.168.69.1;
      option domain-name-servers 192.168.69.1;
      range dynamic-bootp 192.168.69.100 192.168.69.200;
      default-lease-time 31200;
      max-lease-time 62400;
      }



key rndc-key {
      algorithm hmac-md5;
      secret "z+q3DukgZqQ95mzJe0gx7Q==";
      }

zone naso.eu. {
        primary 127.0.0.1;
        key rndc-key;
        }

zone 69.168.192.in-addr.arpa. {
        primary 127.0.0.1;
        key rndc-key;
        }
```

Named.conf (/var/named/chroot/etc)

```
#
# DNS Server Configuration file.
#

options {
      directory "/etc";
      pid-file "/var/run/named/named.pid";
      forwarders {
            151.99.125.2;
            151.99.250.2;
            };
      };

key rndc-key {
      algorithm hmac-md5;
      secret "z+q3DukgZqQ95mzJe0gx7Q==";
      };

controls {
      inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys
                                              { rndc-key; };
      };

zone "." {
      type hint;
      file "/etc/db.cache";
      };

zone "naso.eu" {
      type master;
      file "/var/named/naso.eu.hosts";
        allow-update { key rndc-key; };
      };

zone "69.168.192.in-addr.arpa" {
      type master;
      file "/var/named/192.168.69.rev";
        allow-update { key rndc-key; };
      };
```

Iptables (/etc/sysconfig/iptables)

```
# Generated by iptables-save v1.3.5 on Sat Mar 28 23:07:59 2009

*filter

:INPUT ACCEPT [13469:3166398]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [1755:1744392]

-A INPUT -i eth0 -p icmp -j DROP

-A FORWARD -p tcp -m tcp --sport 21 ! --tcp-flags FIN,SYN,RST,ACK SYN -j
ACCEPT

-A FORWARD -s 192.168.69.0/255.255.255.0 -p tcp -m tcp --dport 21 -j
ACCEPT

-A FORWARD -p tcp -m tcp --sport 21 ! --tcp-flags FIN,SYN,RST,ACK SYN -j
ACCEPT

-A FORWARD -s 192.168.69.0/255.255.255.0 -p tcp -m tcp --sport 1024:65535
--dport 1024:65535 -j ACCEPT

-A FORWARD -d 192.168.69.0/255.255.255.0 -p tcp -m tcp --sport 1024:65535
--dport 1024:65535 ! --tcp-flags FIN,SYN,RST,ACK SYN -j ACCEPT

-A OUTPUT -o eth0 -p tcp -m tcp --dport 80 -j ACCEPT

-A OUTPUT -o eth0 -p tcp -m tcp --dport 443 -j ACCEPT

-A OUTPUT -o eth0 -p tcp -m tcp --dport 53 -j ACCEPT

-A OUTPUT -o eth0 -p udp -m udp --dport 53 -j ACCEPT

COMMIT

# Completed on Sat Mar 28 23:07:59 2009

# Generated by iptables-save v1.3.5 on Sat Mar 28 23:07:59 2009

*mangle

:PREROUTING ACCEPT [17870:3708781]

:INPUT ACCEPT [13489:3173428]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [1788:1751854]

:POSTROUTING ACCEPT [1817:1758268]

COMMIT

# Completed on Sat Mar 28 23:07:59 2009
```

```
# Generated by iptables-save v1.3.5 on Sat Mar 28 23:07:59 2009
*nat
:PREROUTING ACCEPT [6981:968618]
:POSTROUTING ACCEPT [188:12307]
:OUTPUT ACCEPT [191:12575]
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT
# Completed on Sat Mar 28 23:07:59 2009
```

Problema con Webmin

Durante l'utilizzo di Webmin è stata riscontrata la presenza di un bug. Esso si trova all'interno del modulo snort.wbm.

La schermata principale di tale modulo presenta la lista delle regole che è possibile abilitare e disabilitare. Inoltre, viene data la possibilità di accedere al contenuto di ogni singola regola e modificarlo. In realtà, il collegamento che consente di editare la regola è invalido. Il problema nasce dalla definizione della variabile RULE_PATH contenuta nel file di configurazione snort.conf. Per risolvere questo problema è necessario apportare una piccola modifica al file `/usr/local/lib/webmin/snort/index.cgi`.

La linea

```
($rule) = $ruleset =~ /[#]*(\S+)\.rules/;
```

va sostituita con la seguente:

```
($rule) = $ruleset =~ /(\w+-*\w+-*\w+).rules/;
```

Bibliografia

Sicurezza

Andrew Lockhart, "Sicurezza delle reti trucchi e segreti"
2 Edizione, Tecniche Nuove Edizioni

Operating System Market Share
http://marketshare.hitslink.com/report.aspx?qprid=8

DHCP

tu-chemnitx.do, Configurare un server DHCP
http://www.tu-chemnitz.de/docs/lindocs/RH73/RH-DOCS/rhl-cg-it-7.3/s1-dhcp-configuring-server.html

DNS

Isc.org
https://www.isc.org/products/BIND
OpenSkills, Installazione e configurazione di BIND
http://openskill.info/topic.php?ID=112

OpenSkills, Configurazione di BIND
http://openskill.info/topic.php?ID=113

OpenSkills, Configurare named per l'aggiornamento dinamico mediante DHCP
http://openskill.info/infobox.php?ID=766

Dizionario Informatico, DNS
http://www.dizionarioinformatico.com/cgi-lib/diz.cgi?frame&key=dns

Prozone.it, Come funzionano i DNS
http://www.prozone.it/smf/index.php?topic=183

IDS

Andrew Lockhart, "Sicurezza delle reti trucchi e segreti"
2 Edizione, Tecniche Nuove Edizioni

SNORT

Unisa, Snort
http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0001/snort/content.htm

Snort.org
http://www.snort.org/

Andrew Lockhart, "Sicurezza delle reti trucchi e segreti"
2 Edizione, Tecniche Nuove Edizioni

hakin9, Snort-inline come soluzione
Snort Users manual, Martin Roesch, Chris Green, 8 aprile 2003
Snort Users manual, Martin Roesch, Chris Green, 15 settembre 2008
http://www.snort.org/docs/snort_manual/2.8.3/snort_manual.pdf

Wikibooks,"Snort"
http://it.wikibooks.org/wiki/Snort

HTML.it. Snort per Linux
http://sicurezza.html.it/articoli/leggi/961/snort-per-linux/1/

SYSLOG-NG

Unisa, Syslog-NG,
http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0304/Syslog/index.html

FIREWALL e IPTables

Openskills, Linux firewalling introduzione a Iptables
http://openskill.info/topic.php?ID=124

OpenSkills, Overview di Netfilter e Iptables
http://openskill.info/infobox.php?ID=185

FTP
http://digilander.libero.it/amilinux/doc/netfilter-23.html

Iptbales/Netfilter
http://www.netfilter.org/

WEBMIN

Linuxpedia, "Webmin"
http://linuxpedia.netsons.org/index.php?title=Webmin

Webmin.com, "Webmin"
http://www.webmin.com/

Firewall uilder

FWBuilder
http://www.fwbuilder.org/
http://www.fwbuilder.org/slideshows/tutorial_3/slide_1.html

PROXY

Techdown.it
http://www.techtown.it/home/detail.asp?iData=120&iCat=367&iChannel=2&nChannel=Articoli

SQUID e Dansguardian

Squid
http://www.squid-cache.org/

Linuxdidattica.org
http://www.linuxdidattica.org/docs/gcdss/squid/page10.html

Nixcraft, Install Squid Proxy Server on Centos,
http://www.cyberciti.biz/tips/howto-rhel-centos-fedora-squid-installation-configuration.html

Squid content filtering: Block download of music MP3, mpg, mpeg, exec files
http://www.cyberciti.biz/faq/squid-content-filter-block-files/

Squid: deny users accessing a website

http://www.cyberciti.biz/faq/how-deny-access-users-to-particular-websites/

Dansguardian

http://dansguardian.org/