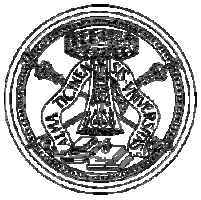


UNIVERSITA' DEGLI STUDI DI PAVIA
SEDE DISTACCATA DI MANTOVA
FACOLTA' DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

UTILIZZO DI SISTEMI LINUX PER LA SICUREZZA DI UNA RETE AZIENDALE

Laureando: Delia Franciosi

Relatore: prof. Giuseppe Federico Rossi

Anno accademico 2004/2005

RINGRAZIAMENTI

Desidero ringraziare in particolar modo il professor Giuseppe Federico Rossi
per la disponibilità e l'aiuto offerti durante lo svolgimento del progetto .
Un particolare ringraziamento va a Emanuele Goldoni per il prezioso supporto offerto.
Ringrazio infine la mia famiglia, Roberto e tutti i miei amici e compagni di studi
che mi hanno sostenuto in questi anni.

INDICE

Pag.

1. INTRODUZIONE SULLA SICUREZZA

La sicurezza informatica è legata strettamente al concetto di protezione del dato. I dati sono importanti perché per i soggetti che li utilizzano e li possiedono assumono un valore. Da qui nasce l'esigenza della protezione. Spesso la perdita di questi dati non è solo economica ma costituisce anche un danno all'immagine dell'azienda.

La compromissione di un dato può assumere varie sfaccettature a seconda dei casi e più precisamente viene detta:

- modifica: quando viene modificata l'integrità;
- generazione: quando vengono aggiunti dati falsi compromettendo l'autenticità;
- intercettazione: quando il dato viene intercettato facendo venir meno la riservatezza;
- interruzione: quando il dato viene distrutto e ciò ne interrompe la fruibilità e disponibilità.

Generalmente ogni tipo di attacco sfrutta un elemento comune: la vulnerabilità, cioè un imprevisto o un errore che consentono ad un intruso l'ingresso nel sistema. Tra le vulnerabilità può anche essere incluso il cosiddetto social engineering, termine usato per indicare la negligenza o scarsa conoscenza di un utente.

Nonostante la parola "hacker" sia utilizzata normalmente come sinonimo di pirata informatico, va precisato che in realtà per "hacker" si intende chiunque si destreggi tra le tecnologie informatiche senza un fine necessariamente negativo, in genere la traduzione che più si addice è quella di "smanettone". I media quindi spesso confondono il termine con "cracker" che invece è il vero pericolo, ovvero colui che si introduce nei sistemi informatici altrui senza permesso. Tra questi si distinguono coloro che agiscono per puro divertimento, dai criminali informatici professionisti veri e propri. In generale chi sferra un attacco informatico può essere semplicemente chiamato "attacker".

I motivi dell'intrusione sono vari: c'è chi lo fa per rubare informazioni, chi per conquistare gloria rispetto ad esempio ai compagni, chi lo fa per danneggiare, chi per ragioni ideologiche in segno di protesta contro la politica o le scelte della società.

Tipologie di intrusi: si differenziano in base ad alcune caratteristiche, che sono da tenere a mente quando ci si occupa della difesa della rete:

-il curioso: ha il solo scopo di "sniffare" i dati e il sistema;

-il malizioso: ha l'obiettivo di bloccare il sistema o modificare pagine web per creare danni e far perdere tempo e denaro a chi si occupa di ripristinare il malefatto;

-l'intruso di alto profilo: lo scopo è quello di rendere popolari le proprie abilità;

-il concorrente: in questo caso l'intruso mira a spiare i dati per poterne beneficiare in termini concorrenziali;

-lo sfruttatore: lo scopo è quello di sfruttare le risorse del sistema per le proprie attività (chat,siti porno,server dns, ecc…).
-il giocatore di cavallina: il fine è quello di sfruttare il sistema come base per sferrare un attacco ad altri sistemi.

Con la crescita di Internet, garantire la sicurezza della rete dei computer è diventata una vera e propria professione. Il sistema può essere attaccato in vari modi.
In questa tesi si analizerranno le varie tipologie di attacco alla rete che possono minare la sicurezza dei dati e gli strumenti che possono servire per difendersi. Si è deciso di analizzare strumenti del sistema operativo Linux in quanto molti sono opensource e di cospicuo interesse e possono contribuire a rendere la rete meno vulnerabile.
L'opensource e Linux stesso si dimostrano una scelta valida per la sicurezza informatica, si sono diffusi velocemente dai piccoli server aziendali fino ad enti militari e governativi. La politica di totale trasparenza ha permesso non solo di evitare la presenza di backdoor (codice che permette di aver accesso alle informazioni sul computer in cui l'applicazione è installata), ma ha portato anche a ricerche approfondite per rilevare le vulnerabilità nel codice in modo di garantire tempi di reazioni immediati.

2. ATTACCHI ALLA RETE

Gli attacchi o i tentativi di intrusione per ottenere l'accesso a un sistema e ai suoi dati possono essere lanciati da due punti differenti:

- da locale: se l'intruso ha accesso alla rete privata
- da remoto: se l'intruso agisce da una rete diversa da quella che vuole attaccare

Distinguiamo l'attacco passivo dall'attacco attivo: nel primo l'attacker si limita ad osservare la vittima e i suoi dati che attraversano la rete, nel secondo invece l'intruso agisce in vari modi, ad esempio fingendo di essere la vittima stessa sulla rete per sottrarre dati sfruttando i servizi di un host o i protocolli utilizzati.

2.1 INTRODUZIONE: PROTOCOLLI DI RETE

Allo scopo di comprendere meglio gli attacchi che verranno descritti è utile tenere presenti alcune informazioni relative ai protocolli di rete. In particolare si farà spesso riferimento ad alcuni campi che sono presenti nelle intestazioni dei pacchetti che vengono trasmessi sulla rete. È necessario perciò avere una idea di come sono composte le header dei pacchetti del protocollo network IPv4 e dei protocolli di trasporto TCP e UDP.

Header IPv4:

<table>
<tr><td>Version</td><td>H_Length</td><td>Type Of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time To Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source IP Address</td></tr>
<tr><td colspan="5">Destination IP Address</td></tr>
<tr><td colspan="5">IP Options</td></tr>
</table>

Spiegazione dei campi:

Campo	**Lun[bit]**	**Descrizione**
Version	4	Versione del protocollo utilizzato (IPv4)
H_Length	4	Lunghezza dell'header in parole di 32 bit
Type of Service	8	Tipologia del servizio
Total Length	16	Lunghezza totale del pacchetto in byte
Identification	16	Identificativo del pacchetto, usato per la frammentazione
Flags	3	Campi per la frammentazione

Frament Offset	13	Esprime la posizione dati del frammento rispetto al datagram originale (in multipli di 8 byte)
Time To Live	8	Tempo di permanenza in secondi del pacchetto sulla rete. Viene decrementato ad ogni nodo attraversato e se raggiunge lo zero viene scartato
Protocol	8	Identifica l'ULP (TCP=6, UDP=17)
Header Checksum	16	Complemento a 1 della somma in complemento a 1 della header vista come parole di 16 bit
Source IP Address	32	Indirizzo IP mittente
Destination IP Address	32	Indirizzo IP destinatario

Lo schema di base può essere esteso con le cosiddette IP Options.

Header TCP:

<table>
<tr><td colspan="3">Source Port</td><td>Destination Port</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Acknowledgement Number</td></tr>
<tr><td>H_Length</td><td>Reserved</td><td>Code Bits</td><td>Window Receiver</td></tr>
<tr><td colspan="3">Checksum</td><td>Urgent Pointer</td></tr>
</table>

Lo schema di base può essere esteso con le cosiddette TCP Options.

Spiegazione dei campi:

Campo	**Lun[bit]**	**Descrizione**
Source Port	16	Numero di porta del processo mittente
Destination Port	16	Numero di porta del processo destinatario
Sequence Number	32	Numero di sequenza del primo byte della parte dati
Ack Number	32	Numero di sequenza del primo byte della parte dati del prossimo pacchetto atteso
H_Length	4	Lunghezza header (in parole di 32 bit)
Reserved	6	Campo riservato
Code Bits	6	Significato del segment (URG, ACK, PSH, RST, SYN, FYN)
Window Receiver	16	Numero di byte (finestra) che il destinatario intende ricevere
Checksum	16	Campo di controllo per parte dati e header
Urgent Pointer	16	Per trasmissione dati "out of band"

Header UDP:

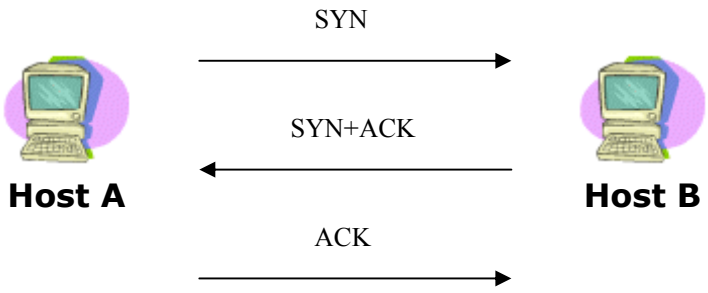
Source Port	Destination Port
Message Length	Checksum

Spiegazione dei campi:

Campo	Lun[bit]	Descrizione
Source Port	16	Numero di porta del processo mittente
Destination Port	16	Numero di porta del processo destinatario
Message Length	16	Lunghezza totale del pacchetto (parte dati e header)
Checksum	16	Campo di controllo per parte dati e header

Threeway handshake:

Verrà anche spesso citata la fase dell'instaurazione di una connessione TCP, in gergo chiamata threeway handshake. Vediamo ora come attraverso quali fasi si stabilisce la connessione:



La threeway handshake inizia con l'invio di un pacchetto con flag SYN attivo da parte dell'host A, dopodichè il server (host B) alloca memoria per conservare le informazioni relative allo stato della connessione. Lo stesso manderà poi un SIN/ACK al mittente (host A) per informarlo che è disponibile ad instaurare la connessione. Poi il server (host B) riceverà a sua volta l'ACK dal mittente (host A). Nel caso in cui non lo riceva entro un certo timeout il pacchetto viene considerato perso, quindi il server (host B) procederà al rinvio del pacchetto SYN/ACK, e il timeout di risposta verrà aumentato.

2.2 FASI DI UN ATTACCO

Il processo di un attacco informatico può essere riassunto in quattro fasi:

a) Hiding (mascheramento)

   Durante questa fase l'attaccante cerca di camuffarsi per non essere individuato ricorrendo a mezzi quali l'utilizzo di sistemi ponte (launchpad) già violati, mascheramento della propria utenza telefonica (dialout), o altro.

b) Information gathering (raccolta di informazioni)

   Per lanciare l'attacco sono necessarie il maggior numero di informazioni possibili riguardanti la rete da attaccare. Si cerca di reperirle attraverso canali standard (database del NIC, WHOIS record, DNS, traceroute, ping, ecc…). Analizziamo ora in successione le tecniche per la raccolta di queste informazioni:

   1. network surveying: lo scopo è di ricercare dati e informazioni importanti, di controllare quali sono le policy di sicurezza e di information disclosure. I dati estratti sono nomi di dominio, denominazioni e funzioni primarie dei server, indirizzi IP, mappe della rete bersaglio, indirizzi e-mail e informazioni sull' ISP dell'azienda.
   2. port scanning: si tratta dell'attività di probing invasivo delle porte del sistema al livello di trasporto. In pratica l'obiettivo è quello di individuare le porte con servizi attivi scandendo gli indirizzi IP precedentemente trovati con la tecnica di network surveying, e ricevere così maggiori dettagli per sferrare l'attacco.
   3. service identification: si tratta dell'analisi dei servizi che sono stati trovati attivi sulle porte prima individuate. Come risultato si ottengono i punti deboli della rete bersaglio.
   4. system identification: si tratta del probing attivo del sistema, per scovare la tipologia di sistema operativo, la sua versione, e altre informazioni. Tra queste tecniche una delle più utilizzate è quella del TCP/IP fingerprinting, la quale può essere realizzata tramite il demone di port scanning nmap. Tutto ciò allo scopo di ampliare la propria visuale della rete da attaccare.

c) System penetration (intrusione)

   Dopo aver recuperato le notizie necessarie è arrivato il momento per l'attaccante di agire. L'attacco si rivelerà sensibilmente diverso a seconda della tipologia della rete, sistema e servizi presenti.

   Esistono 4 tipi di livelli di insicurezza, andiamo ad analizzarli uno per uno.

   1. system insecurity: riguarda una vulnerabilità a livello di sistema operativo o nel software di base. Si parla di vulnerabilità remote nel caso di servizi e programmi applicativi

malconfigurati o di vecchia data sfruttabili per un accesso non autorizzato, oppure di vulnerabilità locali nel caso si tenti di realizzare una scala di privilegi. Tra gli attacchi rivolti a questo tipo di insecurity troviamo quelli volti a eseguire codice arbitrario da remoto (tramite exploit), quelli che riguardano una imprecisa o assente gestione delle eccezioni, o la scorretta implementazione delle politiche di gestione delle credenziali di accesso (attacchi password guessing o brute force).

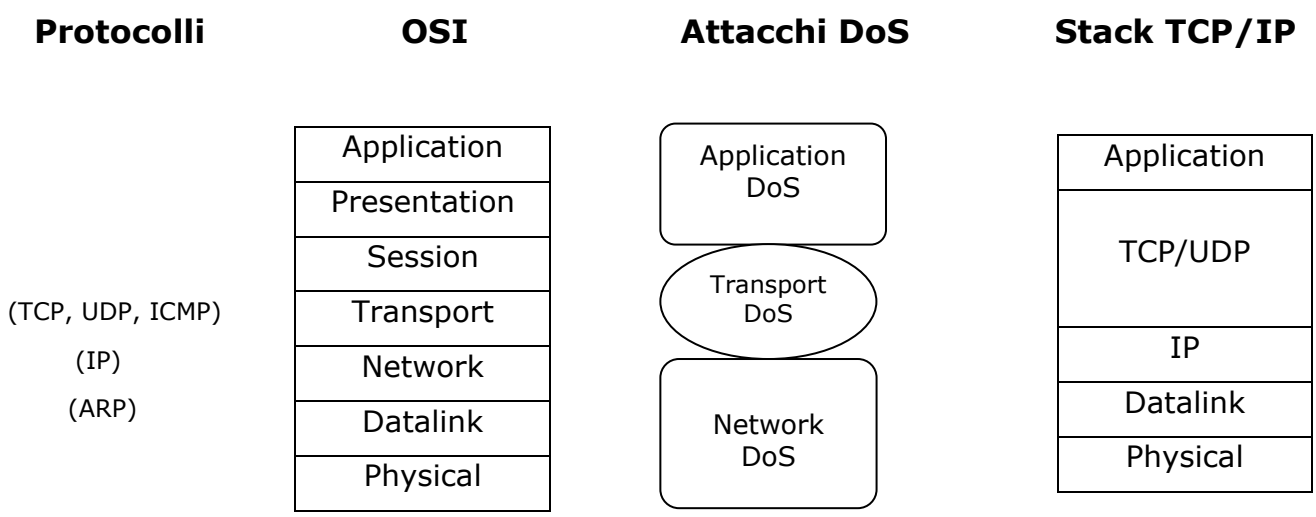
2. network insecurity: riguarda una vulnerabilità connessa alla sicurezza globale della rete. L'insicurezza è dovuta principalmente a come è strutturata la rete, quindi la topologia che lascia punti di accesso, la rete privata, il routing e il firewalling (servizi attivi, default password, policy dei packet filter, ACL) e anche la possibilità di poter effettuare lo sniffing, quindi monitorare il traffico che passa in chiaro sulla rete.
3. application insecurity: riguarda una vulnerabilità del software applicativo. Le insicurezze si manifestano con punti di accesso non autorizzati lasciati aperti da applicazioni internet, dai quali è possibile anche trarre informazioni riservate. Alcuni esempi sono l'SQL injection, il cross-site scripting e session hijacking.
4. procedural insecurity: riguarda una vulnerabilità delle procedure. Possono sfruttare delle falle nella gestione che riguardano relazioni di fiducia tra le macchine o in relazione all'elemento umano (social engineering e human deception).

d) Cleaning (pulizia)

   Dopo che l'attacker ha concluso la sua azione è necessario che cancelli le proprie tracce che lascia sui files di log, su altre parti del file system, e sui dispositivi di monitoraggio della rete (IDS).

2.3 ATTACCO DoS (DENIAL of SERVICE)

Come noto lo stack OSI si compone di sette livelli e in base a questi si possono distinguere vari tipi di DoS. È necessario far presente però che alcuni attacchi ad un determinato livello compromettono livelli più alti, per questo spesso si cerca di risolvere gli attacchi a livello più basso.



L'attacco a interruzione di servizio non prevede l'intrusione ma è volto alla perdita di funzionalità di uno o più servizi, o all'esaurimento delle risorse di sistema per impedirne agli utenti l'utilizzo per un certo periodo di tempo. Lo scopo non è di controllare l'host attaccato, ma quello di impedire allo stesso di svolgere alcune operazioni. Per le aziende che fondano la propria attività sui servizi online ciò costituisce oltre che un danno economico anche d'immagine. Il DoS può causare quindi un esaurimento delle risorse e la saturazione della banda cogliendo le vulnerabilità dei software e dei sistemi operativi e alterando le informazioni delle configurazioni. Si tratta spesso di un attacco facile da attuare ma difficile da combattere.

In realtà alcuni hacker possono usare il DoS come base per poi tentare altri attacchi più seri. Si può indurre un denial of service semplicemente creando un semplice file eseguibile contenente un loop che "pinga" l'indirizzo della vittima, in tal modo esso consuma delle risorse di elaborazione e impedisce temporaneamente l'accesso alla CPU agli altri utenti. Esempio di un file con estensione .bat che crea un loop:

```
:Loop
ping www.SitoRemoto.com
Goto Loop
```

Per fare un altro esempio un attaccante potrebbe creare un file html con lo scopo di far aggiornare dal browser una immagine di dimensioni elevate ogni dieci secondi.

```
<HTML>
<META http-equiv="Refresh" content="10">
<IMG src="http://www.SitoRemoto.com/Enorme.jpg">
</HTML>
```

Tra le varie tipologie di DoS esistono quelli dovuti essenzialmente a errori di programmazione o di configurazione. Uno degli attacchi contro le implementazioni di sistema è il *nuke* che è in grado di terminare connessioni, portare al blocco del sistema o anche al reboot dello stesso. Questo una volta localizzato è facilmente eliminabile, e con un po' di attenzione evitabile. Tra i nuke abbiamo i buffer overflows in lettura causati ad esempio dall'inoculata gestione del parsing, loop vari, e mancanza di timeout nelle operazioni. Per alcune tipologie di DoS invece si può limitare il danni ma non evitare l'attacco. I *flooder*, che sono attacchi invasivi alle risorse del sistema, volti ad esempio alla saturazione della banda, prevedono invece l'utilizzo di maggiori risorse: banda, tempo e altre informazioni oltre all'indirizzo della vittima.
Un buon metro di valutazione della potenza del DoS è il rapporto tra i danni provocati e le risorse impiegate dall'attacker per eseguirlo.
Un firewall comunque è in grado di riconoscere un denial of service notando che è ripetitivo e inoltre che proviene sempre dallo stesso sistema remoto. Contro i nuke un filtro di pacchetti ha effetto a seconda del livello logico nello stack applicativo: se questo è successivo all'applicazione del filtro allora la protezione è efficace. Esistono tuttavia attacchi più complessi e difficili da individuare per un firewall. Contro i flooder il packet filter non può nulla se la banda è saturata a monte del firewall e la quantità di dati sarà tale da bloccare la linea.

Segue una descrizione di alcuni esempi di DoS più celebri:

2.3.1 Ping of death

È un DoS ormai datato e che non arreca più danni e si basa su una vulnerabilità dello stack TCP/IP. Attraverso il comando ping viene inviato ad un sistema un pacchetto ICMP Echo Request. Il mittente riceverà come risposta un ICMP Echo Reply che testimonia che il sistema contattato è attivo sulla rete. L'attacco Ping of Death consiste nell'invio di un pacchetto ICMP Echo Request di dimensioni molto elevate tali da provocare la frammentazione del pacchetto sulla rete. Il riassemblaggio dovrebbe avvenire a destinazione ma il buffer del ricevente non riesce a contenere l'intero pacchetto, e genera un buffer overflow che causerà il blocco della macchina impedendone l'uso per altre operazioni.

2.3.2 Bombardamento di mail

Consiste nel continuo invio di e-mail a un sistema o ad un utente particolare tale da arrivare al riempimento del disco rigido. Come strumento di difesa si può utilizzare Procmail per bloccare o filtrare le e-mail, oppure si può configurare il demone sendmail per bloccare il bombardamento da parte di più utenti.

2.3.3 SYN flood

Un altro DoS remoto da considerare è il SYN Flood (è un flooder), attacco contro le risorse del sistema per esaurirle che sfrutta una debolezza del protocollo TCP/IP. Il flag SYN in un pacchetto permette l'inizio della cosiddetta stretta di mano (threeway handshake) per instaurare la connessione tra mittente e ricevente. L'attacco SYN flood consiste nell'invio di pacchetti con flag SYN attivo e con indirizzo del mittente falsificato. La vittima non riesce a interpretare i pacchetti SYN/ACK del server e li ignora. Il server continuerà quindi a reinviare il pacchetto SYN/ACK. Molte risorse verranno quindi dedicate a queste connessioni senza che esse vengano effettivamente poi terminate.

Un altro modo per generare questo tipo di DoS è l'invio continuo di pacchetto con SYN attivi con lo scopo di esaurire le risorse di memoria del server. Infatti usando una tabella delle connessioni troppo piccola si rischia che sia riempita completamente dall'attacker con l'attacco DoS, altrimenti se è troppo grande l'intruso può cercare di consumare tutta la memoria dell'host.

Per determinare da dove provengono i pacchetti (o da dove sembrano provenire), uno strumento come tcpdump che visualizza il contenuto dei pacchetti può essere utile, poi si può contattare il proprio provider con queste informazioni. I SYN flood possono essere fermati facilmente a livello router o usando un firewall.

2.3.4 Land attack

Questo DoS ormai risolto sfruttava un bug della gestione delle connessioni TCP. L'attacco consiste nell'invio di un pacchetto con il flag SYN attivo e con indirizzo IP e numero di porta del mittente falsificati in modo che coincidano con quelli del destinatario, il quale crede di aver ricevuto il pacchetto da se stesso. Tutto ciò provocava dei rallentamenti del sistema tali a volte da portare allo stallo del sistema.

2.3.5 Teardrop

Anche questo attacco ormai è datato e superato dal kernel 2.0.33 in poi e si basa su una vulnerabilità che coinvolge il riassemblaggio dei pacchetti frammentati a livello IP. Il protocollo IP prevede la frammentazione dei pacchetti a causa delle dimensioni dei livelli delle reti sottostanti. Il kernel deve ricomporre i frammenti del datagram in un certo ordine per ricostruire il pacchetto, per poterli poi passare al livello successivo. Il riassemblaggio a destinazione avviene tramite l'utilizzo di alcuni indici (offset) contenuti negli header dei frammenti che indicano in quale posizione deve essere inserito il frammento per formare il pacchetto originale. L'attacco consiste nella modifica degli

offset per far in modo che la ricomposizione del pacchetto risulti essere sballata generando frammenti sovrapposti parzialmente o interamente, mandando in crisi il sistema.

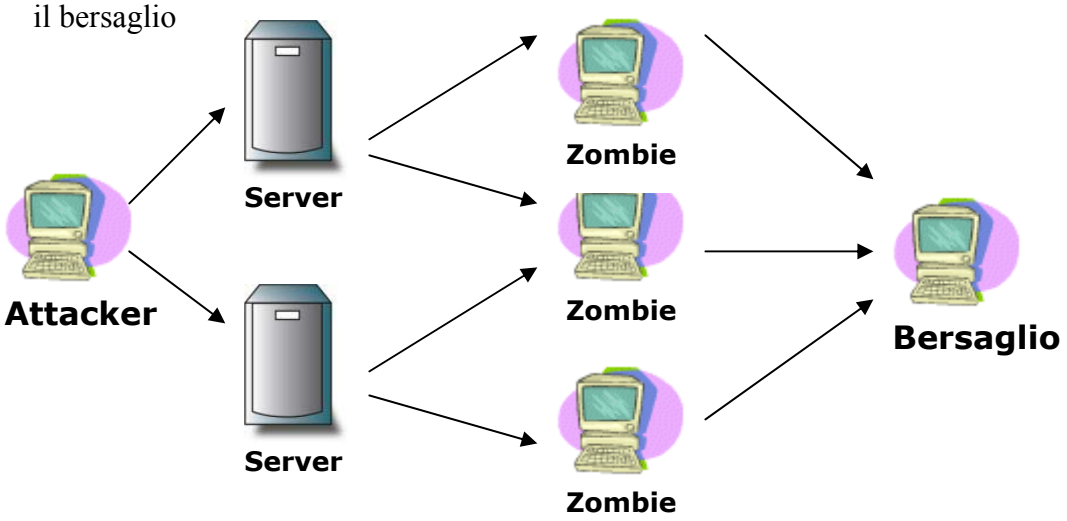
2.3.6 Smurf Icmp

E' un flooder, il tipico attacco Dos ad amplificazione per eccellenza e ha l'obiettivo di saturare la connessione sfruttando il protocollo ICMP e gli indirizzi di broadcast. In sostanza l'attaccante invia un singolo datagramma e il bersaglio ne riceve un numero proporzionale alla quantità di sistemi o reti che amplificano utilizzate dall'attacker. Attraverso il comando ping si può inviare un pacchetto di rete e ricevere una risposta dal computer remoto che accetta la richiesta ICMP (Echo Request). L'output risultante indicherà che il sistema è in funzione e quanto tempo ha impiegato a rispondere. In pratica durante la sua azione l'intruso fa passare la richiesta di ping per una terza parte in modo tale che la risposta sia replicata più volte (relay). L'indirizzo destinatario della richiesta viene impostato a un'intera sottorete piuttosto che all'host singolo e come indirizzo di risposta l'attaccante provvederà a mettere l'indirizzo della sua vittima.
Per difendersi servirà contattare la terza parte perché blocchi l'attacco o mal che vada bloccare sulla propria rete il protocollo ICMP o ancor meglio tentare di convincere l'ISP a bloccare i pacchetti ICMP diretti alla propria rete.

2.3.7 DDoS (Distribuited DoS)

I Dos distribuiti sono originati da più host e diretti verso una vittima predefinita.
La base logica di questi sistemi automatizzati è una rete master/slave controllata dall'attaccante. Nei DDoS si possono distinguere questi elementi:

- la "management", ovvero la postazione di controllo dell'attacker
- i "server"(master, handler) su cui gira il demone per lanciare l'attacco
- gli "zombie"(slave) che dialogano con la rete DDoS
- il bersaglio



L'attaccante quindi per prima cosa installa il demone del DoS (ad esempio con un trojan) sui diversi server, e dopo aver eliminato le proprie tracce dà i comandi ai server stessi, i quali causeranno la negazione di servizio. Comunicando con tale software l'attacker può ordinare di sferrare un attacco flooding congiunto verso un particolare bersaglio. Ogni slave quindi risponde alle richieste del master e porta avanti un attacco sincronizzato verso la vittima comune. Sono stati usati normali exploit per assicurare accessi non autorizzati da remoto. La larghezza di banda risulterà essere direttamente proporzionale al numero di zombie coinvolti e la vittima vedrà arrivare una quantità enorme di pacchetti da indirizzi IP diversi e non riuscirà a gestire il traffico arrivando al blocco del sistema.

Protezione

La mole di dati inviati e la complicata risalita alla sorgente (che avrà indirizzo IP falsificato) renderanno molto ostica la difesa da questo attacco. In questi casi il provider deve riuscire a bloccare i pacchetti con IP falsificato uscenti dalla rete.

I segnali che indicano la presenza di un DDoS sono:

- data link saturo con nessuna riduzione della saturazione
- un numero elevatissimo di connessioni di rete simultanee
- prestazioni di sistema rallentate

Per capire se il data link è saturo, si può pingare un host esterno e tener d'occhio il tempo di latenza della risposta che se molto elevato rispetto al normale (ad esempio di un fattore 10) è indice della disfunzione.

Un buon antitivirus potrebbe rilevare l'agente esterno utilizzato per comandare la nostra macchina, ma ciò non basta, è necessario procurarsi uno sniffer che legga i pacchetti in transito sulla rete.

Oppure attraverso la variazione dinamica dell'indirizzo IP della rete bersaglio si potrebbe redirigere il traffico su una rete fasulla. Comunque la banda passante rimane satura e perciò bisognerebbe agire direttamente sul primo router che collega la rete da dove parte il pacchetto. Per fare ciò sarebbe necessario ottenere l'accordo di tutti gli ISP e carrier internazionali coinvolti, sui quali passa la maggior parte di traffico internet.

2.4 ATTACCO A PREVISIONE DEL NUMERO DI SEQUENZA

Durante una comunicazione, il computer collegato alla rete TCP/IP allega all'interno del proprio pacchetto l'indirizzo IP di destinazione e un numero univoco detto numero di sequenza. Il destinatario accetta i pacchetti che gli vengono inviati solo se l'indirizzo IP e il numero di sequenza sono corretti.

L'attacker esegue l'attacco a previsione del numero di sequenza in due fasi:

1. Durante la prima fase l'attaccante deve riuscire a determinare l'indirizzo IP del server. Se è a conoscenza del nome del dominio del server può individuare l'indirizzo IP semplicemente attraverso il comando ping che nella risposta contiene l'indirizzo IP corrispondente al dominio. Una volta ottenuto l'indirizzo del server l'attaccante potrà cercare di ottenere l'indirizzo degli altri host presenti sulla rete, che hanno in comune parte dell'indirizzo stesso (ad esempio se il server ha un indirizzo di classe C tipo 192.0.0.15 l'attacker cercherà gli indirizzi possibili (fino a 256) esclusivamente cambiando l'ultimo byte).
2. Nella seconda fase l'intruso inizierà a tener d'occhio i pacchetti che si scambiano i computer, e comincerà quindi a prevedere i numeri di sequenza in modo tale da inserirsi nella comunicazione fingendo di essere il mittente del messaggio. In questo modo alcune informazioni sensibili (password, dati, nomi di login, ecc…) che dovrebbero essere riservate potranno essere intercettate. Questo attacco può essere usato oltre che contro il server stesso anche come trampolino di lancio per l'attacco ad altri server sulla rete.

Difesa

Per proteggersi da un attacco a previsione del numero di sequenza è necessario che il router, il firewall e ogni server della rete attivi la funzione di audit-trail. Tramite questo strumento si può osservare quando l'intruso tenta di attraversare il router e il firewall per accedere ai server. L'audit-trail potrebbe mostrare un output ripetuto più volte di questo tipo:

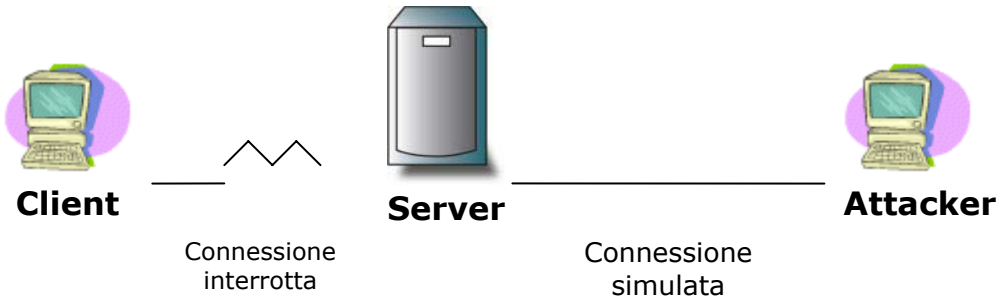
```
Access Denied. IP Address unknown.
```

Appariranno queste righe per ogni tentativo d'accesso dell'intruso, e a seguito di un certo numero di questi accessi negati, verrà generato un avvertimento.

2.5 HIJACKING: DIROTTAMENTO DEL PROTOCOLLO TCP

Noto anche come sniffing attivo (active sniffing), il TCP hijack rappresenta un attacco più serio rispetto a quello a previsione del numero di sequenza. Questo perché l'attaccante non tenterà di indovinare i vari indirizzi IP degli host fino a trovare quello giusto, ma costringerà la rete ad accettare il proprio indirizzo come se fosse lecito. Entrano in gioco perciò sia tecniche di sniffing per recuperare i dati che tecniche di spoofing per far credere di essere il client fidato. L'attacco ha inizio quando il client e il server entrano nello stato di ESTABLISHED (connessione stabilita). In pratica l'attacker acquisirà il controllo di un computer connesso alla rete di interesse e si sostituirà ad esso. Tutto ciò dopo averlo disconnesso e dopo aver sostituito il suo indirizzo IP a quello del computer vittima e inoltre dopo aver cambiato i numeri di sequenza di ogni pacchetto per diventare la nuova destinazione del server. Il TCP hijack risulta essere più pericoloso dell'attacco IP spoofing perché avviene a connessione in corso quindi è possibile intercettare la comunicazione e trarne le

informazioni di interesse. Inoltre consente all'intruso di inserire comandi, e "iniettare" dei dati all'interno della connessione già intrapresa. L'intruso può anche penetrare in un sistema operativo diverso e aggirare il sistema di autenticazione delle password.



Durante l'hijacking si sente spesso parlare di character injection (iniezione di caratteri), in pratica l'intruso dopo aver intercettato la connessione (ad esempio durante una sessione telnet) invia caratteri al server al posto del reale mittente scombussolando la reale comunicazione.
L'attacker ha vari modi d'agire per annullare il client:

- scoordinarlo: durante una connessione attende che il client sia nello stato di ESTABLISHED e prima che anche il server lo diventi, invia un pacchetto col flag RST attivo per resettare la connessione verso il client. L'intruso sarà quindi libero di collegarsi al server spoofando l'indirizzo della vittima. Nel frattempo il client attenderà fino ad andare in timeout che il server entri nello stato di ESTABLISHED e glielo comunichi ma questo non avverrà mai.
- desincronizzarlo: invia i dati in anticipo rispetto al client causando quindi la desincronizzazione della vittima che avrà numeri di sequenza e di acknowledgment errati.
- isolarlo dalla rete attraverso un attacco DOS.
- agire come "Man in the middle" (Uomo nel mezzo), facendo modo di far credere al client e al server di essere in connessione diretta.

Quest'ultima tecnica risulta essere la più proficua perché il client non si accorgerà di nulla e non avrà modo di insospettirsi da lunghi ritardi o interruzioni di comunicazione.
Per combattere questo attacco e lo spoofing è necessario evitare che possano essere portati avanti le tecniche di sniffing, in questo modo la base su cui si pone viene meno.

Spoofing e hijacking con protocollo UDP

Prima di stabilire una connessione nella maggior parte dei casi il client contatta il proprio server DNS per la conversione dell'FQDN (Fully Qualified Domain Name) di una destinazione a indirizzo IP. Per ottimizzare il tempo di collegamento le query DNS utilizzano il protocollo UDP, che è più

veloce ma meno sicuro e affidabile del TCP. Infatti un intruso sniffando la rete può inserirsi nella comunicazione prima che il server DNS risponda al mittente della query e generare una risposta fasulla, per redirigere la vittima verso tutt'altra destinazione.

2.6 SPOOFING

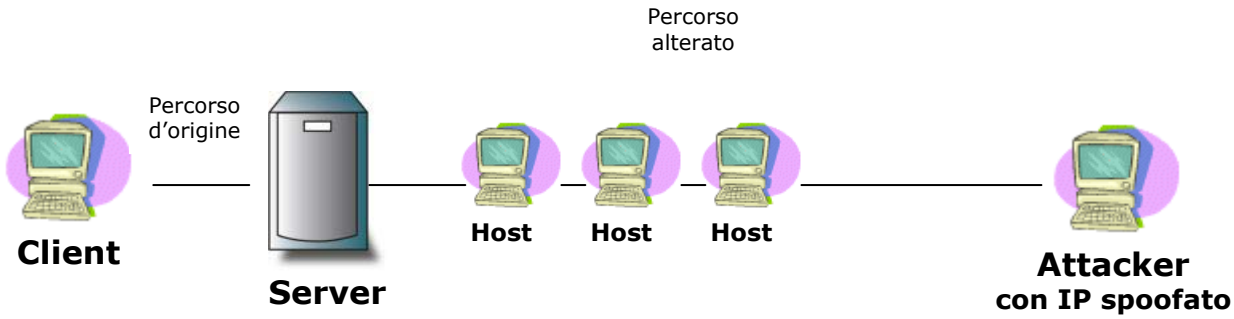
A differenza dello sniffing che è un attacco passivo in quanto si limita solo ad analizzare e trarre informazioni dal traffico sulla rete lo spoofing si rivela un attacco attivo in quanto l'attacker attraverso l'alterazione dei pacchetti convince il server di essere un utente o un sistema fidato e può inviare e intercettare pacchetti indisturbato. Lo spoofing è detto vedente quando l'intruso riesce a sniffare il traffico tra l'host da impersonare e quello che vuole attaccare, mentre è cieco quando ciò non è possibile. Questo attacco sfrutta il fatto che i protocolli TCP e UDP si "fidano" degli indirizzi IP degli interlocutori. In pratica i pacchetti sembrano provenire dall'host vittima mentre in realtà sono inviati dall'attaccante. L'intruso può sfruttare il routing e specificare un percorso diretto per una destinazione e un percorso di ritorno per l'origine. Così facendo può modificare le trasmissioni e intercettarle e comprendere nel percorso di ritorno il proprio sistema.

L'attacco avviene in più fasi:

1. l'attacker dopo aver sniffato la rete per impadronirsi delle informazioni necessarie, sostituisce al proprio indirizzo IP l'indirizzo della vittima valido (campo "Source Address" nell'header IP)
2. attraverso un attacco DoS viene isolato il client per impedire che gli vengano recapitati i pacchetti di risposta del server
3. l'attaccante costruisce poi il percorso di routing desiderato, ovvero il cammino che i pacchetti dovranno intraprendere per arrivare al server e per tornare poi all'host dell'intruso, utilizzando l'indirizzo del client valido come ultimo tratto del percorso per giungere al server
4. l'attacker usufruisce del percorso di origine per inviare al server una richiesta del client valido
5. il server accetta la richiesta credendo di essere in comunicazione con il giusto interlocutore e risponde
6. ogni risposta viene inviata all'host dell'attaccante

Una volta passate queste fasi l'intruso dovrà solamente prestare attenzione ai valori di Sequence Number e Acknowledgement Number sui pacchetti di risposta e rispondere in maniera appropriata.

L'attacco più semplicemente può essere realizzato dopo aver atteso la chiusura della comunicazione del client col server cogliendo il momento per sostituirsi al client stesso.



Esempio:

1. l'attaccante installa uno sniffer per intercettare i dati tra vittima e destinazione da contattare e attende la richiesta di connessione
2. quando inizia la stretta di mano oppure quando questa è terminata l'intruso invia un pacchetto con flag RST attivo alla vittima spoofando l'indirizzo del server e uno al server spoofando l'indirizzo IP della vittima. Ciò per causare una negazione di servizio (DoS) alla vittima che non riesce più a comunicare col server.

2.6.1 Spoofing dei messaggi della posta elettronica

Un intruso utilizzando semplicemente telnet e connettendosi alla porta SMTP può specificare un indirizzo di posta elettronica fasullo ingannando l'host destinatario per sottrarre o falsificare alcuni messaggi.

```
$telnet mail.interfree.it 25
Trying 213.158.72.25...
Connected to mail.interfree.it.
Escape character is '^]'.
220 Welcome to QMAIL TOASTER v 1.0 smtp Server ESMTP
helo interfree.it
250 Welcome to QMAIL TOASTER v 1.0 smtp Server
mail from: <pippo@paperino.it>
250 ok
rcpt to: <mario.rossi@interfree.it>
250 ok
data
354 go ahead
Subject: prova

Ciao da me!
```

```
.
250 ok 1127402779 qp 14206
quit
221 Welcome to QMAIL TOASTER v 1.0 smtp Server
Connection closed by foreign host.
```

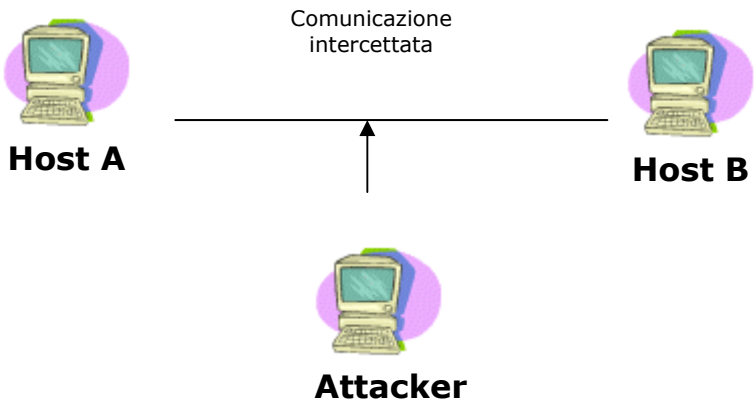
Difesa

Gli attacchi IP spoofing sono difficilmente rilevabili. Attraverso l'audit-trail si effettua una registrazione in un apposito registro di sistema del traffico che passa per la rete. Analizzando questo registro si può venire allo scoperta di "fatti insoliti", indizi di un probabile attacco come ad esempio trovare per pacchetti provenienti da internet che l'indirizzo IP sorgente e destinatario sono entrambi appartenenti alla rete interna. In un attacco IP spoofing infatti in genere l'indirizzo mittente e destinatario di solito appartengono al dominio locale. La soluzione quindi diventa il filtraggio dei pacchetti.

2.7 SNIFFING

Le schede delle interfacce di rete spesso sono settate in modalità promiscua, ciò vuol dire che tutti i pacchetti in transito anche se non destinati alla stessa interfaccia vengono ricevuti. Gli attacker utilizzando appositi programmi detti sniffer sono in grado di monitorare la rete e visualizzare il contenuto dei pacchetti. Se l'intruso ha accesso a un computer della rete basterà che installi lo sniffer per osservare e copiare le informazioni di interesse, altrimenti per prima cosa dovrà violare l'accesso. L'intruso deve però loggare come utente root, questo perché l'uso delle socket raw (system call presenti nei sistemi Unix che permettono di lavorare al livello rete dello stack ISO/OSI) necessarie per lo sniffing è permesso solo all'amministratore. In questi attacchi passivi l'attaccante si limita a sniffare il contenuto dei pacchetti (documenti, e-mail o altro) per estrarre informazioni utili, come password, numeri di carte di credito, ecc… Nel caso si tratti di un attacco attivo cercherà anche di modificare il contenuto del pacchetto.

Gli attacchi passivi a sniffing risultano essere spesso il trampolino di lancio per attacchi attivi a dirottamento o a spoofing IP. In sostanza l'attacker attraverso lo sniffing viene in possesso delle informazioni quali codice utente e password di un utente legittimo, e usufruendo di queste potrà poi aver accesso alla rete.



Difesa

La tecnica dello sniffing risulta essere insufficiente contro protocolli e strumenti che utilizzano la crittografia come SSL e SSH, perchè consente solo l'intercettazione di dati che passano in chiaro. I dati crittografati benché possano essere intercettati presentano un output non leggibile.

2.7.1 Arp Poisoning

L'utilizzo della tecnica di ARP poisoning permette di intervenire in modo attivo nella comunicazione tra client e server.

Prima di addentrarci sull'attacco conviene precisare come funziona il protocollo ARP. Un computer connesso ad una rete è dotato di due indirizzi: IP e MAC. L'indirizzo MAC rappresenta l'indirizzo fisico univoco della scheda di rete della macchina. L'indirizzo IP non risulta sufficiente perché la comunicazione avviene in realtà a livello fisico tra le due schede di rete. Perciò prima dell'invio di dati dall'indirizzo IP si deve cercare di risalire all'indirizzo MAC. Ciò avviene tramite il protocollo ARP, e più precisamente attraverso un pacchetto ARP Request con indirizzo fisico di destinazione broadcast e indirizzo IP della destinazione scelta. In tal modo il pacchetto viene smistato a tutte le schede di rete. In caso la postazione si riconosca come proprietaria dell'indirizzo IP deve rispondere con l'indirizzo MAC corrispondente attraverso un pacchetto ARP Reply. Ogni host è dotato di una ARP cache che contiene le coppie degli indirizzi IP/MAC trovati. Le entry presenti nella cache vengono eliminate dopo un certo periodo di tempo (dai 5 ai 20 minuti).

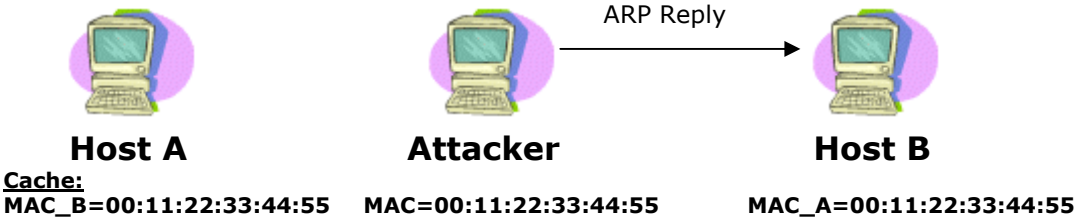
L'attacco è possibile solo in locale dato che i pacchetti ARP non possono transitare attraverso router. L'insicurezza del protocollo ARP sta nel fatto che essendo esso un protocollo stateless, nel caso si verifichi la ricezione di un ARP Reply (senza che ci sia stata una ARP Request) la ARP cache dell'host ricevente viene comunque aggiornata, creando in questo modo una tabella alterata. Va detto che questa falla è stata colmata con l'introduzione del kernel linux 2.4.x che non permette

ARP Reply scorrelate da ARP Request. Tuttavia resta da risolvere il problema delle ARP Request contraffatte (con indirizzo IP e MAC Address ingannevoli).
L'ARP poisoning in genere causa un DoS (negazione di servizio) alla vittima e può inoltre essere usato per sniffare pacchetti in rete.

Esempio:
supponendo il caso della ARP cache che si aggiorna semplicemente con una ARP Reply:

1. per prima cosa l'intruso invia alla vittima una ARP Reply in cui mette il proprio MAC Address facendo credere che sia quello del gateway di uscita della rete locale

2. la stessa cosa viene fatta ai danni del gateway, cioè l'attaccante invia ad esso una ARP Reply col proprio MAC Address al posto di quello dell'host vittima



3. quindi i pacchetti scambiati tra il gateway e l'host in realtà saranno inviati all'intruso, che dopo averli sniffati provvederà a inoltrare i pacchetti alla vere destinazioni per mantenere il contatto

4. per evitare che le associazioni tra indirizzo IP e MAC Address nella ARP cache scadano, i primi due punti dovranno essere ripetuti regolarmente fino alla fine dell'attacco. Una volta concluso l'intruso dovrà fare in modo di ripristinare le associazioni corrette per far in modo che gateway e host vittima riescano a comunicare.

Arp poisoning per decifrare connessioni crittografate

Attraverso questa tecnica si possono ottenere informazioni come password, username, codice html in chiaro. Tutto si svolge in tre passi:

1. la vittima tenta di stabilire una comunicazione crittografata col server SSL. Nel frattempo si inserisce l'attaccante che invia alla vittima un certificato falso da lui generato, e riceve il vero certificato spedito dal server.
2. le session key (per codifica e decodifica dei dati utente) vengono memorizzate all'interno di messaggi cifrati asimmetricamente (presenza di chiave pubblica e privata). Il certificato contiene, oltre alle specifiche dell'ente di produzione, la chiave pubblica necessaria al client per cifrare i messaggi SSL che verranno poi scambiati per negoziare le session key. Non avendo la chiave privata del server l'attaccante deve inviare un proprio certificato alla vittima attraverso lo spoofing (facendo credere che stia comunicando col server). La vittima quindi cifrerà i dati per la negoziazione delle session key con la chiave pubblica dell'attaccante, che riuscirà a decifrare e osservare il contenuto e poi ricifrare con la chiave pubblica del server.
3. Le session key sono due (write e read), una consente al client di codificare i dati mentre il server li decodifica, la seconda serve al client per decodificarli e al server per codificarli. L'attaccante avendo in mano entrambe potrà vedere in chiaro la comunicazione.

La falsificazione del certificato deve essere accurata per non suscitare allarmi perché deve essere:

- inerente al sito contattato dal client, cioè l'host e il dominio (FQDN) devono coincidere con quelli del sito
- la data di emissione e la durata devono essere tali che il certificato non sia scaduto
- deve essere firmato da una certification authority non contestabile, perché il client controlla la firma con quella contenuta nel database del proprio browser.

Contromisure anti sniffing/arp poisoning

Innanzitutto se si dispone ancora di dispositivi hub è meglio sostituirli con switch, meglio se amministrati (managed switch). Questi ultimi permettono di fissare gli indirizzi MAC su ciascuna porta, e non consentono l'uscita di frame dalle porte con indirizzo MAC diverso da uno di quelli configurati. In caso non si disponga di questi switch particolari si può agire a livello software

creando corrispondenze statiche nella ARP cache, in relazione agli host presenti nella rete. Questo è un lavoro sì redditizio ma lungo e faticoso e deve essere fatto per ogni host della rete.

2.8 ATTACCHI A DESINCRONIZZAZIONE ATTIVA

Si basano sul fatto che il ricevente se non riceve pacchetti col numero di sequenza corretto li rifiuta. L'attaccante può sfruttare questo per intercettare la connessione. L'intruso forza entrambe le parti della connessione in uno stato di desincronizzazione. Dopodichè mettendosi su un altro host intercetta i pacchetti e li sostituisce con altri accettabili per entrambi gli host.

2.8.1 Attacco a dirottamento con post-desincronizzazione

Abbiamo detto quindi che l'attacker può essere in grado di intercettare tutti i pacchetti della connessione TCP delle due vittime, e può anche crearne di nuovi e sostituirli a quelli corretti ingannando sia il client che il server. Quindi la comunicazione passerà dal client, all'intruso fino al server e viceversa. Mettiamo che l'attacker sia riuscito nell'intento della desincronizzazione, a questo punto il client invierà al server un pacchetto contente nell'header:

SEG_SEQ=CLT_SEQ

SEG_ACK=CLT_ACK

La prima riga (SEG=segmento dati) indica che il numero di sequenza del pacchetto è il prossimo della serie del client. La seconda riga invece assegna al valore di acknowledgment il prossimo valore di acknowledgment. Data la desincronizzazione il numero attuale di CLT_SEQ non è più uguale al numero precedente di acknowledgment del server (SRV_ACK). Il server quindi scarterà i dati, mentre invece l'attaccante copia il pacchetto rifiutato. Quindi poi l'attacker reinvia questo pacchetto cambiando i dati dell'header in modo che risultino essere:

SEG_SEQ=SVR_ACK

SEG_ACK=SVR_SEQ

La prima riga mostra che il numero di sequenza atteso è corretto perché è quello che il server si aspettava, e quindi il pacchetto verrà accettato ed elaborato.

Dato che l'attaccante intercetta i pacchetti nulla vieta che ne modifichi o elimini tutto o parte del contenuto.

Il client dal canto suo continuerà ad inviare pacchetti e non si accorgerà di nulla perché l'intruso che ottiene le risposte dal server filtrerà il pacchetto in modo da togliere le informazioni che gli servono e manderà al client la risposta del server filtrata opportunamente.

Durante questo attacco in genere si scatena la cosiddetta tempesta di ACK, detta "*TCP ACK storm*". Questo perché un host che riceve un pacchetto non accettabile (con numero di sequenza errato)

risponde con un pacchetto ACK con il numero atteso. Come abbiamo detto in precedenza il primo pacchetto ACK include il numero di sequenza del server, ma il client non lo accetta perché non ha inviato quello di richiesta modificato. Quindi anch'esso genera un pacchetto ACK verso il server, il quale a sua volta genererà un altro pacchetto ACK creando così un ciclo infinito per ogni pacchetto trasmesso. Il ciclo termina nel caso in cui uno dei pacchetti ACK (che non contiene dati) va perso.

2.8.2 Attacco a desincronizzazione iniziale

La differenza tra un attacco a dirottamento e questo tipo di attacco è che nel primo l'attacker agisce a connessione già avvenuta tra client e server mentre nel secondo l'intruso fa la sua comparsa nella fase iniziale di collegamento. L'attaccante quindi interrompe la connessione dal lato server e fa nascere una nuova connessione con un altro numero di sequenza. L'attacco si svolge in quattro fasi:

1. L'attacker intercetta il pacchetto SYN/ACK mandato dal server al client come risposta alla richiesta di connessione.
2. A questo punto invia un pacchetto al server con flag RST attivo per resettare la connessione e poi subito dopo invia il suo pacchetto SYN con i parametri che il server si attende e il numero della porta sulla quale entrare in collegamento. Inoltre il pacchetto dell'attaccante ha un numero di sequenza diverso.
3. Il server alla ricezione del pacchetto RST chiude la connessione e poi la riapre sulla stessa porta all'arrivo del SYN dell'attacker con un numero di sequenza differente. Poi il server risponde con un pacchetto SYN/ACK che invierà al client.
4. Questo pacchetto sarà intercettato dall'intruso che risponderà con un ACK facendo in modo che venga così creata la nuova connessione.

Il client nel frattempo avrà già creato la connessione sincronizzata dopo la ricezione del primo SYN/ACK.

2.8.3 Attacco a desincronizzazione con dati nulli

È una variante del precedente e consiste nell'invio continuo di dati nulli sia al client che al server in modo tale da indurli ad entrare in uno stato desincronizzato a causa dell'alto volume di dati(anche se nulli) facendo in modo di interferire sul buono stato della connessione TCP.

Esempio:

Attacco a sessione telnet

L'attacker può intercettare una sessione telnet in questo modo:

1. prima di sferrare l'attacco si limita a monitorare la sessione intrapresa tra client e server

2. quindi l'attacker invia una sequenza di dati nulli al server contente la sequenza di byte "IAC NOP IAC NOP" (NOP=no operation). Il demone telnet rimuove questi dati dal canale interpretandoli come nulli, mentre la sessione telnet in corso viene interrotta da lato server.
3. si arriva a una sessione telnet desincronizzata
4. per desincronizzare il client l'attacker compie la stessa azione adoperata col server
5. l'attacker ora è libero di intercettare i numeri di sequenza dei pacchetti per continuare a controllare la connessione

Questo tecnica può essere utilizzata solo se la sessione telnet supporta il passaggio di dati nulli. La difficoltà per un intruso è individuare il momento migliore per inviare i dati nulli, dato che in caso di errore non potrà più controllare la sessione telnet vanificando il suo scopo.

3. IDENTIFICAZIONE E DIFESA CONTRO GLI ATTACCHI

3.1 FIREWALL

I firewall sono dei dispositivi hardware o software atti al filtraggio, controllo, modifica, monitoraggio dei pacchetti in transito sulla rete con lo scopo di evitare accessi non autorizzati.
Questi dispositivi tengono traccia dei pacchetti che transitano sulla rete e del loro flusso, e decidono cosa rifiutare, ignorare e accettare, eventualmente registrando sui log di sistema.
Il firewall inoltre può collegare due o più segmenti di rete. In genere si tratta di due reti: una detta interna che è costituita da un certo numero di computer locali e una detta esterna costituita da internet. Spesso il firewall è un componente hardware con due o più schede di rete sul quale gira un ambiente operativo che effettua il packet filtering.

3.1.1 Le security policy

Il firewall viene configurato tramite le cosiddette security policy che sono delle regole impostate dall'amministratore di sistema, le quali consentono di garantire il livello di sicurezza scelto dall'amministratore stesso. In genere vengono modellate a seconda della struttura della rete che devono proteggere e sulle attività svolte in essa e devono riuscire a garantire un livello minimo di riservatezza, disponibilità e integrità. L'idea è quella di lasciare aperti solo i servizi indispensabili all'attività dell'azienda. I dati che soddisfano le security policy vengono fatti passare mentre gli altri vengono rifiutati in ambo le direzioni. Per creare una policy si deve:

1. determinare i servizi necessari
2. determinare gli utenti a cui si vuole garantirli
3. determinare a quali servizi gli utenti possono accedere
4. per ciascun gruppo di utenti decidere come si potrebbe rendere sicuro il servizio
5. scrivere una regola che renda tutte le altre forme di accesso una violazione

Esistono due politiche di base:

- Default Deny Stance: prevede che venga rifiutato tutto ciò che non è permesso esplicitamente
- Default Acceptance Stance: viene permesso tutto ciò che non è espressamente negato.

La prima politica è complessa da realizzare anche se molto efficace, perché è difficile tener presente tutto ciò di cui ha bisogno la rete, mentre la seconda è meno efficace ma più semplice.

Quando si progetta un firewall bisogna stabilire la tipologia di accesso (abilitazione o meno) degli utenti verso i server della rete e i protocolli che possono essere utilizzati. Inoltre è necessario controllare l'accesso al sito web se presente (ad esempio stabilire se è necessario l'acceso tramite sessioni telnet). Come ultimo passo bisogna escludere la possibilità di accesso di eventuali crackers.

3.1.2 Tipologie di firewall

Tra le varie tipologie di firewall la prima distinzione va fatta rispetto al livello di ispezione del firewall e qui troviamo:

- Firewall stateless (Packet Filter)
- Firewall statefull (Statefull Inspection)

3.1.2.1 Firewall stateless (Packet Filter)

I firewall di questo tipo sono in grado di filtrare il traffico fino al terzo livello (rete) controllando selettivamente la comunicazione tra due peer, o tra internet e la rete interna e viceversa, analizzando i pacchetti in transito e instradandoli solo se rispettano alcuni parametri. Questi tipi di firewall consentono o bloccano i pacchetti in base alle caratteristiche della comunicazione, e si occupano solo dei pacchetti presi singolarmente. Le security policy impostate dall'amministratore sono ciò su cui si basa questa tecnologia, ovvero i pacchetti verranno accettati o rifiutati a seconda del rispetto o meno di queste regole. Spesso questa funzionalità è inserita nella tecnologia dei router (screening router), i quali filtrano il contenuto dei pacchetti visionando gli indirizzi IP e le porte sorgente e destinazione o i protocolli utilizzati. Il grosso limite dei Packet Filters consiste nel fatto che non possono comprendere il contenuto di pacchetti in transito perché non trattano il livello applicazione. I firewall filtranti occupano poca CPU, e creano una sopportabile latenza nella propria rete ma non forniscono un controllo a livello di password. Nel caso si intenda far uso di DHCP,o qualche altra assegnazione dinamica dell'IP, si creano dei problemi perchè le regole dei packet filter sono basate sugli indirizzi IP, e quindi è necessario modificarle con i nuovi indirizzi.

Pro:

- economico
- trasparente
- performante

Contro:

- basso livello di sicurezza
- accesso solo ad una parte limitata dell'header IP
- abilità limitata di manipolazione delle informazioni

3.1.2.2 Firewall statefull (Statefull Inspection)

Questi tipi di firewall sono in grado grazie alla facoltà di riconoscere connessioni, trasmissioni e protocolli di consultare dei parametri ulteriori per decidere se inoltrare o meno il pacchetto.
In pratica il firewall è in grado di tener traccia del flusso della comunicazione ottenendo informazioni da tutti i livelli di comunicazione. Lo stato della connessione in corso è di fondamentale importanza per decidere se inoltrare o meno altri tentativi di connessione.
Ad esempio se noi avessimo programmato il firewall in modo che rigetti tutti i pacchetti ICMP in entrata, con la statefull inspection potremmo far procedere i pacchetti di risposta (ICMP Echo Reply) relativi ad una precedente richiesta (ICMP Echo Request) fatta dall'interno.
Vantaggi:

- buon livello di sicurezza
- consapevolezza del livello applicativo e dello stato della comunicazione
- elevate performances
- scalabilità
- trasparenza

La tecnica statefull inspection è stata introdotta in linux dal kernel 2.4 attraverso Netfilter e Iptables.

Esistono tre tipi di firewall a seconda del livello a cui agiscono:

- a livello di rete
- a livello applicazione
- a livello di circuiti

3.1.2.3 Firewall a livello rete

Sono la tipologia di firewall per eccellenza e spesso il nome firewall senza altra specificazione è associato a questa tipologia. Sono normalmente impostati come gateway per le reti. Il packet filtering è possibile grazie alla capacità di questi dispositivi di leggere i pacchetti IP e in particolare le intestazioni (header).
I firewall di questo tipo non possono individuare virus in quanto non agiscono sulla parte dati del pacchetto. Generalmente si tratta di uno screening router che lavora a livello data-link e rete e quindi può filtrare in base a numeri di porta, indirizzi IP, flag TCP,… In particolare nel caso di filtraggio orientato agli indirizzi IP lo scopo è quello di creare una lista nera (black list) contenente gli indirizzi da rigettare. Con questa tipologia di firewall si può agire su:

- indirizzo IP sorgente e destinazione

- tipo di protocollo dei dati (TCP, UDP, ICMP)
- numero di porta (collegate ai servizi)
- pacchetti IP (opzione IP del source routing)
- data e ora

3.1.2.4 Firewall a livello applicazione (Application Proxy)

Sono detti anche Proxy e mettono in comunicazione la rete interna con quella esterna. Essi sono dispositivi che garantiscono maggior protezione alla LAN interna dialogando con server esterni per conto di client interni alla propria rete. Infatti esternamente alla rete viene ricevuto esclusivamente l'indirizzo del proxy e le risposte verrano poi replicate ai client. Alcuni proxy possono memorizzare nella propria cache i dati richiesti, abbassando le richieste di banda e diminuendo il tempo di accesso per il successivo utente che vuole accedere a quegli stessi dati. I protocolli di interesse di cui si può limitare l'uso sono quelli dello strato applicativo e quindi HTTP, FTP, SMTP,…Si ha un server proxy dedicato per ogni servizio applicativo. A differenza dei firewall a livello di rete agiscono sulle informazioni contenute nei dati del pacchetto aumentando la sicurezza e permettendo la conoscenza della comunicazione in corso. Ad esempio i proxy HTTP (web) sono in grado di memorizzare qualsiasi URL visitato, mentre i proxy FTP qualsiasi file scaricato. Possono anche estrapolare parole non comuni dai siti che si visitano o controllare la presenza di virus. I proxy possono inoltre richiedere che l'utente faccia login prima di effettuare la connessione esterna. Questi firewall sono più sicuri dei precedenti perché separano fisicamente la rete esterna e interna, a discapito però della velocità che ne risente. La comunicazione attraverso il proxy richiede infatti due connessioni: una dal client al firewall e l'altra dal firewall al server.

Pro:

- buon livello di sicurezza
- consapevolezza del livello applicativo

Contro:

- ogni servizio richiede un Proxy dedicato
- scalabilità scarsa
- poco performante soprattutto su reti ad alto livello di traffico
- potenzialmente vulnerabile a bug del livello applicativo

3.1.2.5 Firewall a livello circuiti (sessione)

È ancora un firewall a livello applicazione ma in questo caso il proxy ha la funzione di creare il circuito di comunicazione tra client e server senza che le applicazioni ne siano informate. In

sostanza dopo che una connessione ha superato la fase della stretta di mano (handshake), viene registrata in una tabella di comunicazioni ritenute valide. Sulla tabella viene memorizzato:

- l'ID di sessione
- lo stato della connessione (handshake, stabilito, chiuso)
- l'indirizzo IP del mittente
- l'indirizzo IP del destinatario
- l'interfaccia di rete sulla quale giunge il pacchetto

Questa tabella viene utilizzata per far passare dal mittente al destinatario i dati appartenenti alle connessioni presenti nella stessa. In pratica all'arrivo di un pacchetto il firewall controlla se fa riferimento a una riga della tabella e nel caso procede con l'inoltro del pacchetto altrimenti lo scarta.

Una volta che la connessione sarà terminata la riga corrispondente della tabella verrà eliminata.

Un vantaggio di questa tipologia rispetto alla precedente è il fatto che non è richiesto l'utilizzo di client specifici che supportino software proxy. Inoltre è sufficiente gestire un unico server per tutti i protocolli.

I problemi sono:

- difficile collaudo delle regole
- scarse proprietà di controllo degli eventi
- non hanno altre funzionalità

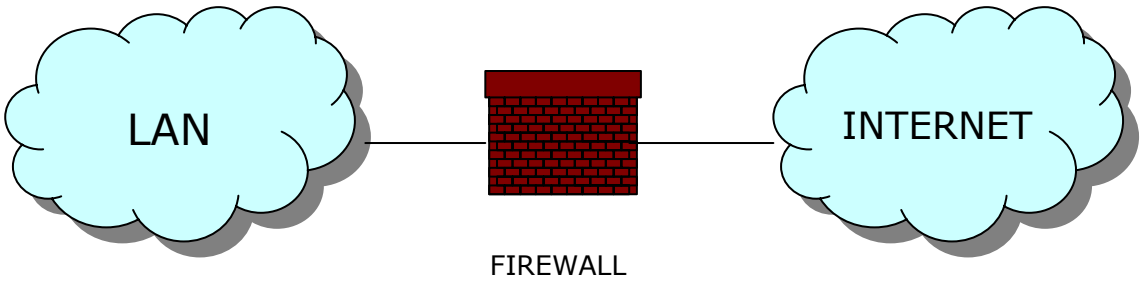
3.1.3 Architettura dei firewall

La combinazione tra screening router, bastion host e firewall proxy per elevate esigenze di sicurezza risulta essere di notevole importanza. È per questo motivo che si sono sviluppate tre architetture che andremo ad analizzare:

- Dual Homed Host
- Firewall Screened Host
- Firewall Screened Subnet

3.1.3.1 Dual Homed Host (Router di controllo)

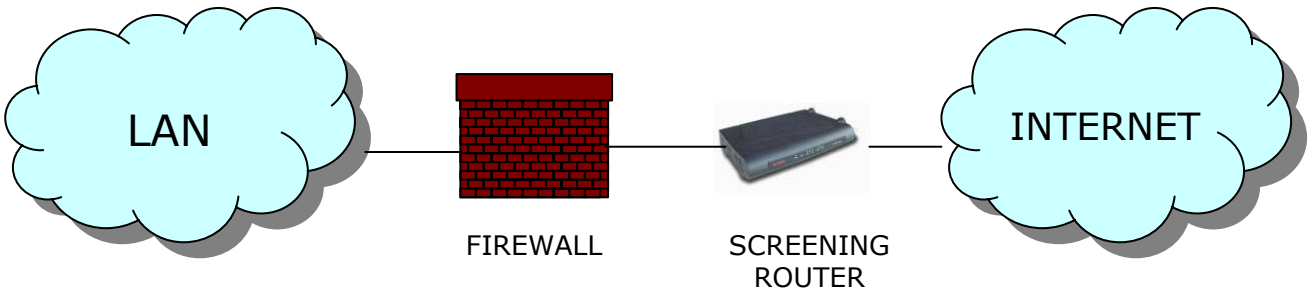
Questa architettura è costituita da un host con due interfacce di rete, una che collega alla rete interna e una che collega alla rete esterna e rappresenta un singolo punto di controllo e di accesso.



A volte questa architettura è presente sottoforma di router con funzioni di filtro. In alcuni casi il firewall Dual Homed Host costituisce di fatto un proxy a livello di applicazione o circuiti. Per il suo funzionamento è necessario disabilitare il routing altrimenti non verrebbe effettuato il packet filtering al livello applicazione perché il pacchetto verrebbe instradato a livello IP. La comunicazione tra le due reti è comunque possibile perché l'instradamento viene fatto a livello applicazione. E' importante comunque che nessuna macchina possa accedere direttamente al Dual Homed Host che rappresenta il punto cruciale su cui si basa la protezione della rete (single point of failure). Nel caso l'attacker riesca ad ottenere un account della rete locale allora tutta la LAN è in pericolo. In particolare nel caso il firewall sia fuori servizio per un guasto, l'attaccante avrà la facoltà di reimpostare l'instradamento per sottoporre la rete ad un attacco. E' intuibile che riabilitando il routing a livello IP, la base su cui si pone questa architettura viene meno. Si tratta comunque di una struttura poco flessibile perché i server pubblici con accesso diretto a internet sono a contatto diretto con il segmento di rete interno.
E' la scelta più utilizzata nel caso di piccole utenze e ditte.

3.1.3.2 Firewall Screened Host (Sistema di controllo)

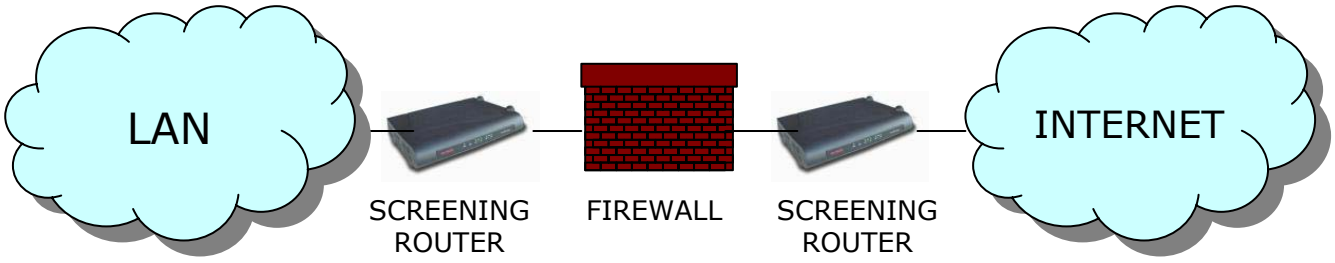
È costituito da una composizione basata su un firewall proxy o un bastion host e uno screening router. In particolare la rete interna sarà collegata al firewall che a sua volta si legherà allo screening router che filtrerà i pacchetti da e per internet. Quindi l'unico sistema raggiungibile da internet risulterà essere la macchina firewall, la quale provvede poi alla comunicazioni con gli altri host interni.



Il firewall quindi non sarà direttamente esposto alla rete esterna ma sarà protetto dallo screening router, mentre la LAN locale per l'accesso a internet vedrà solo il firewall. Per un attacker sarà molto complesso penetrare in questa architettura e arrivare alla rete interna a meno che non riesca a fare login sulla macchina firewall. Nel caso riesca a penetrare dovrà essere in grado di cambiare l'instradamento senza essere scoperto. L'intruso avrà un compito ancora più difficile nel caso non sia garantito l'accesso allo screening router direttamente dalla rete esterna ma attraverso prima il firewall, poi un host interno e poi infine lo screening router.

3.1.3.3 Firewall Screened Subnet (Sottorete schermante o controllata)

In questa architettura che si rivela essere la più sicura, il firewall è protetto dalla rete interna e dalla rete esterna tramite due screening router.



In tal modo il firewall è protetto da un primo screening router (interior router) dai possibili attacchi interni e da un secondo screening router (exterior router) dai possibili attacchi esterni. Quindi la rete interna ed esterna resteranno separate senza avere un collegamento diretto.

Si può anche introdurre il concetto di rete perimetrale (DMZ) con la presenza di uno o più bastion host che possono essere raggiunti da internet per poter offrire i servizi pubblici. La funzione dell'exterior router sarà quindi quella di proteggere la rete perimetrale e la rete interna dal traffico di internet, mentre l'interior router proteggerà la rete interna dalla rete internet e dalla rete perimetrale. Per violare l'architettura l'attaccante dovrebbe essere in grado di riconfigurare la sottorete schermante e la rete interna senza essere scoperto dal firewall.

Questa architettura può inoltre essere realizzata su un unico dispositivo hardware con tre interfacce di rete: una per la rete interna, una per la esterna e una per la rete perimetrale. Il traffico di rete rimarrà separato ma sarà introdotta la debolezza che è presente nell'architettura dual homed host ovvero un singolo punto cruciale in cui si concentra la protezione della rete, che in caso di guasto crea una grave falla di sicurezza, la quale probabilmente non compenserà il risparmio economico e di gestione dato dall'unico dispositivo.

3.1.4 Limiti del firewall

È noto che una gran parte di attacchi informatici proviene dall'interno della rete (insiders attack). Su questi il firewall non può nulla perché non può visionare connessioni che non passano attraverso di esso. Quindi il firewall può garantire solamente una sicurezza "perimetrale".

In particolare non può:

- garantire l'integrità dei dati
- garantire protezione contro ogni tipo di attacco
- autenticare le fonti

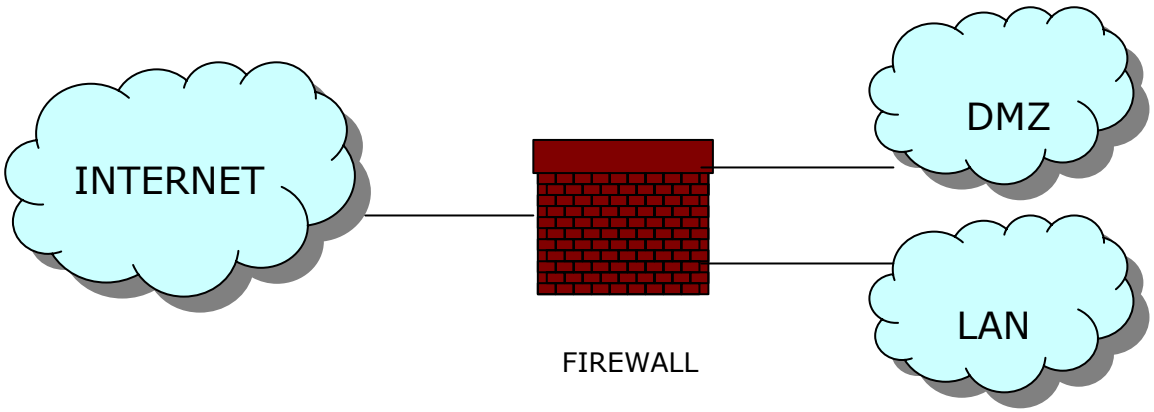
È necessario occuparsi anche di politiche di hardening degli host e dei server. Infatti molti strumenti di attacco potrebbero essere utilizzati dai dipendenti per cercare di aggirare le ACL. Per garantire questo tipo di sicurezza è necessaria una corretta configurazione degli ambienti di lavoro condivisi, delle workstation personali e delle policy aziendali.

3.1.5 DMZ (De-Militarized Zone) e rete militarizzata

La Dmz è una rete posta tra un ulteriore segmento interno protetto e un'altra rete esterna come livello superiore di sicurezza con lo scopo di ridurre il rischio di attacchi alla rete interna nel caso vengano compromessi i punti di accesso a internet. All'interno della rete perimetrale smilitarizzata (DMZ) vengono posti i server pubblici ovvero i server che devono offrire dei servizi all'esterno e sono perciò esposti alla rete internet, e non sono collocati assieme alla rete privata la quale non deve essere visibile dall'esterno. In quest'area vengono generalmente localizzati:

- i web server
- i mail server
- i proxy server (per rendere più sicura la navigazione)

Per proteggere questi dati si può far uso del protocollo SSL.



La rete militarizzata invece coincide con la rete locale privata che conserva i dati privati dell'azienda, viene posta più all'interno e collegata tramite il firewall a internet, ma non è accessibile direttamente dall'esterno in quanto non deve fornire servizi al pubblico. Il firewall in questa zona farà uso di security policy molto restrittive per evitare che la sicurezza dei dati venga minacciata.

3.1.6 NAT (Network Address Translation)

Questa funzionalità garantita anche dal firewall consente la traduzione di indirizzi di rete, ovvero indirizzi IP privati in indirizzi pubblici per modificare, mascherare e dirottare le connessioni di rete. In pratica gli indirizzi dei datagram di rete vengono modificati permettendo così a sistemi privi indirizzi pubblici di essere associati ad uno o più indirizzi IP pubblici, nascondendo i reali indirizzi IP oppure consentendo a più server di rispondere a un unico indirizzo IP. Altra possibilità è l'implementazione di proxy trasparenti. La maggior parte dei firewall sono in grado di offrire anche la NPAT (Network and Port Address Translation) che oltre alla funzionalità di traduzione degli indirizzi IP offre anche la traduzione di porte TCP e UDP.
Non è proprio da considerarsi come tecnica di sicurezza ma comunque consente di aumentare le difficoltà di azione per un attacker, limitando il numero dei possibili obiettivi e nascondendo la reale topologia della rete.
Esistono tre categorie di NAT:
-*IP masquerading* (many to one mapping): consente di mascherare più indirizzi IP di una rete privata in un unico indirizzo IP di una rete pubblica. Questa funzionalità garantisce un risparmio di indirizzi IP pubblici e una maggiore protezione della rete privata.
-*NAT bidirezionale* (one to one mapping): consente ad un host con indirizzo privato, protetto dal firewall, di apparire all'interno della rete pubblica.
-*Port redirect*: consente la redirezione in modo trasparente delle porte TCP o UDP. È utilizzata per pubblicare un unico servizio sulla rete esterna. Un uso possibile di questa tecnologia è la configurazione di un Trasparent Proxy.

Nella tecnologia di Netfilter presente nel kernel Linux, esistono due tipologie di NAT: Source NAT e Destination NAT. Nel primo caso viene modificato l'indirizzo IP sorgente nel datagramma. Ciò avviene subito dopo le procedure decisionali di instradamento. Caso particolare della prima tecnologia è il Masquerading di Linux. Nel secondo caso viene modificato l'indirizzo di destinazione per alterare la destinazione del datagramma prima delle procedure di instradamento. Esempi di questa tecnica sono il port forwarding, il load-sharing e il transparent proxy.

È necessario che Netfilter conosca le regole dei protocolli per lo scambio dei dati, in modo che riesca a modificare i payload e gli indirizzi perciò nel supporto NAT è presente la funzione di protocol checking.

Vantaggi:

- incremento del controllo su connessioni dirette all'esterno, perché i sistemi interni non hanno indirizzi validi al di fuori del segmento di rete interno
- il traffico verso l'interno può essere ridotto nel caso non ci siano associazioni statiche degli indirizzi IP e l'attaccante ha meno possibilità in termini di tempo e traffico
- maschera la topologia della rete interna

Svantaggi:

- sono necessarie informazioni sullo stato della connessione che non sono sempre disponibili
- alcuni protocolli inseriscono gli indirizzi all'interno del payload oltre che nell'intestazione complicando il lavoro(es. FTP, DCC)
- non funziona con protocolli che inseriscono gli indirizzi nel payload e nello stesso tempo garantiscono l'integrità dei dati perché non è possibile per il NAT modificare il payload.
- è necessaria una notevole sincronizzazione

3.2 IDS (INTRUSION DETECTION SYSTEM)

Sono una parte molto importante dell'architettura di sicurezza di rete. Un sistema IDS, noto anche come guardiano di notte, ha la funzione di scovare le intrusioni nel sistema, ovvero individuare e segnalare all'amministratore di sistema la presenza di violazioni del sistema una volta riconosciuto un pattern di attacco noto. Inoltre ha la possibilità di rilevare i servizi offerti dal server. Queste sono le funzioni di base, ma è noto che alcuni IDS possono anche reagire agli attacchi, diventando trasparenti nella rete e lanciando essi stessi dei controattacchi. I dispositivi su cui girano gli IDS vengono detti sensori.

Col passare del tempo si sono create due tipi di applicazioni:

- IDS basati su regole
- sistemi autonomi

I primi confrontano il traffico della rete o le azioni degli utenti con le firme degli attacchi più frequenti nel database di signatures. Una volta effettuato il riscontro e individuata la tipologia di attacco l'IDS segue la procedura impostata dall'amministratore di sistema in quel particolare caso. Da notare che il database di firme deve essere costantemente aggiornato per far fronte anche agli attacchi più recenti.

I secondi riguardano sistemi intelligenti, di autoapprendimento. Il programma viene addestrato a riconoscere eventuali anomalie.

Questa trattazione tratterà solo gli IDS basati su regole. Essi possono essere suddivisi in:

- IDS proattivi
- IDS reattivi

I primi monitorano continuamente la rete o il sistema allo scopo di individuare delle falle di sicurezza e quindi agiscono secondo azioni ben decise preimpostate prima che l'attacco sia stato lanciato.

I secondi sono meno evoluti in quanto non preventivi e si occupano solamente di registrare l'intrusione e allertare l'amministratore di sistema.

Verrebbe scontato chiedersi la differenza tra IDS e firewall, ebbene i sistemi di ID non hanno lo scopo di bloccare il traffico di passaggio, ma quello di rivelare le intrusioni al sistema.

L'IDS deve essere configurato al meglio perché possa agire nel migliore dei modi, in particolare deve conoscere: il o i sistemi operativi, l'attuale versione, le eventuali patch o service pack installati, lo stack di rete, ….

3.2.1 Tipologie di IDS

Esistono tre tipi principali di IDS che si differenziano in base ad alcune caratteristiche:

- Host IDS (HIDS)
- Network IDS (NIDS)
- Stack IDS (SIDS)

3.2.1.1 Host IDS

Gli host IDS hanno lo scopo di monitorare un singolo sistema ad esempio un server controllando processi, applicazioni, e in particolare i files di sistema e i log. In pratica devono individuare violazioni di accesso effettuate da cracker che sono riusciti ad impossessarsi di un account utente che parrebbe legittimo. Quest'azione sarebbe quasi impossibile da rilevare per un amministratore ma non per un IDS, il quale è in grado di tener d'occhio i log in tempo reale e monitorare particolari tipi di attività specifiche come modifiche ai permessi sui files e rilevare trojan e backdoor in maniera efficiente. Devono essere posizionati in prossimità della macchina da sorvegliare, in particolare sullo stesso segmento di rete in modo che vengano attraversati dagli stessi pacchetti che giungono all'host controllato. Il difetto degli host IDS è che non lavorano in tempo reale, quindi rilevano le intrusioni solo dopo un certo periodo di tempo. Possono anche essere installati come modulo software sul sistema da monitorare. In genere per individuare un'intrusione analizzano i log di sistema, l'integrità dei files locali e monitorano i pacchetti destinati all'host controllato.

Esempi: Tripwire, HostSentry, PortSentry.

Un HIDS può segnalare:

- comportamenti anomali, ovvero situazioni in cui i comportamenti degli utenti non possono essere considerati normali (sintomo che qualche cracker si è impossessato di un account utente)
- anomalie basate sul tempo, quando ad esempio un utente si collega in orari non usuali
- anomalie locali, quando un utente esegue una login con indirizzi IP generalmente non utilizzati.

Alcuni HIDS (HostSentry) non fanno uso dei file di log, in quanto essi possono essere facilmente cancellati durante un attacco, ma utilizzano propri sistemi di registrazione crittografati.

Alcuni (tra cui Tripwire) analizzano l'integrità dei file e directory confrontandone le caratteristiche(data di accesso, modifica, checksum,…) con le entry di un database di riferimento, allertando in caso di incongruità.

3.2.1.2 Network IDS

I network IDS hanno una visione più larga perché si occupano del monitoraggio della rete, sniffando i pacchetti TCP/UDP in transito. Sono trasparenti, lavorano in tempo reale, sorvegliano l'intera rete o segmenti di essa e hanno un database proprio di firme (sequenze di pacchetti note) associate a tentativi di intrusione. Quindi ogni pacchetto che passa per la rete sarà sniffato (ne sarà osservato il contenuto) e sarà confrontato con il database di firme.

I NIDS si compongono di tre sottosistemi:

- il packet decoder, il quale si occupa di catturare i pacchetti in transito sulla rete
- il detection engine, che fa il raffronto del traffico con il database di firme
- il logging/alert , che registra il traffico anomalo e avvisa l'amministratore

La posizione degli IDS deve essere strategica, devono essere posizionati ad esempio vicino ai router in modo che siano attraversati da tutto il traffico di rete. Sono stati riscontrati problemi con reti a stella in quanto non si riesce ad accedere a più flussi. Un calo di prestazioni hardware e software si registra in presenza di traffico troppo elevato.

Esempi: Snort, RealSicure ISS.

La maggior parte degli IDS presenta degli elementi in comune:

- l'interfaccia di rete è settata in modalità promiscua, in modo che venga catturato tutto il traffico di rete in transito
- l'utilizzo di filtri specifici per sottoporre il traffico al confronto col database di signatures (firme). Il motore di ricerca per scovare gli attacchi è principalmente di tre tipi:a ricerca di modelli, frequenza o anomalie.

Sono in sostanza degli sniffer e perciò hanno bisogno di analizzare tutto il traffico di rete, ciò significa che, nel caso che un IDS debba occuparsi di monitorare l'intera rete, si deve far uso di hub o meglio ancora di switch con la funzione di PortMirroring attivata (in modo che i pacchetti possano essere sniffati anche se non destinati a quel segmento di rete).

3.2.1.3 Stack IDS

Gli stack IDS si basano sul controllo a basso livello del traffico di rete, in particolare del livello dello stack TCP/IP. Prima che le applicazioni o il sistema operativo accedano ai dati, i SIDS agiscono controllando i datagrammi e decidono quali azioni permettere e quali no. Questa tipologia di IDS rappresenta una via di mezzo con i filtri a pacchetto che mantengono informazioni di stato.

3.2.2 Necessità ed efficacia degli IDS

I sistemi di rilevamento delle intrusioni risultano essere un valido e soprattutto utile strumento per gli amministratori di sistema, che sfruttandone la tecnologia potrà evitare di tener d'occhio per 24 ore su 24 i terminali su cui sono stati lanciati strumenti di analisi del traffico di rete, quali ad esempio tcpdump, arpwatch, dsniff. L'analisi di questi output alla ricerca di attacchi o attività anomale potrà essere evitata a favore di strumenti indipendenti e automatizzati in grado di segnalare le anomalie. Infatti quando vengono rilevate attività sospette, viene avvisato l'amministratore di sistema tramite un alert che può coinsistere in una e-mail, un segnale acustico o una finestra popup contenente la segnalazione. Eventuali riscontri potranno poi essere effettuati con gli strumenti di analisi prima citati, ma è indubbio che i sensori IDS offrano un notevole vantaggio.

Gli IDS hanno un problema fastidioso ovvero l'insorgenza di falsi allarmi in quanto l'algoritmo di rilevamento delle anomalie compie un'analisi statistica che raffrontata con i normali comportamenti ne individua gli scostamenti. Gli errori generati sono:

- falsi positivi: un'azione anomala viene considerata come un'intrusione. Per evitarli bisogna gestire sofisticatamente i files di configurazione.
- falsi negativi: una intrusione viene considerata come un'azione legittima.
- subversion errors: sono generati a causa dell'azione dell'intruso per far creare all'IDS dei falsi positivi allo scopo di portare a termine la sua azione.

Ecco alcuni parametri da tener d'occhio per essere sicuri della bontà di un IDS:

- percentuali di falsi negativi (attacchi che non sono stati rilevati) sul totale degli eventi
- percentuali di falsi positivi (eventi innocui considerati come attacchi) sul totale degli eventi
- potenza di elaborazione (per le prestazioni)
- tipi di protocolli gestiti
- livello di difficoltà per essere mantenuti e aggiornati
- costi di gestione e apprendimento

3.2.3 Tecniche per sottrarsi al controllo di un IDS

La maggior parte delle tecniche di elusione si basano sulla generazione di una notevole quantità di traffico non normale che allerta il programma e lo costringe a prendere dei provvedimenti reiterati continuamente per condurlo ad un buffer overflow o all'esaurimento delle risorse. Per altre lo scopo è quello di desincronizzare il traffico ricevuto dall'IDS da quello ricevuto dal sistema monitorato. I pacchetti analizzati saranno quelli del protocollo TCP perché più degni di attenzione.

Tra le tecniche per evitare il controllo di un IDS oltre al comune attacco DoS troviamo:

- inserzione
- aggiramento
- ambiguità

3.2.3.1 Inserzione

L'attacker inserisce pacchetti falsi che vengono ricevuti dall'IDS ma non dall'host in modo da rendere impossibile all'IDS il riconoscimento della sequenza di attacco. Esempi di pacchetti falsi:

- checksum IP scorretto
- TTL errato perché breve
- header IP errato
- DF flag (Don't fragment)
- fragmentation overlap
- conoscenza del MAC Address dell'IDS
- dati fuori finestra
- …

3.2.3.2 Aggiramento

Se l'IDS non è stato posizionato correttamente, alcuni pacchetti potrebbero non arrivare o essere scartati perché si ritiene che non raggiungano l'host controllato (numero di hop e timeout diversi)

3.2.3.3 Ambiguità

Possono essere creati messaggi ambigui (ad esempio flag degli header posti in diverse combinazioni) che venono ricevuti dall'IDS ma non dall'host o viceversa. Questo perché le implementazioni dello stack TCP/IP possono essere differenti.

3.2.4 Problemi

Quando il dispositivo IDS non funziona a causa di un guasto, o perché è sotto attacco DoS, o per qualsiasi altro motivo si seguono due strade:

-fail open: la rete rimane senza sorveglianza e continua a lavorare come se niente fosse

-fail closed: la rete viene bloccata fino al ripristino dell'attività dell'IDS

Gli IDS devono essere continuamente aggiornati perché la simulazione dei protocolli non è affatto semplice e il database delle firme corre il rischio di diventare obsoleto. Gli HIDS sono più semplici da realizzare e svolgono in genere un lavoro migliore rispetto ai NIDS soprattutto a causa della difficile gestione in particolare anche dovuta al traffico elevato. Naturale conseguenza è che gli IDS

da soli non sono in grado di scongiurare il rischio di intrusioni nel nostro sistema, perciò devono essere integrati con altre tecniche di difesa.

3.2.5 Integrazione con altri strumenti di difesa

Un IDS può essere integrato con altri strumenti al fine di riuscire a garantire una sicurezza migliore. Oltre al firewall e alla sua architettura (comprensiva di interior e exterior router e DMZ), sono importanti le ACL (Access Control List), le policy di sicurezza in generale, le politiche di Hardening del sistema, sistemi crittografici, e l'educazione alla sicurezza di tutti coloro che dovranno avere a che fare con la rete. Non sono da escludere anche *honeypot/honeynet*, i cosiddetti sistemi civetta ovvero reti fasulle create da programmi per far credere all'intruso di essere penetrato nella rete. In questo modo l'attaccante rimarrà imprigionato in questo ambiente fasullo e verrà riconosciuto in breve tempo.

3.3 VALUTAZIONE ACCESSO AI SERVIZI DI RETE

In un sistema Linux sono presenti una serie di servizi di rete che in caso di errata configurazione o in caso non siano strettamente necessari possono contribuire ad aumentare le vulnerabilità del sistema. È consigliabile perciò disattivare i servizi che non servono e configurare al meglio i restanti. I servizi di rete sono necessari affinché possano essere inviate informazioni attraverso la rete tramite l'uso di regole ben stabilite. I servizi in genere sono elencati nel file `/etc/services`

Esempio di una parte del file `/etc/services`:

```
[…]
ftp                     21/tcp
fsp                     21/udp          fspd
ssh                     22/tcp          #SSH Remote Login Protocol
ssh                     22/udp          #SSH Remote Login Protocol
telnet                  23/tcp
smtp                    25/tcp          mail
[…]
```

La prima colonna indica il nome del servizio, quella in mezzo il numero di porta e il protocollo utilizzato e l'ultima a destra indica gli alias per il servizio in questione.

Il programma che si occupa di gestire le richieste del servizio e che agisce in background è detto demone. Questo è avviato assieme agli altri in modo automatico nel momento del boot del sistema e rimane in esecuzione fino alla chiusura del sistema stesso. Un altro modo per avviare i servizi può essere fatto attraverso il demone `xinetd`, il quale è in ascolto su molte porte e si prende in carico di avviare il processo indicato.

3.3.1 Disabilitazione servizi di rete

Quasi tutti i servizi sono avviati tramite il processo `xinetd` o tramite uno script di avvio nella directory `/etc/init.d`. Nel momento in cui viene effettuata una connessione a una porta `xinetd` si occupa di attivare il programma relativo al servizio richiesto e passa ad esso la connessione. La configurazione di `xinetd` è presente nel file `/etc/xinetd.conf`. Nella directory `/etc/xinetd.d` sono contenuti i files che danno l'indicazione delle porte su cui il demone deve mettersi in ascolto e i relativi servizi. Ogni file, che normalmente prende lo stesso nome del servizio, contiene le informazioni necessarie per la propria configurazione. Alcuni servizi sono disattivati di default, nel caso si volesse attivarli è sufficiente editare il file di configurazione apposito e mettere `disable=no` al posto dell'attuale `disable=yes`.

Esempio: stralcio di `/etc/xinetd.d` riferito a `ipop3`:

```
#default: off
#description: The POP3 service allows remote users to access their
mail using an POP3 client
#such as Netscape Communicator, mutt, or fetchmail.
service pop3
{
        disable = yes
        socket_type = stream
        wait = no
        user = root
        server = /usr/sbin/ipop3d
        log_on_succes += HOST DURATION
        log_on_failure += HOST
}
```

Modificando in tal modo la dicitura `disable = yes` alcuni servizi non sicuri come (`/etc/xinetd.d/rlogin` e `rsh`) possono essere disabilitati semplicemente editando il file di configurazione interessato.

3.3.2 Tcpwrapper

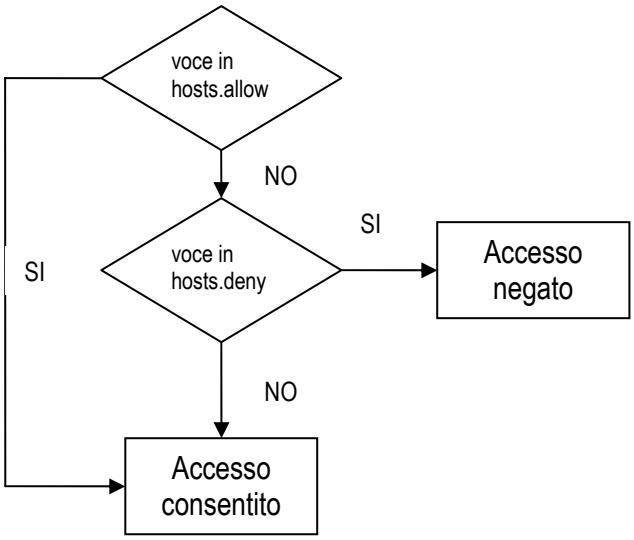
Allo scopo di permettere o meno la disponibilità e l'accesso di un servizio in rete in modo selettivo viene fatto uso dei TCP wrappers. Nelle prime versioni veniva usato il demone wrapper tcpd, ora il supporto wrapper tcp è stato integrato in `xinetd`. I tcp wrappers effettuano dei controlli sulla provenienza della connessione tramite access list prima che sia effettuata stabilita la connessione stessa ad un servizio.

I servizi di rete verranno abilitati solo per gli host consentiti e negati agli altri, come descritto nei files di configurazione `hosts.allow` e `hosts.deny` presenti nella directory `/etc`. In particolare in `hosts.allow` vengono inseriti gli host e gli utenti a cui il servizio è consentito mentre in `hosts.deny` quelli da rifiutare. Non è possibile elencare tutti i singoli indirizzi che potrebbero tentare una connessione alla rete, ma si possono specificare intere sottoreti e gruppi di indirizzi. Durante la scansione di `hosts.allow` e `hosts.deny` il demone si arresta quando trova una voce che corrisponde all'indirizzo IP della macchina della connessione. Si procede verificando i tentativi di connessione nel seguente ordine:

- se l'indirizzo si trova nel file `hosts.allow`, viene consentita la connessione e `hosts.deny` non viene controllato

- se non si trova nel file `hosts.allow` si procede all'analisi di `hosts.deny`, e in caso la voce sia trovata la connessione viene rifiutata
- se la connessione in corso non corrisponde a nessuna delle entry viene comunque consentita.

Funzionamento del tcp wrapper:



I due files sono composti come segue:

`<service>:<client> [:action]`

`<service>` è la lista dei servizi di cui si vuole controllare l'accesso, `<client>` permette di specificare gli host o le reti e `<action>`(opzionale) indica l'azione da seguire quando la regola viene matchata.

È possibile seguire due politiche di default di base:

- la prima permette l'accesso di default, quindi vengono inserite le righe di limitazione di interesse nel file `hosts.deny`, mentre il file `hosts.allow` risulterà vuoto
- la seconda è mirata a negare l'accesso a tutti: in particolare viene inserita la riga

  `ALL:ALL`

  nel file `hosts.deny` in modo da escludere tutti gli utenti da tutti i servizi. L'accesso sarà consentito solo agli utenti indicati nel file `hosts.allow`, che ricordiamo viene analizzato per primo, in caso contrario si applicherà la regola di default nel file `hosts.deny`.

Purtroppo tale strumento non è valido per tutti i servizi di rete e rappresenta un semplice mezzo di protezione, ma viene surclassato dall'introduzione del firewall. Quest'ultimo si rende necessario per proteggere a monte i pc della rete locale, per rendere flessibili le politiche di accesso in base ai servizi lavorando ad un livello più basso (livello rete), e impostare diverse regole di filtro per il traffico esterno e interno alla rete.

3.4 NETWORK SCANNING

La parte fondamentale per un attacco è ricavare più informazioni possibili sulla rete bersaglio, operando una attenta e precisa scansione. In particolare è necessario venire a conoscenza della topologia della rete, degli host, dei sistemi operativi, dei servizi offerti,… Inoltre è importante tentare di scoprire quali sono e dove sono posti i sistemi di difesa della rete obiettivo, come firewall o IDS. Tutto ciò è necessario per scoprire i punti deboli e sferrare un attacco mirato senza essere scoperti. Da notare poi che la giustizia americana non considera il network scanning come reato perché non mette a repentaglio la disponibilità della rete.

Va precisato che gli strumenti che andremo ad analizzare possono essere utilizzati nel bene o nel male, infatti può farne uso sia un attaccante per acquisire informazioni per lanciare l'attacco, sia l'amministratore di sistema per scovare tentativi di intrusione.

3.4.1 Network surveying

Con network surveying si intendono gli strumenti atti a raccogliere informazioni sulla rete. Tra questi quelli che analizzeremo sono:

- web
- whois e host
- traceroute

3.4.1.1 Web

La prima fonte per una ricerca dei dati di interesse per penetrare nel sistema è il web, infatti molte aziende pubblicano informazioni di natura sensibile sulla propria homepage.

3.4.1.2 Whois e host

La seconda fonte è costituita dal database dell'InterNic, un'organizzazione che si occupa di gestire i domini internet, il cui database può essere visionato tramite più vie: indirizzi web, quali www.arin.net o www.ripe.net (ente preposto per l'assegnazione degli indirizzi IP per l'Europa) oppure con il comando `whois`. Le informazioni che si possono ricavare sono i domini registrati da un'organizzazione e gli indirizzi di rete corrispondenti, la sede della società, i numeri di telefono e fax, i nomi dei titolari e i loro indirizzi e-mail. L'informazione più importante è comunque costituita dall'indirizzo dei DNS server, che può essere preziosa nel caso in cui il server DNS consenta di effettuare all'esterno un zone transfer, ovvero il trasferimento su connessione TCP (normalmente la connessione avviene su UDP) dell'intero database del dns utilizzato per la traduzione tra hostname e indirizzi IP. Generalmente la zone transfer è utilizzata dai dns secondari per aggiornare il database

e bisogna garantire tramite delle access list che questa operazione venga fatta esclusivamente da essi.

Esempio:

L'attacker usa il proprio client DNS per risolvere l'hostname.

```
$host www.nomehost.it
www.nomehost.it has address 1.1.1.3
```

Dopo aver individuato l'indirizzo IP viene fatta una query al database del RIPE.

Esempio:

```
$whois –h whois.ripe.net 1.1.1.3
```

Che visualizzerà in output informazioni come la sede della società, numeri telefonici, fax, indirizzi email, nomi dei titolari

3.4.1.3 Traceroute

Le prime due fonti sono accessibili pubblicamente e sono utili per iniziare ad addentrarsi nel privato della rete bersaglio. Qui viene utile il comando `traceroute` che consente di costruire una mappa dettagliata della rete intorno, mandando più pacchetti alla destinazione scelta, ognuno con un ttl (time to live) incrementato di uno rispetto al precedente. Ogni host che sarà attraversato dal pacchetto risponderà con un ttl exceeded. In questo modo si ricostruirà la mappa di rete che interpone la nostra postazione a quella bersaglio. Se l'host bersaglio è interno gli indirizzi precedenti saranno anch'essi della rete interna e in particolare si scoveranno quelli dei firewall e dei router. Il comando `traceroute` consente anche di specificare il protocollo di interesse, quindi anche ICMP e UDP con la specifica anche delle porte in questo ultimo caso. Permette di specificare inoltre il percorso che dovranno seguire i pacchetti (source routing). Per ogni host attraversato sono indicati inoltre espressi in millisecondi i tempi di percorrenza totali, di andata e di ritorno.

Esempio: stralcio dell'output del comando traceroute

```
$traceroute –n www.lugman.org
Traceroute to www.lugman.org (62.94.8.185), 30 hops max, 38 byte packets
1 192.168.2.1 0.610 ms 0.486 ms 0.505 ms
2 192.168.100.1 155.384 ms 180.925 ms 173.967 ms
3 217.141.110.199 82.752 ms 63.607 ms 181.068 ms
[…]
```

3.4.1.4 Ping

Si tratta di uno strumento che viene utilizzato per verificare il funzionamento della rete. Viene inviato un pacchetto ICMP Echo Request al segmento di rete indicato, e in caso di risposta l'host contattato invierà un ICMP Echo Reply e ciò significa che è funzionante. L'output della risposta a schermo è composto le dimensioni in bytes del pacchetto inviato con indirizzo di appartenenza, e una serie di parametri tra cui il tempo impiegato per l'andata e il ritorno. Verranno continuamente visualizzate le righe di output a schermo perché il test viene ripetuto regolarmente. Per interrompere il test e vedere le statistiche della trasmissione effettuata, quali tempo minino medio e massimo di ricezione è necessario premere la combinazione di tasti CTRL+C.

Esempio:

```
$ping 192.168.0.20
64 bytes from 192.168.0.20: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=0.056 ms
```

In caso di risposta di questo tipo:

```
connect: Network is unreachable
```

significa che il pacchetto PING non riesce ad essere instradato, nel caso sarebbero da verificare le regole di routing.

Altra risposta potrebbe essere:

```
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable
```

nel caso in cui l'interfaccia non risponde significa o che il routing non è corretto oppure che il pacchetto è stato bloccato dal firewall magari attraverso la disabilitazione del protocollo ICMP.

Opzioni:

`-R <indirizzo_ip>`: registra il percorso seguito dal pacchetto sia all'andata che al ritorno visualizzando gli indirizzi IP attraversati (al max ne visualizza 9 perciò è più indicato il comando `traceroute`).

3.4.2 Port scan e Network status

Andremo ora ad analizzare due tools, uno per fare il port scanning ovvero per la scansione delle porte aperte sulla rete e l'altro per visualizzare le connessioni di rete attive. Tra questi analizzeremo:

- Nmap (Network security scanner)
- Netstat

3.4.2.1 Nmap (Network security scanner)

Il port scanning `nmap`, acronimo di Network mapper, è uno strumento che consente di scandire le reti TCP/IP. È considerato uno dei metodi indispensabili per il recupero di informazioni necessarie nell'ambito di un security audit o di un penetration test. È utilizzato per identificare gli host attivi, determinare il tipo e la versione del sistema operativo e fare uno scan delle porte per individuare i servizi attivi.

Può accettare in ingresso sia un indirizzo di rete, sia un file contente le liste degli indirizzi da controllare.

Per verificare quali sono gli host attivi è sufficiente digitare da linea di comando:

```
$nmap <target>
```

`<target>` è un indirizzo IP, un hostname o un elenco di indirizzi

Per quanto riguarda il TCP/IP fingerprinting, `nmap` consente attraverso l'opzione `-o` di venire a conoscenza della tipologia e versione di sistema operativo presente.

```
$nmap -o <target>
```

Tipologie di scanning principali:

- TCP connect() scan: essa è il tipo di scansione di default di nmap. Viene utilizzata la chiamata di sistema connect() per instaurare una connessione sulle porte in ascolto. La porta sarà considerata non raggiungibile quando la porta non è in ascolto. Lo scan sarà segnalato nei log e perciò facilmente rilevabile.
- TCP SYN scan: si tratta di uno scan half open, perchè la connessione avviene a metà. Viene spedito un pacchetto con flag SYN attivo e come risposta si otterrà un SYN/ACK nel caso che la porta sia in ascolto, mentre un RST in caso contrario. Quando si riceve il SYN/ACK viene inviato un RST per concludere la connessione. Questa tecnica è certamente più silenziosa dell'altra e inoltre necessita dei privilegi di root.
- Stealth FIN scan: questa tecnica utilizza un pacchetto col flag FIN attivo. Come indicato nella RFC 793 (Transmission Control Protocol pagina 64), le porte chiuse devono rispondere allo scan con un pacchetto RST, mentre quelle aperte devono ignorare il pacchetto. Se lo scan risponde trovando delle porte aperte, allora non si tratta di una macchina Windows, la quale non rispetta lo standard dello RFC. Altri sistemi che violano lo standard sono Cisco, BSDI, HP/UX, MVS e IRIX.
- Xmas Tree scan: questo scan invia un pacchetto con flag FIN, URG e PUSH attivi. Per il resto vale ciò che è stato detto per il FIN scan

- Null scan: questa tecnica disattiva tutti i flag. Per il resto vale ciò che è stato detto per lo scan FIN e lo Xmas Tree.
- Ping scanning: questo scan manda pacchetti ICMP Echo Request all'indirizzo IP indicato con lo scopo di sapere quali sono gli host attivi sulla rete. Può altresì mandare un pacchetto TCP ACK,e se la risposta è un RST significa che la macchina è attiva. In alternativa è possibile mandare un pacchetto SYN e ricevere come risposta un SYN/ACK o un RST. Di default venie usata sia la tecnica ICMP che quella ACK.
- UDP scan: questa tecnica è volta allo scopo di individuare quali porte UDP sono aperte sull'host. Vengono inviati pacchetti UDP delle dimensioni di 0 byte ad ogni porta del bersaglio. Se come riposta si ottiene un pacchetto ICMP Port Unreachable allora si può dedurre che la porta è chiusa, in caso contrario è da considerarsi aperta. Lo scan UDP risulta essere lento.
- ACK scan: questo scan viene utilizzato in genere per scovare le regole che fanno capo al firewall ed è utile per stabilire se si ha di fronte un firewall stateless o statefull. Viene inviato un pacchetto ACK casuale alle porte indicate e come risposta si può avere un pacchetto RST oppure un pacchetto ICMP Port Unreachable o nulla. Nel primo caso la porta viene identificata come non filtrata, mentre negli altri casi è detta filtrata. Generalmente `nmap` non mostra le porte non filtrate perciò se nell'output non sono presenti porte significa che lo scan è stato compiuto e sono stati restituiti dei pacchetti RST. Lo scan chiaramente non visualizza porte aperte.
- RPC scan: questa tecnica di scanning viene utilizzata assieme agli altri metodi di port scan di `nmap`. Vengono trovate tutte le porte TCP e UDP aperte e vengono mandati poi pacchetti NULL allo scopo di determinare se si tratta di porte RPC, il programma e la versione utilizzata.

Gli scan stealth sono meno performanti ma di solito non lasciano tracce nei log, mentre scansioni del tipo TCP connect() inseriscono delle entry nei files di log (tipicamente `var/log/messages`).

Opzioni principali:

`-sT`: TCP connect() scan (default se non privilegiato)

`-sS`: TCP SYN scan (default con privilegi di root)

`-sF`: Stealth FIN scan (solo con privilegi di root)

`-sX`: Xmas Tree scan (solo con privilegi di root)

`-sN`: Null Scan (solo con privilegi di root)

`-sP`: Ping scanning

-sU: UDP scan (richiede privilegi di root)

-sA: ACK scan

-sR: RPC scan

-o: TCP fingerprinting

3.4.2.2 Netstat

Attraverso questo comando si può avere una visione generale delle porte con servizi attivi in ascolto. Inoltre si può determinare da dove provengono le connessioni.

Opzioni:

-a(all) mostra tutte le porte

-t mostra le porte TCP

-u per le porte UDP

-p(solo per root) mostra il PID del processo

-n mostra l'indirizzo numerico

Le intestazioni delle colonne dell'output di netstat evidenziano:

Proto - il protocollo utilizzato dalla connessione

Recv-Q - il numero dei byte non ancora copiati del programma utente

Send-Q - il numero di byte non riconosciuti dall'host

Local Address - indica l'indirizzo locale di provenienza

Foreign Address - indica l'indirizzo verso cui la connessione è diretta e dopo l'indicazione ":" viene messa la porta utilizzata

State - indica lo stato della sessione della porta

PID - indica l'id del processo e nome del programma del processo utilizzato

Esempio:

```
$netstat -atndp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address         Foreign Address    State       PID/Program name
tcp   0      0      0.0.0.0:21            0.0.0.0:*          LISTEN      32522/oftpd
tcp   0      0      217.133.222.121:39126 80.17.149.253:6667 ESTABLISHED 6146/irssi
```

In questo esempio si può vedere che la prima riga dell'output riguarda il servizio ftpd che è in ascolto(LISTEN) su tutte le interfacce(0.0.0.0) sulla porta 21. La seconda riga riguarda una connessione IRC stabilita (ESTABLISHED) verso il server irc.oltrelinux.com.

Tabella degli State:

STATO	DESCRIZIONE
ESTABLISHED	La porta ha stabilito una connessione
SYN_SENT	La porta sta cercando di instaurare una connessione
SYN_RECV	La richiesta di connessione è stata ricevuta dalla rete
FIN_WAIT1	La porta è chiusa e la connessione in via di conclusione
FIN_WAIT2	La connessione è chiusa ed è in attesa di conferma dall'altra parte
TIME_WAIT	La porta attende la conferma della chiusura
CLOSED	La porta non è in uso
CLOSE_WAIT	La parte remota chiude la connessione e attende conferma dall'altra parte
LAST_ACK	La porta e la connessione sono chiuse in attesa di conferma finale
LISTEN	La porta è in attesa di una connessione in entrata
CLOSING	Entrambe le estremità sono chiuse ma non tutti i dati sono stati inviati
UNKNOWN	Lo stato della porta è sconosciuto

3.4.3 Sniffing

Ora analizzerò i principali sniffer del mondo Linux, i quali hanno lo scopo di visualizzare i dettagli relativi ai pacchetti che sono in transito sulla rete. In particolare vedremo:

- Tcpdump
- Ethereal

3.4.3.1 Tcpdump

Si tratta di uno strumento per l'analisi del traffico della rete. Permette di visualizzare il contenuto dei pacchetti intercettati dall'interfaccia di rete ottenendo una versione sintetica o anche il contenuto in esadecimale. Inoltre è possibile applicare dei filtri ai pacchetti. Per interpretare il contenuto dei pacchetti è necessario avere una buona conoscenza dei protocolli utilizzati e della loro struttura.

```
$tcpdump [opzioni] [espressione]
```

Le opzioni più importanti sono:

`-i (interface)`: per specificare l'interfaccia di rete su cui ascoltare, se non specificata di default tcpdump si mette in ascolto sulla prima (`lo` sta per loopback)

`-n (numeric)`: per non convertire gli indirizzi IP di rete e il numero delle porte in nomi

`-s n_byte (split)`: permette di definire il numero di byte del pacchetto da visualizzare(di default 68 byte)

`-w file (write)`: l'output viene salvato nel file specificato

`-r file (read)`: per leggere l'output salvato precedentemente nel file

`-x (exa)`: per leggere i pacchetti in formato esadecimale

`-F file (filter)`: permette di inserire l'espressione di filtro per i pacchetti contenuta in un file

Esempi:

```
$tcpdump –i eth0
```

Visualizza su terminale l'output dei messaggi intercettati dall'interfaccia

```
$tcpdump –n –i eth0
```

Come prima con la differenza che gli indirizzi di rete e le porte vengono indicate numericamente

```
$tcpdump –x –i eth0
```

I dati dei pacchetti vengono espresso in forma esadecimale

3.4.3.2 Ethereal

È un tool grafico simile a tcpdump ma più potente. Dopo averlo lanciato da riga di comando basta andare nel menù "Capture" e premere "Start" e scegliere la scheda di rete da sniffare, o anche tutte tramite "Any". Verranno quindi raccolte le informazioni che passano sulla rete e verranno classificate in base al protocollo. Dopo aver premuto "Stop" verranno visualizzati i dettagli dei pacchetti che sono transitati, al contrario di tcpdump che visualizza solo le intestazioni del pacchetto.

3.5 I LOG E L'ANALISI FORENSE

Dopo aver analizzato i mezzi di difesa del sistema, è necessario raccogliere e gestire i files di log.

Per visionare le attività di rete e capire se qualche intruso è entrato nel sistema e di conseguenza sapere che operazioni ha svolto all'interno, è necessario verificare se il nostro sistema presenta delle anomalie. Uno dei metodi per farlo consiste nell'analisi dei files di log, che sono semplicemente files di testo dove viene registrato ciò che succede nel sistema. Le informazioni che vengono raccolte comprendono errori generati dalle applicazioni, il numero di riavvii del sistema, password cambiate, ecc. L'accurata analisi di questi files può evidenziare un'eventuale intrusione, sempre che l'attacker non sia stato in grado prima di cancellare le proprie tracce modificando i log. Di solito i log sono memorizzati sui dischi fissi dei computer o in ambienti di raccolta specifici in caso di dispositivi proprietari tramite alcune tecnologie tra cui syslog e trap SNMP. Di solito in questo modo si ha disorganizzazione perché i log sono memorizzati in posti diversi in base al dispositivo di produzione, per cui la consultazione e revisione risulta essere lenta e complessa, anche per i dati che risultano essere disomogenei.

L'analisi forense dei log consiste nell'estrazione di dati significativi da database contenenti informazioni che appaiono come eterogenee. La tecnica non si limita solamente a visionare le tracce di eventi che sono accaduti, ma cerca di intervenire sul problema e in qualche modo di riuscire ad evitarlo. L'analisi forense è basata sulla centralizzazione dei dati per gestire i dati da unica postazione, sulla raccolta di dati da più strumenti tipo firewall, IDS, router e sistemi operativi e sulla correlazione degli stessi. Data l'eterogeneità dei dati, e le diverse scale di allarme dei vari dispositivi, è necessaria la normalizzazione dei dati. La correlazione dei dati garantisce una vista logica omogenea.

Il data mining è la fase volta all'estrazione di informazioni aggregate tramite delle associazioni nascoste individuate tra informazioni raccolte nel database. Vengono forniti dei report che forniscono dettagli che risultano utili per la sorveglianza della rete (drill down).

Sarà presente poi un sistema di notifica che in caso di eventi anomali potrà emettere un trap snmp, spedire un sms o una e-mail in modo da far scattare le contromisure manuali o automatiche. Naturalmente essendo dotati di una real time console si riduce il tempo necessario per ripristinare l'anomalia. Abbiamo così creato un SIM (security information management).

Log fondamentali sono `utmp`, che registra informazioni su chi sta usando la macchina in quel momento e `wtmp` che contiene le date degli ultimi login. Questi log riguardano come detto i login e sono salvati in formato binario. Vengono interrogati tramite i comandi `who`, `w` e `lastlog`.

Per esempio per vedere la lista degli ultimi tre utenti (nome utente, terminale, data e ora e indirizzo IP) che han fatto login sulla macchina:

```
$who /var/log/wtmp | tail –n 3
mainman pts/3 Jun 25 11:51 (:0)
root    tty1  Jun 25 11:52
mainman pts/4 Jun 25 11:53 (192.168.0.1)
```

I files di log si trovano generalmente nella cartella `/var/log/` e vengono gestiti dal demone `syslogd`, il quale garantendo una gestione centrale, si occupa di ricevere i messaggi dagli altri programmi e di scrivere sui log di riferimento. Può essere utilizzato come sistema centralizzato anche verso vari sistemi in rete, accettando messaggi provenienti da altre macchine per consentire un'unica revisione. Oltre a fornire una organizzazione più efficiente, grazie alla centralizzazione per un eventuale attaccante risulterà difficile modificare i file di log che saranno presenti in un'unica postazione. A volte questa postazione può svolgere unicamente la funzione di loghost, in questo modo non figurando con altri servizi, essa sarà difficilmente attaccata.

3.5.1 Redirezione dei log su un loghost

Per reindirizzare i log su un'unica postazione è necessario modificare il file di configurazione `/etc/syslog.conf`.

Nel file sono presenti due colonne di informazioni: in quella di sinistra sono presenti una lista di messaggi e priorità separati da punto e virgola, la colonna destra indica il file di log dove vanno registrati i messaggi. Per indirizzare il file di log all'host prescelto occorre indicare al posto della dicitura del nome del file, il nome della macchina preceduta dal carattere @. Ad esempio @logmachine. È utile indicare un alias per la macchina loghost in modo da rendere più semplici eventuali successive modifiche della destinazione.

Output del file `syslog.conf`:

```
$cat syslog.conf
#Log all kernel messages to the console.
#Logging much else clutters up the screen.
#kern.*                                        /dev/console
#Log anything (except mail) of level info or higher.
#Dont'log private authentication messages!
*.info;mail.none;authpriv.none;cron.none    /var/log/messages
#The authpriv file has restricted access
authpriv.*                                     /var/log /secure
#Log all the mail messages in one place
```

```
mail.*                                          /var/log/maillog
#Log cron stuff
cron.*                                          /var/log/cron
#Everybody gets emergency messages
*.emerg                                         *
#Save news errors of level crit and higher in a special file.
uucp,news.crit                                  /var/log/spooler
#Save boot messages also to boot.log
local7.*                                        /var/log/boot.log
```

File modificato con la dicitura logmachine:

```
#The authpriv file has restricted access
authpriv.*                                      @logmachine
#Log all the mail messages in one place
mail.*                                          @logmachine
```

3.5.2 Formato dei files di log

Ogni riga rappresenta un messaggio registrato da un programma o un servizio.

Esempio:

May 03 10:08:02 Prova CRON[1968]: (pam_unix) session opened for user mail by (uid=0)

May 03 10:08:02 Prova CRON[1968]: (pam_unix) session closed for user news

May 03 10:08:02 Prova CRON[1968]: (pam_unix) session closed for user mail

May 03 10:17:01 Prova CRON[2028]: (pam_unix) session opened for user root by (uid=0)

Il messaggio è costituito da:

- data e ora di registrazione
- nome della macchina da cui proviene
- nome del programma o servizio di riferimento
- numero del processo tra parentesi quadre del programma o servizio
- breve messaggio di testo

Nel caso si noti un numero notevole di tentativi di connessione a un servizio dall'esterno è probabile che un intruso sia tentando di penetrare nel sistema.

3.5.3 Analisi dei files di log

Per visualizzare i log di sistema basta fare uso di programmi come `more` o `less`. Per visionare le ultime righe si può usare il comando `tail`. Spesso questo comando viene usato con l'opzione `-f`, la quale permette la visualizzazione dell'output in tempo reale man mano che vengono registrate nuove entry nel file.

```
$tail -f /var/log/messages
```

In tal modo modo è più facile rilevare un'intrusione.

Essendo questi files lunghi, si può usare il comando `cat` in combinazione con `grep` per ricercare all'interno del file le parole di interesse.

Esempio mouse:

```
$cat /var/log/XFree86.0,log | grep mouse

(II) LoadModule: "mouse"
(II) Loading /usr/X11R6/lib/modules/input/mouse_drv.o
(II) Module mouse: vendor = "The XFree86 Project"
(==) ATI(0): Silken mouse enabled
```

3.5.4 Logwatch

Questo programma permette una più semplice analisi dei files di log e la generazione di report. È modulare, facilmente adattabile e configurabile, non funziona in real-time, infatti una volta eseguito analizza i log e manda il report tramite e-mail (normalmente all'utente root). Logwatch segnala i tentativi di intrusione, e gli eventi anomali sul sistema, ma non può nulla nel caso che l'attaccante abbia fatto in tempo a cancellare le proprie tracce dai log.

Configurazione

Il file di configurazione `logwatch.conf` si trova nella directory `/etc/log.d`

Le impostazioni di default sono:

`LogDir = /var/log` – indica dove risiedono i files di log

`TmpDir = /tmp` – indica la directory in cui vengono messi i files temporanei

`MailTo = root` – indica l'utente al quale deve essere indirizzata la mail col report. In caso di utente locale basta il nome, in caso di utente esterno si può indicare l'indirizzo e-mail completo

`Print = No` – se settato a yes redireziona l'output a video anzichè inviare la mail

`Save = /tmp/logwatch` – se impostato, l'output viene salvato nel file indicato

`Archives = Yes` – indica di cercare anche nei log archiviati

`Range = yesterday` – indica quando deve essere effettuata l'analisi dei log. Può essere "`yesterday`" per il controllo dei files del giorno prima, "`today`" per il controllo dei files del giorno, "`all`" per analizzare tutti i log

`Detail = Low` – indica il livello di dettaglio dei report. Può essere "`Low`", "`Med`" o "`High`"

`Service = All` – indica per quali servizi deve essere effettuato il controllo dei log. "All" indica tutti, ma si possono specificare servizi singolarmente

`LogFile = messages` – indica uno specifico file di log da analizzare

`mailer = /bin/mail` – indica quale mailer utilizzare

Esempi:

```
$logwatch --print --detail High --archives --range All
```

Stampa a video invece che inviare via mail, con il massimo dettaglio, includendo anche i log archiviati, tutti i messaggi di ogni data

```
$logwatch --save logwatch.txt --range Today
```

Salva sul file `logwatch.txt` l'output relativo alla giornata corrente

Per migliorare il monitoraggio è consigliabile dirigere tutto il traffico ad un unico host con solo quella funzione, mandare più copie su più computer per maggiore affidabilità, criptare files importanti e limitare l'accesso alle macchine dove risiedono i log.

3.6 CRITTOGRAFIA: INTRODUZIONE

La crittografia è una vera e propria scienza atta a cifrare messaggi in modo da garantire la riservatezza e impedire ad estranei la leggibilità di informazioni private. Assieme alla crittoanalisi fa parte della crittologia, la scienza che studia i messaggi segreti. Mentre la crittografia si occupa di rendere illeggibili i messaggi agli intrusi, la crittonalisi si occupa di violare la riservatezza dei messaggi. Per cifrare e decifrare i messaggi scambiati vengono utilizzate delle chiavi che saranno in possesso del mittente e del destinatario. Gli algoritmi di crittografia si suddividono in simmetrici e asimmetrici. La crittografia a chiave simmetrica utilizza la medesima chiave sia per codifica che decodifica. In caso di crittografia pubblica sono presenti due tipi di chiavi: una privata e una pubblica. La chiave pubblica del destinatario viene utilizzata dal mittente per codificare il messaggio, in questo modo il ricevente utilizzando la propria chiave privata riuscirà a decodificare il messaggio.

3.7 PASSWORD: INTRODUZIONE

Per garantire la sicurezza ed evitare accessi indesiderati è necessario adottare una politica intelligente di password. Queste sono facilmente identificabili attraverso attacchi basati su dizionari, facendo uso di una lista di parole comuni (wordlist) o usando il metodo della forza bruta tentando svariate combinazioni. Molto spesso le password sono facilmente scopribili, infatti vengono usati generalmente nomi di città, di persona, date di nascita, numeri di telefono. Sarebbe più prudente adottare password alfanumeriche (mixed case) e con caratteri speciali come asterischi, spazi, parentesi, segni di interpunzione. Importante non utilizzare come password la parola usata come nome utente.

Altro fattore da considerare è l'utilizzo della password in applicazioni con autenticazione in chiaro, come ad esempio ftp, telnet, pop3. Una soluzione è costituita dall'utilizzo di applicazioni che fanno uso della crittografia. Infatti OpenSSH può benissimo essere usato al posto di telnetd e ftpd e fa uso di algoritmi crittografici sia per l'autenticazione(RSA) che per la comunicazione (3DES o Blowfish). OpenSSH è in ascolto sulla porta 22 e viene utilizzato attraverso i comandi `ssh` e `sftp`.

Altro strumento di protezione è SSL che è utilizzato dai browser web per transizioni sicure e per l'autenticazione al server pop nel client di posta.

3.8 VPN (Virtual Private Network): INTRODUZIONE

Si tratta di una rete privata virtuale cioè una rete in cui i datagrammi interni passano attraverso reti esterne senza che ciò sia visibile dall'interno. È chiaro che le comunicazioni appariranno all'esterno criptate per evitare che siano lette da estranei. La rete virtuale è quindi spesso crittografata e accetta il traffico solo verso e da entità note. Sono spesso usate per connettere attraverso Internet personale che lavora da casa ad un rete interna di una azienda. Le VPN offrono vantaggi economici perché consentono di collegare diverse reti disperse in internet garantendo autorizzazione, accesso e accounting senza l'utilizzo di dispendiosi collegamenti dedicati.

Vantaggi:

- segretezza delle comunicazioni per impedire l'individuazione dei sistemi che stanno comunicando e che vengano intercettati i dati trasmessi
- forniscono strati di sicurezza a protocolli che non usano filtri di pacchetti o proxy

Svantaggi:

- trasmettono su canali pubblici, perciò in caso di compromissione la rete privata non sarà più al sicuro
- si rende necessario proteggere un altro segmento di rete, magari realizzando una rete perimetrale con un punto di accesso e controllo alla rete interna separato.

4. SPERIMENTAZIONE PRESSO IL LABORATORIO RETI

In questa parte della trattazione verremo a conoscenza di come utilizzare uno strumento del mondo Linux per la gestione della sicurezza aziendale. In particolare verrà utilizzato il tool "`iptables`" per la configurazione di un semplice firewall.
La realizzazione della parte pratica è stata eseguita su macchine con sistema operativo Linux della distribuzione Fedora Core 3.

4.1 IMPLEMENTAZIONE DI UN FIREWALL TRAMITE IPTABLES

4.1.1 La tecnologia Netfilter/iptables

Netfilter è uno strato software del kernel Linux che si occupa della gestione di tutte le politiche di Packet Filtering, di Network Address (e Port) Translation (NAT e NAPT) e manipolazione dei pacchetti.
Iptables si basa sul concetto di tabelle, catene e regole. Più precisamente una tabella è formata da catene, le quali a sua volta sono formate da regole. Nella tecnologia iptables esistono tre tabelle:

- nat: consente l'effettuazione del NAT di indirizzi IP sorgente e destinazione, e delle porte per pacchetti TCP e UDP
- filter: consente il filtraggio dei pacchetti che vengono generati o ricevuti dal firewall oppure che lo attraversano
- mangle: consente l'alterazione dell'header IP (per esempio campi TTL e TOS) per aumentare la priorità o marcare alcuni pacchetti con lo scopo di monitorarli (log)

Ogni tabella è costituita da delle catene predefinite che consistono in una lista di regole che possono corrispondere a un insieme di pacchetti.
In particolare per la tabella nat esistono le catene:
PREROUTING: per operazioni di NAT appena il pacchetto raggiunge il firewall
OUTPUT: per operazioni di NAT dei pacchetti generati in locale
POSTROUTING: per operazioni di NAT una volta che i pacchetti sono usciti dal firewall

Per la tabella filter le catene sono:
INPUT: contiene le regole di filtraggio per i pacchetti che hanno come destinazione il firewall

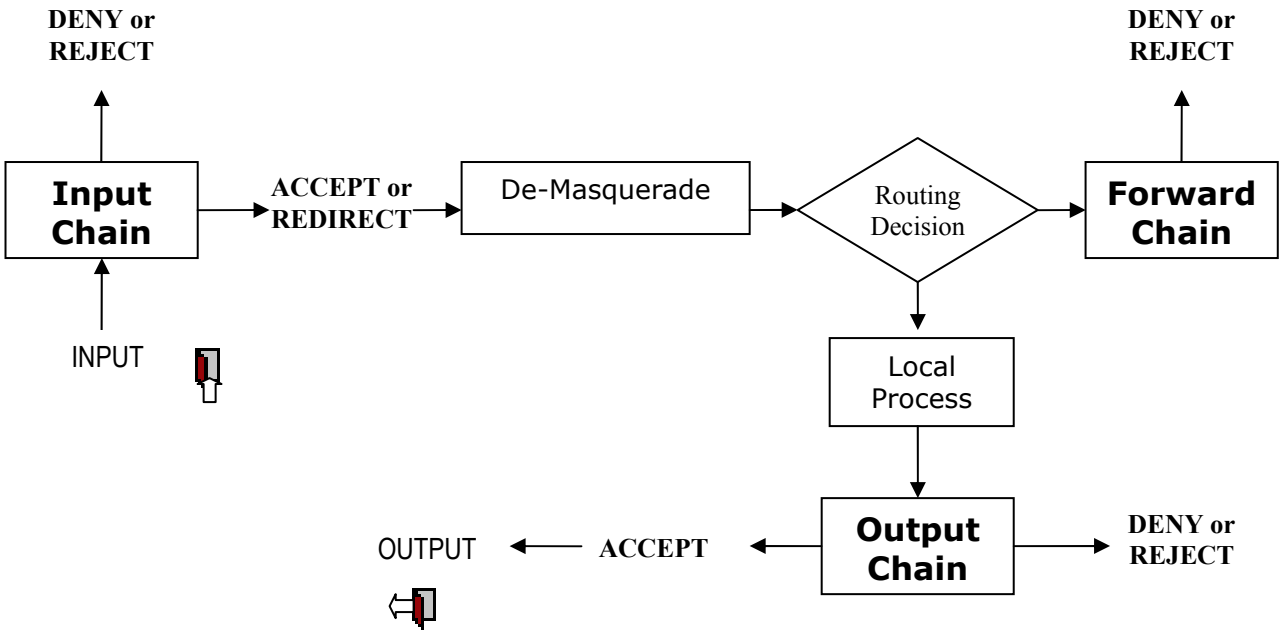
FORWARD: contiene le regole di filtraggio per i pacchetti che attraversano il firewall (ad esempio percorso LAN-Internet e viceversa)
OUTPUT: contiene le regole di filtraggio per i pacchetti che sono generati dal firewall in locale.
Le catene di INPUT e OUTPUT sono utilizzate per la costruzione di un personal firewall, mentre la catena di FORWARD viene utilizzata per gestire le regole di filtraggio di un gruppo di computer.

Per la tabella mangle le catene sono:
PREROUTING: per l'alterazione dei pacchetti in ingresso
OUTPUT: per l'alterazione dei pacchetti in uscita generati in locale

Dunque un pacchetto può seguire essenzialmente tre strade:

1. Il pacchetto ha come destinazione il firewall quindi attraversa la catena nat PREROUTING e poi la catena filter INPUT e poi passato all'applicazione destinataria
2. Il pacchetto è generato in locale dal firewall quindi attraversa la catena filter OUTPUT e poi la catena nat POSTROUTING
3. Il pacchetto attraversa il firewall ad esempio quando dalla LAN deve passare su Internet o viceversa. Quindi attraversa la catena nat PREROUTING poi filter FORWARD e nat POSTROUTING.

Stralcio dello schema dell'architettura iptables:



La base del firewall è costituita dalle catene della tabella filter. Le catene INPUT e OUTPUT si occupano della protezione del firewall mentre la catena FORWARD contiene le regole per la protezione della LAN aziendale.

Sintassi di una regola iptables:

```
iptables <table> <command> <match> <target>
```

In particolare:

`<table>`: per selezionare la tabella (nat, filter o mangle) di appartenenza della regola

`<command>`: l'azione che deve essere svolta. È possibile modificare cancellare o inserire una catena o una regola

`<match>`: regole per l'analisi dell'header del pacchetto

`<target>`: viene decisa la sorte (accettazione, eliminazione, modifica o log) del pacchetto che soddisfa la condizione di match

Il pacchetto percorrerà le catene e quando farà match con una delle regole verrà decisa la sua sorte

I target di iptables:

Il valore di un target (specificato tramite l'opzione –j o --jump) può essere il nome di una catena user-defined per demandare il filtraggio del pacchetto ad essa o uno dei valori speciali:

ACCEPT: per accettare il passaggio del pacchetto

DROP: per scartare il pacchetto (il mittente non viene avvertito che il pacchetto non è stato elaborato)

REJECT: per rifiutare il pacchetto (il mittente viene avvertito attraverso un messaggio ICMP)

QUEUE: per passare il pacchetto a user-space, cioè a qualche applicazione in attesa

RETURN: per passare alle regole della catena chiamante senza proseguire nella catena corrente

LOG: per registrare l'header del pacchetto in un file di log (Tramite l'opzione --log-prefix si può inserire un messaggio che comparirà nel log)

MIRROR: per rispedire il pacchetto al mittente

SNAT, DNAT, MASQUERADE: per funzioni di NAT

L'approccio generale nella configurazione di un firewall consiste nel bloccare tutto ciò che non viene esplicitamente permesso. Viene definita l'azione di default da applicare come DROP o REJECT e vengono consentite con ACCEPT solo quelle regole considerate lecite.

Per specificare la policy di default DROP per le tre catene della tabella filter si usano i seguenti comandi:

```
iptables –P INPUT DROP
iptables –P OUTPUT DROP
```

```
iptables –P FORWARD DROP
```

La policy di default verrà eseguita una volta che il pacchetto avrà attraversato riga per riga le regole della catena senza aver trovato una corrispondenza. L'ordine delle regole perciò risulta fondamentale, infatti appena il pacchetto fa match con una delle entry della catena viene eseguita l'azione corrispondente e non viene proseguita la ricerca.

Operazioni sulle catene:

Possono essere create delle catene ulteriori a discrezione dell'amministratore.

Per creare una nuova catena si usa l'opzione –N (--new-chain):

```
iptables –N <nome catena>
```

Per cancellare la catena che deve essere vuota e non può essere una delle predefinite (built-in), si usa l'opzione –X (--delete-chain):

```
iptables –X <nome catena>
```

Per cambiare la politica di una catena si usa l'opzione –P (--policy):

```
iptables –P <nome catena> <target>
```

Per visualizzare le regole contenute in una catena si utilizza l'opzione –L (--list):

```
iptables –L <nome catena>
```

Per visualizzare le regole di tutte le catene:

```
iptables -L
```

Se si vuole eliminare tutte le regole inserite nella catena si usa l'opzione –F (--flush):

```
iptables –F <nome catena>
```

Per azzerare i contatori dei pacchetti e dei byte che sono passati attraverso il firewall si fa uso dell'opzione -Z

```
iptables –Z
```

Ci sono molti modi per manipolare le regole presenti nelle catene:

Per aggiungere (appendere) una regola ad una catena si usa l'opzione –A (--append):

```
iptables –A <nome catena> <parametri> -j <target>
```

Per inserire una regola in una determinata posizione della catena si usa l'opzione –I (--insert):

```
iptables –I <nome catena> <parametri> -j <target>
```

o se viene specificata la posizione attraverso un numero la regola sarà inserita esattamente nella posizione indicata altrimenti sarà inserita di default nella prima posizione.

Per sostituire la regola in una determinata posizione si usa l'opzione –R (--replace)

```
iptables –R <nome catena> <parametri> -j <target>
```

o se viene specificata la posizione attraverso un numero sarà sostituita la regola indicata.

Per cancellare una regola si usa –D (--delete)

```
iptables –D <nome catena> <parametri> -j <target>
```

in questo caso è importante che i parametri corrispondano a quelli specificati nella regola che si vuole eliminare. Se viene specificata la posizione attraverso un numero sarà eliminata la regola indicata.

Opzioni (parametri) per il filtraggio:

È possibile invertire il significato del flag attraverso il simbolo ! da specificare dopo il flag stesso.

L'opzione –s (--source) si usa per indicare indirizzi IP mittenti, mentre per indicare indirizzi IP destinatari l'opzione è –d (--destination). L'indicazione 0/0 indica qualsiasi IP.

L'opzione –p (--protocol) si usa per indicare il protocollo (TCP,UDP o ICMP). Con –p ALL si intendono tutti i protocolli

L'opzione –i (--in-interface) serve per selezionare l'interfaccia di ingresso, mentre per l'interfaccia di uscita si usa –o (--out-interface). Il flag –i può essere utilizzato solo in relazione alle catene di INPUT, FORWARD e PREROUTING, mentre il flag –o in relazione a OUTPUT, FORWARD e POSTROUTING

L'opzione --dport si usa per indicare la porta di destinazione, mentre per quella sorgente --sport

L'opzione --tcp-flags si usa per filtrare attraverso i flag del protocollo TCP

L'opzione --icmp-type si usa per filtrare pacchetti particolari del protocollo ICMP

Possono essere caricati altresì dei moduli aggiuntivi per avere ancora maggiori possibilità di filtraggio. Vediamo i moduli che si possono caricare:

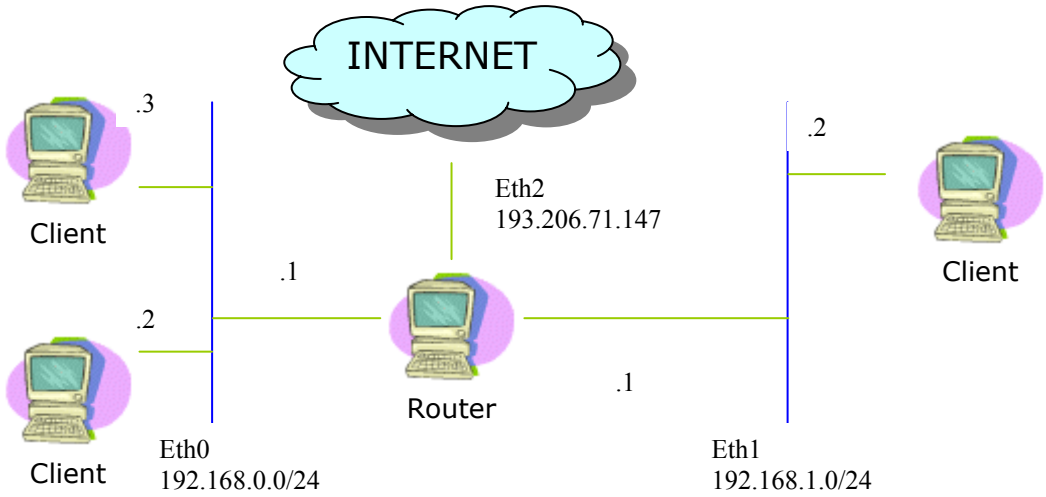
mac (--mac-source): per analizzare i pacchetti con un certo MAC Address (si usa in PREROUTING, INPUT e FORWARD)

limit (-m limit): serve per limitare il numero di registrazioni nei log

state (-m state): per filtrare i pacchetti a seconda dello stato della connessione. Stati possibili: NEW (il pacchetto crea una nuova connessione), ESTABLISHED (il pacchetto appartiene a connessione preesistente), RELATED (il pacchetto è correlato ma non a una connessione preesistente), INVALID (il pacchetto è associato a una connessione sconosciuta)

4.1.2 Rete di test ed analisi dello script

Vediamo lo schema della rete a disposizione per la parte pratica in laboratorio:



Dopo aver configurato la rete soprastante sono passata alla fase di implementazione di un semplice firewall. Sul router è stato creato lo script contenente le regole iptables per creare il firewall. Si tratta di uno script con delle funzioni per l'attivazione, la disattivazione, il ripristino del firewall e la visualizzazione delle regole.

Si è deciso di consentire ai pc l'accesso a servizi molto usati quali ftp per il trasferimento di files, telnet e ssh per il login da remoto. Teniamo presente che telnet non è uno strumento sicuro al 100% perché le trasmissioni viaggiano in chiaro, per cui è sempre consigliabile utilizzare il servizio ssh che fa ampio uso della crittografia. Tuttavia telnet è uno strumento tuttora ancora molto utilizzato, per questo motivo si è deciso di consentirne l'uso.

Per dimostrare come il firewall sia in grado di bloccare le più svariate applicazioni è stata aggiunta al firewall una regola in grado di bloccare la trasmissione di dati in streaming alla rete locale di un video lanciato sulla postazione del pc firewall. Come ulteriore aggiunta si è deciso di utilizzare una funzione di NAT allo scopo di consentire l'accesso a Internet anche alla rete locale privata.

Per salvare la configurazione del firewall impostata tramite iptables si può digitare:

```
$iptables-save > /etc/sysconfig/iptables
```

Inoltre per fare in modo che il firewall venga avviato al boot si può inserire il collegamento allo script nel file `/etc/rc.d/local`.

Definizione variabili

Per comodità vengono definite alcune variabili che saranno utilizzate nello script.

Con inet indichiamo l'interfaccia esterna collegata a internet:

```
inet="eth2"
```

Con lan1 e con lan2 indichiamo l'interfaccia verso la prima e la seconda sottorete locale:

```
lan1="eth0"
lan2="eth1"
```

Con lan1_add indichiamo l'indirizzo IP privato della prima sottorete locale e con lan2_add quello della seconda:

```
lan1_add=192.168.0.0/24
lan2_add=192.168.1.0/24
```

Infine con inet_add indichiamo l'indirizzo IP pubblico

```
inet_add=193.206.71.147
```

Moduli iptables da caricare in memoria:

Vediamo ora quali sono i moduli che ci serve siano caricati in memoria:

```
modprobe ip_tables
modprobe ipt_LOG
modprobe ipt_state
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe iptable_nat
modprobe ipt_MASQUERADE
```

Oltre al modulo principale iptables, abbiamo caricato `ipt_LOG` per consentire la registrazione nei log dei messaggi, `ipt_state` allo scopo di poter analizzare i pacchetti attraverso il modulo state, `ip_conntrack` e `ip_conntrack_ftp` per il connection tracking (tracciamento connessioni) i quali si occupano di tener traccia dello stato delle connessioni, e infine i moduli che consentono il nat e masqueranding degli indirizzi IP.

Configurazione di alcune flags del kernel

Ora ci occupiamo di modificare il comportamento della nostra macchina in rete attraverso la variazione di alcuni parametri localizzati nella directory `/proc/sys/net/ipv4/` e usati dal kernel per prendere delle decisioni in merito ai pacchetti che riceve o trasmette

Attraverso la riga sottostante viene abilitato il forwarding per consentire il passaggio dei pacchetti da un'interfaccia all'altra:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

La riga successiva disabilita le risposte al broadcast ICMP (per evitare l'attacco Smurf ICMP che provoca un DoS):

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Ora attiviamo la protezione contro attacchi Syn flooding (inondazione di pacchetti TCP con flag SYN attivo per esaurire le risorse del sistema attaccato):

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

La regola successiva disabilita l'ecn (Explicit Congestion Notification protocol) ovvero la notifica di congestione in quanto essa non è sempre supportata essendo recente:

```
echo 0 > /proc/sys/net/ipv4/tcp_ecn
```

La riga sottostante serve ad abilitare il "reserved path filter", ovvero la verifica della coerenza dell'indirizzo IP mittente con l'interfaccia su cui è stato ricevuto e le regole delle tabelle di routing (antispoofing):

```
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
```

La regola successiva rifiuta i pacchetti IP con source routing, utilizzati allo scopo di chiedere che la risposta passi attraverso determinati routers :

```
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
```

La regola sottostante fa in modo di ignorare i pacchetti ICMP per redirigere il traffico (indicare un diverso percorso da seguire):

```
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
```

La regola successiva disabilita il logging di messaggi di risposte errate mandate da alcune apparecchiature di rete:

```
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

L'ultima regola consente l'abilitazione della registrazione nei log per pacchetti strani (ad esempio con indirizzo IP falsificato):

```
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
```

Regole iptables

Come primo passo eliminiamo le eventuali regole iptables e catene utente precedentemente impostate e resettiamo i contatori necessari per le statistiche

```
iptables -F
iptables -F -t nat
iptables -F -t mangle
```

```
iptables -X
iptables -Z
```

Di default si è deciso di utilizzare la politica "Default Deny Stance", ovvero viene bloccato tutto e concesso solo quello che è esplicitamente permesso. Quindi:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Catene utente

Per comodità di gestione sono state definite delle catene utente in cui inserire le regole relative al traffico che passa sulla rete.

Come prima cosa creiamo una catena per la gestione del traffico ICMP:

```
iptables -N icmp_chain
```

Con le due regole successive creiamo una catena per il traffico che passa dalla lan1 (192.168.0.0/24) alla lan2 (192.168.1.0/24) e una per il traffico nel senso inverso:

```
iptables -N lan1lan2
iptables -N lan2lan1
```

Ora creiamo le catena per il traffico che va dalle nostre sottoreti locali a internet:

```
iptables -N lan1inet
iptables -N lan2inet
```

Infine creiamo le catene per il traffico da internet alle nostre reti locali:

```
iptables -N inetlan1
iptables -N inetlan2
```

Catena POSTROUTING (Tabella NAT)

In questa catena inseriamo una regola che non è proprio una regola di sicurezza ma è utile perché consente il mascheramento dei nostri indirizzi privati nell'unico indirizzo IP pubblico a disposizione. Tramite questa regola quindi si potrà consentire alla nostra rete privata l'accesso ai servizi del web. In pratica i pacchetti della rete privata verranno modificati in modo che l'indirizzo sorgente corrisponda a quello IP pubblico e quindi ne verrà consentita l'uscita sull'interfaccia di rete inet destinata a internet.

```
iptables -t nat -A POSTROUTING -o $inet -j MASQUERADE
```

Catena FORWARD

La prima regola qui sotto mostrata delega il traffico ICMP alla catena utente icmp_chain:

```
iptables -A FORWARD -p ICMP -j icmp_chain
```

Tramite le regole successive demandiamo il traffico sulle interfacce alle catene utente precedentemente definite:

```
iptables -A FORWARD -i $lan1 -o $inet -j lan1inet
iptables -A FORWARD -i $lan2 -o $inet -j lan2inet
iptables -A FORWARD -i $lan1 -o $lan2 -j lan1lan2
iptables -A FORWARD -i $lan2 -o $lan1 -j lan2lan1
iptables -A FORWARD -i $inet -o $lan1 -j inetlan1
iptables -A FORWARD -i $inet -o $lan2 -j inetlan2
```

Catena icmp_chain

Questa catena contiene le regole necessarie per la gestione dei pacchetti del protocollo ICMP. Come è possibile vedere dalle regole si è deciso di consentire solamente i seguenti pacchetti ICMP:

```
iptables -A icmp_chain -p ICMP --icmp-type echo-request -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type echo-reply -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type time-exceeded -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type destination-unreachable
-j ACCEPT
```

Le prime due regole consentono il comando ping (in particolare la richiesta e la risposta) per consentire che sia possibile verificare il funzionamento della rete. La terza regola accetta gli ICMP che segnalano che il pacchetto inviato è stato scartato perché il TTL (Time To Live) ha raggiunto lo zero, e quindi il tempo di permanenza del pacchetto sulla rete è scaduto. L'ultima regola indica di accettare i pacchetti che segnalano che la destinazione contattata non è raggiungile o non è disponibile al momento.

Catena lan1lan2

La prima riga elimina tutti i pacchetti in transito dalla prima sottorete locale alla seconda che hanno un indirizzo sorgente che non corrisponde a quello che dovrebbe essere l'indirizzo mittente e cioè 192.168.0.0/24 (lan1), essendo ciò indizio di un attacco di tipo IP spoofing:

```
iptables -A lan1lan2 -s ! $lan1_add -j DROP
```

Le successive righe consentono ai pc situati sulla sottorete lan1 di poter accedere a servizi telnet, ftp e ssh attivi sui pc situati sulla sottorete lan2:

```
iptables -A lan1lan2 -p TCP --dport 23 -j ACCEPT
```

```
iptables -A lan1lan2 -p TCP --dport 21 -j ACCEPT
iptables -A lan1lan2 -p TCP --dport 20 -j ACCEPT
iptables -A lan1lan2 -p TCP --dport 22 -j ACCEPT
```

La regola sottostante consente il passaggio dei pacchetti che sono relative a connessioni già stabilite e correlate a connessioni preesistenti, in modo che siano consentite le risposte a connessioni già iniziate e precedentemente concesse dal firewall:

```
iptables -A lan1lan2 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola successiva permette la registrazione nei log dei pacchetti di questa catena scartati i quali saranno riconoscibili facilmente grazie al prefisso impostato:

```
iptables -A lan1lan2 -j LOG -log-prefix "Pacchetti lan1lan2 scartati: "
```

L'ultima riga scarta tutti i pacchetti che arrivano in fondo a questa catena:

```
iptables -A lan1lan2 -j DROP
```

Catena lan2lan1

La prima riga elimina tutti i pacchetti in transito dalla seconda sottorete locale alla prima che hanno un indirizzo sorgente che non corrisponde a quello che dovrebbe essere l'indirizzo mittente e cioè 192.168.1.0/24 (lan2), essendo ciò indizio di un attacco di tipo IP spoofing:

```
iptables -A lan2lan1 -s ! $lan2_add -j DROP
```

Le successive righe consentono ai pc situati sulla rete lan2 di poter accedere a servizi telnet, ftp e ssh attivi sui pc situati sulla sottorete lan1:

```
iptables -A lan2lan1 -p TCP --dport 23 -j ACCEPT
iptables -A lan2lan1 -p TCP --dport 21 -j ACCEPT
iptables -A lan2lan1 -p TCP --dport 20 -j ACCEPT
iptables -A lan2lan1 -p TCP --dport 22 -j ACCEPT
```

Questa riga consente il passaggio dei pacchetti che sono relative a connessioni già stabilite e correlate a connessioni preesistenti, in modo che siano consentite le risposte a connessioni già iniziate e precedentemente concesse dal firewall:

```
iptables -A lan2lan1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola successiva permette la registrazione nei log dei pacchetti di questa catena scartati i quali saranno riconoscibili facilmente grazie al prefisso impostato:

```
iptables -A lan2lan1 -j LOG -log-prefix "Pacchetti lan2lan1 scartati: "
```

L'ultima regola scarta tutti i pacchetti che arrivano in fondo a questa catena:

```
iptables -A lan2lan1 -j DROP
```

Catena lan1inet

Questa catena contiene le regole relative al traffico che passa dalla prima sottorete locale (lan1) e va verso l'interfaccia connessa a internet (inet).

La prima regola si occupa di eliminare i pacchetti che hanno un indirizzo sorgente diverso da 192.168.0.0/24 (lan1_add) perché sono un indizio di attacco IP spoofing:

```
iptables -A lan1inet -s ! $lan1_add -j DROP
```

Le regole successive permettono ai pc sulla rete lan1 di poter accedere ai servizi dns e http:

```
iptables -A lan1inet -p TCP --dport 53 -j ACCEPT
iptables -A lan1inet -p UDP --dport 53 -j ACCEPT
iptables -A lan1inet -p TCP --dport 80 -j ACCEPT
```

La regola sottostante indica di accettare i pacchetti che sono relativi a connessioni già stabilite o relative a connessioni preesistenti e quindi consente le risposte a connessioni già accettate dal firewall:

```
iptables -A lan1inet -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola successiva permette la registrazione nei log dei pacchetti di questa catena scartati i quali saranno riconoscibili facilmente grazie al prefisso impostato:

```
iptables -A lan1inet -j LOG --log-prefix "Pacchetti lan1inet scartati: "
```

L'ultima regola indica che i pacchetti non hanno fatto match con nessuna regola precedente della catena vanno scartati:

```
iptables -A lan1inet -j DROP
```

Catena lan2inet

Questa catena contiene le regole relative al traffico che passa dalla seconda sottorete locale (lan2) e va verso l'interfaccia connessa a internet (inet).

La prima regola si occupa di eliminare i pacchetti che hanno un indirizzo sorgente diverso da 192.168.1.0/24 (lan2_add) perché sono un indizio di attacco IP spoofing:

```
iptables -A lan2inet -s ! $lan2_add -j DROP
```

Le regole successive permettono ai pc sulla rete lan2 di poter accedere ai servizi dns e http:

```
iptables -A lan2inet -p TCP --dport 53 -j ACCEPT
iptables -A lan2inet -p UDP --dport 53 -j ACCEPT
```

```
iptables -A lan2inet -p TCP --dport 80 -j ACCEPT
```

La regola sottostante indica di accettare i pacchetti che sono relativi a connessioni già stabilite o relative a connessioni preesistenti e quindi consente le risposte a connessioni già accettate dal firewall

```
iptables -A lan2inet -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola successiva permette la registrazione nei log dei pacchetti di questa catena scartati i quali saranno riconoscibili facilmente grazie al prefisso impostato:

```
iptables -A lan2inet -j LOG --log-prefix "Pacchetti lan2inet scartati: "
```

L'ultima regola indica che i pacchetti non hanno fatto match con nessuna regola precedente della catena vanno scartati:

```
iptables -A lan2inet -j DROP
```

Catena inetlan1

Questa catena contiene le regole che gestiscono i pacchetti provenienti dall'interfaccia verso internet e destinati alla prima rete locale (lan1).

La prima regola si occupa di scartare i pacchetti che hanno come indirizzo sorgente l'indirizzo 192.168.0.0/24 (lan1_add) che indicherebbe che i pacchetti provenienti da internet hanno come mittente l'indirizzo della nostra sottorete privata, in pratica si tratterebbe di un attacco IP spoofing:

```
iptables -A inetlan1 -s $lan1_add -j DROP
```

La regola successiva indica di lasciar passare i pacchetti di risposta che si riferiscono a connessioni precedentemente accettate dal firewall :

```
iptables -A inetlan1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola sottostante permette la registrazione nei log dei pacchetti che verranno successivamente scartati. Essi saranno facilmente individuabili perchè preceduti dal prefisso indicato.

```
iptables -A inetlan1 -j LOG --log-prefix "Pacchetti inetlan1 scartati: "
```

L'ultima regola si occupa di scartare i pacchetti che non sono stati prima accettati:

```
iptables -A inetlan1 -j DROP
```

Catena inetlan2

Questa catena contiene le regole che gestiscono i pacchetti provenienti dall'interfaccia verso internet e destinati alla seconda sottorete locale (lan2).

La prima regola si occupa di scartare i pacchetti che hanno come indirizzo sorgente l'indirizzo 192.168.1.0/24 (lan2_add) che indicherebbe che i pacchetti provenienti da internet hanno come mittente l'indirizzo della nostra rete, in pratica si tratterebbe di un attacco IP spoofing:

```
iptables -A inetlan2 -s $lan2_add -j DROP
```

La regola successiva indica di lasciar passare i pacchetti di risposta che si riferiscono a connessioni precedentemente accettate dal firewall:

```
iptables -A inetlan2 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La regola sottostante permette la registrazione nei log dei pacchetti che verranno successivamente scartati. Essi saranno facilmente individuabili perchè preceduti dal prefisso indicato.

```
iptables -A inetlan2 -j LOG --log-prefix "Pacchetti inetlan2 scartati: "
```

L'ultima regola si occupa di scartare i pacchetti che non sono stati prima accettati:

```
iptables -A inetlan2 -j DROP
```

Catena INPUT

La catena INPUT si occupa della gestione dei pacchetti che entrano nel pc che fa da firewall.

Prima di tutto abilitiamo l'interfaccia di loopback che viene utilizzata da alcune applicazioni per il corretto funzionamento e quindi non rappresenta alcun pericolo:

```
iptables -A INPUT -i lo -j ACCEPT
```

Ora registriamo nei log i tentativi di attacco IP spoofing dall'esterno, in particolare non accettiamo pacchetti che hanno come indirizzo sorgente indirizzi utilizzati normalmente come range di indirizzamento privato e quindi non instradabili su internet:

```
iptables -A INPUT -i $inet -s 192.168.0.0/16 -j LOG --log-prefix "Spoofing: "
iptables -A INPUT -i $inet -s 172.16.0.0/12 -j LOG --log-prefix "Spoofing: "
iptables -A INPUT -i $inet -s 10.0.0.0/8 -j LOG --log-prefix "Spoofing: "
```

Con le regole successive scartiamo i pacchetti relativi ad attacchi IP spoofing precedentemente loggati:

```
iptables -A INPUT -i $inet -s 192.168.0.0/16 -j DROP
```

```
iptables -A INPUT -i $inet -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i $inet -s 10.0.0.0/8 -j DROP
```

Ora passiamo i pacchetti ICMP alla catena utente icmp_chain:

```
iptables -A INPUT -p ICMP -j icmp_chain
```

Le due regole successive consentono alla nostra rete privata di poter accedere al servizio ssh:

```
iptables -A INPUT -p TCP -i $lan1 --dport 22 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 22 -j ACCEPT
```

Le regole sottostanti consentono l'utilizzo del servizio ftp alla nostra rete privata (la porta 21 è per la connessione e la porta 20 per il trasferimento dati):

```
iptables -A INPUT -p TCP -i $lan1 --dport 21 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 21 -j ACCEPT
iptables -A INPUT -p TCP -i $lan1 --dport 20 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 20 -j ACCEPT
```

Tramite le regole qui sotto consentiamo alla nostra rete privata di poter far uso del servizio telnet:

```
iptables -A INPUT -p TCP -i $lan1 --dport 23 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 23 -j ACCEPT
```

Questa riga consente i pacchetti che sono relative a connessioni già stabilite e correlate a connessioni preesistenti, in modo che siano consentite le risposte a connessioni già iniziate precedentemente concesse dal firewall:

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Infine registriamo nei log i pacchetti che sono arrivati in fondo a questa catena:

```
iptables -A INPUT -j LOG --log-prefix "Pacchetti INPUT scartati: "
```

Dato che la policy di default della catena INPUT è impostata a DROP, non è necessario inserire qui la regola per scartare i pacchetti che arrivano in fondo alla catena.

Catena OUTPUT

La catena OUTPUT si occupa della gestione dei pacchetto in uscita dal pc che fa da firewall.

Come prima cosa ci occupiamo di abilitare l'interfaccia di loopback utilizzata da alcune applicazioni per il corretto funzionamento e quindi non rappresenta alcun pericolo:

```
iptables -A OUTPUT -o lo -j ACCEPT
```

Ora vediamo la regola che ci permette di escludere la sottorete lan1 dalla possibilità di sfruttare il servizio di videostreaming consentito dall'applicazione VLC media player installata come server sul pc firewall e come client sugli altri. Questa applicazione consente lo streaming trasmettendo sulla porta UDP 1234. Lanciando un video in streaming sulla postazione server sulla sottorete lan2

ne sarà consentita la visualizzazione mentre sulla sottorete lan1 sarà bloccata. Questa regola è stata inserita non con uno scopo di sicurezza particolare ma solo per dimostrare come un firewall può bloccare le applicazioni:

```
iptables -A OUTPUT -o $lan1 -p UDP --dport 1234 -j DROP
```

Tutto il resto dei pacchetto in uscita dal firewall viene accettato tramite le regole sottostanti:

```
iptables -A OUTPUT -o $lan1 -j ACCEPT
iptables -A OUTPUT -o $lan2 -j ACCEPT
iptables -A OUTPUT -o $inet -j ACCEPT
```

La regola successiva ci permette di registrare nei log i pacchetti scartati che saranno riconoscibili attraverso il prefisso specificato:

```
iptables -A OUTPUT -j LOG --log-prefix "Pacchetti OUTPUT scartati:
"
```

Non è necessario poi inserire la regola che scarta i pacchetti in quanto la catena di OUTPUT ha già impostata come policy di default il target DROP, per cui tutto ciò che non è stato qui accettato sarà scartato.

4.2 PROVE DI FUNZIONAMENTO DEL FIREWALL

4.2.1 Telnet

Vediamo alcuni esempi con telnet per verificare come agiscono le regole iptables del firewall.

1) Se nella catena utente lan1lan2 che gestisce i pacchetti che vanno dalla sottorete 192.168.0.0/24 alla 192.168.1.0/24 lasciamo invariata la regola che permette il tentativo di connessione telnet:

```
iptables -A lan1lan2 -p TCP --dport 23 -j ACCEPT
```

e non modifichiamo le regole sulla catena lan2lan1 che consentono la risposta alla connessione, ovvero:

```
iptables -A lan2lan1 -p TCP -m state --state ESTABLISHED,RELATED
```

allora la connessione telnet verrà stabilita correttamente come si può vedere dal risultato ottenuto sulla shell:

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
Connected to 192.168.1.2 (192.168.1.2).
Escape character is '^]'.
Fedora Core release 3 (Heidelberg)
```

```
Kernel 2.6.11-1.27_FC3 on an i686
login:
```

Infatti utilizzando sul pc che richiede la connessione lo sniffer Ethereal per catturare i pacchetti in transito sulla interfaccia eth0 (lan1) otteniamo in output le informazioni relative ai pacchetti transitati:

No. Time Source Destination Protocol Info

1 0.000000 192.168.0.2 192.168.1.2 TCP 60698 > telnet [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=2610115 TSER=0 WS=2

No. Time Source Destination Protocol Info

2 0.000322 192.168.1.2 192.168.0.2 TCP telnet > 60698 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=2613530 TSER=2610115 WS=2

No. Time Source Destination Protocol Info

3 0.000358 192.168.0.2 192.168.1.2 TCP 60698 > telnet [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=2610116 TSER=2613530

Come possiamo vedere la connessione viene stabilita, ovvero viene mandato il primo pacchetto TCP con flag SYN attivo a cui segue un SYN, ACK di risposta e infine l'ultimo pacchetto ACK che fa stabilire la connessione.

2) Se invece al posto della riga precedente mettiamo questa:

```
iptables -A lan1lan2 -p TCP --dport 23 -j DROP
```

oppure se togliamo tale riga per cui il pacchetto giunge all'ultima regola della catena:

```
iptables -A lan1lan2 -j DROP
```

otteniamo usando telnet da shell per contattare il client sull'altra rete questa risposta che indica che la connessione è bloccata:

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
```

Infatti utilizzando lo sniffer Ethereal sul pc che chiede la connessione per catturare i pacchetti in transito sulla interfaccia eth0 (lan1) otteniamo:

No. Time Source Destination Protocol Info

1 0.000000 192.168.0.2 192.168.1.2 TCP 46546 > telnet [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=1672511 TSER=0 WS=2

No. Time Source Destination Protocol Info

2 2.998951 192.168.0.2 192.168.1.2 TCP 46546 > telnet [SYN] Seq=0 Ack=0 Win=23360 Len=0 MSS=1460 TSV=1675511 TSER=0 WS=2

Quindi come vediamo dal risultato il nostro client continua a mandare pacchetti TCP con flag SYN per tentare la connessione telnet ma non riceve risposta.

3)Se invece inseriamo sempre al posto della regola precedentemente esaminata:

```
iptables -A lan1lan2 -p TCP --dport 23 -j REJECT --reject-with tcp-reset
```

oppure togliamo la regola e lasciamo che il pacchetto arrivi all'ultima regola della catena così impostata:

```
iptables -A lan1lan2 -p TCP -j REJECT --reject-with tcp-reset
```

si ottiene come risposta:

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
telnet: connect to address 192.168.1.2: Connection refused
telnet: Unable to connect to remote host: Connection refused
```

questo perchè viene mandato dal firewall un pacchetto di risposta TCP con flag RST attivo che rigetta la connessione.

Con Ethereal impostato per catturare i pacchetti su eth0 (lan1) si ottiene:

No. Time Source Destination Protocol Info

1 0.000000 192.168.0.2 192.168.1.2 TCP 37430 > telnet [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=2176293 TSER=0 WS=2

No. Time Source Destination Protocol Info

2 0.000148 192.168.1.2 192.168.0.2 TCP telnet > 37430 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0

Come possiamo notare dopo il pacchetto SYN per la richiesta di connessione viene spedito un pacchetto RST,ACK di risposta per resettare la connessione.

Se invece ci mettiamo a filtrare il traffico di risposta della controparte e quindi lan2lan1, possiamo ottenere i seguenti risultati:

1) Lasciando invariate le regole consentendo la risposta a connessioni già stabilite e correlate a connessioni preesistenti, quindi nel nostro caso la risposta (del pc su lan2) al tentativo di connessione telnet da parte del pc su lan1:

```
iptables -A lan2lan1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

otteniamo come output (sempre sul pc che tenta la connessione):

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
```

```
Connected to 192.168.1.2 (192.168.1.2).
Escape character is '^]'.
Fedora Core release 3 (Heidelberg)
Kernel 2.6.11-1.27_FC3 on an i686
login:
```

Il risultato che si ottiene con Ethereal è lo stesso di quando si riesce a stabilire la connessione.

2) Se la regola viene cambiata in:

```
iptables -A lan2lan1 -p TCP -m state --state ESTABLISHED,RELATED -
j REJECT --reject-with tcp-reset
```

oppure se la regola viene tolta e quindi il pacchetto passa alla regola successiva, ovvero:

```
iptables -A lan2lan1 -p TCP -j REJECT --reject-with tcp-reset
```

La connessione lato lan1 rimarrà bloccata:

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
```

mentre sul pc situato sulla lan2 verrà mandato un pacchetto RST dal firewall quando tenterà di rispondere alla richiesta di connessione come dimostrato da Ethereal:

No. Time Source Destination Protocol Info
1 0.000000 192.168.0.2 192.168.1.2 TCP 51059 > telnet [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=6721179 TSER=0 WS=2

No. Time Source Destination Protocol Info
2 0.000162 192.168.1.2 192.168.0.2 TCP telnet > 51059 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=6724580 TSER=6721179 WS=2

No. Time Source Destination Protocol Info
3 0.000056 192.168.0.2 192.168.1.2 TCP 51059 > telnet [RST] Seq=1 Ack=596986302 Win=0 Len=0

3) Nel caso invece in cui fosse stato usato il target DROP la connessione veniva comunque bloccata:

```
[root@labreti3 ~]# telnet 192.168.1.2
Trying 192.168.1.2...
```

Ma sul pc situato sulla lan2 da Ethereal si ottiene un risultato diverso:

No. Time Source Destination Protocol Info
3 0.000127 192.168.0.2 192.168.1.2 TCP 43529 > telnet [SYN] Seq=0 Ack=0 Win=5840 Len=0 MSS=1460 TSV=8269646 TSER=0 WS=2

No. Time Source Destination Protocol Info

4 0.000166 192.168.1.2 192.168.0.2 TCP telnet > 43529 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=8273044 TSER=8269646 WS=2

No. Time Source Destination Protocol Info

5 2.997233 192.168.0.2 192.168.1.2 TCP 43529 > telnet [SYN] Seq=0 Ack=0 Win=23360 Len=0 MSS=1460 TSV=8272646 TSER=0 WS=2

Come si può vedere in questo caso il pc viene contattato per stabilire una connessione telnet col primo pacchetto SYN, al quale risponde con un pacchetto SYN, ACK ma questa volta non riceve nessun pacchetto RST ma viene raggiunto di nuovo da una richiesta di connessione con pacchetto SYN del pc su lan 1.

4.2.2 Stream video

Ora proviamo ad impostare le regole del firewall in modo che consenta l'applicazione in videostreaming VLC media player, la quale trasmette sulla porta 1234 del protocollo UDP.

Il video lanciato dall'applicazione lato server (che nel nostro caso è situata sul pc router) sarà visibile in tempo reale dalla postazione client. Ciò è consentito da questa regola:

```
iptables -A OUTPUT -o $lan1 -j ACCEPT
```

la quale permette l'uscita dei pacchetti dal firewall (il nostro router) al pc che deve fruire dell'applicazione videostreaming.

Utilizzando lo sniffer Ethereal sull'interfaccia eth0 (lan1) durante la trasmissione abbiamo il seguente risultato:

No. Time Source Destination Protocol Info

1 0.000000 192.168.0.1 192.168.0.2 UDP Source port: 32776 Destination port: 1234

No. Time Source Destination Protocol Info

2 0.037969 192.168.0.1 192.168.0.2 UDP Source port: 32776 Destination port: 1234

No. Time Source Destination Protocol Info

3 0.075948 192.168.0.1 192.168.0.2 UDP Source port: 32776 Destination port: 1234

L'output indica che i pacchetti vengono correttamente spediti e accettati.

Se invece vogliamo bloccare la fruizione del video dobbiamo inserire la seguente regola prima della precedente e quindi le regole saranno:

```
iptables -A OUTPUT -o $lan1 -p UDP --dport 1234 -j DROP
iptables -A OUTPUT -o $lan1 -j ACCEPT
```

In questo caso il video non sarà trasmesso e infatti Ethereal non catturerà nessun risultato perchè i pacchetti trasmessi verranno scartati dal firewall.

4.2.3 Nmap

Proviamo ora a fare un port scanning tramite nmap:

```
[root@labreti3 ~]# nmap 192.168.0.1

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2005-10-12
14:41 CEST
Interesting ports on 192.168.0.1:
(The 1656 ports scanned but not shown below are in state:
filtered)
PORT   STATE  SERVICE
20/tcp closed ftp-data
21/tcp open   ftp
22/tcp open   ssh
23/tcp open   telnet
MAC Address: 00:50:FC:CD:65:EE (Edimax Technology CO.)

Nmap run completed -- 1 IP address (1 host up) scanned in 20.420
seconds
```

Come si può vedere risultano solo le porte concesse in apertura il resto delle porte appare filtrato grazie all'azione del firewall.

4.2.4 Ping frammentato

Se spediamo un pacchetto con una dimensione superiore a 1500 byte (MTU massima per la ethernet), esso suddiviso in frammenti, che il firewall è comunque in grado di riconoscere e gestire.

```
[root@labreti3 ~]# ping -s 2000 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 2000(2028) bytes of data.
2008 bytes from 192.168.1.2: icmp_seq=0 ttl=63 time=5.38 ms
2008 bytes from 192.168.1.2: icmp_seq=1 ttl=63 time=3.98 ms
2008 bytes from 192.168.1.2: icmp_seq=2 ttl=63 time=3.98 ms

--- 192.168.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 3.982/4.451/5.389/0.665 ms, pipe 2
```

Con lo sniffer Ethereal si ottiene:

```
No.    Time       Source          Destination     Protocol Info
     3 0.000266   192.168.0.2       192.168.1.2       ICMP     Echo (ping) request
No.    Time       Source          Destination     Protocol Info
     4 0.000279   192.168.0.2       192.168.1.2       IP       Fragmented IP protocol (proto=ICMP
0x01, off=1480)
No.    Time       Source          Destination     Protocol Info
```

5 0.004159 192.168.1.2 192.168.0.2 ICMP Echo (ping) reply
No. Time Source Destination Protocol Info
6 0.004617 192.168.1.2 192.168.0.2 IP Fragmented IP protocol (proto=ICMP 0x01, off=1480)

Dal risultato si vede che i pacchetti ICMP vengono frammentati e spediti senza che essi vengano bloccati dal firewall. Infatti effettuando il tracciamento delle connessioni (connection tracking) tutti i frammenti saranno ricomposti prima che raggiungano il codice che si occupa di filtrare i pacchetti, perciò non c'è motivo di temere i frammenti.

4.2.5 Ping con source routing

Ora vediamo una prova di source routing, ovvero forziamo il percorso che il pacchetto deve seguire e vediamo come si comporta di conseguenza il firewall.

```
C:\>ping -j 192.168.1.1 192.168.1.2

Esecuzione di Ping 192.168.1.2 con 32 byte di dati:

Richiesta scaduta.
Richiesta scaduta.
Richiesta scaduta.
Richiesta scaduta.

Statistiche Ping per 192.168.1.2:
     Pacchetti: Trasmessi = 4, Ricevuti = 0, Persi = 4 (100% persi)
```

Tramite l'opzione –j del comando ping (eseguito su macchina con sistema operativo Windows XP) abbiamo richiesto l'invio di un pacchetto ICMP con destinazione 192.168.1.2 vincolando il passaggio per l'host 192.168.1.1. Il firewall non accetta il source routing, che viene ritenuto pericoloso, infatti nella configurazione dei flag del kernel avevamo disabilitato l'accettazione del source routing:

```
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
```

Dalla postazione firewall possiamo vedere tramite uno sniffer come tcpdump i pacchetti che arrivano sull'interfaccia interessata (eth0):

```
[root@labreti2 ~]# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:09:41.216566 IP 192.168.0.2 > 192.168.1.1: icmp 40: echo request seq 6400
```

```
16:09:46.233404  IP  192.168.0.2  >  192.168.1.1:  icmp  40:  echo
request seq 6656
16:09:51.733598  IP  192.168.0.2  >  192.168.1.1:  icmp  40:  echo
request seq 6912
16:09:57.233736  IP  192.168.0.2  >  192.168.1.1:  icmp  40:  echo
request seq 7168

4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

Il risultato indica che le richieste ICMP vengono inviate ma non raggiungono la destinazione e non ottengono risposta perché bloccate dal firewall.

5. CONCLUSIONI

In questa tesi si è cercato di affrontare un tema molto complesso e vasto proponendo una carrellata degli strumenti che è possibile utilizzare per la sicurezza in ambito del sistema operativo Linux. È chiaro che non è stato possibile prendere in esame tutti i vari strumenti ma solo quelli che sembravano più interessanti e utili.

Ovviamente ogni strumento va plasmato con l'architettura della rete di cui si dispone, con le esigenze e con le security policy che si vogliono rispettare.

Quando si decide di proteggere la rete è necessario cercare di integrare i vari strumenti di difesa: quindi oltre al firewall che garantisce una sicurezza perimetrale, sarà necessario dotarsi di un IDS per rilevare le intrusioni e tramite sistemi di audit trail informare l'amministratore degli eventi anomali. Tra questi spicca il NIDS Snort, che oltre alla funzione di Intrusion Detection può funzionare in modalità sniffer o packet logger.

Utile anche uno strumento per la scansione delle vulnerabilità tra cui si può segnalare Nessus, un programma che contiene test per verificare che il sistema non sia a rischio per permettere eventualmente di correggere le falle riscontrate.

Abbiamo visto come i tools di Linux permettono di avere il controllo su vari aspetti del sistema e quindi la loro conoscenza approfondita risulta utile e redditizia.

Non dimentichiamo che gli stessi utenti del sistema devono essere istruiti riguardo le problematiche della sicurezza affinché con errori banali apparentemente innocui non causino danni che possano mettere a rischio il sistema.

Nonostante tutti questi accorgimenti ci sarà sempre qualcuno in grado di colpire la nostra rete, come abbiamo visto esistono un'infinità di attacchi che è possibile sferrare, e purtroppo le tecniche di cracking si stanno evolvendo.

Pensare di progettare una rete impenetrabile resta un'utopia ma almeno quello che si può fare è tentare attraverso l'utilizzo di tutti questi metodi analizzati (e altri ancora) di rendere la vita più difficile a chi tenta di minare il nostro sistema di sicurezza.

6. APPENDICE A: lo script del firewall

Ecco lo script completo del firewall implementato:

```
# !/bin/bash
# Questo script configura un firewall di base
# Author: Delia Franciosi(C) 2005

# Definizione delle variabili:
# interfaccia verso internet
inet="eth2"
# interfaccia prima sottorete locale
lan1="eth0"
# interfaccia seconda sottorete locale
lan2="eth1"
# indirizzo di lan1
lan1_add=192.168.0.0/24
# indirizzo di lan2
lan2_add=192.168.1.0/24
#indirizzo IP pubblico
inet_add=193.206.71.147

# Funzione per abilitare il firewall
firewall_on(){
# Caricamento dei moduli necessari:
# modulo iptables
modprobe ip_tables
# modulo per la registrazione nei log
modprobe ipt_LOG
# modulo per l'analisi dei pacchetti attraverso modulo state
modprobe ipt_state
# moduli per il connection tracking (tracciamento connessioni)
modprobe ip_conntrack
modprobe ip_conntrack_ftp
# moduli per il NAT
modprobe ip_nat_ftp
modprobe iptable_nat
modprobe ipt_MASQUERADE

# Configurazione dei flag del kernel necessari:
# abilitazione dell'IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
# disabilitazione delle risposte al broadcast ICMP (per evitare lo smurf)
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# protezione contro attacchi syn flooding
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
# disabilitazione della notifica di congestione non sempre supportata
echo 0 > /proc/sys/net/ipv4/tcp_ecn
# verifica coerenza indirizzo IP mittente con interfaccia(antispoofing)
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
# rifiuto dei pacchetti IP con source routing, i quali chiedono che la risposta
# passi da determinati routers
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route
# rifiuto degli ICMP per redirigere il traffico
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
# disabilitazione del logging di messaggi di errore apparecchiature di rete
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# abilitazione di log per pacchetti con indirizzo IP falsificato
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians

# Flushing delle regole
iptables -F
```

```
iptables -F -t nat
iptables -F -t mangle
# Cancellazione delle chain user-defined
iptables -X
# Azzeramento dei contattori
iptables -Z
# Definizione delle policy di default: blocchiamo tutto
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# Definizione di catene utente
# Creazione catena per la gestione dei pacchetti ICMP
iptables -N icmp_chain
# Creazione catena per il traffico dalla lan1 alla lan2
iptables -N lan1lan2
# Creazione catena per il traffico dalla lan2 alla lan1
iptables -N lan2lan1
# Creazione catena per il traffico dalla lan1 a internet
iptables -N lan1inet
# Creazione catena per il traffico dalla lan1 a internet
iptables -N lan2inet
# Creazione catena per il traffico da internet alla lan1
iptables -N inetlan1
# Creazione catena per il traffico da internet alla lan2
iptables -N inetlan2

# Autorizzazione al mascheramento degli indirizzi IP privati
iptables -t nat -A POSTROUTING -o $inet -j MASQUERADE

# Catena FORWARD (pacchetti che attraversano il firewall):
# passaggio dei pacchetti icmp alla catena apposita
iptables -A FORWARD -p ICMP -j icmp_chain
# permesso di passaggio dei pacchetti sulle interfacce
iptables -A FORWARD -i $lan1 -o $inet -j lan1inet
iptables -A FORWARD -i $lan2 -o $inet -j lan2inet
iptables -A FORWARD -i $lan1 -o $lan2 -j lan1lan2
iptables -A FORWARD -i $lan2 -o $lan1 -j lan2lan1
iptables -A FORWARD -i $inet -o $lan1 -j inetlan1
iptables -A FORWARD -i $inet -o $lan2 -j inetlan2
# registrazione nei log dei pacchetti scartati
iptables -A FORWARD -j LOG --log-prefix "Pacchetti FORWARD scartati: "

# Catena icmp_chain
# I pacchetti ICMP accettati sono:
iptables -A icmp_chain -p ICMP --icmp-type echo-request -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type echo-reply -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type time-exceeded -j ACCEPT
iptables -A icmp_chain -p ICMP --icmp-type destination-unreachable -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A icmp_chain -j LOG --log-prefix "Pacchetti ICMP scartati: "
# il resto dei pacchetti viene scartato
iptables -A icmp_chain -p ICMP -j DROP

# Catena lan1lan2
# blocco dello spoofing
iptables -A lan1lan2 -s ! $lan1_add -j DROP
# concessione telnet, ftp e ssh
iptables -A lan1lan2 -p TCP --dport 23 -j ACCEPT
iptables -A lan1lan2 -p TCP --dport 21 -j ACCEPT
iptables -A lan1lan2 -p TCP --dport 20 -j ACCEPT
iptables -A lan1lan2 -p TCP --dport 22 -j ACCEPT
# concessione dei pacchetti relativi a connessioni già stabilite o relative a
```

```
# connessioni preesistenti
iptables -A lan1lan2 -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables –A lan1lan2 –j LOG –log-prefix "Pacchetti lan1lan2 scartati: "
# il resto dei pacchetti vengono scartati
iptables -A lan1lan2 –j DROP

# Catena lan2lan1
# blocco dello spoofing
iptables -A lan2lan1 -s ! $lan2_add -j DROP
# concessione telnet, ftp e ssh
iptables -A lan2lan1 -p TCP --dport 23 -j ACCEPT
iptables -A lan2lan1 -p TCP --dport 21 -j ACCEPT
iptables –A lan2lan1 –p TCP --dport 20 –j ACCEPT
iptables -A lan2lan1 -p TCP --dport 22 -j ACCEPT
# concessione pacchetti relativi a connessioni già stabilite o relative a
# connessioni preesistenti
iptables -A lan2lan1  -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A lan2lan1 -j LOG --log-prefix "Pacchetti lan2lan1 scartati: "
# il resto dei pacchetti viene scartato
iptables -A lan2lan1 -j DROP

# Catena lan1inet
# blocco dello spoofing
iptables -A lan1inet -s ! $lan1_add -j DROP
# concessione dns e http
iptables -A lan1inet -p TCP --dport 53 -j ACCEPT
iptables -A lan1inet -p UDP --dport 53 -j ACCEPT
iptables -A lan1inet -p TCP --dport 80 -j ACCEPT
# concessione pacchetti relativi a connessioni già stabilite o relative a
# connessioni preesistenti
iptables -A lan1inet -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A lan1inet -j LOG --log-prefix "Pacchetti lan1inet scartati: "
# il resto dei pacchetti viene scartato
iptables -A lan1inet –j DROP

# Catena lan2inet
# blocco dello spoofing
iptables -A lan2inet -s ! $lan2_add -j DROP
# concessione dns e http
iptables -A lan2inet -p TCP --dport 53 -j ACCEPT
iptables -A lan2inet -p UDP --dport 53 -j ACCEPT
iptables -A lan2inet -p TCP --dport 80 -j ACCEPT
# concessione pacchetti relativi a connessioni già stabilite o relative a
# connessioni preesistenti
iptables -A lan2inet -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A lan2inet -j LOG --log-prefix "Pacchetti lan2inet scartati: "
# il resto dei pacchetti viene scartato
iptables -A lan2inet –j DROP

# Catena inetlan1
# blocco dello spoofing
iptables -A inetlan1 -s $lan1_add -j DROP
# concessione pacchetti relativi a connessioni già stabilite o relative a
# connessioni preesistenti
iptables -A inetlan1 -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A inetlan1 -j LOG --log-prefix "Pacchetti inetlan1 scartati: "
# il resto dei pacchetti viene scartato
iptables -A inetlan1 –j DROP
```

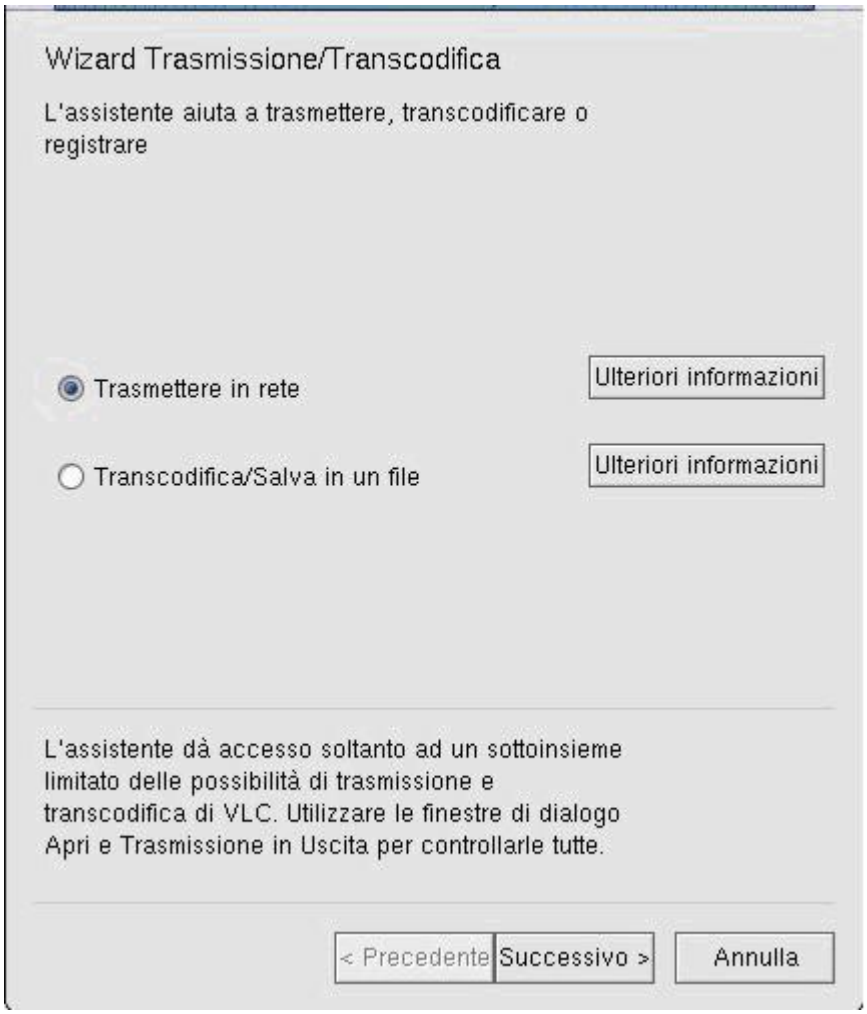
```
# Catena inetlan2
# blocco dello spoofing
iptables -A inetlan2 -s $lan2_add -j DROP
# concessione pacchetti relativi a connessioni già stabilite o relative a
# connessioni preesistenti
iptables -A inetlan2 -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A inetlan2 -j LOG --log-prefix "Pacchetti inetlan2 scartati: "
# il resto dei pacchetti viene scartato
iptables -A inetlan2 –j DROP

# Catena INPUT (pacchetti che arrivano al firewall)
# abilitazione dell'interfaccia di loopback per le applicazioni locali
iptables -A INPUT -i lo -j ACCEPT
# logging dei tentativi di attacco IP spoofing controllando che
# sull'interfaccia esterna non arrivino pacchetti con IP riservati
# a reti private
iptables -A INPUT -i $inet -s 192.168.0.0/16 -j LOG --log-prefix "Spoofing: "
iptables -A INPUT -i $inet -s 172.16.0.0/12 -j LOG --log-prefix "Spoofing: "
iptables -A INPUT -i $inet -s 10.0.0.0/8 -j LOG --log-prefix "Spoofing: "
# blocco dei pacchetti IP spoofing
iptables -A INPUT -i $inet -s 192.168.0.0/16 -j DROP
iptables -A INPUT -i $inet -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i $inet -s 10.0.0.0/8 -j DROP
# passaggio dei pacchetti icmp alla catena apposita
iptables -A INPUT -p ICMP -j icmp_chain
# concessione ssh dall'interno
iptables -A INPUT -p TCP -i $lan1 --dport 22 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 22 -j ACCEPT
# concessione ftp dall'interno
iptables -A INPUT -p TCP -i $lan1 --dport 21 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 21 -j ACCEPT
iptables -A INPUT -p TCP -i $lan1 --dport 20 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 20 -j ACCEPT
# concessione telnet dall'interno
iptables -A INPUT -p TCP -i $lan1 --dport 23 -j ACCEPT
iptables -A INPUT -p TCP -i $lan2 --dport 23 -j ACCEPT
# concessione pacchetti in entrata che hanno già stabilito una connessione o
# relativi a connessioni preesistenti
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A INPUT -j LOG --log-prefix "Pacchetti INPUT scartati: "

# Catena OUTPUT (pacchetti che escono dal firewall)
# abilitazione dell'interfaccia di loopback per le applicazioni locali
iptables -A OUTPUT -o lo -j ACCEPT
# disabilitazione per la lan1 della fruizione dello stream output vlc
iptables -A OUTPUT -o $lan1 -p UDP --dport 1234 -j DROP
# concessione dei pacchetti in uscita
iptables -A OUTPUT -o $lan1 -j ACCEPT
iptables -A OUTPUT -o $lan2 -j ACCEPT
iptables -A OUTPUT -o $inet -j ACCEPT
# registrazione nei log dei pacchetti scartati
iptables -A OUTPUT -j LOG --log-prefix "Pacchetti OUTPUT scartati: "
}

# funzione per disabilitare il firewall
firewall_off(){
# Flushing delle regole
iptables -F
iptables -F -t nat
iptables -F -t mangle
```

```
# Cancellazione delle chain user-defined
iptables -X
# Azzeramento i contatori
iptables -Z
# Ripristino delle policy di default: consentiamo tutto
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT

}

# funzione per visualizzare le regole inserite
firewall_display(){
iptables -L -v
}

# parametri per lanciare il firewall
case "$1" in
on) firewall_on;;
off) firewall_off;;
restart) firewall_off
         firewall_on;;
display) firewall_display;;
*) echo "Usage: $0 {on|off|restart|display}";;
esac
```

7. APPENDICE B: Implementazione dello streaming tramite VLC media player

Il metodo più facile per utilizzare lo streaming con VLC consiste nell'utilizzo dell'interfaccia grafica: wxwindows per Windows e GNU/Linux. Lo "Streaming/Transcoding Wizard" ci guida passo passo attraverso facili menù nel processo di streaming dei nostri media sulla rete e nel salvataggio degli stessi. Quindi sul pc che utilizzeremo come VideoLAN server (il nostro router) utilizzeremo questo strumento per consentire lo streaming.

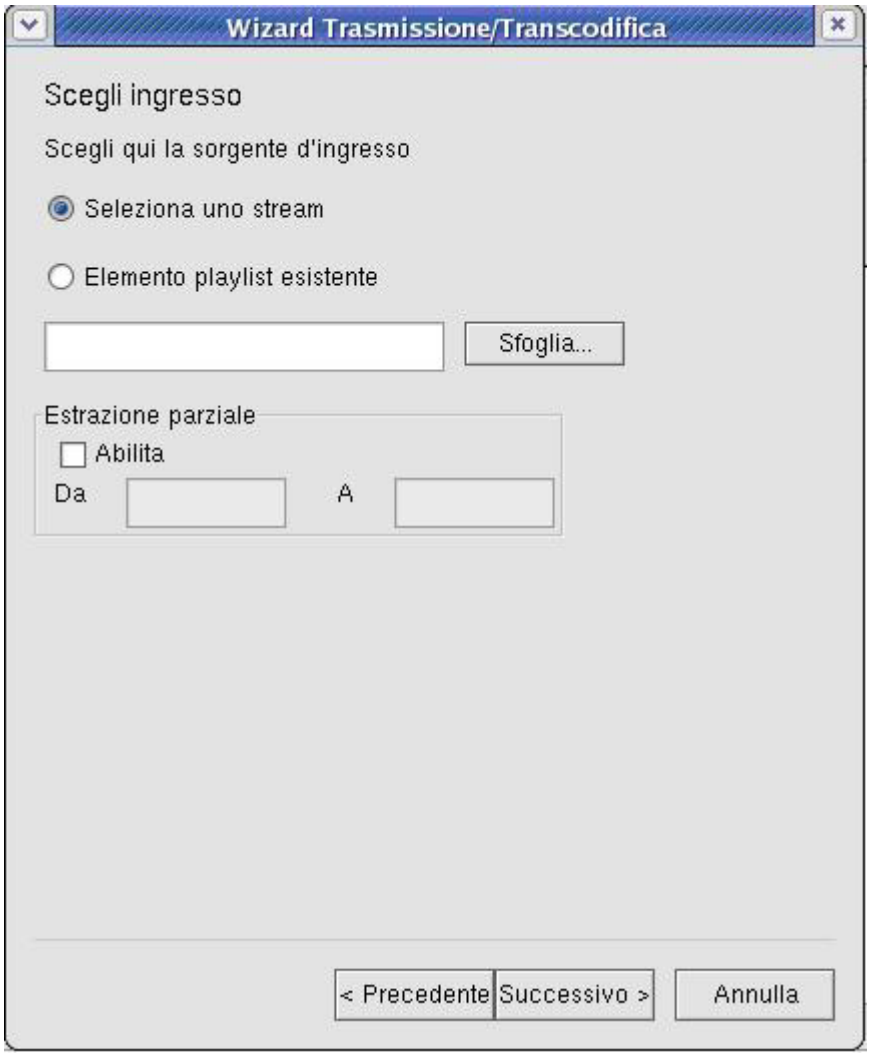
Per lanciare lo *Streaming/Transcoding Wizard*, è necessario aprire il menù "File" e selezionare Wizard. Quindi appare la seguente finestra:



Ora è necessario scegliere tra i due tasks:

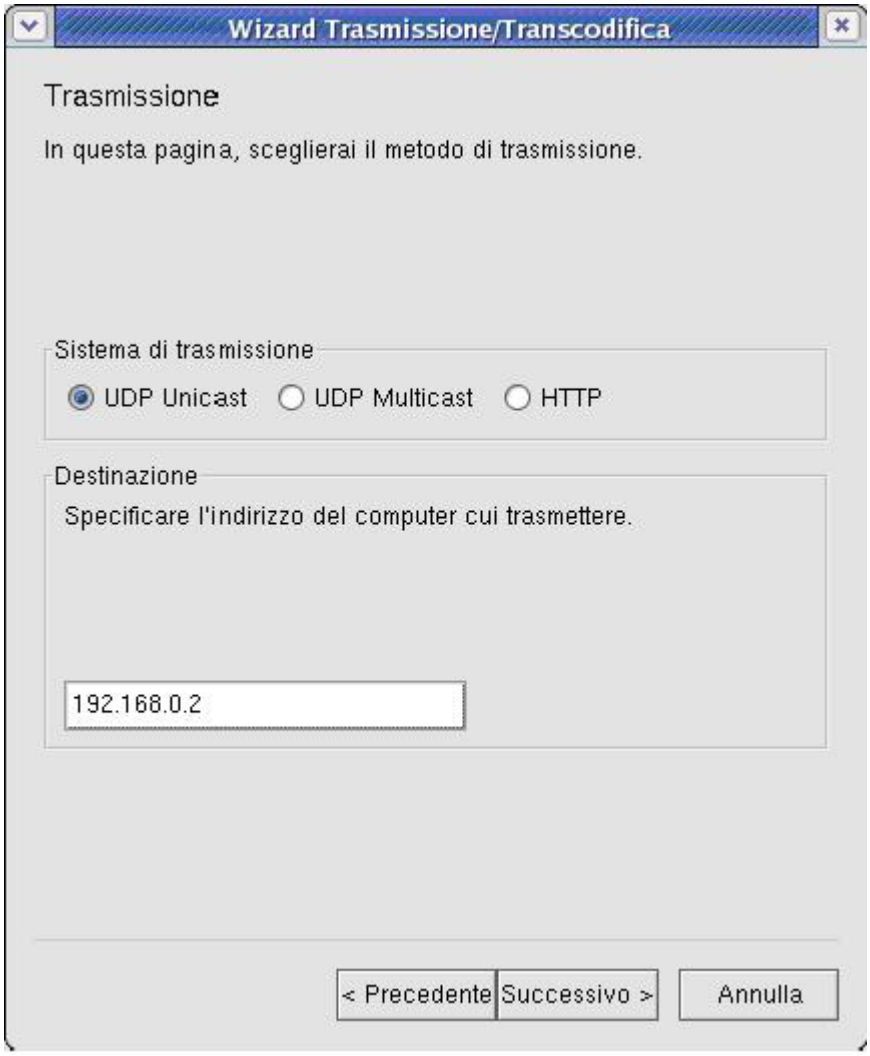
- *Trasmettere in rete*: per operare lo streaming di media sulla rete.
- *Transcodifica/Salvataggio in un file*: per modificare la codifica audio e/o video di un file il suo bitrate, o il suo metodo di incapsulamento.

Quindi dopo aver scelto il task trasmettere in rete e aver premuto il tasto “Successivo” ci appare la seguente finestra:



Ora quindi bisogna selezionare uno stream (un file, uno stream di rete, un disco…) selezionando da una playlist esistente o come nuovo attraverso il tasto “Sfoglia”. L’abilitazione del checkbox dell’estrazione parziale consente di leggere solo una parte dello stream (“da”… “a” in secondi).

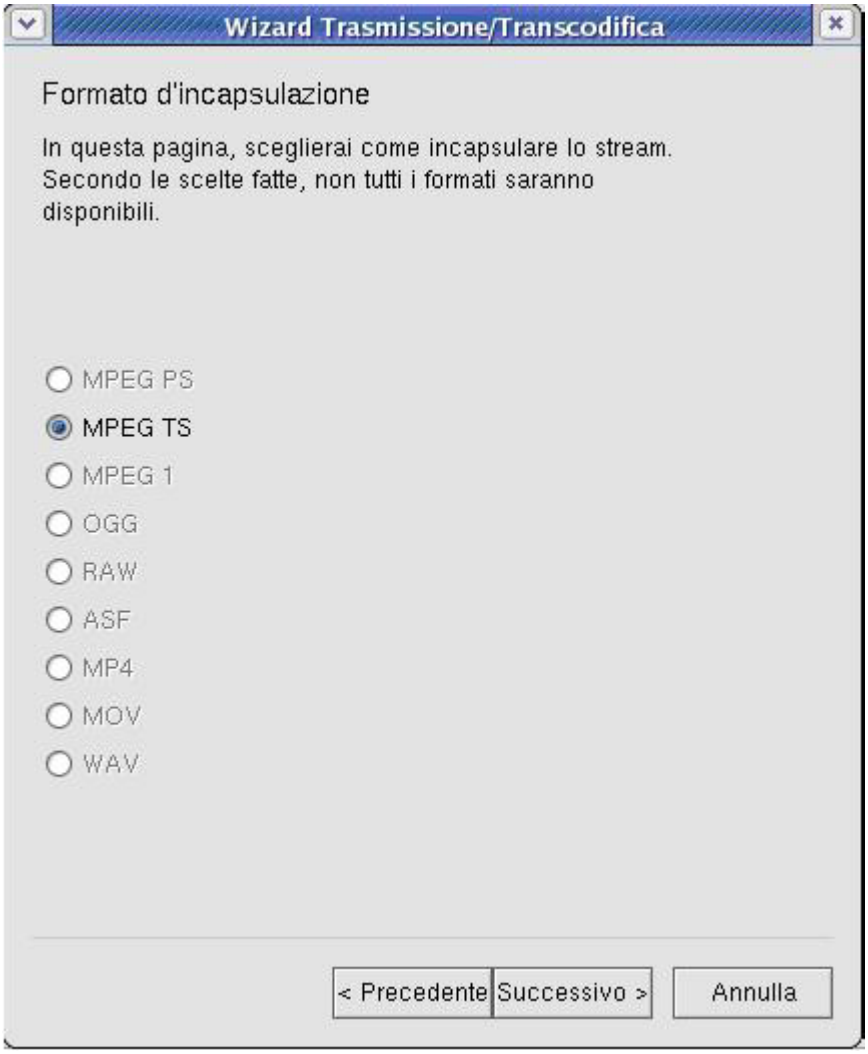
Ora vediamo la successiva finestra:



Come possiamo vedere qui è possibile selezionare il metodo di trasmissione:

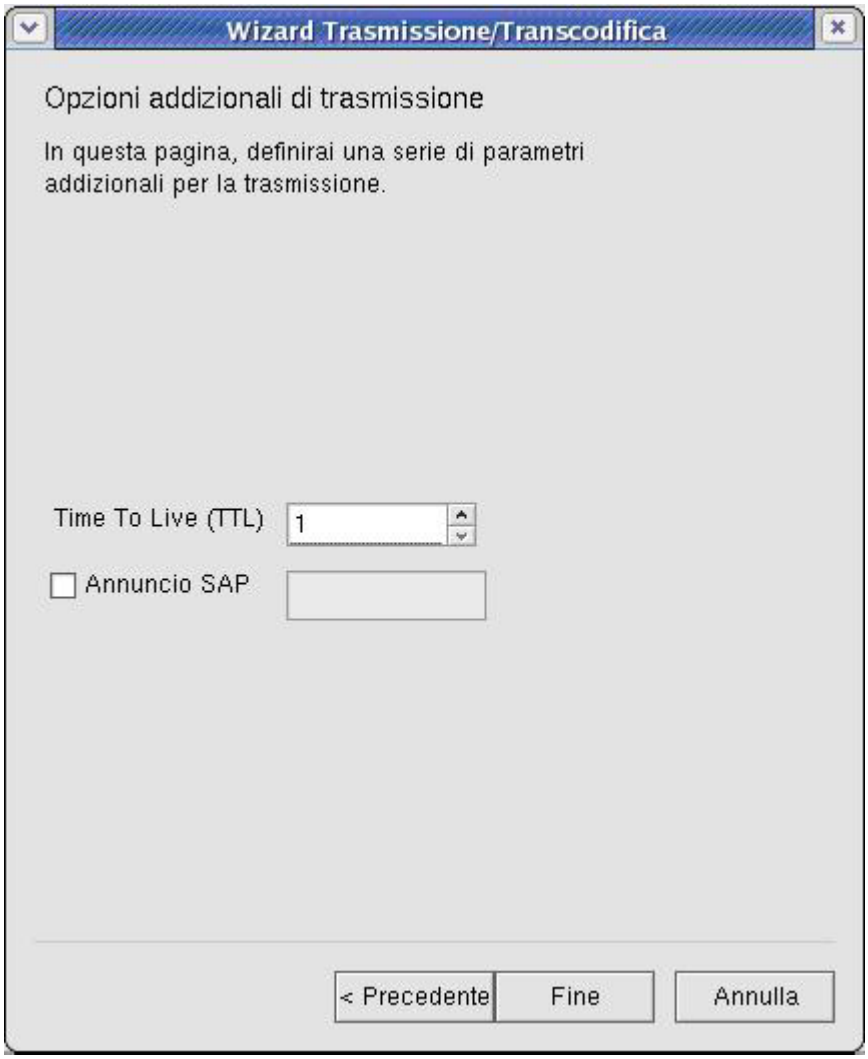
- *UDP Unicast*: permette lo stream a un singolo computer inserendo l'indirizzo IP nell'apposito campo di testo "Destinazione".
- *UDP Multicast*: permette lo stream a più computer utilizzando il multicast inserendo l'indirizzo IP del gruppo multicast nel campo di testo apposito.
- *HTTP*: permette lo stream attraverso il protocollo HTTP. Se si lascia il campo "Destinazione" vuoto, VLC starà in ascolto su tutte le interfacce di rete del server sulla porta 8080. Si specifica l'indirizzo, la porta e il percorso su cui stare in ascolto attraverso la seguente sintassi: [ip][:port][/path].

La finestra successiva sarà:



La finestra consente di scegliere il formato di incapsulamento. Gli streaming UDP richiedono il formato MPEG TS. Lo streaming HTTP può essere usato con MPEG PS, MPEG TS, MPEG 1, OGG, RAW o ASF.

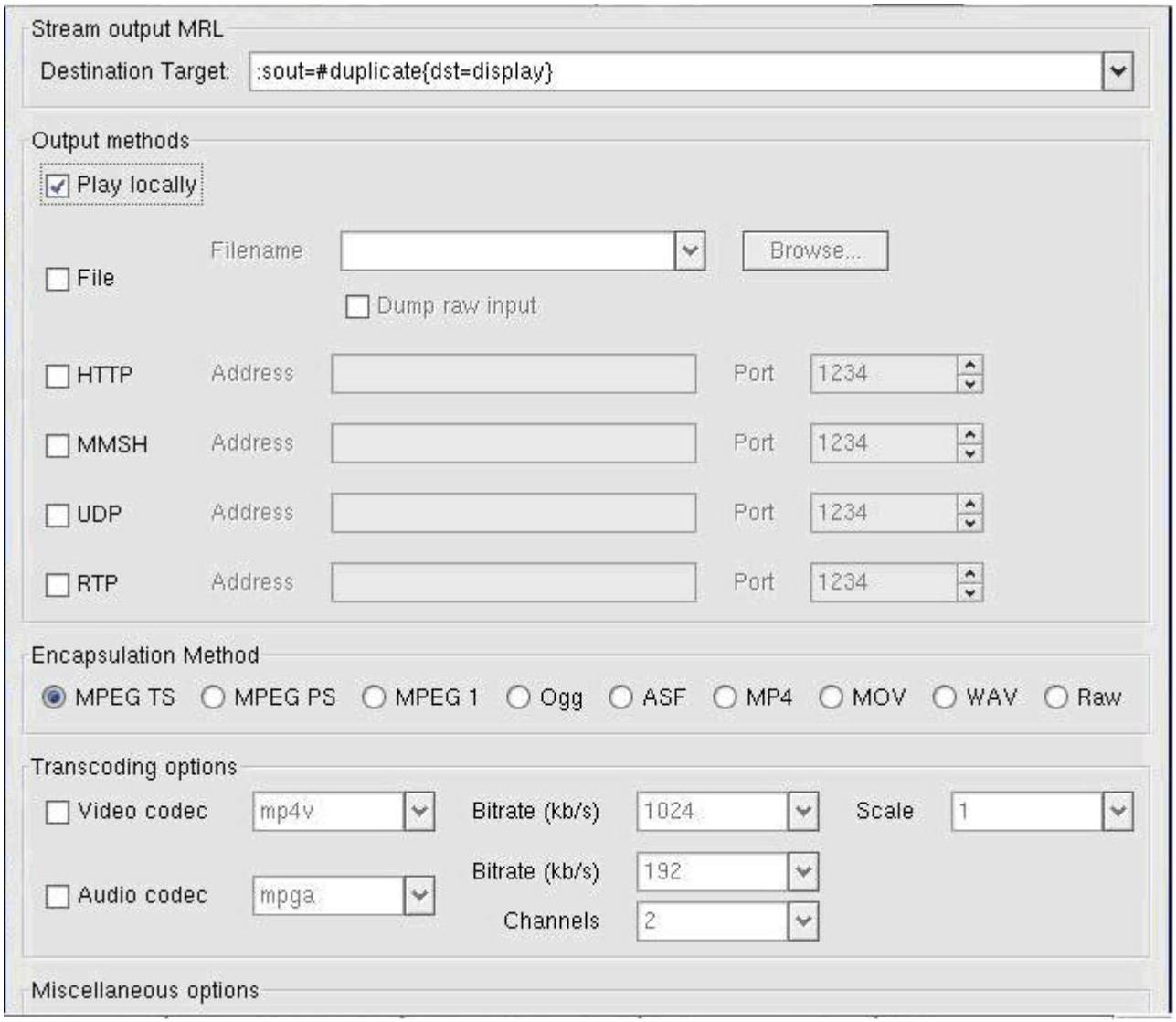
Ultima finestra:



Ulteriori opzioni che è possibile specificare sono:

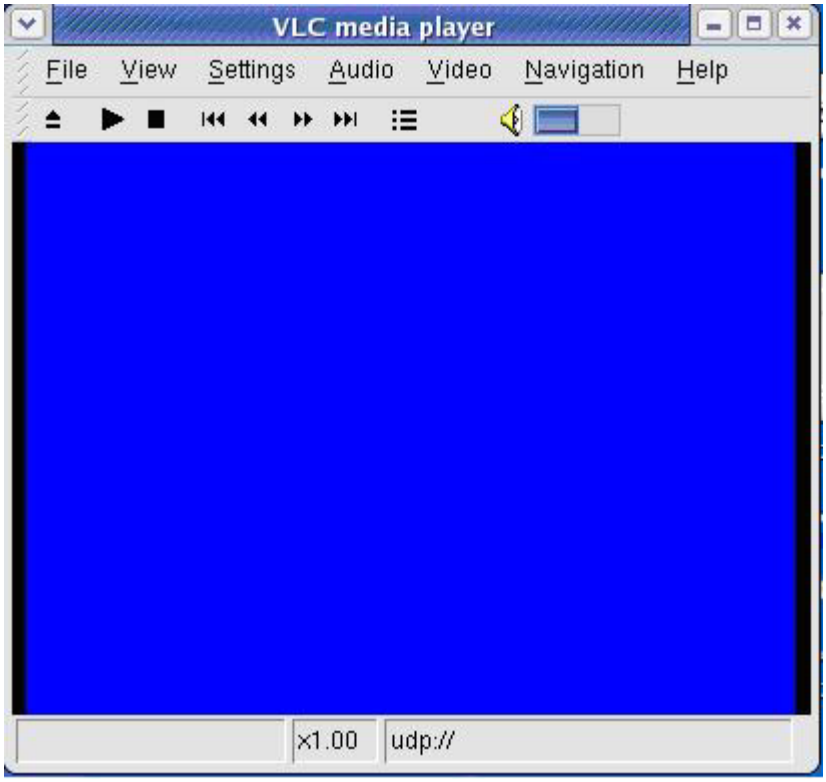
- *Time To Live (TTL)*: quest'opzione setta il numero di routers che lo stream può attraversare, per l'UDP unica e multicast.
- *SAP Announce*: Per avvertire dell'arrivo dello stream sulla rete usando l'UDP streaming utilizzando il protocollo SAP, immettendo il nome dello stream nel campo di testo. Non è utilizzabile con lo streaming HTTP.

Sul pc client sarà sufficiente aprire il menù “File” e selezionare “Apri sorgente di rete” e sulla scheda e abilitare il checkbox “Trasmissione in uscita” e premere il bottone “Impostazioni” per visualizzare la seguente finestra:



Abilitando il checkbox “Play Locally” si consentirà la riproduzione dello stream in locale.
Sotto vediamo l’interfaccia grafica in attesa della riproduzione:

Ed ora VLC trasmette il video mandato in streaming tramite il protocollo UDP sulla porta 1234:



8. RIFERIMENTI BIBLIOGRAFICI

Libri e dispense:

"Hacker Proof sicurezza in rete 2/ed" Kris Jamsa, McGraw-Hill

"Internet e reti di calcolatori 2/ed" di James F.Kurose, Keith W.Ross, McGraw-Hill 2003

Dispense del corso di Reti di Calcolatori (MN), prof. G. F. Rossi

"Manuale di configurazione di base della rete nel sistema operativo Linux" di Luca Capisani

Manpages di vari comandi Linux

Materiale online:

http://www.netfilter.org/ "Sito ufficiale di Netfilter/Iptables"

http://www.commedia.it/ccontavalli/

Iptables for Fun - Implementare un firewall in Linux di Carlo Contavalli

http://www.lugman.org/ "Linux User Group Mantova"

Introduzione alle reti in ambiente GNU/Linux e al firewalling con iptables di Grespan Lorenzo

http://www.valtellinux.it/ "Sito dedicato al mondo linux"

http://a2.pluto.it/ "Appunti di informatica libera"

http://openskills.info/ "Sito dedicato al mondo Linux"

http://sicurezza.html.it/ "Sito dedicato alla sicurezza"

Attacchi in una rete LAN e *Attacchi DoS* di Yan Raber

http://www.securelab.it/ "Sito dedicato alla sicurezza"

http://it.tldp.org/ "PLUTO Project – Italian Linux Documentation Project"

Security-HOWTO di Kevin Fenzi e Dave Wreski

Firewall-HOWTO di Mark Grennan

http://www.videolan.org/vlc/ "VLC media player"

VLC Streaming-Howto di Alexis de Lattre, Johan Bilien, Anil Daoud, Clément Stenac, Antoine Cellerier

Riviste:

Articoli su riviste dedicate al mondo Linux:

Linux & C. anno 4 numero 24-25-26, anno 5 numero 30

Red Hat Magazine anno 1 numero 3, anno2 numero 3-4-5

Linux pratico anno 2 numero 7, anno 4 numero 15-16-18

Hackers & C. anno 2 numero 8

Inter.net n.71-72-73