NetShield

Un firewall per la sicurezza di una rete

David Vincenzetti
vince@dsi.unimi.it

Stefano Taino
taino@dsi.unimi.it

Dipartimento di Scienze dell'Informazione
Università degli studi di Milano

Maggio 1994 [†]

ABSTRACT

La rete Internet sta diventando sempre più popolare e viene ormai usata da molti enti accademici e società commerciali. L'utilizzo dei computer collegati in rete fornisce innegabili vantaggi per la propria attività ma presenta anche dei rischi. Il problema della computer security, trascurato in passato, sta divenendo una delle questioni più importanti.

In questo paper descriviamo un *firewall*, uno strumento per proteggere un intero network di calcolatori collegato ad Internet. Il firewall di cui parliamo esiste come prototipo ed è stato creato dagli autori.

[†] Published in the proceedings of the i2u Convention, Milano, May 1994

1 - Introduzione

In questi ultimi anni, la rete Internet ha subito una grande espansione tanto che è stata definita la “rete di tutte le reti”. Oggi conta qualcosa come dodici milioni di utenti e interconnette quattromila altre reti in tutto il mondo. Offre servizi di ogni tipo: posta elettronica, accessi a banche dati, conferenze telematiche in tempo reale, e collega enti e istituzioni del mondo accademico, industriale e culturale.

Il principio fondamentale di Internet è lo scambio di informazioni tra il maggior numero possibile di soggetti, senza limitazioni geografiche, nazionali o politiche. Chiunque può accedervi stipulando un abbonamento, qualsiasi struttura può collegarsi, con poche formalità, al *resto del mondo*.

Nel momento in cui si decide di connettere il proprio computer ad Internet si accede, quasi sempre gratuitamente, a tutti i servizi che Internet ci mette a disposizione. Collegandosi ad Internet aumenta la nostra visibilità del mondo esterno, cioè degli altri *site* connessi alla rete, ma aumenta anche la visibilità del *nostro* site da parte delle macchine esterne. Potenzialmente un enorme numero di persone può accedere al nostro network. Nell'eventualità che avvenga una *intrusione informatica*, nell'eventualità, cioè che qualcuno riesca ad accedere al nostro sistema senza esserne preventivamente autorizzato, possono verificarsi alcune conseguenze:

- Nessuna conseguenza. L'hacker in questione è penetrato nel nostro sistema solamente per scopi teoretici. Molto raro.
- Lettura o “furto” di informazioni. Questa eventualità può portare a ulteriori conseguenze se le informazioni sono di tipo riservato per l'organizzazione.
- Modifica o distruzione dei dati. Le modifiche dei dati possono essere finalizzate ad un furto o ad un danno economico per l'azienda; la distruzione dei dati, di solito eseguita a *basso livello*, viene compiuta distruggendo il contenuto di uno o più dischi.
- Installazione di cavalli di troia, virus, backdoor. Anche se a prima vista può sembrare vero il contrario l'istallazione di programmi “parassiti” sul nostro sistema, come per esempio una *backdoor*, è l'ipotesi più sfavorevole [2]. Se il nostro hacker si limita ad installare una backdoor c'è la possibilità che non ci si accorga per nulla della violazione delle nostre misure di sicurezza e che da quel momento in poi l'hacker possa accedere al nostro sistema in qualsiasi momento, nonostante eventuali misure di sicurezza aggiuntive.
- Limitazioni di utilizzo. Rallentamenti o inoperabilità per quanto riguarda le nostre macchine e il nostro network. Ciò è causato dall'hacker che utilizza intensamente le nostre risorse per i suoi scopi. Una eventualità abbastanza comune è quella di un hacker che sfrutta illegalmente la potenza di calcolo di macchine altrui per effettuare il *cracking* delle password; i programmi che vengono utilizzati per questo scopo sono estremamente CPU bound.

Ogni azione di questo tipo comporta dei danni per il site vittima dell' attacco. Tali danni possono essere limitati ad alcune ore di lavoro extra “per rimediare all'accaduto” oppure costituire una ingente perdita economica e d'immagine per l'azienda.

La sicurezza base offerta dalle varie implementazioni del sistema operativo Unix non è più da considerarsi soddisfacente. Una nuova classe di hacker è in grado di realizzare attacchi sempre più avanzati e sofisticati. In particolare il network è il punto più debole dell'intera struttura, visto che proprio tramite esso è più facile che si verifichino violazioni informatiche.
C'è bisogno di uno strumento che ci permetta di sfruttare la rete e, nello stesso momento, ci garantisca una certa "chiusura" rispetto al mondo esterno. L'obbiettivo di ogni sistema per la sicurezza è quello di ottenere una rete il più possibile aperta, raggiungendo un compromesso tra le necessità di interoperabilità del sistema e i requisiti minimi per un network sufficientemente sicuro.
Per fare ciò proponiamo *NetShield*, un firewall atto ad aumentare drasticamente la sicurezza di un intero network. Un firewall system è la soluzione che un numero sempre più elevato di organizzazioni impiega per proteggere la propria subnet dal mondo esterno. NetShield costituisce lo strumento tecnico a cui vanno affiancate una adeguata formazione del personale e la costituzione di una "policy" che descriva ciò che è esplicitamente permesso ai fini della sicurezza del proprio network.
Questo documento illustrerà la parte prettamente tecnica e funzionale di NetShield. Bisogna tenere conto che questa è solamente una parte del lavoro. Altri documenti illustreranno tutti i prerequisiti e le procedure da svolgere periodicamente per conservare l'integrità e la sicurezza del sistema e del network.

2 - Definizioni

Nel seguito del documento con il termine "internal network" indicheremo la nostra LAN mentre con i termini "external network" o "public network" indicheremo il mondo esterno, cioè la rete Internet. "Router" indicherà un dispositivo che opera sui pacchetti IP al livello di network layer con possibilità di "packet filtering". Due classici esempi di router sono i Cisco e i Wellfleet router. Con "gateway" indicheremo una macchina con sistema operativo Unix che funge da "ponte" tra i due network al livello di application layer.
Il nostro discorso sarà condotto dalla parte di colui che deve proteggere una rete, e con il termine "hacker" si vorrà intendere colui che vuole, al contrario, penetrare nel sistema che noi abbiamo intenzione di proteggere. Alle volte, quando questo non generi confusione, con il termine "firewall" o "macchina firewall" si intenderà il sistema fisico su cui opera il software del firewall system.

3 - Il firewall

Un firewall può essere definito come un'insieme di dispositivi atti a proteggere un'organizzazione connessa ad un public network. Il principio alla base del firewall è il seguente:

è molto più facile proteggere un piccolo numero di sistemi piuttosto che centinaia o migliaia di macchine. Tramite un firewall la sicurezza di *tutta* la nostra LAN viene ricondotta alla sicurezza di un ristrettissimo numero di nodi, tipicamente una o due.
L'obbiettivo del firewall è di prevenire l'accesso non autorizzato alle risorse dell'internal network e di evitare che informazioni riservate possano essere trasferite attraverso la rete.
E' interessante osservare la diminuizione dell' "area a rischio" relativa al proprio network quando in quest'ultimo sia istallato un firewall. Nel caso che la nostra LAN sia connessa direttamente ad Internet l'internal network è soggeto a potenziali attacchi da parte degli innumerevoli hacker provenienti dal mondo esterno. Non è ovviamente detto che la nostra LAN sia vulnerabile a tutti gli attacchi provenienti dal "public network"; però, se non si fa uso di un firewall, è senz'altro necessario occuparsi della security di ogni singolo calcolatore all'interno della LAN affinchè un buon livello di sicurezza possa essere raggiunto.
Come dice il proverbio, la sicurezza di una catena è pari a quella del suo anello più debole: la sicurezza della nostra LAN, se tutti gli host sono direttamente accessibili dall'esterno, è pari a quella del nostro calcolatore meno protetto e si può tranquillamente affermare che nella maggioranza dei casi in ogni organizzazione vi è almeno un host che risulta "molto debole". Il fatto che questo host più debole degli altri, venga compromesso può sembrare non allarmante ma bisogna considerare che nel momento in cui l'hacker è riuscito ad entrare effettivamente su una delle nostre macchine allora la situazione diventa molto sfavorevole nei nostri contronti. Esistono, infatti, tecniche di attacco molto efficaci che permettono di compromettere *tutti* gli host di una LAN quando si sia ottenuto il pieno controllo di una sola della macchine componenti la LAN. Insomma, una volta che un hacker è riuscito a diventare *root* su una delle nostre macchine gli ulteriori attacchi ai nostri calcolatori proverranno dall'interno anzichè dall'esterno e avranno una maggiore probabilità di riuscire. Bisogna, quindi, evitare a tutti i costi che anche una sola delle nostre macchine venga compromessa e ottenere questo risultato senza l'ausilio di un firewall è cosa assai costosa in termini di denaro e di tempo dedicato alla security administration.

Alcuni servizi di Unix possiedono delle intrinseche debolezze e rendono difficile un controllo da parte dell'amministratore. I servizi NIS e NFS, entrambi basati su RPC, possono essere "ingannati" molto facilmente ed è possibile, così facendo, che qualunque hacker abbastanza smaliziato possa ottenere il nostro file delle password o montare remotamente un nostro disco senza averne l'autorizzazione [2].
Oltre agli attacchi che sfruttano oggettive debolezze del sistema operativo Unix bisogna tenere conto anche dell'inesperienza e della sprovedutezza di alcuni utenti. Per esempio molti utenti fanno un uso improprio degli *r* tools*, cioè dei servizi di rete Berkeley (*rlogin* e *rsh*) che consentono di personalizzare l'accesso dall'esterno alla propria macchina. Oppure eseguono il comando *xhost +*, permettendo a *chiunque* di connettersi al proprio X server.

Con un firewall l'area a rischio è ridotta o al solo firewall oppure a un ristretto numero di macchine "trusted". In caso di attacco il firewall permette, nella maggior parte dei casi, di bloccare l'attacco in corso fornendo dettagliate informazioni all'amministratore di sistema. Nel caso peggiore, il firewall può essere compromesso e quindi l'area a rischio si allarga all'intero internal network. Una volta guadagnato l'accesso al firewall un hacker può utilizzarlo come base di partenza per attaccare gli altri calcolatori della rete interna. In questo scenario l'ultima speranza è quella che l'hacker lasci delle tracce sufficienti per

la sua identificazione. Se il firewall è completamente distrutto l'intera LAN sarà esposta a potenziali attacchi rendendo problematica la ricostruzione dell'attacco.
In generale un firewall può essere visto come un sistema che riduce l' area a rischio della nostra rete ad un singolo, massicciamente protetto, punto di difesa. Questa potrebbe sembrare una soluzione un'pò rischiosa ma l'esperienza pratica ha dimostrato la validità della scelta che costringe l'hacker a concentrare il suo attacco esclusivamente sul firewall con elevata probabilità di essere scoperto.

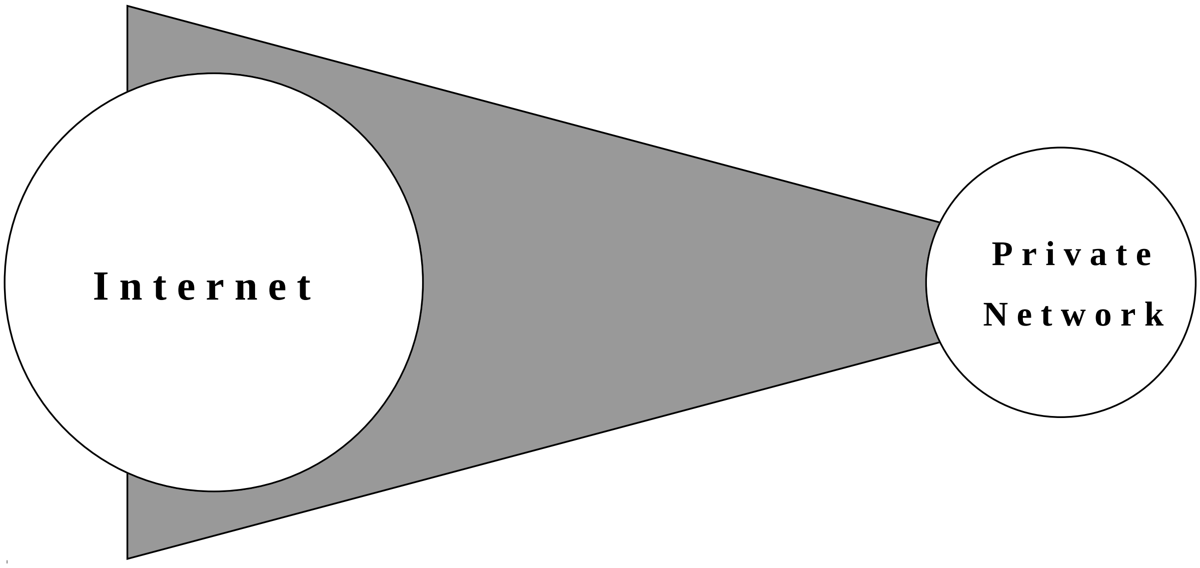


Figura [3.1] - "Area a rischio" senza firewall

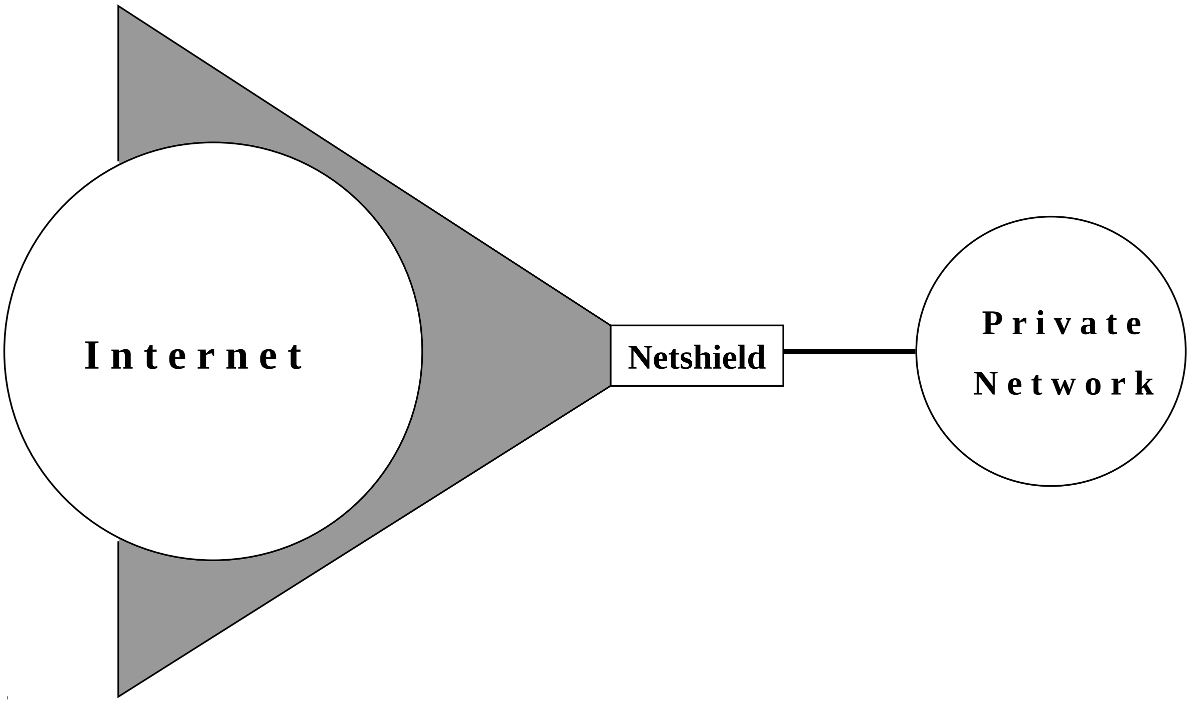


Figura [3.2] - "Area a rischio" con firewall

3.1 Firewall = router

Nel caso più semplice un firewall è costituito da un router. Questa soluzione è abbastanza comune, visto che per attuarla è sufficiente utilizzare le "access list" già incorporate in quasi tutti i router. Il router è utilizzato per effettuare uno "screening" dei pacchetti stabilendo quale debba essere il traffico inbound e outbound consentito. Le restrizioni sono usualmente di tipo "asimmetrico" e provvedono ad escludere soprattutto il traffico generato dall'external network verso l'internal network. Questa scelta può essere effettuata sulla combinazione di più criteri: il protocollo usato, l'indirizzo IP sorgente e destinazione, il "port number" destinazione.

Una configurazione comunemente adottata ha le seguenti caratteristiche:

- Traffico ICMP inbound
  Permesso con l'opzione di disabilitare la possibilità di "echo requests" in caso di tentativi esaustivi su tutti gli indirizzi dell'internal network finalizzati allo studio della topologia della rete.
- Traffico UDP inbound
  Negato per tutti i pacchetti con "destination port number" minore di 1024 tranne per i servizi di DNS e NTP. Opzionalmente si possono anche bloccare alcune porte che corrispondono a servizi UDP sopra la porta 1024. Per esempio è buona norma bloccare la porta 2049 utilizzata dai daemon NFS. Il meccanismo di autenticazione di NFS, infatti, e' assai debole ed è possibile, per un hacker esperto, riuscire a montare i nostri dischi anche se la macchina da cui viene condotto l'attacco non compare in `/etc/exports`.
- Traffico TCP inbound
  Vietato per tutti i pacchetti con "destination port number" minore di 1024 tranne per i servizi SMTP e NNTP per i calcolatori adibiti a quelle funzioni. Opzionalmente si possono anche filtrare alcune porte che corrispondono a servizi TCP sopra la porta 1024. Per esempio è buona norma bloccare la porta 6000, che corrisponde al server `X`, per scongiurare la possibilita' che si possa spiare dall'esterno la sessione di un utente.
- Traffico outbound
  Nessuna restrizione, completamente consentito. Nel caso dei Cisco router è necessario utilizzare la keyword `established`.

Una tipica access-list per il Cisco router è evidenziata nella figura seguente. Supponiamo che il network da proteggere sia 149.132.*.* e che l'indirizzo del router sia 149.132.1.1; supponiamo, inoltre, che i servizi SMTP, DNS, NNTP e NTP siano supportati, rispettivamente, dalla macchine 149.132.1.25, 149.132.1.53, 149.132.1.119 e 149.132.1.123.

!!!! ICMP inbound: disable pings

```
access-list 102 deny icmp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255 eq 8
```

!!!! UDP inbound

```
! permit DNS
access-list 102 permit udp 0.0.0.0 255.255.255.255 149.132.1.53 0.0.0.0 eq 53
! permit NTP
access-list 102 permit udp 0.0.0.0 255.255.255.255 149.132.1.123 0.0.0.0 eq
123
! deny when port less than 1024
access-list 102 deny udp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255 lt
1024
! deny NFS
access-list 102 deny udp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255 eq
2049

!!!! TCP inbound
! permit SMTP
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.25 0.0.0.0 eq 25
! permit NNTP
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.119 0.0.0.0 eq
119
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255
established
! deny when port less than 1024
access-list 102 deny tcp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255 lt
1024
! deny X
access-list 102 deny tcp 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255 eq
6000

!!!! permit all
access-list 102 permit ip 149.132.0.0 0.0.255.255 0.0.0.0 255.255.255.255
access-list 102 permit ip 0.0.0.0 255.255.255.255 149.132.0.0 0.0.255.255
```

Figura [3.3] - Access list di un Cisco router

Purtroppo un approccio di questo tipo presenta alcune debolezze e non è in grado di garantire un adeguato livello di sicurezza e interoperabilità:

- Esistono alcuni servizi (servizi RPC, X11 ecc.) che operano su port number dinamici e maggiori di 1024, quindi esposti a potenziali attacchi.
- Non esiste alcuna possibilità attraverso molti dei router comunemente usati (per esempio Cisco router) di fare un "monitoring" *attivo*, cioè registrare i tentativi di connessione non autorizzati e prendere le dovute azioni atte a contrastare l'attacco.
- Nessuna possibilità di accounting a livello utente.

- Interfaccia per la configurazione di un router generalmente poco user-friendly che si traduce spesso in errori di configurazione; diventa quindi possibile che il nostro sistema, da noi ritenuto sicuro, in realtà sia facilmente accessibile dall'esterno [1].
- Un hacker abbastanza smaliziato può lanciare un "server" che ascolti su una porta > 1024 e accedere alla subnet indisturbato. Se si volesse cambiare le ACL del router per bloccare tutto il traffico TCP entrante, anche quello con destination port > 1024, si potrebbe andare incontro ad alcuni problemi. Per esempio il procollo FTP fa sì che il server effettui delle connessioni al client su porte dinamicamente scelte > 1024. È possibile bloccare il traffico TCP > 1024 *e* usare FTP solamente se si ricorre a ftp client particolari capaci, ad esempio, di funzionare in *passive* mode [6].

3.2 Firewall = router + application gateway

Affiancando un gateway al router si ottiene una configurazione molto più stabile e si risolvono i problemi sopra menzionati. In questo caso il router ha la funzione di rendere il gateway l'unico punto di contatto tra l'internal e l'external network; come misura aggiuntiva le ACL sul router sono configurate in maniera da proteggere il gateway. Quest'ultimo ha la funzione di autenticare, registrare e controllare ogni connessione.
Una tipica access-list per il Cisco router è evidenziata nella figura seguente. Supponiamo che il network da proteggere sia 149.132.*.* e che l'indirizzo di NetShield sia 149.132.1.2. Le ACL sono più semplici perchè ci si appoggia al software istallato sul gateway. NetShield, infatti, si occupa di fare il "forwarding" delle connessioni all'application layer e regola completamente il traffico sia in entrata che in uscita.

!!!! UDP inbound

```
access-list 102 permit udp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 53
access-list 102 permit udp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 123
```

!!!! UDP outbound

```
access-list 102 permit udp 149.132.1.2 0.0.0.0 0.0.0.0 255.255.255.255
```

!!!! TCP inbound

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 21
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 23
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 25
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 53
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0 eq 119
access-list 102 permit tcp 0.0.0.0 255.255.255.255 149.132.1.2 0.0.0.0
established
```

!!!! TCP outbound

```
access-list 102 permit tcp 149.132.1.2 0.0.0.0 0.0.0.0 255.255.255.255
```

Figura [3.4] - Access-list di un Cisco router affiancato ad un gateway

Il gateway diventa quindi l'unico punto di contatto tra la rete esterna e quella interna; attraverso il gateway un utente interno o esterno può accedere all'altra parte del network in modo completamente trasparente, previa la sua identificazione e autenticazione da parte del gateway stesso.

Nel disegno e nello sviluppo di NetShield abbiamo tenuto presenti le seguenti esigenze:

- Supporto dei servizi TELNET, FTP, SMTP e NNTP.
- Possibilità per l'amministratore di controllare l'accesso ad ogni servizio basato sullo user id, indirizzo IP, ora del giorno, giorno della settimana, port number sorgente e destinazione e direzione della connessione.
- Capacità per il sistema di agire come "mail gateway" per tutta la LAN. Come misura di sicurezza addizionale tutti i riferimenti ai nomi dei calcolatori interni sono convertiti al solo nome del gateway per l'e-mail in uscita.
- Tramite opportuna configurazione del DNS ogni informazione riguardo la tipologia e la topologia della LAN è preclusa a utenti e macchine esterne.
- Un sistema di "logging" e accounting molto dettagliato per ogni servizio è configurabile a seconda del livello di informazioni che si vogliono avere.
- Notifica immediata di ogni attività sospetta attraverso e-mail, *syslog* ecc. "Report" giornaliero con archiviazione automatica di tutti gli eventi che si sono generati.

4 - Controllo degli accessi

Il firewall può essere visto come un "supervisore di attività" per la rete su cui si pone. Attraverso il firewall "passano" tutte le connessioni tra internal ed external network e quindi il firewall è in grado di decidere quali connessioni autorizzare, quali registrare e quali negare. Il paradigma alla base del sistema di controllo è il seguente: quello che non è espressamente permesso viene negato. Il controllo degli accessi è governato da una tabella di accesso che definisce e regola tutto il traffico inbound / outbound. La sintassi della tabella è elastica e semplice ed è possibile speficare "contraints" individuali per ogni utente, specificare controlli sulla data e il giorno della settimana, associare delle "action" ad ogni collegamento.

La tabella d'accesso è, fondamentalmente, divisa in due parti. Nella prima parte ci sono delle regole generali che vengono applicate ai collegamenti in entrata o in uscita per tutti gli utenti. Nella seconda parte c'è una lista di record, uno per ogni utente della subnet, che contengono alcuni dati relativi all'utente (data di ultimo login, provenienza dell' ultimo login, sistemi di autenticazione supportati dall'utente) e, anche, un set di *limitazioni* che vanno a "sovrascrivere" le regole specificate nella prima parte della tabella. Attualmente, l'accesso e modifica dei record è possibile solamente collegandosi fisicamente alla macchina firewall. In una prossima release tramite un opportuno programma client sarà possibile

per un amministratore di una subnet accedere via rete alla tabella di configurazione di NetShield senza avere necessariamente un account sul gateway. In questo modo ogni singolo sysadm potrà gestire gli accessi alla rete a seconda delle esigenze dei suoi utenti aumentando il controllo e la sicurezza dei collegamenti.
Un esempio di tabella d'accesso è il seguente:

```
#
# A SAMPLE FIREWALL CONFIGURATION.
# (suppose our network is .secure.org, David Vincenzetti and
# Stefano Taino are trusted users, John Bad is an untrusted
# user)
#

# we don't want connections from these bad sites
deny ALL from bad.site.com, bad2.site.com to *.*.*.* {
        exec = '/usr/local/bin/hacker';
}

# allow telnet connections from everywhere to out subnet
allow telnet from *.*.*.* to *.secure.org {
# permit connections when between 8 a.m.  and 8 p.m.
        time = [8, 20];
# permit connections on working days only
        day = 1, 2, 3, 4, 5;
# perform RFC391 checking
        rfc931;
# exec something upon connecting
        exec = '/usr/local/bin/logmng %s > /etc/nsx.log 2>&1 &';
# look for sensitive strings
        grepin = '^(passwd|nasagov|goblin)$';
        grepout = '^(WARNING.*YOU ARE SUPERUSER)$';
}

# allow ftp connections from everywhere to our ftp site
allow ftp from *.*.*.* to ftp.secure.org {
        ;
}

# allow all outgoing connections. we trust our users...
allow ALL from 149.132.*.* to *.*.*.* {
# ...but we still look forward interesting strings!
        grepin = '^(shutdown|mdformat|crack)$';
}

#
```

```
# User records.  The constraints defined inside user records
# override the ones defined in the general rules.  This way
# it is possibile to set constraints on a user basis.
#

# personal record for David Vincenzetti
David.Vincenzetti {
        lastlogin = ghost.secure.org, 762901439;
# can choose between different auth methods
        auth = safeword;
# Can only connect from some systems
        from:  149.132.*.*,
               home.david.net,
               work.money.com;
# further logging facility
        exec = '/staff/vince/bin/personal';
        timeout = 30;
}

# personal record for Stefano Taino
Stefano.Taino {
        lastlogin = goblin.sm.secure.org, 763372844;
# can choose between different auth methods
        auth = s/key, secureid;
        from:  149.132.*.*,
               work.moremoney.com;
        timeout = 30;
}

# personal record for John Bad
John.Bad {
        lastlogin = untrusted.edu, 760144823;
        auth = s/key;
# John can NOT escape from the subnet.
        to:  149.132.*.*;
        rfc931;
        grepin = ALL;
# 100 ms delay for each block of data
        delay = 100;
        exec = '/usr/local/bin/hacker';
        time = [10, 19];
        day = 1, 2, 3, 4, 5;
}
```

Figura [4.1] - Esempio di configurazione di NetShield

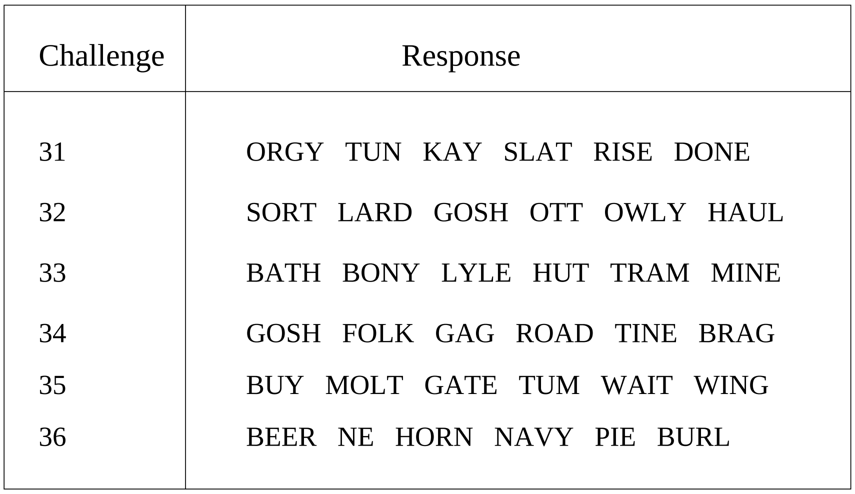
Come si vede usando i *personal record* è possibile trattare differentemente utenti diversi. Settando in maniera opportuna i campi `from:` e `to:` si può effettivamente controllare la provenienza di un utente e limitare il traffico inbound/outbound. Inoltre è possibile definire delle *action* da eseguire in determinate condizioni o rallentare alcune connessioni per evitare che alcuni utenti possano esportare velocemente grosse quantità di dati appartenenti all'organizzazione. I sistemi di autenticazione al firewall sono definiti nel campo `auth`.

Inoltre il gateway effettua un controllo atto a smascherare un tipo d'attacco abbastanza comune: il cambiamento fraudolento di IP address di una macchina interna. Se, per esempio, abbiamo un PC nella nostra rete e abbiamo saggiamente deciso che tale PC non debba essere abilitato a chiamate verso l'esterno della LAN, un utente sufficientemente smaliziato potrebbe cambiare l'IP address del PC facendolo diventare quello di una macchina "trusted" e poi chiamare tranquillamente il mondo esterno fingendosi un altro host. Questo attacco è molto comune anche quando abbiamo a che fare con x-terminal, terminali intelligenti configurabili con un indirizzo IP. Per rendere vano questo tipo di attacco viene mantenuto dal gateway un database con tutti gli indirizzi Ethernet (MAC) di tutte la macchine della LAN e per ogni indirizzo IP viene sempre controllato il corrispondente indirizzo Ethernet.

5 - Sistemi di autenticazione

L'evoluzione dei sistemi integrati per l'elaborazione delle informazioni ha reso critico il problema dell'identificazione degli utenti. L'identificazione di un utente da parte di un calcolatore è generalmente eseguita attraverso l'uso di uno "username" e di una "password". La password è mantenuta segreta e deve essere difficile da indovinare. Solamente l'utente deve conoscere la password associata al proprio username. È però vero che a volte le password si possono indovinare secondo alcune ben note tecniche di "password cracking". Inoltre quando username/password sono utilizzati attraverso dei canali di trasmissione dati come la rete Internet c'è la possibilità che vengano catturati perchè viaggiano in chiaro come l'utente li digita sulla tastiera. Questo sistema di autenticazione ci è apparso molto debole anche alla luce di alcuni recenti episodi di "hacking" basati su tecniche di "eavesdropping" delle password. Ci è quindi sembrato importante aumentare la resistenza del sistema di autenticazione ad attacchi di questo tipo. Abbiamo realizzato questa idea utilizzando "one time password" e le smart card della Security Dynamics$^{(tm)}$ ed Enigma Logic$^{(tm)}$.

Le one time password sono password che possono essere utilizzate una sola volta; così facendo un attacco come quelli precedentemente citati diventa completamente innocuo. Abbiamo scelto di utilizzare `S/Key`, un programma disponibile pubblicamente che implementa il concetto di one time password in modo molto flessibile e facile [3]. La password segreta dell'utente non viaggia mai attraverso la rete durante la procedura di login. Nessuna informazione segreta relativa alla password è custodita sul firewall e l'algoritmo per la generazione delle chiavi è pubblicamente conosciuto. Un utente prima di collegarsi deve generare, utilizzando un programma apposito sul proprio calcolatore, la sequenza di “challenge/response” e custodire la lista fino alla scadenza di tutte le one time password. Ogni challenge/response è utilizzabile una sola volta. Esse vengono generate in base alla password segreta che l’utente inserisce una sola volta nell’applicazione. Ovviamente l’unico rischio è legato alla perdita della lista e al fatto che qualcuno riesca a sapere lo username e il sistema a cui sono associate le one time password.



Challenge	Response
31	ORGY TUN KAY SLAT RISE DONE
32	SORT LARD GOSH OTT OWLY HAUL
33	BATH BONY LYLE HUT TRAM MINE
34	GOSH FOLK GAG ROAD TINE BRAG
35	BUY MOLT GATE TUM WAIT WING
36	BEER NE HORN NAVY PIE BURL

Figura [5.1] - Esempio di lista di one time password con S/Key

Un metodo più sicuro ma più costoso è implementato attraverso l’uso di *smart card*, device hardware delle dimensioni di una piccola calcolatrice tascabile, simili a normali carte di credito ma capaci di elaborazione propria. Un vantaggio evidente offerto da questa soluzione è che, mentre una password fissa può essere rubata senza che il proprietario se ne accorga, la scomparsa di un token è immediatamente evidente. Usando un digital token c’è una relazione molto stretta, *verificabile*, tra l’account con il quale vengono compiute delle azioni e il suo legittimo prorietario; non è più possibile, quindi, che un utente affermi: “Non sono stato io, qualcuno deve aver rubato la mia password!”.

Ci sono diverse tipologie di “token”:

- Challenge-response: il computer visualizza un sequenza random di caratteri. L’utente la digita sulla smart card che restiuisce un “response” che l’utente inserisce nel calcolatore.

- Sincroni: la smart card visualizza una password random da inserire nel calcolatore. La password è generalmente *time dependent*.

- Semi sincroni: agisce come i token sincroni ma a volte per ristabilire una sincronizzazione ha bisogno di una procedura di challenge/response.

Enigma Logic offre un modello di smart card di tipo challenge/response. L'utente fornisce il proprio username, il software sul firewall propone un numero casuale chiamato *challenge* e richiede la password. Se l'utente fornisce la corretta password dinamica, tramite l'utilizzo del token ottiene l'accesso al sistema. Il challenge e, di conseguenza, la password, sono ogni volta differenti.

La Security Dynamics offre *SecurID*, che supporta token di tipo sincrono *time based*. La smart card possiede al suo interno un unico *seed* utilizzato per l'operazione di encryption. Ogni 60 secondi visualizza un valore numerico basato sul seed la data del giorno e l'ora. Il software presente sul firewall deve a sua volta conoscere il seed, lo username dell'utente e l'algoritmo di cifra. Alcuni modelli richiedono l'inserimento di un PIN (Personal Identification Number) prima di poter utilizzare il token. Questa misura aggiuntiva previene il furto del token stesso.

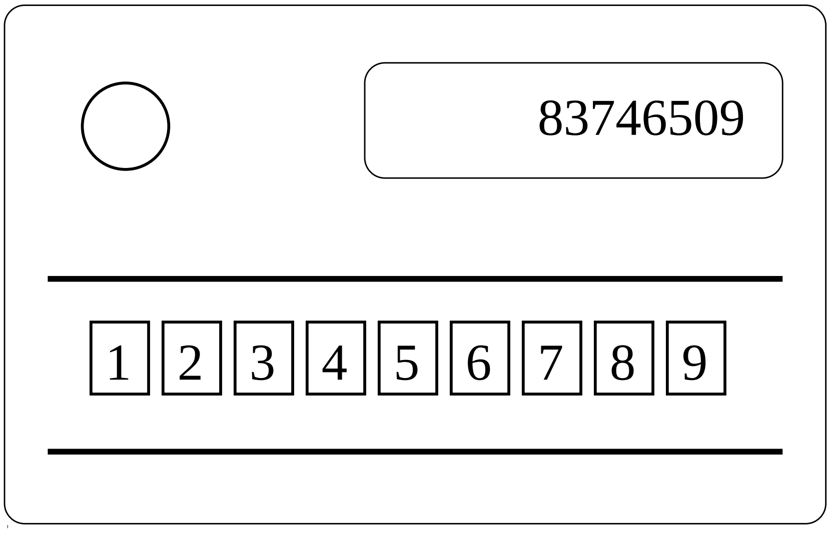


Figura [5.2] - Esempio di Smart-card

6 - Servizi supportati

6.1 Telnet e Ftp

Per permettere l'accesso ai servizi telnet/ftp inbound/outbound attraverso il firewall senza comprometterne la sicurezza è stato sviluppato un proxy server. Il server esercita un controllo sulle connessioni basandosi sulle regole stabilite nella tabella degli accessi e richiede un'autenticazione da parte dell'utente tramite uno degli strumenti descritti nel paragrafo precedente. Gli utenti su un lato del firewall che vogliono accedere ad un computer sull'altro lato devono prima connettersi al gateway ed autenticarsi. Tutte le connessioni e i byte trasferiti sono registrati secondo le specifiche esigenze. Opzionalmente è possibile disabilitare il sistema di autenticazione per i collegamenti dall'interno verso l'esterno. Nel caso di *ftp* è possibile abilitare o disabilitare alcune operazioni (es: `put`, `get`) riducendo il rischio che qualche utente possa esportare dei dati di particolare importanza per l'organizzazione.
Consideriamo il seguente esempio: l'utente "David.Vincenzetti" vuole stabilire una connessione telnet dalla macchina `abc.foobar.it` al calcolatore `goblin.secure.org`, interno alla nostra subnet `secure.org`.
L'unica possibilità è chiamare la macchina `NetShield.secure.org`, perchè una connessione diretta a goblin.secure.org sarebbe impossibile per l'azione di packet filte@iffgfffffst Internet ed in

un utente sul "public network" chiede informazioni per il nostro dominio può vedere solo quello che noi abbiamo stabilito nei file di configurazione sulla macchina gateway perchè le connessioni al name server interno sono bloccate.

6.3 NNTP

NNTP è il protocollo per il trasferimento delle news. Per non degradare le performance e per ridurre al minimo la presenza di software sulla macchina gateway abbiamo sviluppato un daemon che agisce come “ripetitore” tra il calcolatore da cui riceviamo le news e il nostro news server locale. Il daemon è in grado di regolare gli accessi tramite la stessa sintassi usata per le connessioni telnet e ftp. Lo stesso concetto può essere applicato ad altri servizi come ad esempio gopher, WWW ecc.

7 - Requisiti di NetShield

La configurazione della macchina gateway deve corrispondere a uno standard di sicurezza molto elevato. Il suo ruolo all'interno del sistema è molto importante, la sua “compromissione” potrebbe automaticamente invalidare la sicurezza dell'“internal network”. A questo scopo, valgono alcune regole per il raggiungimento dell'obbiettivo:

- Disabilitazione, attraverso opportuna modifica del kernel della della macchina gateway, di fare un “forwarding” diretto dei pacchetti IP sull'internal network.
- Nessun supporto di protocolli per il “routing” dinamico (RIP, IGMP, OSPF ecc).
- Unica login di `root` con possibilità di collegamento interattivo o da console o tramite apposito programma client che supporta autenticazione e comunicazione tramite tecniche crittografiche. Nessuna altra forma di connessione che fornisca accesso ad una “shell” sul gateway è supportata.
- Disabilitazione dei servizi NIS, NFS ecc.
- Massima restrizione sui permessi di accesso per file e directory del file system.
- Eliminazione di parti del sistema operativo non strettamente necessarie al funzionamento del firewall.

Oltre a queste regole di carattere generale è prevista l'installazione di tre packages per un monitoring *attivo* dei possibili eventi sospetti. Questi tools coprono aspetti differenti per la sicurezza di un calcolatore con sistema operativo Unix collegato in rete.

`ATP` (Anti Tampering Program) è un programma che ha il compito di assicurare l'integrità di una macchina Unix. In pratica ATP è in grado di “fotografare” lo stato del sistema e successivamente di controllare che non siano state affettuate alterazioni, deliberate o causate da eventi esterni, su tutti i file che hanno una importanza critica per il sistema [8]. ATP è particolarmente utile per evitare che il nostro sistema venga attaccato e che siano istallate delle “backdoor” su di esso.

`NetTrace` opera un “logging” di tutto il traffico inbound e outbound analizzando gli “header” dei pacchetti IP,TCP e UDP alla ricerca di connessioni insolite.Una tipica registrazione di collegamento permette di avere le seguenti informazioni:

- data del collegamento

- host sorgente/destinazione
- source/destination port number
- primi byte della sessione

Inoltre, opzionalmente, si può abilitare la “feature” di analisi automatica del contenuto della parte dati dei pacchetti TCP e UDP per la ricerca di precise “firme” che mettano in evidenza un tentativo di attacco.

L’output di syslog per il monitor di tutti gli eventi di sistema e di rete può assumere spesso dimensioni considerevoli da rendere spesso inefficiente una sua scansione manuale o con l’ausilio dei classici comandi Unix. A questo scopo viene affiancato un programma specializzato nell’estrarre e mettere in evidenza possibili eventi critici con una determinata serie di “action”.

8 - Conclusioni

In risposta ad una classe di hacker sempre più agguerrita e specializzata abbiamo sviluppato un firewall in grado di vanificare gli attacchi che attualmente vengono condotti sfruttando le debolezze intrinseche del sistema operativo Unix e di alcuni protocolli di rete. La nostra azione si è sviluppata seguendo quattro strategie differenti:

- “Packet filtering”, per la sua capacità di protezione di un elevato numero di calcolatori da eventuali attacchi.
- Identificazione e autenticazione dell’utente tramite protocolli non suscettibili ai comuni attacchi tradizionali. Si utilizzano, infatti, solamente dispositivi di one time password e digital token.
- “Monitoring” attivo per ottenere informazioni dettagliate riguardo la provenienza e le tecniche di attacco degli hacker.
- Mantenimento dell’integrità dei calcolatori con controllo automatico e sistematico dei file system.

L’unione e le sinergie tra queste azioni realizzano un valido compromesso tra il mantenimento dell’interoperabilità dei sistemi e un buon livello di sicurezza cui oggi deve tenere in giusta considerazione ogni organizzazione che decida di collegarsi ad Internet.

9 - Bigliografia

[1] D. Brent Chapman, "Network (In)Security Through IP Packet Filtering", Proceedings of the 3rd USENIX Security Symposium, September 1992

[2] David Vincenzetti & Stefano Taino, "Sulla (In)Sicurezza di Unix in ambiente TCP/IP" Conferenza AICA 1993, Gallipoli (LE).

[3] Phil Karn, Neil M. Haller and John S. Walden, Bellcore, S/Key software kit, available via anonymous ftp from `thumper.bellcore.com /pub/nmh/skey/*`

[4] Marcus J. Ranum, "Thinking about Firewalls", Proceedings of Second International Conference on System and Network Security and Management (SANS-II), April 1993

[5] Bill Cheswick, "The Design Of a Secure Internet Gateway", Proceedings of the 3rd USENIX Security Symposium, September 1992

[6] S. Bellovin, "Firewall-Friendly FTP", RFC 1579, 2/18/94

[7] M. St. Johns, "Authentication server", RFC 931, 1/1/1985

[8] David Vincenzetti & Massimo Cotrozzi, "ATP: Anti Tampering Program", Proceedings of the 4th USENIX Security Symposium, October 1993