Firewall:
strategie, implementazioni ed architetture.

Autore:
Armando Leotta
armyz@idic.caos.it

Internet è una società e come tale è afflitta dai più disparati tipi di molestie commesse da utenti più o meno maliziosi.
Così come è possibile incappare nel classico teppista di strada che imbratta i muri o legge e/o strappa la posta altrui senza apparente motivo, sulla Rete delle Reti non è raro leggere articoli sui sempre più frequenti tentativi *(hack attempts)* d'intromissione da parte di gente smaliziata che per innumerevoli e totalmente differenti motivazioni trovano questo lato di Internet alquanto affascinante.
Come conseguenza, nuove realtà devono essere necessariamente progettate tenendo conto della sicurezza della rete stessa e dato che gli obiettivi (*targets)* più appetibili per un *hacker* sono i sistemi "vecchi" (vecchio in informatica è tutto ciò che non è prontamente aggiornato) con *bugs* e non protetti si viene a delineare sempre più marcatamente la figura professionale d'esperto in sicurezza delle reti.
Per motivi di retaggio culturale e tecnici tutti i termini informatici non di facile e di chiara traduzione saranno lasciati inalterati di proposito e scritti in *corsivo*.

Definizione di firewall

Il *firewall* è un componente di rete che serve a proteggerne una parte rispetto al resto.
Esso è collocato solitamente tra una rete interna (privata) e la rete WAN esterna (solitamente Internet) per evitare accessi indiscriminati alla rete interna da parte di *host* collocati all'esterno.
Generalmente, i compiti del firewall sono svolti da un elaboratore opportunamente configurato con almeno due schede di rete, una per ciascuna rete "visibile" da questa macchina (una "vede" la privata , l'altra Internet).

Strategie

Least priviledge

Consiste nel più importante principio di sicurezza: concedere i privilegi strettamente ed esclusivamente se necessari.
Dare accesso come *root* su un sistema UNIX a tutti gli utenti che hanno accesso su quella macchina non è solo senza alcun senso ma è anche masochistico.
Concedere ad un utente che genera solo traffico WEB la possibilità di utilizzare l'FTP o IRC può essere solo fonte di possibili problemi, situazione non giustificata peraltro dalle esigenze.
Unix è altamente configurabile per quanto riguarda i diritti/attributi dunque piuttosto che comunicare la *pass* di *root* ad un utente che deve lanciare un programma con un diverso *groupid* è sicuramente referibile vagliare la possibilità di cambiare gli attributi al programma o all'utente stesso (magari assegnadolo a quel gruppo).

Defense in depth

Consiste nell'adottare più meccanismi di sicurezza.
Solitamente è preferibile non affidarsi ad unica soluzione la quale, se scardinata, comprometterebbe la nostra rete, dati, reputazione, immagine etc etc.
Per esempio, quando introdurremo il concetto di *packet filtering* vedremo che è una buona idea porne due in cascata (anche se è normale che si occupino di regole (*rules*) diverse ha senso configurare il secondo ridondante in modo da scartare tutto ciò che il primo non avrebbe mai dovuto far passare).

Choke point

Consiste nell'imporre un unico varco per il traffico di dati sia entrante che uscente.

È chiaramente più sicuro concentrare il nostro sforzo per controllareun' unica macchina che raccoglie tutto il traffico da e per Internet piuttosto che controllare tutte le macchine.
Immaginate grandi aziende con oltre 2 o 3 mila *workstastion* (e immaginate anche che per renderle sicure devono anche essere aggiornate!)

Weakest Link

Dopo *least priviledge* un altro dogma della sicurezza delle reti : una catena è tanto robusta e sicura quanto lo è il suo anello più debole.
Possiamo pensare di investire molti soldi nella sicurezza, nelle migliori soluzioni commerciali e/o public-domain ma chiaramente un punto meno sicuro di un altro ci sarà sempre..
Sarà cura dei responsabili della sicurezza far si che sia il meno debole possibile dato che presumibilmente sarà obiettivo di tutti gli attacchi.

Ad esempio, proteggere o addirittura vietare il servizio Telnet e non porre alcuna protezione sull'FTP significherebbe trovarsi una syslog piena connessioni indesiderate sulla porta 21.
Lo stesso avviene quando si ha un'eccezionale riservatezza dei dati, ottime configurazioni delle varie macchine esposte ma nessuna protezione sulle macchine vere e proprie, magari *unattended*.

Fail-Safe Stance

Consiste nella logica di prevedere , in seguito ad un eventuale attacco andato a buon fine , di provocare una situazione di errore tale da bloccare comunque ogni tentativo di accesso (autorizzato o meno) piuttosto che consentirlo a tutti.
Ciò significherebbe anche negare l'accesso ad utenti autorizzati finchè i sopralluoghi , le scansioni delle log ed il successivo ripristino della macchina non avvengano ma ciò è una situazione di norma accettabile se si pensa al danno che si è sicuramente evitato.

A tal proposito, esistono due posizioni in merito: *Default Deny Stance* e *Default Permit Stance.*
Default Deny Stance
Questa posizione stabilisce che qualsiasi cosa che non abbia riscontro nelle *rules* del *firewall* sia proibita per default
Il tutto rientra nella logica: se non conosco può arrecarmi danno.
È estremamente restrittiva e necessita di più tempo per la configurazione ed il tuning, pena il traffico bloccato per il malcapitato utente interno che magari non aveva menzionato il servzio di news *tra le richieste effettuate al* net-admin.
Default Permit Stance

Per contro, esiste la logica opposta: permettere tutto ciò che non è espressamente vietato.
Chiaramente è da valutare di volta in volta, da scenario a scenario la metodologia da adottare.
Mi piace sperare che non sia mai adottato niente del genere se non per test.

Terminologie inerenti ai firewall

Bastion host

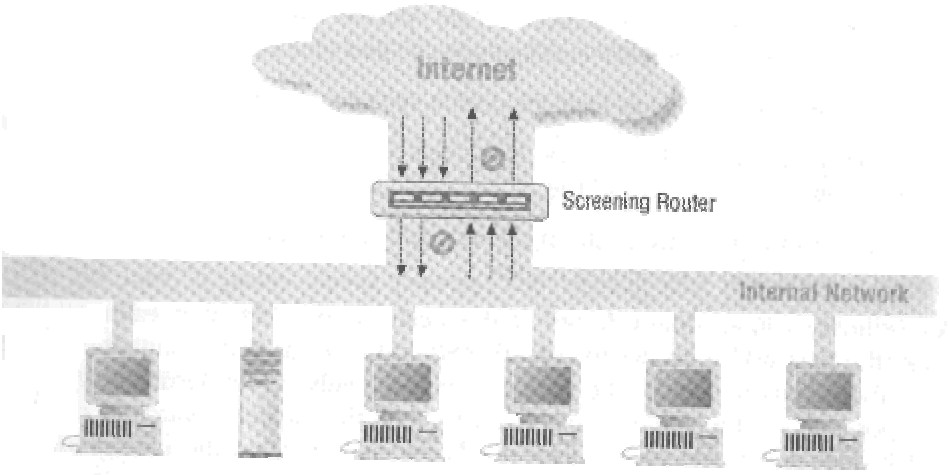
Si tratta di un computer che deve essere particolarmente protetto in quantointrinsecamente vulnerabile essendo punto di raccolta del traffico interno (privato) e di inoltro verso l'esterno il tutto coordinato solitamente da untcpd wrapper che monitorizza,filtra e inoltra le richieste ai vari servizi consentiti.

Multi-homed host

Si tratta di un computer con più schede di rete.

Packet filtering

È un dispositivo atto a permettere o a bloccare il transito dei pacchetti .
Concettualmente questa azione viene solitamente compiuta durante il cosiddetto*routing* (instradamento) da una rete ad un'altra (o da una rete interna ad Internet e viceversa).
Come avremo modo di vedere in dettaglio, la tecnica del*packet filtering* (nota come *screening*) è accompagnata da un set di *rules* (regole) le quali costituiranno le effettive*policies* della nostra rete (bloccare un servizio preciso, o una*well known port* , etc).
Essa può aver luogo in un *router*, in un *bridge*, o in un *host* dedicato.
Il router implementato in un firewall che effettua *packet filtering* si dirà *screening router*.



Perimeter Network

Consiste in un ramo di rete interposta tra la rete interna (protetta) e la rete esterna (solitamente Internet, dunque insicura).
Ha chiaramente lo scopo di creare un livello di sicurezza in più dato che si "allontana" la zona da proteggere da quella potenzialmente fonte di attacchi.
Una *perimeter network* è anche chiamata in gergo **DMZ**, *De-Militarized Zone* , zona demilitarizzata (in seguito alla separazione Nord-Sud Corea).

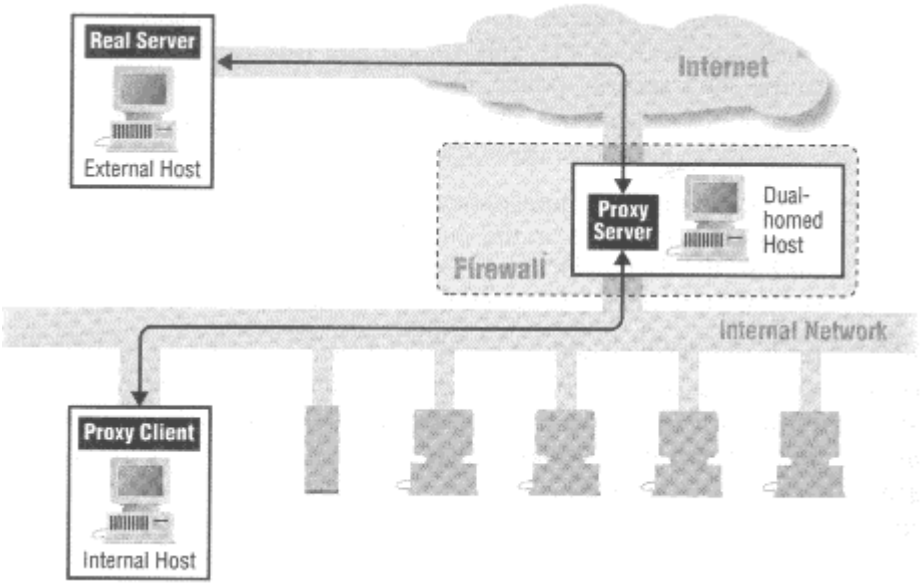
Proxy Server

Non è parte integrante del *firewall* in sè ma è sicuramente un componente che permette ai*security admin* di gestire, controllare e monitorare il traffico *inbound/outbound* (entrante / uscente) focalizzando i loro sforzi su queste macchine.
I client (originariamente dovevano essere appositiproxy client, adesso quasi tutti i generici client permettono l'utilizzo tramite proxy server) dialogano esclusivamente con il proxy server , sarà cura di quest'ultimo inoltrare o meno (al contrario dei firewall , i proxy possono entrare nel merito del singolo utente, estensione del fileetc etc) i pacchetti verso i server originali e riconsegnare la risposta alclient.
Il tutto è chiaramente trasparente :

- il server ignora che la richiesta gli arriva da unproxy e non da un classico client;

- il client ignora di non contattare direttamente il server di destinazione e di dialogare esclusivamente con una macchina locale.

Altra importante feature del proxy è la cache: è possibile dimensionare, configurare e mantenere un servizio di cache per le risorse più accessate (si pensi al traffico http) con un conseguente minor spreco di banda e maggiore velocità (ad esempio se la pagina web da visitare è stata già visitata recentemente quest'ultima verrà caricata dalla cache).



Implementazione del Packet Filtering

Quando un pacchetto arriva ad un'interfaccia di uno screening router alcune informazioni contenute nel suo header vengono esaminate per decidere se inoltrare o bloccare il pacchetto:

- *IP source address*
- *TCP or UDP source port*
- *IP destination address*
- *TCP or UDP destination port*
- *Protocol*
- *ICMP message type*

Inoltre, una volta arrivato il pacchetto , il *router* sarà in grado di:

- sapere su che interfaccia di rete è arrivato il pacchetto
- sapere su che interfaccia dovrà inoltrare **o meno** il pacchetto (consultando le sue tabelle di routing).

È a questo punto che il funzionamento di un *router* classico si differenzia da uno *screening router* .

Un *router ordinario* una volta che viene raggiunto da un pacchetto su una delle sue interfaccie di rete guarda il *destination address* di ogni pacchetto e consultando le *routing tables* trova il percorso migliore per inoltrare il pacchetto.
Solo ed esclusivamente in base all'indirizzo di destinazione ilrouter ha due possibilità :

1. trova un'occorrenza sulle tabelle di *routing* e lo instrada scegliendo il miglior percorso a disposizione;
2. non trova alcuna occorrenza e non potendolo instradare manda un **ICMP** " *destination unreachable*" al mittente.

Lo *screening router*, invece di "limitarsi" a decidere come instradarlo, entra nel merito del pacchetto e decide se gli è permesso l'instradamento o meno.
Il tutto nel pieno rispetto delle *rules*.
Ad esempio è possibile configurare le suddette regole per :

- bloccare tutte le connessioni entranti ad eccezione delle richieste SMTP (per consentire il ricevimento della posta);
- bloccare tutte le connessioni provenienti da host ritenuti non sicuri o non graditi;
- consentire **FTP**,**email**, **HTTP** ma bloccare servizi pericolosi come il **TFTP**,**X** window system, e i **"remote" service** tipo rlogin,rexec,rcp,rsh etc).

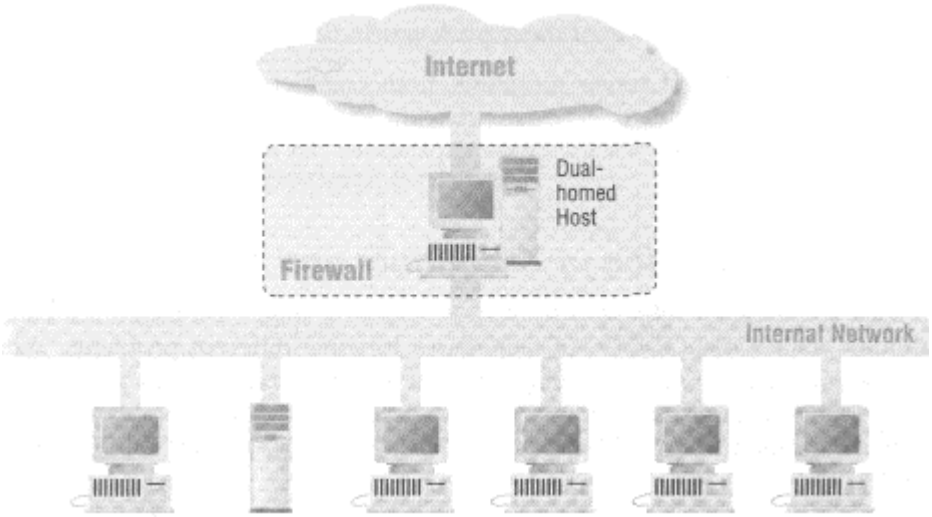
Sarà compito del *security admin* decidere se sovraccaricare un unico router per entrambe le funzioni o dedicarne uno espressamente al *packet filtering*.

Combinazioni di tecniche,tecnologie e architetture

Il progetto di un firewall non è semplice soprattutto perchè raramente è possibile (e conveniente) applicare una singola tecnica per proteggere adeguatamente tutte le macchine e i vari servizi.
Alcuni di questi infatti hanno delle peculiarità che fanno propendere per alcune soluzioni invece di altre.
Ad esempio, **Telnet** e **SMTP** si prestano particolarmente all'implementazione del *packet filtering* mentre altri per protocolli come ad esempio **Gopher**,**WWW**,**Archie**,**Ftp** risulta più efficiente una soluzione con *proxy*.
Infatti, la maggior parte dei firewall in uso adottano un approccio misto.

Dual-Homed Host Architecture

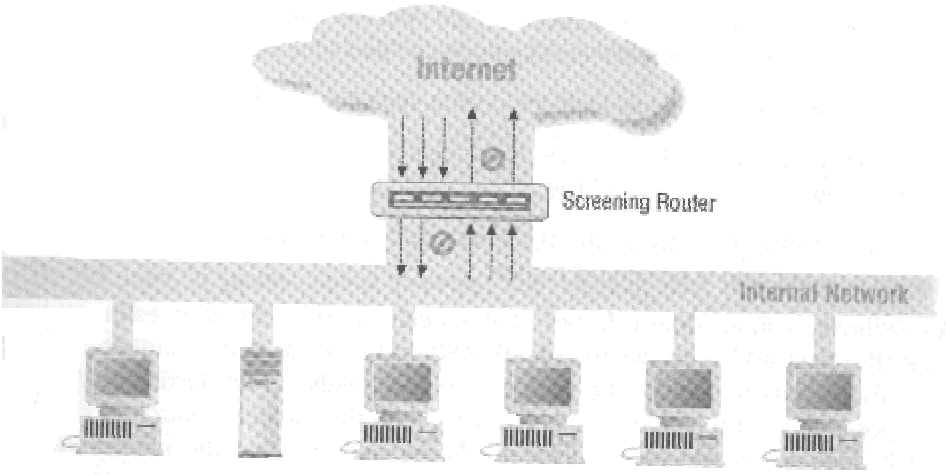
È una delle architetture più semplici e funzionali: essa prevede un multi-homed host che è fisicamente tra la parte sicura (interna) e la parte critica (WAN o internet) alle quali è connesso tramite le diverse interfaccie di rete (in caso di due interfaccie si dirà *Dual-homed*).



In questo caso l'opzione di routing IP tra le due (o più) schede deve essere disabilitata in modo da poter entrare nel merito del tipo di traffico.
Questa architettura viene implementata solitamente congiuntamente ad un proxy in modo da permettere agli utenti interni di poter "uscire" sulla rete pubblica senza il rischio sicuramente elevato di consentire loro di effettuare login sulla macchina multi-homed.
È infatti preferibile limitare al massimo le utenze su una macchina così delicata come la multi-homed in questa configurazione.

Screened Host Architecture

A differenza della precedente architettura che vedeva i servizi forniti da un host collegato a più network , nella *screened host architecture* i servizi vengono assicurati da un *host* collegato esclusivamente alla rete interna tramite un apposito *screening router*.

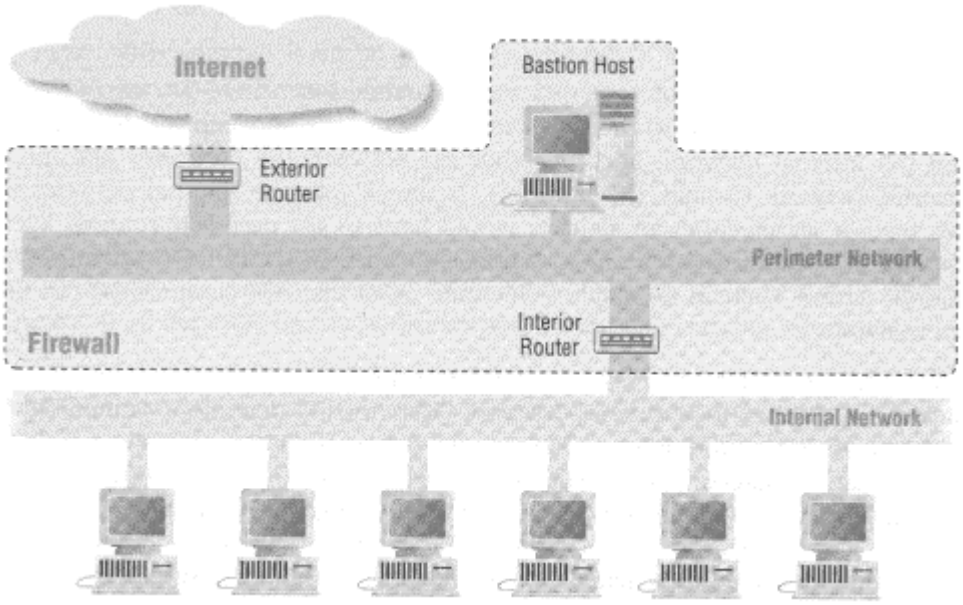


Il bastion host è nella rete interna ed un'apposita configurazione dello screeningrouter lo rende unico host visibile e raggiungibile dall'esterno.
Inoltre il packet filtering effettuato dal router può farsi carico di bloccare tutti quei tentativi di connessione diretta che utenti interni possono effettuare provando ad aggirare ilproxy.
Tutto deve passare dal bastion host, sia traffico entrante (opportunamente controllato e filtrato dalrouter) che uscente (controllato sia dal bastion host che dal router stesso).
Come al solito alcuni protocolli si prestano più di altri ad una soluzione piuttosto che ad un'altra.
Esistono due punti "deboli" :

1. eccessiva esposizione del *bastion host*. Esso, per quanto possa essere protetto dallo *screening router*, sarà sempre esposto ad attacchi proprio per come è strutturata la rete e, una volta compromesso, nulla lo separerà dalla restante rete interna ormai totalmente indifesa;
2. nel caso di compromissione dello screeningrouter, tutta la rete sarà in mano agli hackers.

È per queste ragioni che diviene sempre più frequente ricorrere ad una variante, la *screened subnet*.
Screened Subnet Architecture

Questa architettura prevede un livello di sicurezza aggiuntivo tramite la creazione della *perimeter network* che isola ulteriormente la rete privata da proteggere dalla rete esterna.
Infatti interponendo questo ramo di rete tra il *bastion host* e la rete interna si riduce drasticamente le conseguenze di un eventuale compromissione del *bastion host* stesso.



La configurazione più semplice prevede l'utilizzo di due *screening router* entrambi connessi alla *perimeter network*: uno è disposto tra la rete esterna e la perimeter, l'altro tra quest'ultima e la rete interna.
Con questa disposizione per poter raggiungere le macchine private un eventuale *hacker* dovrebbe compromettere entrambi gli *screening router*.
Inoltre, l'unico *host* visibile dall'esterno è ancora una volta il *bastion host* ma in questo caso,un eventuale *exploit* su esso non comprometterebbe l'intera rete interna dato che è separata dal secondo *screening router*.
In questa configurazione il router che si affaccia sulla rete pubblica viene chiamato anche *exterior router* mentre quello che separa la *perimeter* dall'interna prende anche il nome di *interior router*.
È inoltre possibile adottare perimeter multipli in modo da creare dei livelli di sicurezza progressivi dislocando i servizi più delicati dopo l'ennesimo screening router allontanandolo sempre più dalla potenziale fonte di attacchi che è la rete esterna. Chiaramente è una scelta più costosa da gestire e da realizzare ed è opportuno realizzarla esclusivamente se le *filter rules* dei vari router sono diverse tra loro.

Perimeter network

Un altro aspetto che rende utile l'implementazione di una o più *perimeter network* è dovuto al cosiddetto *packet sniffering*. È una tecnica spesso messa in atto da *hackers* che una volta compromesso una macchina (ed aver guadagnato accesso *root*) istallano pacchetti che esaminano e loggano il traffico all'interno della rete sia essa ethernet, fast ethernet, token-ring o FDDI spesso accompagnati dai cosiddetti ***rootkit*** i quali sostituiscono i comandi cruciali unix per nascondere le tracce dell'intrusione (esempio lampante il ***ps***, proc status NON riporta il pacchetto in questione).
In quest'ottica soltanto il traffico all'interno della *perimeter* potrebbe essere tracciato ma non il traffico sulla rete privata.

Il compito dell'***interior router*** (a volte chiamato *choke router*) è di proteggere la rete interna sia dalla rete pubblica che dalla *perimeter*.È infatti a lui demandato il carico maggiore del *packet filtering*.
Invece, **l'*exterior router*** (chiamato anche *access router*) oltre ad avere solitamente le stesse regole per il traffico entrante dell'*interior* tende ad essere abbastanza permissivo per il traffico uscente dato che verrà raggiunto da richieste già controllate dall'*interior* e valutate anche dal *bastion host*.
Inoltre è responsabile della sicurezza delle macchine nella *perimeter* , primo fra tutti proprio il *bastion host*.

Combinazioni di tecniche

È possibile scegliere un'architettura che meglio si presta alle esigenze tecniche (hardware disponibile,phisical planning, IP disponibili,subnet già funzionanti da tempo) ed amministrative (tempi,budget,formazione,etc) proprie della rete in questione.
Esistono comunque delle variazioni ormai abbastanza frequenti ed apprezzate :

- utilizzare più di un bastion host con indubbio alleggerimento del carico di lavoro per ciascuna macchina ma si ha una macchina cruciale in più da proteggere;
- unificare interior e exterior router in un unico dispositivo ;
- utilizzare più exterior routers (solitamente quando si hanno più connessioni alla rete pubblica o più WAN);
- utilizzare più perimeter networks;
- utilizzare dual-homed host all'interno di Screened Subnet.

Altre fortemente sconsigliate :

- implementare un'unica macchina per bastion host e interior router (se viene compromesso il bastion host nulla separerà la rete privata dall'attacco;
- implementare più interior router (difficile da configurare e da tenere allineati ed in alcuni casi potrebbero essere la causa del transito di dati importanti sull perimeter e se il bastion host ha uno sniffer il gioco è fatto).

Scenario

Vogliamo adesso creare un firewall : motivazioni proprie della rete in oggetto , costi, hardwareetc ci fanno propendere ad esempio per una *Screened Subnet Architecture*.

Oltre alle macchine che costituiscono propriamente ilfirewall supponiamo di volere una mail server, un news server, un dns server e vari client.
Ognuno di questi servizi interni viene assicurato tramite connessione diretta (epacket filtering) o indiretta (reindirizzata tramite service proxy sul bastion host).
Supponiamo che tutti gli utenti interni (ritenuti fidati) non provino abypassare il firewall e che non sia necessario uno strumento di controllo anche all'interno della rete privata (cosa comunque di facile attuazione,).
Si assume che l'assegnazione degli indirizzi IP,Subnet, gateway siano configurati e instradati correttamente verso la rete pubblica (compito quest'ultimo del fornitore di accesso).
Per le macchine interne è possibile specificare indirizzi IP privati sfruttando le potenzialità di un servizio diproxy; se si vuole visibilità "diretta " dall'esterno è necessario specificare un indirizzo IP assegnato dal fornitore o dal NIC(Network Information Center)
Per ulteriori dettagli esaminare le RFC (Request for Comments, sono degli standard per Internet) numero 1597 e successivi (1627 e 1918).
Inoltre, condizione di estrema rilevanza, supponiamo di utilizzare diversi Net ID (identificativo rete) per laperimeter e l'internal network in modo da poter rilevare pacchetti con presunta origine interna sull'interfaccia esterna e viceversa (*source address forgery attack*).

TCP & Packet filtering

Packet filtering è il miglior approccio per le connessioni relative a servizi basati sul traffico TCP.
Ragionando in termini di richiesta/risposta (dunquedirezionalità del traffico) stabilire una connessione viene trattata come una richiesta e la trasmissione su una connessione già stabilita come una risposta.
Intuitivamente, è una politica accettabile in quanto se unhost interno tenta una connessione verso un servizio esterno potrebbe essere considerato implicito il consentireall'host interno (source) di ricevere le risposte da questohost.
In altre parole questa logica implica che il nostrofirewall dovrebbe ammettere il traffico TCP entrante scaturito da una rischiesta di connessione di unhost interno verso la rete pubblica ma bloccare le richieste di (effettiva prima) connessione provenienti dall'esterno.

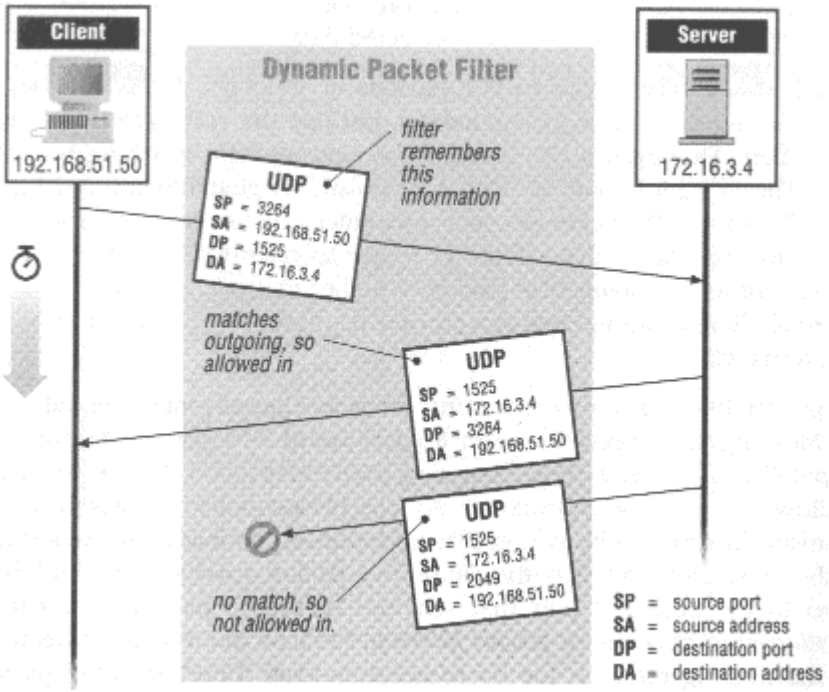
Tutto ciò e' possibile sfruttando le caratteristiche del protocollo TCP durante la connessione
Viene anche detto Standard SYN filtering. La fase connection-establishment del TCP e' la cosiddetta three-way handshake: SYN, SYN+ACK, ACK.

Dunque possiamo creare delle rules per ammettere tutto il traffico TCP entrante, tranne pacchetti che cercano di iniziare la sequenza dithree-way handshake cioe' SYN settato ma NON ACK.
La prima connessione NON ha il flag ACK settato mentre tutte gli altri pacchetti si.
In questo modo anche se utilizzassimo solo questa politica sullo screening del traffico TCP entrante i pacchetti costruiti da qualsivoglia malintenzionato sarebbero scartati dall'host di destinazione(in quanto non preceduti da un SYN ne' aspettati).

.

Tutto ciò non è possibile implementarlo per il trasporto UDP in quanto essendo un protocolloconnectionless e unreliable è privo di questo meccanismo delthree-way handshake ed è dunque insicuro.

Alcuni firewall permettono le cosiddette dynamic-filtering rules che prevedono in seguito ad una richiesta interna di connession UDP verso l'esterno l'aggiornamento dinamico dellerule in modo da consentire di ricevere la risposta. Questo viene effettuato tenendo traccia dei dati sorgenti (source port & address) i quali verranno utilizzati per aggiornare la tabella in modo da abilitare il transito di pacchetti diretti a quell'IP e a quella porta (tale permesso è chiaramente valido limitatamente nel tempo, una sorta di Timeto Live).



In quanto il protocollo UDP/IP è NON sicuro, i suoi pacchetti potrebbero non arrivare affatto o arrivare multipli(ad esempio su una Ethernet quando un router rigenera il pacchetto credendo che esso sia stato vittima di una collisione); inoltre ogni pacchetto UDP è indipendente e non fanno parte dei cosiddettivirtual circuit come quelli TCP.
Tutti questi motivi e la facilità di spoofing di pacchetti UDP fanno spesso propendere per un blocco delle connessioni UDP entranti.

ICMP

I pacchetti ICMP non hanno source o destination port address mahanno solo un campo che identifica il tipo di messaggio. Inoltre essi NON prevedono ACK bit.
Nel packet filtering sarà proprio il tipo di messaggio che verrà preso in considerazione.
I più "pericolosi" sono ICMP PING e REDIRECT (0 e 8) che possono essere usati maliziosamente per effettuare un *Denial of Service (ICMP* Ping Flood) e un tentativo di cambiare il routing rispettivamente.

TELNET / SSH

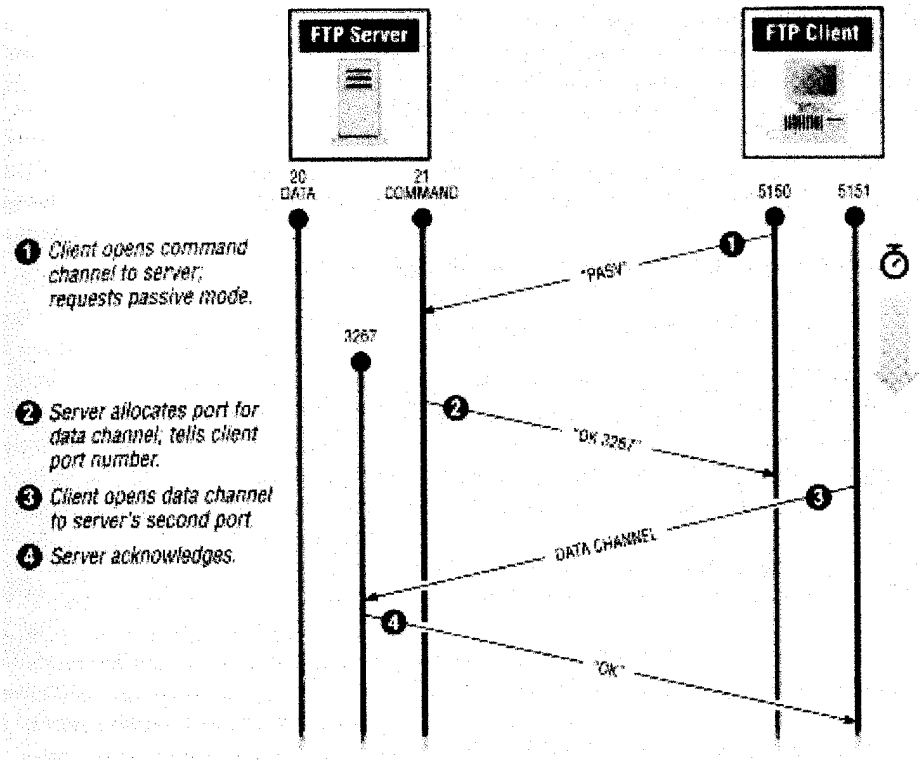
Se volessimo fornire un servizio telnet per utenti esterni dovremmo configurarlo appositamentesull'exterior router e configurare opportunamente il bastion host, unico host in grado di accettare connessioni dall'esterno.
Tale servizio viene comunque ritenuto altamente insicuro in quanto le autenticazioni avvengono tramite trasmissioni in chiaro e dunque potenzialmente a rischio disniffing.
Nel caso di effettiva ed assoluta necessità di poter usufruire di un terminale remoto è preferibile ricorrere alla Secure Shell (ssh) la quale prevede sessioni automaticamente etrasparentemente criptate (è possibile ad esempio usare il solito metodo di autenticazione tramite password o adottare RSA per lo scambio/verifica delle chiavi mentre la connessione è cifrata tramite algoritmi consentiti dal server ma scelti di volta in volta dalclient come DES,TRIPLE DES,Blowfish).
Ad ogni modo la funzione di encrypt viene eseguita prima dell'autenticazione in modo da evitare qualsiasi di trasmissione in chiaro.

Nel nostro scenario è possibile configurare SOCKS service sul proxy del bastion host (ulteriore controllo sulla location, login e password).

FTP

Avendo sconsigliato l' implementazione del Telnet daemon e shell utenti ha senso vietare anche gli account FTP . Ragionamento diverso per gli account anonymous che possono essere abilitati magari con alcuni accorgimenti:

- ✓ numero massimo di connessioni simultanee molto basso;
- ✓ numero massimo di connessioni simultanee per IP rigidamente configurato a l;
- ✓ log accessi;
- ✓ controllo della password (e-mail) specificata durate l'anonymous login : verificare che appartiene quanto meno al dominio del richiedente, e se l'utente ha l'identd attivo (porta 113) verificarne lo userid;
- ✓ idle time durante la connessione molto basso;
- ✓ controllo periodico del virtual file system visibile;
- ✓ valutare la possibilita ' di implementare il server ftp su una macchina sulla perimeter connessa al bastion host;
- ✓ blocco delle richieste di negoziazione porte tramite comando PORT che specifichino un IP diverso dal richiedente (origine di DoS, spesso usato per spedire delle connessioni sulla porta 139 di ignari utenti i quali si vedrebbero "attaccati" dal nostro server FTP!).



Se le nostre macchine hanno IP privati o unrouted dobbiamo sfruttare le potenzialita' di un ftp proxy server il quale tra l'altro ci permette di controllare ed entrare nel merito anche del tipo di traffico (ad esempio vietare la ricezione di file da siti ritenuti compromessi o insicuri, materiale pornografico, pirata, etc).
Invece, con IP regolarmente assegnati e routed possiamo optare per un packet filtering fermo restando l'utilizzo dei client in PASsiVe mode.

Inoltre ogni ftp client deve essere configurato per lavorare nel cosiddetto PASV Mode.

SMTP

Vogliamo canalizzare tutte le connessioni SMTP entranti verso un unico e protetto SMTP server e inibiamo categoricamente la possibilità di utilizzo da parte dei nostri client interni di altri server SMTP ritenuti "sicuri".
Tutto ciò e' realizzabile configurando un server SMTP sul bastion host ed utilizzare dei record Mail eXchange nella configurazione del DNS server (MX record) in modo da dirigere tutto il traffico delle mail entranti sul bastion host il quale farà il relay su un secondo (protetto ed interno) SMTP.
Per quanto riguarda il traffico uscente si predisporranno i client interni opportunamente in modo da garantire che tutte le mail uscenti puntino al bastion host il quale, in base all'indirizzo di destinazione, le inoltrerà verso la rete pubblica o verso il server SMTP privato ed interno.

NNTP

E' verosimile e conveniente usare un NEWS server esterno piuttosto che implementarne uno sul bastion host.
Un servizio in meno in esecuzione e' un possibile punto di attacco in meno; inoltre questi server tendono ad ingrandire a dismisura lo spazio su disco richiesto con conseguenti esigenze di ridimensionamento hardware.
Per il lato client si può ipotizzare un news proxy server disponibile sul bastion host oppure, solo ed esclusivamente se le esigenze dell'ente/società ci obbligano ad avere un proprio news server privato è possibile implementare un NNTP server interno e protetto il quale effettua il relay al servizio NNTP disponibile sul bastion host il quale a sua volta lo inoltra verso l'esterno.
Il servizio e' disponibile sulla well known port 119, i client ed il traffico server<->server coinvolge porte maggiori di 1023.

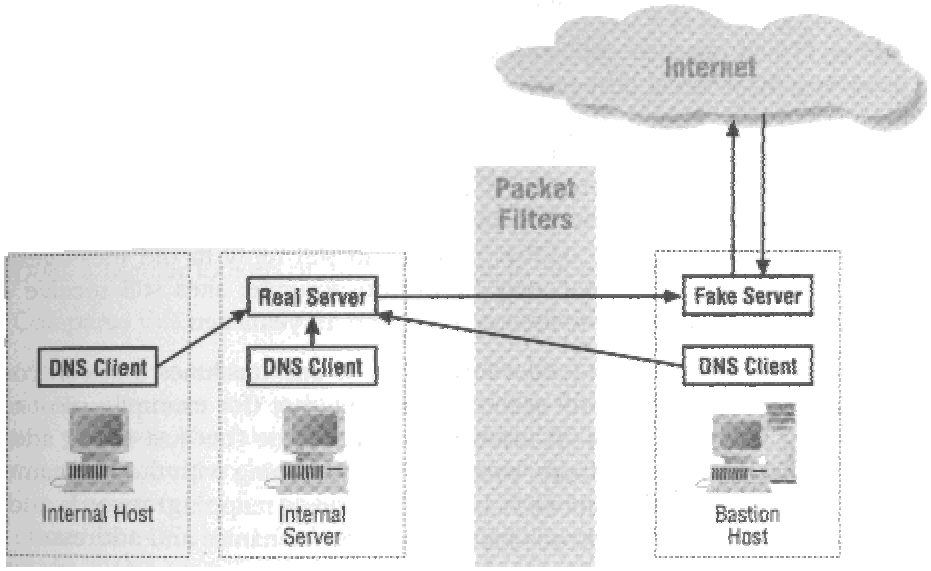
HTTP

Supponiamo che vi siano le seguenti esigenze:
- ✓ web server intranet aperto a tutti magari con zone riservate (ream);
- ✓ web server pubblico aperto a tutti;
- ✓ accesso verso web server esterni dipendente dalla macchina richiedente (IP).

Un web server interno che non abbia alcuna visibilita' dall'esterno e' di semplicissima attuazione:
in base al tipo di servizio offerto (consultazione classica, query database, etc) , alla quantità di traffico e alla mole di dati da trasferire è possibile configurare un web server nella rete privata, con IP privato (o unrouted, RFC 1918) al quale tutte le macchine possono accedervi direttamente specificando IP e porta (se diversa dalla well known port 80).
Ogni tipo di eventuale sicurezza e' demandata al server stesso dato che tutti possono accedervi essendo nella subnet locale e privata.
Se la mole di dati è notevole e si vuole utilizzare al tempo stesso un ulteriore livello di autenticazione e sicurezza e' possibile configurare i client forzandoli ad interpellare il servizio proxy disponibile sul bastion host per qualsiasi richiesta, interna o esterna che sia.
In questo modo, oltre ai ream configurati sul web server possiamo sfiuttare le potenzialità di autenticazione, di oscuramento e di caching proprie del proxy.
Sarà compito di quest'ultimo inoltrare le richieste opportunamente (riconosce che l'indirizzo di destinazione e' locale dunque le inoltra localmente).
Inoltre per sfruttare al meglio questa configurazione sarebbe opportuno configurare il web server privato in modo che possa accettare connessioni solo dal bastion host: chiaramente se il tipo di traffico generato e le esigenze siano tali da giustificare tale configurazione.
Per le richieste di connessioni verso l'esterno (client interni che tentano una connessione verso siti web fuori dall'intranet) e' sufficiente configurare le policy del proxy in modo da consentire l'inoltro dei soli pacchetti provenienti da macchine abilitate verso l'esterno.
Come già menzionato e' possibile entrare nel dettaglio del traffico generato vietando la connessione a siti ritenuti pericolosi, immorali, illegali o censurare alcune estensioni di file (ad esempio .mp3 per *presunta* violazione copyright etc...).
Discorso a parte merita il server visibile dall'esterno.
Innanzitutto, eseguire il server senza i privilegi di root, configurare accuratamente il virtual file system pubblico ed evitare categoricamente di mettere il server all'interno della rete privata.
Inoltre limitare l'accesso anonimo ai soli file desiderati dunque configurare la directory non visualizzabile (questo in modo da evitare che possano vedere e sfogliare parti indesiderate del filesystem: in unix ,chmod o+x-r directory).
Usando il proxy si risolve il problema di web server che non risiedono sulla porta standard ma i veri problemi derivano dai programmi esterni invocati dal client (dato che vengono eseguiti localmente).
Da un punto di vista tecnico e' indispensabile tenere un server talmente accessato e critico su una macchina totalmente dedicata e connessa solo al bastion host.
Inoltre e' auspicabile un'istallazione accurata e distribuita di tutte le applicazioni usualmente invocate dal client http.
L 'unica soluzione seria, proficua ed attuabile e' una formazione continua del personale in modo da dar loro i mezzi per una prima valutazione dei rischi derivante dall'utilizzo di programmi esterni quali documenti postscript (che e' un linguaggio di programmazione a tutti gli effetti con tutti i comandi file oriented) , video file, movie file, etc ...).

DNS

Solitamente si ha l'esigenza di mascherare gli host interni per maggiore sicurezza o per necessita'.
Inoltre se si utilizzano IP privati questa tecnica è utilizzata congiuntamente ad una funzionalità NAT (Network Address Translation) per rendere tali IP raggiungibili.



L 'idea chiave che rende questo approccio funzionale e' quello di avere un dns server e client sulla stessa macchina ma che non si "parlano": infatti il client dirigerà le proprie richieste ad un altro dns server in un'altra macchina.
Occorre istallare un DNS server sul bastion host in modo tale da renderlo visibile al mondo esterno e configurarlo "authoritative" per il nostro dominio.
In effetti quello che questo server sà è quello che noi vogliamo il mondo esterno sappia: nomi e indirizzi dei gateway, i record MX etc. Ci si riferisce solitamente a questo server come DNS public server.
Un altro DNS server ,anche questo authoritative (al contrario del pubblico, questo lo e' veramente), si deve istallare su una macchina interna e configurare con tutti i record necessari al server per funzionare correttamente.
Questo sarà il nostro "vero" DNS server e verrà configurato, tra l'altro, per fare il forward al DNS pubblico di query che non e' in grado di risolvere (per esempio su una macchina unix una fwd line su etc/named.boot).
A questo punto è necessario configurare tutti i DNS client o resolver (per esempio etc/resolv.conf) inclusi quelli sulla macchina che ospita il DNS server pubblico in modo da utilizzare il DNS server interno. E' proprio questa la chiave.
Si avranno le seguenti possibilità:

- ✓ un client interno che chiede di risolvere un host interno contatta il dns privato il quale fornisce la risposta;
- ✓ un client interno che chiede di risolvere un host esterno contatta il dns privato il quale chiede al pubblico il quale a sua volta propaga la richiesta su Internet al DNS server assegnato dal fornitore d'accesso che si occupera' di fornire (o di far pervenire) la risposta e la catena viene ripercorsa in senso opposto fino al DNS client richiedente;
- ✓ un client esterno interroga il dns pubblico e riceve le risposte "mascherate" che noi vogliamo che il mondo esterno conosca.

Bisogna pubblicare false informazioni per ciascuna macchina visibile dall'esterno.
Infatti molti server su internet (ad esempio anonymous FTP server) vogliono conoscere anche il nome della macchine e non solo dell'ip anche se solo per tener traccia delle connessioni.
Nei DNS record, A record (name->address mapping) e PTR record (address->name mapping) sono i record coinvolti nel lookup degli indirizzi e nomi.
Aggiungere per ciascun host visibile i PTR record relativi puo' non essere sufficiente dato che sempre piu' server eseguono il cosiddetto double-reverse lookup e si rifiutano di continuare la connessione in caso di insuccesso.
Ad esempio, in un double-reverse lookup, un resolver :
.esegue un reverse lookup per tradurre un indirizzo IP nel nome host;
.esegue un lookup sul nome host ottenuto per determinare l'indirizzo IP presunto;
.confronta l'indirizzo IP di partenza con il presunto.
Il nostro server pubblico deve dunque fornire dei dati validi per ogni host del dominio visibile dall'esterno.
Per ciascun IP il server pubblico deve contenere un PTR record ad un hostname "fake" e un A record che effettua il bind di questo fake hostname all'indirizzo IP di partenza.
Per esempio, per un indirizzo 195.31.100.30 dovremo creare un PTR con il nome host-195.31.100.30.my.net ed il corrispondente A record che garantisce la risoluzione da host-195.31.100.30.my.net a 195.31.100.30.
Questa tecnica garantisce che in caso di double-reverse lookup i nostri client non avrebbero comunque alcun problema dato che soddisfiamo il doppio controllo pienamente.

Per quanto riguarda il packet filtering configurato sul router interno dobbiamo specificare di permettere UDP-based query e TCP-based DNS transfer zone e risposte dal server interno a quello sul bastion host e viceversa.
Invece, sul router esterno dobbiamo permettere il traffico sia UDP che TCP (well known port 53) dal bastion host (server DNS pubblico) ai DNS esterni alla nostra rete e viceversa.

Analisi

Rivediamo il firewall appena descritto alla luce delle strategie e dei principi discussi precedentemente.

Least priviledge

Lo si puo' vedere ad esempio nella scelta adottata per l'implementazione dell' SMTP .Infatti raccogliere sul bastion host tutte le mail uscenti piuttosto che permettere l'inoltro direttamente verso l'estemo è un'applicazione di tipo least priviledge dato che ci permette un controllo maggiore sul traffico in questione.
In altri termini abbiamo raggiunto lo stesso scopo ma con un controllo maggiore e rischi minori.

Defense in depth

La ridondanza delle regole sui due screening routers sono un esempio di defense in depth.
Infatti l'interior e l'exterior router hanno spesso qualche regola di filtering in comune, il che è una sicurezza in più dato che un determinato pacchetto che deve essere bloccato in ingresso dall'exterior router non avrebbe motivo di arrivare all'interior.

Choke point

Esempio pertinente e' proprio l'architettura da noi adottata: tutto il traffico I/O tra client interni e rete pubblica passa dalla perimeter .
Inoltre l'utilizzo di proxy service sul bastion host accentua ancor di più la nostra propensione a rispettare questa strategia

Weakest link

Un eventuale punto debole della nostra struttura potrebbe essere il bastion host, proprio per sua definizione essendo l'unica macchina raggiungibile dall' esterno e quindi soggetta agli attacchi più disparati.
Comunque in caso di compromissione del bastion host l'eventuale intruso potrà creare più danno ai client che vorranno connettersi verso l'esterno che infastidire realmente risorse interne data la presenza della perimeter e dell'interior router (vedi defense in depth).

Fail-safe stance

Si riscontra facilmente nel packet filtering:

Direction : 110
Source address : Any
Dest. Address : Any
Protocol : Any
Source Port : Any
Dest Port : Any
Ack Set : Any
Action : Deny

Con questa regola per ultima siamo sicuri che se non si ha un "match" con le precedenti (il meccanismo di parsing e' sequenziale) il pacchetto verrà scartato.
Dunque una connessione non prevista ne' espressamente configurata (un nuovo tipo di attacco ?) verrà bloccata.
Sarà compito del security admin controllare periodicamente le log ,analizzarne il contenuto ed eventualemte apportare le modifiche del caso sulle tabelle di filtering.

Conclusione

Un buon hacker puo' sempre trovare un punto debole nella nostra struttura:sara' compito del security admin tenere sempre aggiornate le versioni del software utilizzato e tenersi informato riguardo alle evoluzioni delle tecniche di hacking.
Un ex-hacker è potenzialmente (etica a parte) il miglior security manager.