Sicurezza
La sicurezza delle reti

Le problematiche di sicurezza di una rete sono distinte da quelle legate alla sicurezza dei sistemi informativi o dei singoli calcolatori e richiedono strumenti ad hoc. Ciononostante per garantire la sicurezza globale delle informazioni è necessario che gli strumenti per la garanzia della sicurezza nei calcolatori, nei sistemi operativi e nelle reti siano in grado di interagire in modo sinergico al fine di permettere la più elevata possibile garanzia.

Quando si fa riferimento ad un evento atto a violare la sicurezza delle informazioni trasmesse all'interno della rete, si parla di procedura di attacco.

Gli attacchi alla sicurezza della rete si possono classificare secondo due grandi categorie:

- MINACCE PASSIVE: definite anche intercettazioni che rappresentano i tentativi da parte di terzi di accedere alle informazioni trasmesse durante una comunicazione.
- MINACCE ATTIVE: in cui l'accesso alle informazioni trasmesse da parte di un'entità non autorizzata è seguito dall'alterazione delle informazioni stesse e dalla trasformazione delle stesse in modo da trasmettere informazioni false.

Sulla base di tali considerazioni si può intuitivamente comprendere come l'implementazione di tecniche di protezione e la definizione dei servizi da loro offerti coinvolga in modalità diverse l'architettura di rete.

È dunque necessario, innanzi tutto, comprendere come agiscono i diversi protocolli rispetto alle molteplici problematiche di protezione dell'informazione trasmessa tenendo presente che la *suite* protocollare *TCP/IP*, sulla quale si basa la rete Internet, come noto, risulta essere, universalmente la più diffusa.

Una volta apprese le caratteristiche architetturali, la progettazione di sistemi di sicurezza ad hoc per lo scenario considerato deve prevedere un'analisi dei rischi al fine di poter individuare la tecnica migliore sia dal punto di vista dell'efficienza, sia dal punto di vista strettamente economico.

Inoltre, la conoscenza delle tecniche d'attacco consente all'amministratore di sistema di proteggere i propri sistemi prevenendo gli attacchi, ovvero adottando le misure necessarie a ridurre i fattori di rischio di esposizione.

Una fra le più comuni tecniche di protezione della rete in termini di sicurezza è rappresentata dal *firewall*. Un *firewall* si può definire come un oggetto che consente l'implementazione di una politica di sicurezza. Ovviamente per poter implementare una adeguata politica di sicurezza mediante l'aiuto di un *firewall* è necessario comprendere quali strumenti e quali tecniche vengono comunemente adottati dagli attaccanti per penetrare all'interno delle reti e per meglio comprendere la tecnologia dei *firewall* e le operazioni che svolge è necessario conoscere gli oggetti con cui un *firewall* interagisce.

Particolare importanza in tema di sicurezza è ricoperta dalle tecniche di crittografia e cifratura che costituiscono uno strumento potente ed estremamente importante per garantire la protezione delle informazioni trasmesse all'interno della rete.

In un contesto di crittografia convenzionale l'elemento fondamentale è la chiave, condivisa fra due entità, che consente di cifrare e decifrare le informazioni, ma come si può intuitivamente comprendere la distribuzione e la protezione delle chiavi costituisce a sua volta uno degli elementi di debolezza del sistema e quindi che a sua necessita di una particolare attenzione e cura in termini di sicurezza.

Nella crittografia a chiave pubblica si dispone di una coppia di chiavi, una per la cifratura e l'altra per la decifratura. Una delle due chiavi è di dominio pubblico mentre l'altra è mantenuta segreta da parte del soggetto che ha generato la coppia.

Spesso nella gestione della sicurezza di rete le due alternative vengono combinate per garantire maggiori funzionalità o una maggiore efficienza ad un sistema per la protezione delle informazioni. In particolare la chiave pubblica è utilizzata per la gestione di applicazioni di firma digitale che danno la possibilità di autenticare la sorgente delle informazioni inviate.

La presente sezione di tale modulo formativo ha dunque lo scopo di affrontare le problematiche ora descritte, al fine di garantire una conoscenza esaustiva dei vari aspetti legati alla sicurezza delle reti di telecomunicazioni, fornendo poi ulteriori spunti su alcuni degli aspetti descritti nella relativa sezione presente negli approfondimenti.

La sicurezza telematica

La presenza di molti servizi Internet standard sempre più richiesti ed utilizzati dagli utenti fa intuitivamente comprendere come tenda esponenzialmente a crescere la probabilità che, in alcuni casi, fornire un determinato servizio possa rendere la nostra rete vulnerabile rispetto ad alcune tecniche di attacco che mirano alla violazione o addirittura alla distruzione delle informazioni trasmesse all'interno della rete cui l'utente accede.

In questa unità didattica ci occuperemo dei principali servizi forniti in Internet e cercheremo di comprendere quali sono i loro principali problemi di sicurezza.

Parlando di servizi sicuri, tipicamente ci si riferisce a servizi che forniscono due tipi di garanzie:

- il servizio non può essere utilizzato in nessun modo se non per le operazioni previste.
- non è possibile leggere e/o falsificare le transazioni che avvengo attraverso il servizio.

Tali garanzie non implicano che si possano eseguire transazioni con il servizio continuando ad essere al sicuro. Per esempio, si potrebbe utilizzare un HTTP (*HyperText Transfer Protocol*) sicuro per effettuare il *download* di un *file*, ed essere sicuri che si stia effettivamente effettuando il *download* del *file* a cui si è interessati, e che nessuno lo stia modificando nel transito. Ma non si possono avere garanzie che il *file* non contenga dei virus o programmi dannosi.

È possibile anche utilizzare servizi insicuri in modo sicuro, ma ciò richiede maggiore cautela. Ad esempio, la posta elettronica attraverso il protocollo SMTP (*Simple Mail Transfer Protocol*) è un classico esempio di un servizio insicuro.

Tutte le volte che si valuta la sicurezza di un servizio, bisogna contestualizzare le valutazioni al proprio ambiente e tener conto delle proprie configurazioni; non è interessante la sicurezza in astratto di un servizio.

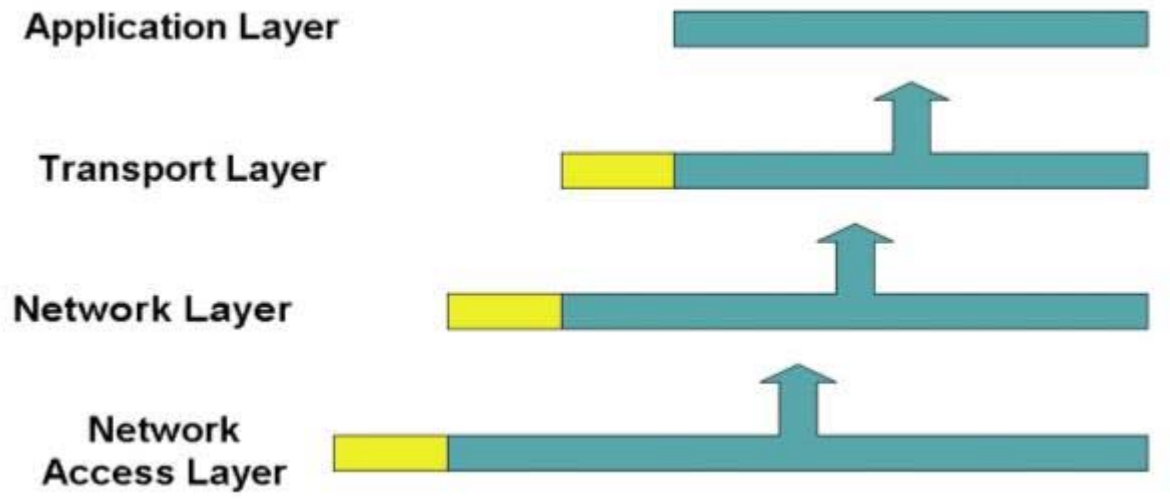
Sicurezza nei protocolli TCP/IP

Per comprendere le tecniche di *packet filtering*, una delle possibili tecniche adottate dai *firewall* per difendere la propria rete, è necessario comprendere come siano composti i pacchetti da ciascuno strato *software* che costituisce l'architettura *TCP/IP*:

- *Application Layer* (FTP, HTTP, eccetera);
- *Transport Layer* (*TCP* o UDP);
- *Internet Layer* (IP).

Ad ogni livello un pacchetto si compone di due parti: l'intestazione (*header*) e i dati (*payload*). L'intestazione contiene informazioni rilevanti per il protocollo, mentre il *payload* contiene i dati.

La costruzione del pacchetto avviene in base al meccanismo che prevede che ciascuno strato aggiunga proprie informazioni di controllo al campo dati ricevuto dalo strato soprastante. Questo procedimento che consiste nel ricevere un pacchetto da un protocollo di livello superiore e nell'aggiungere a tale pacchetto una propria intestazione viene detto incapsulamento.



Il protocollo IP

Il protocollo IP fornisce il servizio di *internetworking* in maniera non connessa e non riscontrata. Il trasporto dei pacchetti IP può avvenire con l'impiego di differenti reti fisiche basate su diverse tecnologie (locali, geografiche).

La maggior parte dei pacchetti IP sono di tipo *unicast* (sono spediti verso un unico *host* di destinazione).IP prevede anche la trasmissione e l'indirizzamento *multicast* (spediti ad un gruppo di *host*) oppure di tipo *broadcast* (indirizzati a tutti gli *host* che possono riceverli nell'ambito della rete logica di appartenenza del mittente).

Lo scopo del *multicasting* è quello di migliorare l'efficienza. Un pacchetto di tipo *multicast* è un singolo oggetto. Se diversi *host* desiderano la stessa informazione, un pacchetto di tipo *multicast* consente di spedire loro diverse informazioni trasmettendo una sola copia del pacchetto, anzichè spedire un pacchetto ciascuno.

Si noti che gli indirizzi di *multicast* e di *broadcast* debbono essere intesi come indirizzi di destinazione, e non come indirizzi di origine. Altrimenti, gli indirizzi di origine di tipo *multicast* e di tipo *broadcast* potrebbero essere utilizzati da un attaccante che sta utilizzando una macchina di destinazione per amplificare l'attacco.

L'attaccante probabilmente non sarebbe in grado di raggiungere un grande numero di *host* senza usare questo genere di scorrettezza. Non c'é interesse ad ottenere informazioni di tipo *broadcast* da altre reti, poiché non sono rilevanti per la propria organizzazione: potrebbero essere altresì potenzialmente dannosi. Un *firewall* quindi deve rifiutare i pacchetti destinati ad un indirizzo di *broadcast* e i pacchetti il cui indirizzo di origine sia un *multicast* o un *broadcast*.

L'intestazione del pacchetto IP include un campo *Options* che solitamente non viene utilizzato. Il campo opzioni IP è stato progettato per utilizzare informazioni speciali o per gestire istruzioni che

non avevano un proprio campo specifico nell'intestazione. In pratica, le opzioni IP sono usate raramente eccezion fatta per i tentativi di attacco.

La più comune opzione IP che un *firewall* è costretto a controllare è l'opzione di *source routing*. Il *source routing* consente al mittente del pacchetto di specificare il percorso che il pacchetto dovrebbe seguire per giungere a destinazione, piuttosto che consentire ad ogni *router* lungo il cammino di usare la propria *routing table* per decidere a quale *router* successivo consegnare il pacchetto. Il *source routing* è stato progettato per sovrascrivere le istruzioni presenti nelle *routing table*. Lo scopo del *source routing* è di aggirare i *router* che possiedono *routing table* guaste o non corrette. In pratica, il *source routing* viene comunemente utilizzato solamente dagli attaccanti che tentano di aggirare le misure di sicurezza costringendo i pacchetti a seguire cammini inaspettati.

Alcuni sistemi di protezione seguono l'approccio di scartare tutti quei pacchetti che hanno le opzioni IP impostate, senza analizzarle; tale approccio solitamente non causa grossi problemi.

Una delle caratteristiche del protocollo IP è la sua capacità di dividere un pacchetto di grandi dimensioni, che altrimenti non potrebbe attraversare una rete (a causa delle limitazioni imposta dalle diverse porzioni di reti fisiche attraversate) in pacchetti più piccoli chiamati frammenti, che possono attraversare la rete. I frammenti vengono quindi riassemblati nell'*host* di destinazione.

Qualunque *router* può decidere di frammentare un pacchetto. Un *flag* nell'intestazione IP può essere utilizzato per evitare che un *router* frammenti un pacchetto. In passato tale *flag* non era molto utilizzato, perchè un *router* che necessità di frammentare un pacchetto ma è impossibilitato a farlo è costretto a scartare il pacchetto, cosa peraltro meno desiderabile della frammentazione stessa. Per apprendere la MTU (*Maximun Transmission Unit*) che può essere utilizzata lungo un cammino viene utilizzato un sistema che fa uso del *flag* suddetto.

La tecnica per l'individuazione della massima MTU è un modo che consente di determinare qual'è il più grande pacchetto che può essere spedito ad una macchina senza subire frammentazione. Pacchetti grandi non frammentati consentono di avere un efficienza maggiore rispetto a pacchetti piccoli. Perciò, la massima efficienza dipende dalla conoscenza di quanto possono essere grandi i pacchetti. Al fine di scoprire tale limite massimo, i sistemi spediscono pacchetti impostando il *flag* che vieta la frammentazione e attendono messaggi di errore. Se si verifica un errore, la macchina riduce la dimensione dei pacchetti, altrimenti la aumenta.

Dal punto di vista della sicurezza il problema che si incontra con la frammentazione sta nel fatto che solo il primo frammento contiene le informazioni relative ai protocolli di più alto livello che i sistemi di *firewalling* devono controllare per decidere se far passare o meno un pacchetto. In principio, un approccio comune era quello di consentire il passaggio a tutti i frammenti facendo il controllo solamente sul primo. Questo approccio era considerato sicuro perché se il *firewall* decideva di scartare il primo frammento, l'*host* di destinazione non poteva essere in grado di riassemblare tutto il contenuto. Se il pacchetto originale non può essere ricostruito, il pacchetto parzialmente riassemblato non può essere accettato.

I problemi con i pacchetti frammentati ancora oggi persistono. Se si consente il passaggio a tutti i pacchetti eccetto il primo, l'*host* di destinazione mantiene tali frammenti in memoria per un certo periodo, in attesa di ricevere il pezzo mancante; questo consente ad un attaccante di usare i pacchetti frammentati in un attacco di tipo DoS. Quando l'*host* di destinazione rinuncia ad assemblare un pacchetto, spedisce un messaggio ICMP di tipo *packet reassembly time expired* in risposta al mittente, tale messaggio informa l'attaccante dell'esistenza dell'*host* e del motivo per cui la connessione non può essere stabilita (presenza del *firewall*).

Inoltre, gli attaccanti possono usare pacchetti frammentati in modo speciale per nascondere delle informazioni. Ogni frammento contiene i riferimenti che indicano dove i dati iniziano e finiscono.

Normalmente, ogni frammento inizia dopo la fine di quello precedente. Comunque un attaccante può costruire pacchetti nelle parti in cui i frammenti si sovrappongono. Questo ovviamente non accade in condizioni normali; può accadere solamente nel caso di errori o di attacchi.

I sistemi operativi differiscono nelle modalità con cui gestiscono i frammenti che si sovrappongono. Poichè tali frammenti non sono normali, molti sistemi operativi li gestiscono male e possono riassemblarli in pacchetti non validi. Tre tecniche di attacco sono possibili grazie ai frammenti che si sovrappongono:

- Semplici attacchi di tipo DoS contro sistemi che gestiscono male i frammenti che si sovrappongono.
- Attacchi di tipo *Information-hiding*. Se un attaccante sa che sono stati installati sistemi che rilevano virus, individuano le intrusioni, o altri sistemi che sono attenti al contenuto dei pacchetti allora può costruire frammenti che nascondono il reale contenuto del pacchetto.
- Attacchi che prelevano informazioni da servizi che non dovrebbero essere accessibili. Un attaccante può costruire un pacchetto con un'intestazione valida nel primo frammento e quindi sovrapporla con il prossimo frammento. Poichè un *firewall* non si aspetta intestazioni nei frammenti successivi al primo, non analizza tali frammenti.

Se non è possibile eseguire il riassemblaggio dei pacchetti nel *firewall* la cosa migliore da fare è scartare tutti i frammenti. Tale approccio potrebbe distruggere connessioni che avrebbero potuto andare a buon fine ma, in ogni caso, tra i due mali questo è certamente il minore.

Il protocollo TCP

Il protocollo *TCP* è il protocollo di strato trasporto più comunemente usato in Internet. Ad esempio, i servizi Telnet, FTP, SMTP, NNTP e HTTP sono tutti servizi basati sul protocollo *TCP*. Il protocollo *TCP* fornisce alle applicazioni connessioni affidabili e bidirezionali tra due *host*.
Il protocollo *TCP* è affidabile nel senso che garantisce lo strato applicativo, cioè:

- La destinazione riceve i dati applicativi nello stesso ordine in cui sono stati spediti.
- La destinazione riceve tutti i dati applicativi.
- La destinazione non riceve dati duplicati.

Il protocollo *TCP* è anche in grado di chiudere una connessione nel caso in cui non riesca a garantire le tre precedenti proprietà. Ad esempio, se vengono persi i pacchetti *TCP* nell'ambito di una sessione, il protocollo *TCP* proverà a ritrasmettere i pacchetti prima di chiudere la connessione definitivamente.

Queste garanzie implicano ritardi sui tempi di *setup* (i due lati di una connessione debbono scambiarsi delle informazioni prima che possano realmente spedire dei dati) ed influenzano le prestazioni (i due lati di una connessione debbono tenere traccia dello stato della connessione).

Il protocollo *TCP* è bidirezionale nel senso che dal momento in cui viene stabilita una connessione tra un *client* ed un *server*, il *server* ha la possibilità di rispondere al *client* sulla stessa connessione.

Per bloccare una connessione *TCP* è sufficiente bloccare il primo pacchetto di tale connessione (quello che contiene la *flag* SYN=1). Senza il primo pacchetto, qualunque altro pacchetto successivo al primo non può essere riassemblato in uno *stream* sul lato ricevente. Qualunque altro pacchetto, successivo al SYN iniziale, indifferentemente dalla direzione in cui viaggia, è contraddistinto dal bit di ACK impostato ad 1.

Il riconoscimento dei pacchetti di apertura della connessione consente il rafforzamento delle politiche di sicurezza, dal momento che, ad esempio, permette ai *client* interni di connettersi ai

server esterni, e può vietare ai *client* esterni di connettersi ai *server* interni.

Le opzioni presenti in un pacchetto *TCP* sono:

- URG (*URGent*).
- ACK (*ACKnowledgement*).
- PSH (*PuSH*).
- RST (*ReSeT*).
- SYN (*SYNchronize*).
- FIN (*FINish*).

I *flag* URG e PSH vengono utilizzati per identificare dati particolarmente critici; PSH comunica al ricevente di interrompere il *buffering* e consegnare i dati allo strato applicativo, mentre URG identifica i dati che il mittente considera genericamente importanti. In pratica entrambi non sono implementati in maniera affidabile, quindi i *firewall* possono tranquillamente trascurarli. Potrebbe essere utile scartare i pacchetti con i bit URG e PSH impostati nei casi in cui fossero indirizzati a delle destinazioni che non li gestiscono.

I *flag* ACK e SYN vengono utilizzati per implementare il protocollo *Three-way Handshake*. Il *flag* SYN è impostato ad 1 nei primi due pacchetti che vengono utilizzati per stabilire una connessione.

I *flag* RST e FIN vengono utilizzati per chiudere le connessioni. Il *flag* RST viene utilizzato per una chiusura brutale, mentre il *flag* FIN viene utilizzato per una chiusura concordata tra i due lati della connessione.

Si deduce quindi che gli unici due *flag* interessanti per un *firewall* sono ACK e RST:

- ACK perché consente di rilevare in maniera affidabile il primo pacchetto della connessione.
- RST perché fornisce un modo utile per chiudere una connessione senza dover spedire messaggi di errore.

Si possono pensare diversi attacchi che coinvolgono l'impostazione ad 1 di alcuni *flag* che normalmente non vengono impostati. Molte implementazioni *TCP/IP* rispondono erroneamente a strane combinazioni di *flag*, bloccando ad esempio la macchina. Altre implementazioni rispondo a tali pacchetti ma non effettuano il *logging* del pacchetto, consentendo agli attaccanti di non essere rilevati, eccetera.

Il protocollo *TCP* garantisce alle applicazioni che riceveranno i dati nell'ordine corretto, ma nulla garantisce al protocollo *TCP* che i pacchetti arriveranno nell'ordine corretto. Al fine di poter ricostruire correttamente i pacchetti ricevuti, il protocollo *TCP* identifica i pacchetti attraverso un numero, chiamato numero di sequenza. All'inizio di una connessione tra due *host*, ciascun *host* seleziona un numero da cui iniziare, tali numeri vengono scambiati attraverso il protocollo *Three-way Handshake*.

Un attaccante, per poter dirottare una connessione, deve indovinare i corretti numeri di sequenza. Poiché tali numeri vengono semplicemente incrementati durante una connessione, è facile per un attaccante prevedere i numeri di sequenza futuri. D'altra parte, tale operazione è molto difficile se non si ha la possibilità di osservare i numeri di sequenza iniziali stabiliti durante l'apertura della connessione; i numeri di sequenza iniziali dovrebbero essere scelti in modo casuale. In alcune implementazioni *TCP/IP* i numeri di sequenza sono predicibili.

Per poter dirottare una connessione predicendo i numeri di sequenza, un attaccante deve:

- avere la possibilità di costruire i pacchetti *TCP/IP*;
- conoscere il numero di sequenza iniziale di una connessione;
- conoscere l'esistenza di una connessione interessante;
- avere informazioni precise sull'istante di tempo in cui una connessione è iniziata;
- avere la possibilità di rispondere in modo che nessuno possa rilevare la sua presenza.

Per anni tale attacco è stato considerato un attacco puramente teorico, che non comporta rischi reali. Attualmente è molto diffuso ed esistono programmi di libero dominio che ne rendono possibile l'attuazione.

Analisi dei rischi

Una volta che sono ben chiare le caratteristiche di gestione della sicurezza nell'architettura protocollare che caratterizza la nostra rete è opportuno, prima di affrontare l'effettiva implementazione di strumenti e servizi per la protezione dell'informazione, effettuare un'analisi dei rischi. In effetti, la sicurezza in ciascun sistema deve essere valutata rispetto ai rischi. Il processo che consente di determinare quali controlli siano appropriati e realizzabili dal punto di vista economico è molto spesso complesso ed a volte soggettivo.

Ci sono diversi approcci per quel che riguarda la fase di analisi dei rischi, ma essi possono essere facilmente ricondotti ad uno dei due seguenti tipi:

- Analisi dei rischi di tipo quantitativo. Questo approccio impiega due elementi fondamentali: la probabilità che si verifichi un evento disastroso e le perdite stimate che possono essere associate a tale evento.
- Analisi dei rischi di tipo qualitativo. Questo approccio è quello più largamente utilizzato. Ci si basa sull'individuazione delle risorse da proteggere, dei possibili attaccanti e delle possibile tecniche di attacco.

COSA PROTEGGERE

I dati

I dati di una organizzazione possiedono tre caratteristiche che necessitano di essere protette:

- Segretezza. Si desidera che altri non possano leggerne il contenuto.
- Integrità. Si desidera che altri non possano modificarli.
- Disponibilità. Si desidera che siano sempre accessibili.

Anche se i propri dati non sono particolarmente segreti, bisogna sempre preoccuparsi delle conseguenze che si verificherebbero in seguito alla loro modifica. In questo caso si genererebbe la perdita di fiducia da parte di utenti e/o dei clienti rispetto alle tecnologie e alle politiche di amministrazione e quindi una perdita di fiducia nell'organizzazione.

Le risorse

Le risorse elaborative di un'organizzazione sono considerate pregiate e, come tali, l'organizzazione deve tutelarle evitando che gli attacchi dall'esterno ne sfruttino le capacità per scopi maliziosi.

La reputazione

Un attaccante che riesce a penetrare all'interno di un sistema si presenta in Internet con l'identità dell'organizzazione che è riuscito ad attaccare.

È noto come sia possibile comporre e spedire messaggi di posta elettronica senza ottenere l'accesso ad un certo *server*, ma è molto più facile farlo dopo essere penetrati all'interno del sistema stesso. I messaggi che provengono dal sito attaccato ed inviati dall'attaccante non sono distinguibili da quelli inviati dalle persone realmente autorizzate.
Attacchi di tale genere riducono la fiducia verso l'organizzazione.

DA COSA PROTEGGERE

Le prerogative di un *hacker* sono:

- evitare l'individuazione e la cattura;
- nascondere la propria identità mediante un *nickname*;
- nascondere la propria collocazione geografica.

Se ottengono l'accesso su un sistema, tentano certamente di conservarlo.

Joyrider

I *joyrider* sono persone annoiate che cercano dei divertimenti. Essi violano i sistemi perché pensano di trovarvi cose interessanti o perché trovato eccitante la possibilità di usare le risorse di altri.

Vandali

I vandali sono coloro che cancellano i dati nei siti. Sono invisi anche alle persone che fanno parte dell'*underground*.

Scorekeeper

A questa categoria appartengono i collezionisti di successi. Non sono interessati solo alla qualità dei sistemi violati, ma anche alla quantità.

Spia

Appartengono a questa classe le persone cha praticano il furto di informazioni, direttamente o indirettamente convertibili in valore economico (informazioni relative a carte di credito, schede/ricariche telefoniche, eccetera).
Le precauzioni che governi e organizzazioni attuano per proteggere informazioni sensibili sono complesse e costose (schermi elettromagnetici, controllo degli accessi ossessivo, eccetera).

Strategie per la sicurezza della rete

L'approccio più semplice possibile per quel che riguarda la sicurezza è quello di considerare quali strategie adottare per ridurre al minimo i rischi.
È bene partire dal presupposto che non esiste un approccio o una strategia che possa risolvere tutti i problemi. Non esiste nulla che possa fornire una protezione perfetta e non esiste neanche una strategia che sia in grado di risolvere tutti i problemi di gestione.

Sicurezza attraverso l'*obscurity*

Un'altra semplice strategia di sicurezza è quella a cui comunemente ci si riferisce con il termine *security through obscurity*. Con questa strategia, un sistema si considera sicuro semplicemente perché si suppone che nessuno sia a conoscenza della sua esistenza. Tuttavia, esistono molti modi per venire a conoscenza dell'esistenza di un *host*.

Ci sono diversi modi per ottenere informazioni sensibili da una macchina. Ad esempio, conoscendo l'*hardware*, il *software* e la versione del sistema operativo di un *host*, è possibile individuare le tecniche da utilizzare per accedervi. In molti casi la versione del proprio sistema operativo viene rivelata al *server* al momento del *login*.

Si inviano informazioni sensibili anche quando ci si connette con macchine esterne alla propria rete. Ad esempio, quando si effettua una connessione ad un *server* HTTP, il *client* comunica la versione del browser e del sistema operativo utilizzati.

A lungo termine, quindi, la scelta della tecnica di *obscurity* non si rivela molto efficace.

Host security

Una strategia molto utilizzata è quella che si basa sulla sicurezza a livello di *host*. Con questa strategia, si rafforza la sicurezza di ciascun *host* separatamente, e ci si sforza di evitare o alleviare tutti i problemi di sicurezza sui singoli *host*. Tale soluzione presenta difficoltà di scalabilità, al crescere del numero degli *host* e al crescere della varietà degli *host* stessi (*hardware* diverso, sistemi operativi diversi, applicazioni diverse, configurazioni eterogenee, eccetera).

La sicurezza a livello di *host* dipende fortemente dalle competenze di chiunque abbia un accesso privilegiato ad ogni macchina.

Una sicurezza a livello di *host* può essere molto appropriata per piccoli siti oppure per siti con elevati requisiti di sicurezza.

Network security

Al crescere della consistenza degli ambienti di elaborazione la soluzione basata sulla sicurezza a livello di *host* diviene sempre meno attuabile e gestibile; per questo motivo molte organizzazioni e reti di *computer* adottano una strategia a livello di rete.

In tal caso si concentra l'attenzione sul controllo degli accessi alla rete ed ai servizi offerti. Gli approcci a livello di rete includono la realizzazione di *firewall* per la protezione delle reti e dei sistemi interni, l'uso di meccanismi di autenticazione forte e l'uso della cifratura per proteggere i dati particolarmente sensibili.

Un sito può ottenere importanti vantaggi utilizzando un approccio basato sulla rete. Infatti, un singolo *firewall* può proteggere molte macchine da attacchi che provengono da reti esterne, senza preoccuparsi del tipo di sicurezza a livello dei singoli *host*.

Least privilege

Molto probabilmente, il principio fondamentale per la sicurezza è quello dei privilegi minimi. Tale principio prevede che utenti, amministratori, programmi, sistemi, dovrebbero possedere solamente i privilegi necessari per eseguire uno specifico *task*. Il principio dei privilegi minimi è un importante principio per limitare l'esposizione agli attacchi e per limitare i danni.

Tutti gli utenti non necessitano in generale di accedere ad ogni servizio Internet. Ogni utente probabilmente non necessita di modificare o leggere ogni *file* in un sistema. Ogni utente probabilmente non necessita di conoscere la *password* di amministratore di una macchina. Ogni amministratore di sistema probabilmente non necessita di conoscere le *password* di amministrazione di tutti i sistemi. Molti sistemi operativi non sono configurati con privilegi minimi, anche per semplificare l'avviamento all'uso della macchina da parte dell'utente.

Ci possono essere due problemi nel momento in cui si decide di applicare il principio dei privilegi minimi. Innanzitutto, può essere difficile applicarlo a causa dell'esistenza di programmi e/o di protocolli non progettati per supportare tale schema. In secondo luogo, si può correre il rischio di impostare in un sistema un numero di privilegi inferiore a quelli minimi.

Il tentativo di applicare il principio dei privilegi minimi sulle persone piuttosto che sui programmi potrebbe rivelarsi controproducente. Si può predire abbastanza facilmente quali permessi siano necessari per un *mail server* (il paradosso a cui si può andare incontro è di trasformare involontariamente i propri utenti in potenziali nemici della propria rete).

Defense in depth

Un secondo principio di sicurezza è basato sulla difesa in profondità. In altri termini ci si affida a diversi meccanismi, anche per motivi di *fault tolerance*.
Si possono prevedere diversi meccanismi che forniscono *backup* e ridondanza:

- meccanismi di *network security* (*firewall*);
- meccanismi di *host security*;
- meccanismi di sicurezza per gli utenti.

Tutti questi meccanismi sono importanti e possono essere molto efficienti, ma è bene utilizzarli in maniera combinata.

Choke point

Un *choke point* obbliga un attaccante ad utilizzare un canale o un accesso obbligato. Per quanto riguarda la sicurezza nelle reti, il *firewall* tra una rete privata ed Internet (assumendo che esista un unico cammino che le interconnetta) costituisce un *choke point*; tutti coloro che desiderano attaccare la rete privata debbono passare per il *firewall*.

Un *choke point* è inutile se ci sono modi per aggirarlo; un attaccante proverà ad entrare dalla via di accesso meno sicura e meno sorvegliata.

Un *choke point* costituisce un meccanismo di sicurezza centralizzato e come tale sarà più agevole l'esercizio, la manutenzione, la configurazione.

Link più debole

Un'assunzione fondamentale per quanto riguarda la sicurezza è che una catena di sicurezza è tanto forte quanto il suo anello più debole. Gli attaccanti più scaltri cercano di individuare il punto più debole in una rete e si concentrano solo ed esclusivamente su di esso.

È necessario che l'amministratore sia a conoscenza dei punti deboli delle proprie difese per poterli eliminare quanto più possibile e per controllare attentamente quelli non eliminabili.

Strategie di configurazione

Un altro fondamentale principio di sicurezza è quello delle configurazioni *fail safe*, cioè quelle configurazioni che continuano ad essere sicure anche a seguito di un errore e/o di un *failure*. Se, infatti, falliscono, dovrebbero lasciare gli accessi completamente bloccati.
Ci sono due approcci principali che si possono seguire rispetto alle politiche ed alle strategie da adottare per quel che riguarda la sicurezza:

- *Default deny*. Si specifica solamente ciò che è consentito e si vieta qualunque altra cosa.
- *Default permit*. Si specifica solamente ciò che è proibito e si abilita qualunque altra cosa.

Dal punto di vista della sicurezza, l'approccio migliore è quella relativo al *default deny*, mentre, probabilmente, dal punto di vista degli utenti l'approccio migliore è quello del *default permit*.

TUTTO CIÒ CHE NON È ESPRESSAMENTE CONSENTITO DEVE ESSERE PROIBITO

Tale approccio ha senso da un punto di vista della sicurezza in quanto è *fail safe*.
Con questo approccio, si proibisce di *default* qualunque cosa; per poter individuare cosa è consentito occorre:

- esaminare i servizi necessari agli utenti;
- considerare le implicazioni relative alla sicurezza con l'erogazione di tali servizi;
- permettere solamente i servizi che si conoscono, che possono essere forniti in maniera sicura e che sono strettamente necessari.

I servizi in questo modo vengono abilitati in maniera controllata.

TUTTO CIÒ CHE NON È ESPRESSAMENTE PROIBITO DEVE ESSERE PERMESSO

In questo caso, possono rilevarsi alcune malfunzioni in alcuni servizi:

- NFS non è permesso attraverso il *firewall*;
- l'accesso al WWW è consentito solamente agli utenti che sono stati adeguatamente istruiti sulle possibilità di attacco che provengono dal *Web*;
- gli utenti non possono installare servizi non autorizzati.

Tale approccio richiede di specificare cosa sia ritenibile pericoloso. Ciò che si considera pericoloso viene vietato, mentre si permette qualunque altra operazioni che non si considera pericolosa. Ipotizzare quali siano tutti i rischi in un sistema o in Internet è un'impresa impossibile. Finché non si hanno notizie del rischio derivante dall'utilizzo di determinati servizi, tali servizi non verranno inseriti nell'elenco.

L'utente spesso reagisce a questo approccio limitativo, cercando nuovi modi per accedere ai servizi chiusi.

Architetture per reti sicure

Non esiste una terminologia completa e consistente per le architetture e componenti di *firewall*. Per quanto riguarda i *firewall* sicuramente si può schematizzare quanto segue:

- *Firewall*: un componente o un insieme di componenti che limitano l'accesso tra una rete protetta ed Internet.
- *Host*: un *computer* connesso ad una rete.
- *Bastion host*: un *computer* che deve essere reso molto sicuro in quanto potrebbe essere oggetto di attacchi.
- *Dual-homed host*: un *computer* che ha almeno due interfacce di rete.
- *Network address translation*: una procedura mediante la quale un *router* modifica i pacchetti che lo attraverso cambiando gli indirizzi di rete in base ad una opportuna politica.
- Pacchetto: l'unità fondamentale di comunicazione in Internet.
- *Packet filtering*: l'azione intrapresa da un dispositivo per controllare in maniera selettiva il flusso dei dati proveniente e/o diretto verso la rete.

- Rete perimetrale: una rete aggiunta (interposta) tra una rete protetta ed una rete esterna (Internet) al fine di fornire un ulteriore livello di sicurezza. Una rete perimetrale viene qualche volta chiamata DMZ, *De-Militarized Zone* (Zona DeMilitarizzata, riferimento alla zona che separa le due Coree).
- *Proxy*: un'applicazione *software* che dialoga con *server* esterni per conto dei *client* interni.
- *Virtual Private Network* o VPN: una rete che trasporta pacchetti, appartenenti ad una rete privata implementata sull'infrastruttura pubblica, che non possono essere decifrati dagli attaccanti.

Firewall

Per meglio comprendere la tecnologia dei *firewall* e le operazioni che svolge è necessario conoscere gli oggetti con cui un *firewall* interagisce: i pacchetti e i protocolli che sono stati utilizzati per assemblare tali pacchetti. In questa unità didattica vengono illustrate le principali problematiche relative alla sicurezza dei protocolli comunemente adottati in Internet.

Un *firewall* è un oggeto che consente l'implementazione di una politica di sicurezza. Per poter definire un'adeguata politica di sicurezza, e per poterla successivamente implementare mediante l'aiuto di un *firewall* è necessario comprendere quali strumenti e quali tecniche vengono comunemente adottati dagli attaccanti per penetrare all'interno delle reti.

La conoscenza delle tecniche di attacco consente all'amministratore di sistema di proteggere i propri sistemi prevenendo gli attacchi, ovvero adottando le misure necessarie a ridurre i fattori di rischio di esposizione.

Un buon amministratore di reti e sistemi deve essere un po' *hacker*.

Personal firewall

Si tratta di un programma progettato per proteggere adeguatamente un *computer* quando questo è collegato ad una rete. Un *personal firewall* analizza i canali di comunicazione, negando l'elaborazione del traffico ritenuto rischioso sia in ingresso sia in uscita. Di seguito si analizzano le caratteristiche di alcuni prodotti molto diffusi e si riassumono le caratteristiche comparate, in una tabella.

Tiny Personal Firewall

Tiny Personal Firewall è un prodotto facile da configurare ed utilizzare e protegge completamente un *computer* dagli attacchi. *Tiny Personal Firewall* include dei *wizard* semplici per il rilevamento delle intrusioni che individuano attività sconosciute e chiedono all'utente di impostare i parametri del *firewall*.

Per proteggere il *computer* da cavalli di Troia o applicazioni non autorizzate, *Tiny Personal Firewall* include degli *application filter*. Appositi *wizard* rilevano i tentativi di connessione alle porte di comunicazione e creano delle regole di *filtering* in base all'indicazioni dell'utente.

Per garantire che dei cavalli di Troia non si nascondano all'interno di applicazioni viene utilizzata la firma digitale con algoritmo MD5.

I *log file* generati possono essere salvati localmente oppure trasmessi ad un *server Syslog*.

Norton Personal Firewall

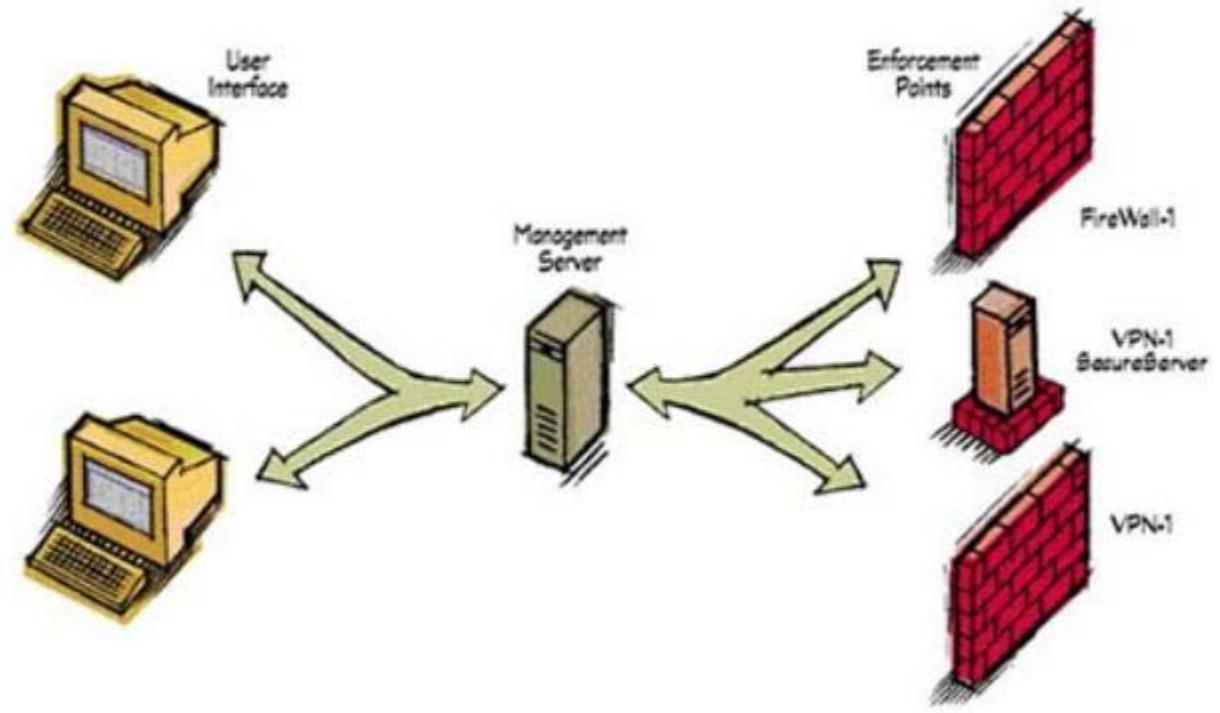
Norton Personal Firewall controlla tutte le connessioni tra il *computer* e la rete. Fornisce dei *tool* e dei *wizard* per la configurazione automatica delle regole di *filtering*.

Zone Alarm

Personal Firewall simile ai precedenti per quel che riguarda protezione e *tool* di configurazione.

Esempio di firewall commerciali

Una soluzione per la sicurezza di un'organizzazione deve essere in grado di dichiarare una politica a livello di organizzazione, distribuirla e ricevere i *log*. Deve inoltre consentire all'organizzazione di controllare l'intera infrastruttura di sicurezza (i *firewall* dell'organizzazione, le reti private virtuali) da un unico punto di amministrazione.



Esistono diversi prodotti che soddisfano i requisiti di sicurezza e che forniscono i *tool* per la protezione delle reti private delle organizzazioni. Si analizzano a titolo di esempio le caratteristiche di due prodotti commerciali molto diffusi: *Cisco PIX* e *Checkpoint FIREWALL* 1

Firewall Cisco

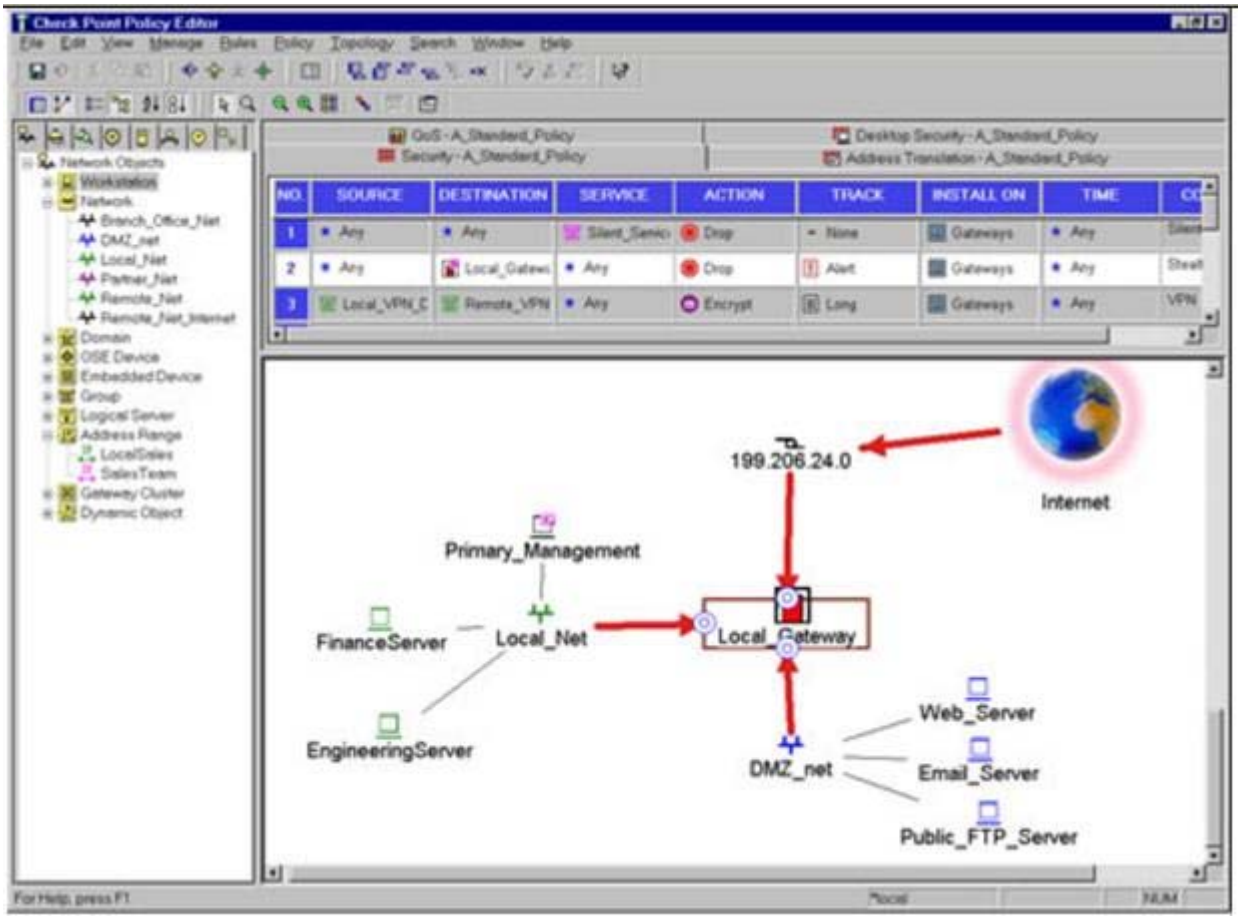
Le principali caratteristiche sono:

- *Context-Based Access Control*: fornisce agli utenti interni un controllo di accesso sicuro per tutto il traffico attraverso il *firewall*.
- Rilevamento delle intrusioni: fornisce il monitoraggio, l'intercettazione e la risposta in tempo reale agli abusi nella rete rilevando un vasto insieme di attacchi comuni.
- *Proxy* di autenticazione: fornisce meccanismi di autenticazione e autorizzazione degli utenti per quel che riguarda le comunicazioni di rete e/o *dial-up*.
- Rilevamento e prevenzione di attacchi di tipo DoS: difende e protegge le risorse del *router* da attacchi comuni.
- Assegnazione dinamica delle porte.
- Blocco degli *applet Java*.

- Supporto per reti VPN, cifratura IPSec e qualità del servizio.
- *Alert* in tempo reale.
- Funzionalità di *auditing* dettagliati: memorizza la data, l'*host* di origine, l'*host* di destinazione, le porte, la durata e il numero totale di *byte* trasmessi.
- *Logging* degli eventi: consente agli amministratori di rilevare in tempo reale, potenziali buchi di sicurezza o altre attività non standard effettuando il *logging* dei messaggi di errore di sistema su un *Syslog server*.
- Funzionalità di gestione del *firewall*: *tool* di configurazione che offre la possibilità di definire passo passo le azioni necessarie per la protezione della rete.
- Strategie di *filtering* del traffico base ed avanzate.
- Ridondanza/*fileover*: dirotta automaticamente il traffico ad un *router* di *backup* nell'eventualità in cui il *firewall* vada in errore.
- Funzionalità NAT.
- Regole per il *filtering* temporizzato.

***Checkpoint Firewall*-1**

Un *firewall Cisco* è un dispositivo *hardware* per la protezione di una rete, *Checkpoint Firewall*-1 è invece un'applicazione *software*. La *console* di *management* di *Checkpoint Firewall*-1 fornisce una singola interfaccia grafica per definire e gestire molti elementi di una rete. Tutte le definizioni degli oggetti sono condivise tra tutte le applicazioni.



Gli amministratori della sicurezza possono selezionare la locazione degli oggetti oppure modificarne le caratteristiche utilizzando l'*editor* visuale per la definizione delle politiche di sicurezza.
Checkpoint Firewall-1 fornisce anche un *editor* visuale per i *log* che consente un'analisi in tempo reale delle informazioni relative al *tracking*, al monitoraggio e all'*accounting* di tutte le connessioni.

Il modulo per la generazione dei *report* permette agli amministratori di trasformare i dettagliati *log* del *firewall* in *report* di gestione che rappresentano le informazioni mediante tabelle e grafici.

I *firewall* proteggono le organizzazioni in Internet fornendo accessi sicuri: garantendo che utenti validi possano accedere alle risorse di rete di cui hanno bisogno. Determinare chi sia un utente valido è compito del sistema di autenticazione; mentre determinare quali risorse un utente possa accedere è compito del sistema di autorizzazione (*Access Control*).

Le regole per l'*Access Control* determinano quali utenti possono accedere alle risorse. Per fornire meccanismi di *Access Control*, un *firewall* richiede una comprensione profonda dei servizi e delle applicazioni utilizzati in rete. *Checkpoint Firewall*-1 fornisce tecnologie di ispezione di tipo *statefull*.

Le funzionalità di *content security* di *Checkpoint Firewall*-1 estendono le ispezioni dei dati fino al livello applicativo per proteggere gli utenti nei confronti di:

- Virus.
- Oggetti *ActiveX*.
- *Applet Java*.

Checkpoint Firewall-1 ha inoltre la possibilità di mascherare la visibilità interna di una rete attraverso meccanismi di NAT.

Checkpoint Firewall-1 cura inoltre tutti gli aspetti relativi alle prestazioni per non introdurre ritardi

penalizzanti nell'instradamento dei pacchetti che sono abilitati a transitare.

PGP - Pretty Good Privacy

Le problematiche legate alla sicurezza della rete coinvolgono molti aspetti che sono, di fatto, diversi tra loro e condizionano in vario modo le strategie attuate per garantire un'implementazione di rete sicura. Si parla, infatti, di sicurezza telematica, sicurezza dei protocolli, sicurezza dei sistemi informativi. In tale scenario, un aspetto di particolare importanza è rappresentato dalla necessità di proteggere le informazioni sensibili durante la trasmissione delle stesse all'interno della rete. Al fine di perseguire quest'obiettivo sono stati progettati strumenti opportuni che si basano su algoritmi di crittografia di consolidata efficienza. In particolare, in questi ultimi anni, il sistema di protezione denominato PGP, *Pretty Good Privacy*, ha conseguito un enorme successo di mercato in virtù delle sue caratteristiche di particolare efficienza e di facile reperibilità a livello mondiale.

È dunque necessario per completezza della trattazione relativa agli aspetti di sicurezza delle reti, descrivere quali siano le principali caratteristiche e le relative potenzialità di tale servizio.

PGP può essere definito come un servizio d'autenticazione e d'amministrazione confidenziale delle informazioni utilizzato per la gestione sia di applicazioni di memorizzazione di *file*, sia della posta elettronica.

In effetti, come rilevato nella precedente sessione si tratta sostanzialmente dello sforzo di una singola persona, *Phil Zimmermann*, il quale, nell'implementazione di questo nuovo servizio di crittografia, ha seguito alcuni fondamentali aspetti, che si sono poi rilevati l'elemento chiave del successo del PGP a livello mondiale.

Gli elementi in questione possono essere brevemente riassunti attraverso il seguente elenco:

1. Selezione dei migliori algoritmi di crittografia esistenti da utilizzare come elementi basilari per la definizione del nuovo servizio.
2. Definire un'efficiente integrazione di tali algoritmi in un'applicazione di tipo *general-purpose* ossia in grado di operare in modo indipendente rispetto dall'*hardware* ed al sistema operativo utilizzato dall'utente.
3. Una volta conclusa l'implementazione del servizio, garantire la libera fruizione dell'intero pacchetto applicativo via Internet, corredato da un contributo di documentazione che sia il più completo ed il più aggiornato possibile.
4. Definire accordi commerciali con le più importanti compagnie al fine di fornire piena compatibilità al prodotto, garantendo anche un suo costo commerciale quanto più possibile ridotto.

In effetti, il perseguimento di tali obiettivi ha consentito al PGP di riscuotere un immediato successo pochi anni dopo la sua nascita con ritmi di crescita impressionati.
Le ragioni che hanno consentito quest'ampia diffusione a livello mondiale possono essere così riassunti:

1. Il PGP è disponibile in modo gratuito e facilmente reperibile in Internet (si rimanda al sito citato nella bibliografia del presente modulo dove viene offerta la possibilità di fruire liberamente di un esaustiva documentazione sul PGP, oltre che della possibilità di scaricare il relativo *software* applicativo) sia per ambiente *Window* sia per ambiente *Unix/linux*.
2. Si basa su algoritmi di crittografia di provata e consolidata affidabilità: in particolare si utilizza RSA per l'operazione di crittografia delle chiavi pubbliche, IDEA (*International Data Encryption Algorithm*) per le operazioni di crittografia convenzionale ed MD5 per le codifiche *hash*.
3. È caratterizzato da un elevato livello di applicabilità in relazione alle più diverse aree di

utlizzo.
4. Non essendo controllato da nessun ente governativo o di standardizzazione lo rende inconsciamente enormemente appetibile.

Dal punto di vista dei servizi il PGP è in grado di fornirne cinque fondamentali:

1. Autenticazione.
2. Trattamento confidenziale.
3. Compressione.
4. Compatibilità col servizio di posta elettronica.
5. Segmentazione.

Tali servizi possono essere spiegati in modo semplice e schematico attraverso la seguente tabella che ne rappresenta le funzioni, i relativi algoritmi utilizzati, descrivendone le principali caratteristiche.

Funzione	**Algoritmo utilizzato**	**Descrizione**
Message Encryption	IDEA, RSA	Il messaggio è criptato utilizzando IDEA con una singola chiave di sessione generata dal trasmettitore. La chiave di sessione è poi sottoposta ad operazione di crittografia attraverso RSA con la chiave pubblica del beneficiario e inclusa nel messaggio.
Firma digitale	RSA, MD5	Utilizzando MD5 si crea un codice *hash* del messaggio. Il messaggio viene dunque sottoposto ad operazione di crittografia con RSA con la chiave privata del trasmettitore ed incluso nel messaggio.
Compressione	ZIP	Il messaggio viene compresso utilizzando ZIP al fine di ottenere una trasmissione ed una memorizzazione più efficiente.
Compatibilità con e-mail	Conversione Radix 64	Al fine di garantire trasparenza in caso di applicazioni *e-mail* il messaggio criptato può essere convertito in formato ASCII utilizzando Radix 64.
Segmentazione		Al fine di garantire il rispetto della massima dimensione ammessa per il messaggio il PGP realizza funzioni di segmentazione e riassemblaggio.