

Sicurezza

Le 3 caratteristiche di un sistema sicuro:

1. Integrità => le modifiche possono essere effettuate solo da persone autorizzate
2. Confidenzialità => i "segreti" sono protetti, solo chi è autorizzato può accedere ai dati
3. Disponibilità => le risorse sono sempre reperibili dalle persone autorizzate

Asset: le risorse che si vogliono proteggere.

Un sistema presenta problemi di sicurezza poichè il codice è scritto dall'uomo che compie circa 10/15 errori ogni 1000 righe di codice e, in genere, per correggere uno di questi errori se ne introducono altri 2. Questi errori vengono chiamati BUGS, e lo sfruttamento di essi per compiere un attacco **Exploiting**.

Le possibili vulnerabilità possono derivare da:

- errori a livello del codice
- politiche di sicurezza errate
- errori di configurazione
- errori di progettazione

per scoprire questi errori si fa uso di reverse engineering.

I tipi di attacchi possono essere di questo tipo:

- intercettazione => sniffing del traffico
- interruzione => interruzione di un servizio (DoS)
- modificazione => modifica dei dati senza autorizzazione
- fabbricazione => creazione di cloni a fini di frode [siti simili (phishing), aggiunta di record nei db]

Un attaccante deve avere 3 requisiti fondamentali:

1. Metodo => conoscenze, abilità, tool e tanto tempo; l'obiettivo in genere è il guadagno
2. Opportunità
3. Motivazione

Rischio: possibilità che un attacco venga effettuato con successo.

Un attacco avviene quando una minaccia si presenta ad una vulnerabilità: eseguo un servizio buggato e del codice sfrutta questo bug per compiere atti illeciti.

Quindi quello che si tenta di fare è o eliminare la vulnerabilità o bloccare la minaccia. La cosa migliore è compiere entrambe le operazioni.

Per ottenere l'effetto voluto:

- prevenendo l'attacco, quindi blocco la minaccia o chiudo la vulnerabilità
- complicando sensibilmente l'attacco (criptazione)
- rendo l'asset vulnerabile meno allettante
- creo un altro asset fittizio più allettante
- individuo l'attacco e cerco di sistemare

Per una politica di sicurezza ottimale si adottano tutte le strategie insieme.

Si può paragonare un asset con una fortezza: essa veniva protetta con fossati, cancelli, trappole.

Allo stesso modo un asset può avere più livelli di sicurezza: firewall, antivirus, politiche di accesso ai file,

Ma **la sicurezza costa**, quindi non sempre è possibile proteggerci su tutti i fronti; per questo si adottano politiche diverse in base al **rischio**, ovvero in base a **quanto e cosa perdo** se l'attacco va a buon fine.

Se non si può adottare una politica di sicurezza completa, si può sempre cercare di rendere difficile l'attacco con:

- **Crittografia:** se usata opportunamente riesce a garantire le tre proprietà fondamentali, anche se vi sono alcuni problemi spiegati in seguito.
- **Controllo del/nel Software:** politiche di accesso integrate nel software stesso (dbms, sistemi operativi), separazione della memoria grazie al sistema operativo, programmi aggiuntivi quali antivirus, intrusion detection utility, password checker, ...
- **Controlli Hardware:** device progettati per la sicurezza come firewall, IDS (intrusion detection system), smartcard, lettori biometrici (impronte digitali, iride, ...), ...

La crittografia è un potentissimo tool contro una grande quantità di minacce. Essa si occupa di individuare algoritmi per l'offuscamento dei dati (problemi matematici non risolvibili in tempo polinomiale ma esponenziale), rendendo difficile, se non impossibile, violare la confidenzialità e l'integrità di un sistema. Tali algoritmi sono però inutili se non supportati da un protocollo.

Il protocollo è una serie di passi da eseguire in ordine e correttamente da tutti i soggetti interessati alla comunicazione.

La cifratura è il processo che preso in input un messaggio in chiaro (PlainText), ne genera uno cifrato detto testo in codice (ChiperText).

L'insieme degli algoritmi utilizzati, l'ordine delle procedure e lo scambio delle chiavi è detto protocollo crittografico.

Le garanzie della crittografia sono:

1. confidenzialità (o segretezza)
2. integrità
3. autenticità
4. non ripudiazione

La controparte della crittografia è la crittoanalisi che si occupa di trovare gli algoritmi per tornare dal messaggio cifrato al messaggio in chiaro.

Due tipi di protocolli crittografici:

1. simmetrici: chiave unica sia per cifrare che per decifrare (chiave privata)
2. asimmetrici: chiave pubblica per cifrare il messaggio e chiave privata per decifrarlo (PGP)

La peculiarità dei sistemi crittografici è che l'insieme dei messaggi sono pubblici, le uniche cose da tenere strettamente confidenziali sono le chiavi.

Da questo deriva il problema di come scambiarsi le chiavi in luoghi geograficamente distanti o problema del **Key Exchange**. Scambiarsi le chiavi via mail non è una buona idea, dato che viaggierebbero in chiaro e si renderebbe più facile il reperimento delle chiavi.

Così nel 1976 **Diffie e Hellman** studiarono un protocollo che chiamarono appunto diffie-hellman. Tale protocollo rende possibile lo scambio di chiavi senza che nessuna delle entità coinvolte ne abbia mai stabilita una. La sicurezza del protocollo si basa sulla difficoltà computazionale del calcolo del **logaritmo discreto**:

Alice genera e comunica pubblicamente a Bob un numero primo **P** molto elevato (1024 bit ~ 300 cifre), un numero random **a** che rimarrà privato ed un generatore **G** più piccolo di P che ha la seguente proprietà:

$$A = g^a \text{ mod } P$$

Allo stesso modo Bob farà:

$$B = g^b \text{ mod } P$$

Alice e Bob si scambiano i due numeri **A** e **B**.

Alice calcolerà:

$$K_a = (B^a) \text{ mod } P$$

mentre Bob

$$K_b = (A^b) \text{ mod } P$$

Poiché $K_a = K_b$ Alice e Bob hanno la chiave per comunicare in segretezza.

Questo algoritmo soffre però del Man In The Middle.

La crittografia conferisce alle nostre comunicazioni in primo luogo confidenzialità. Ma spesso è molto importante garantire integrità (documenti legali); è come se volessimo porre un sigillo al nostro documento e non volessimo che nessuno modifichi il contenuto. Le **funzioni di hash** rientrano nella categoria delle funzioni a una sola via (One-Way Functions) ovvero le funzioni per cui è facile calcolare $y = f(x)$, ma molto difficile o addirittura impossibile calcolare $x = f^{-1}(y)$.

Le peculiarità delle funzioni di hash sono:

1. dimensione dell'output fissata, non dipendente dall'input, chiamato hash value, o message digest o checksum
2. unicità dell'output: a due input diversi corrispondono output diversi
3. dall'output è impossibile tornare all'input

Le funzioni di hash possono servire come **checksum** o come **firma digitale**.

Checksum: si calcola l'hash sul dato e si confronta con l'hash di chi ha rilasciato il dato. Se i due hash coincidono non ci sono state modifiche. Obiettivo: **MDC** (Modification Detection Code)

Firma Digitale: la firma digitale è a tutti gli effetti equivalente a una firma a mano. È un protocollo asimmetrico infatti per firmare digitalmente un documento devo:

1. calcolare l'hash value sul messaggio
2. cifrare tale message digest con la mia chiave privata
3. rendere pubblica la chiave pubblica

Tale firma non garantisce confidenzialità poichè l'unica cosa ad essere cifrata è il message digest. Obiettivo: **MAC** (Message Authentication Code).

Per il riconoscimento di una chiave buona da una cattiva, poichè potrebbe accadere che il possessore di una chiave la smarrisca, esistono delle entità esterne atte a certificare il possessore della chiave. Tali entità vengono dette Certification Authority.

Due algoritmi crittografici: **DES** (Data Encryption Standard) e **RSA** (Rivest Shamir Adelman).

DES:

Il des è stato uno dei primi protocolli sviluppati per il governo americano tra gli anni 1970 ed il 1972. Con il passare degli anni si arrivò nel 1999 al Triple DES e nel 2002 ad AES.

DES è un sistema di cifratura a blocchi a chiave privata.

RSA:

rsa venne creato nel 1978 da **Rivest**, **Shamir** ed **Adelman**. Il problema matematico su cui si basa è la fattorizzazione di grossi numeri interi. Fa uso di una coppia di chiavi, una per cifrare e una per decifrare. A causa della simmetria matematica di tali funzioni le chiavi possono essere usate anche in maniera invertita.

Il sistema operativo offre ai propri utenti dei servizi di sicurezza tra cui:

- controllo degli accessi
- gestione delle identità
- flusso dell'informazione
- audit e protezione dell'integrità

Il sistema operativo, per rendere possibili questi meccanismi di sicurezza, deve conoscere chi sta utilizzando il sistema; l'utente deve quindi autenticarsi. L'autenticazione avviene in realtime, sia claimant (user) che verifier (sisop) sono attivi sullo stesso canale di comunicazione.

Vi è una grossa differenza tra autenticazione e identificazione, in quanto per identificazione si intende il nome utente, per autenticazione la password.

I paradigmi con cui un principal può autenticarsi sono:

1. ciò che sai
2. ciò che hai
3. ciò che sei

Combinando più di questi paradigmi si ha una sicurezza più elevata; per esempio al bancomat si usa il paradigma ciò che sai (PIN) insieme a ciò che hai (TESSERA).

La **password** è una parola, una frase, un insieme di bit conosciuta sia dall'utente che dal sistema. Per lo storage delle password i sistemi adottano metodi differenti: dal banale salvataggio in un file della password in chiaro, dal più complicato nel quale le password vengono cifrate e al file in cui sono contenute vengono applicate opportune politiche di accesso.

Gli attacchi alle password sono un altro dei principali attacchi informatici.

Essi possono essere di diversa natura. L'attaccante può basarsi su un dizionario delle parole maggiormente utilizzate. Usare un attacco di tipo brute force, ovvero provare tutte le possibili combinazioni di numeri/lettere, anche se occorre per questo tipo di attacco parecchio tempo e molta potenza di calcolo (computer in parallelo).

Un attacco molto più subdolo ma molto efficace è quello che prende il nome di Social Engineering, ovvero farsi credere un'altra persona per semplicemente chiedere la password; per questo tipo di operazione serve una estrema bravura e una buona dote recitativa. Altro metodo, spesso sottovalutato, ma in molti casi il più semplice, è leggere la password direttamente da un post-it, o da qualche documento del malcapitato. Infatti spesso gli utenti, per paura di dimenticare la password la scrivono in posti facilmente reperibili per tutti.

I sistemi migliori per proteggerci dal password guessing sono sicuramente quello di scegliere password complicate, in grado di garantirci una certa robustezza agli attacchi.

Un metodo per aggirare il problema degli attacchi alle password è costituito dalle One Time Password, ovvero password valide durante una unica sessione, quindi anche in caso di reperimento della password da parte di un hacker, essa sarà inutilizzabile poiché già scaduta.

Ci sono diversi metodi per la generazione della one-time-password:

- usando un algoritmo matematico per la generazione della nuova password
- basato sulla sincronizzazione tra un server che fornisce la password e il client
- utilizzando un algoritmo che genera una password random
- utilizzando una lista di password

Si possono utilizzare anche sistemi di ChallengeResponse. Questo significa che l'utente deve essere in grado di rispondere a una "domanda" proposta dal server.

Inoltre, un altro sistema per ovviare al problema del brute force potrebbe essere quello di ritardare il meccanismo di autenticazione, in modo che anche per riuscire a indovinare password semplici potrebbero occorrere anni.

Altro problema potrebbe derivare dalla non fidatezza del sistema. C'è la possibilità che il sistema non sia sicuro, sia solo una "copia" della nostra schermata di login che intercetta la nostra password per poi renderla utilizzabile da altri. Il problema è molto più grave su internet dove la copia di siti internet è sempre più diffusa in modo che l'utente distratto rilasci le proprie credenziali al sito sbagliato (phishing).

Subito dopo l'autenticazione dell'utente, l'operazione intrapresa è quella del controllo degli accessi. Questa serve per stabilire a quali risorse l'utente può accedere e fa parte della politica di sicurezza del sistema. Vi sono due tipi di politiche:

- Discrezionali
- Mandatorie

DAC: Discretionary Access Control, permette al proprietario di gestire autonomamente le proprie risorse, di creare quindi da se una politica di sicurezza adeguata. Nei sistemi Unix-Like viene adottata questa tecnica di controllo degli accessi.

MAC: Mandatory Access Control, imposta le restrizioni dall'alto, l'amministratore decide le politiche di sicurezza per tutti gli utenti i quali non possono gestire la politica di sicurezza nemmeno sulle risorse della quali sono proprietari. Questo tipo di politica di sicurezza è in genere adottata nei sistemi militari.

Nei sistemi Windows NT il processo che controlla se un utente ha determinati permessi si chiama **Reference Monitor**, ed è parte del sistema operativo. Il reference monitor deve essere invocato a ogni azione compiuta dall'utente, e il suo avvio non può fallire.

TCB (Trusted Computer Base) è il nome che viene dato a tutti quei componenti hardware e software che si occupano di garantire la sicurezza. Questi componenti devono avere un distacco dal resto del sistema in modo da non poter essere intaccati da eventuali virus. Inoltre è molto importante l'assenza di bug, poichè se si riesce a infettare il TCB, si può infettare tutto il sistema.

Le politiche di sicurezza riguardanti le autorizzazioni possono essere:

Access Control List: utilizzare spesso nei sistemi unix-like. Una ACL è una lista di permessi associati ad un oggetto. La lista specifica chi o che cosa ha l'autorizzazione di accedere all'oggetto e quali operazioni può svolgere su di esso (rwx).

Access Control Matrix: Le ACM danno priorità ai principals; infatti sono tabelle sulle cui righe ci sono tutte le entità, e sulle colonne gli oggetti.

Capability: è una specie di ticket che può generare solo il sistema operativo e che garantisce certi privilegi ad un principals su di un oggetto.

Un hacker malintenzionato ha molto più interesse ad exploitare una parte kernel proprio per le politiche di sicurezza che vengono adottate lato software. Quindi vi è una necessita di proteggere il sistema operativo. Chi compie questo tipo di protezione è l'hardware.

La prima cosa che l'hardware deve risolvere è il Confinement Problem, cioè deve garantire che nessun processo scriva o acceda ad aree di memoria altrui. La sicurezza è garantita dalla **separazione fisica** e dal **bit di stato**. Per separazione fisica si intende che nessuna applicazione può accedere ad aree di memoria del sistema operativo (MMU). I metodi per controllare gli accessi in memoria sono il **Segment Addressing** e il **Bit di Stato** (user e kernel mode).

Malware: è un programma progettato per compiere azioni maligne e indesiderate sul maggior numero di macchine possibili. Anche programmi che ospitano bugs rientrano nella definizione. Vi sono diversi tipi di malware che hanno nomi diversi in base al loro comportamento.

Virus: è un programma che si riproduce iniettando il proprio codice in altri programmi benigni. Il termine deriva proprio dal nome dei virus biologici poichè agiscono nello stesso modo. Un virus è quindi un programma maligno che ha bisogno di un vettore di trasporto per poter compiere il proprio lavoro. Il ciclo di vita di un virus è:

1. Infezione
2. Replicazione
3. Danneggiamento
4. Propagazione

Worm: la caratteristica del worm è l'autoreplicamento. È simile al virus nel comportamento e nella presenza di codice dannoso, ma a differenza del primo, il worm non ha bisogno di un vettore di trasporto. Il compito del worm è quello di infettare il maggior numero di macchine possibile utilizzando anche internet, infatti il mezzo preferito è la mail. In genere il worm utilizza tecniche di social engineering per riuscire a convincere il ricevente ad aprire la mail e infettare così anche la propria rete.

Trojan: ispirati al mitologico cavallo di troia, operano esattamente allo stesso modo. La differenza rispetto a un virus/worm è che il trojan non si autodiffonde, ma serve che un intervento umano attivi il trojan. Non è un compito difficile perchè basta “*nascondere*” il trojan in un altro programma.

RootKit: un rootkit è un software maligno che permette all'attaccante di agire come amministratore all'interno del sistema.

Un malware ha un grosso bisogno di rimanere “*anonimo*” sul computer in modo tale da poter compiere il proprio lavoro. In modo particolare i virus devono stare parecchio attenti, durante la fase infettiva, a non farsi rintracciare. I programmi che hanno il compito di rintracciare malware sono detti **antivirus**. Il buono o cattivo funzionamento è basato principalmente sulla completezza e l'aggiornamento del database contenente le **Signature** o firme, le quali sono delle porzioni di codice che riconoscono univocamente un malware. Il mettere signature troppo corte provoca come effetto collaterale la generazione di troppi falsi positivi. Il procedimento di confronto tra porzione di codice e signature si chiama **pattern matching**.

La prima soluzione adottata per l'offuscamento del codice di un virus fu quella di criptare il codice e di lasciare in chiaro solo la funzione di decriptaggio. Tale soluzione non fu risolutiva poichè bastò impostare come firma la stessa funzione di decriptaggio.

Il problema iniziò con l'avvento dei virus **oligomorfici**, virus in grado di generare diverse varianti della funzione di decriptaggio, che complicarono notevolmente la gestione delle signature, se pur non resero impossibile il lavoro. L'idea che si ebbe successivamente fu quella di decifrare il virus e analizzare il payload. Fu forse proprio da questa idea che si iniziò a pensare alla necessità per i prodotti antivirus di avere un ambiente virtuale nel quale testare il comportamento di un potenziale virus. Tale ambiente virtuale diventò indispensabile con l'avvento dei virus **polimorfici**. Tali virus sono in grado di generare infinite varianti della funzione di decriptaggio rendendo definitivamente impossibile il popolamento del database delle signature. I virus polimorfici vengono fatti girare nell'ambiente virtuale e appena il virus viene decriptato si analizza il payload in modo da riconoscere il comportamento maligno.

L'ultima frontiera dei virus sono i virus **metamorfici**, una famiglia di virus che si propaga in forma di codice sorgente, e sfrutta un compilatore presente sul computer ospite. Prima di compilarci il virus aggiunge e toglie del sorgente da se stesso in modo da generare un virus sempre diverso, non avendo così alcuna parte costante ed a ogni infezione corrisponde un corpo completamente diverso.

Esistono inoltre diversi metodi di infezione.

Il metodo **Fast Infector** che comporta l'infezione massiva di tutto ciò che si può infettare. Addirittura si tenta di utilizzare l'antivirus per propagarsi nei file analizzati.

Sparse Infector è un metodo che comporta una lenta e graduale infezione; il virus infetta solo file di una determinata categoria (range di dimensione, tipo di file, ...) in modo da limitare il più possibile le probabilità di scoperta.

Amored Virus è un metodo di creazione del virus; esso consiste nel rendere difficilissimo il reverse engineering, il tracciamento e il disassemblamento del virus. Spesso riescono a ingannare l'antivirus facendo credere a quest'ultimo di essere su una posizione del disco differente da quella in cui sono realmente.

Mpack è un tool molto utilizzato per la generazione di “cloni”. Esso preso in input un programma ne genera uno dallo stesso comportamento ma da un codice diverso, offuscato, introducendo svariate linee di codice inutili e molto complicate, deviando il flusso dei dati continuamente in modo da rendere molto complicato il procedimento del reverse engineering.

Un qualsiasi computer, acceso, ma offline non rappresenta un potenziale target poichè le uniche persone che possono accedere ad esso devono necessariamente essere davanti al computer. Se però colleghiamo il computer alla rete, il problema dell'accesso cambia. Inoltre per la proprietà di opaqueness della rete, ovvero che è difficile capire se stiamo comunicando con il vicino di casa o con una persona dall'altro capo del mondo, un attacco potenzialmente potrebbe partire da svariati chilometri di distanza. Connetterci alla rete porta infiniti vantaggi, ma anche qualche svantaggio.

Per effettuare un attacco sulla rete, dato che lo scopo è il reperimento di informazioni, l'attaccante si può paragonare ad un rapinatore: si studia la preda, si studiano i punti deboli, le vie di fuga e gli accorgimenti da adottare per non farsi catturare.

L'inizio di un attacco, è il **port scan** del target, con lo scopo di reperire le informazioni sulle porte aperte, quindi sui servizi attivi, informazioni sul sistema operativo e le applicazioni in uso e informazioni sulle versioni del sistema operativo e delle applicazioni.

L'output di un port scan permette all'attaccante di conoscere l'esatto panorama del target, per quanto riguarda i punti accessibili. Esistono diversi tipi di port scanning: il **TCP port scanning**, che compromette l'effetto sorpresa, poichè si viene scoperti instaurando con il target una connessione, il **SYN Stealth scanning**, che ci permette di non essere rintracciati durante lo scan o l'**Idle Scanning**, ovvero si utilizzare la macchina di un'altra persona per fare il port scan. Dopo il port scan avviene il reperimento di informazioni tramite **social engineering**, **dumpster diving** (ricerca nella spazzatura), **fingerprint**, **eavesdrop** (origliare una comunicazione) o **wiretap** (in modalità passiva è simile al eavesdrop, in modalità attiva consente la deviazione e la modifica della comunicazione). Tutto il processo di reperimento di informazioni prende il nome di **Intelligence**.

Dopo aver reperito tutte le informazioni si passa all'attacco vero e proprio. Le principali possibilità sono:

- impersonificare un host
- DoS
- Accesso a informazioni sensibili

L'impersonificazione è una tecnica molto utilizzata nella rete Internet. Per farlo abbiamo diverse possibilità tra cui indovinare identità e autenticazione del target, recuperare identificazione e autenticazione da una precedente sessione (wiretap), aggirare o disabilitare i meccanismi di autenticazione, usare un target che non richiede autenticazione o usarne uno dei quali siamo in possesso dei dati di autenticazione.

Con spoofing si intende la situazione in cui un utente o un programma riesce a mascherarsi come un altro ottenendo così una serie di vantaggi come accessi non autorizzati. Tale tecnica viene divisa in diverse tipologie:

- Il **masquerade** nel quale un host cerca di passare per un altro. Generalmente vengono utilizzate URL fasulle ma molto simili alle originali (www.poste.it con www.posteitaliane.it). Una tecnica molto simile è quella del phishing o dns cache poison per dirottare dei siti.
- Il **session hijacking** è il dirottamento di una connessione. Durante la navigazione la connessione, durante l'ultimo passo, viene dirottata su un sito pirata (acquisto online, la connessione viene dirottata SOLO al momento dell'acquisto reperendo così dati sensibili o direttamente i soldi dell'acquirente).
- Nel **Man In The Middle** la connessione viene dirottata sin dall'inizio facendo passare tutto il traffico da un nodo che si trova appunto nel mezzo della connessione.

Il **Denial Of Service** o negazione del servizio è un tipo di attacco che mira a violare la proprietà della disponibilità. Sovraccaricando di lavoro il servizio con **smurf**, **flood**, **syn** e altri tipi di **request**, il servizio sarà così impegnato a gestire le richieste fittizie che non sarà più in grado di gestire alcuna richiesta.

I meccanismi di difesa contro gli attacchi di rete sono svariati.

Il **firewall**, letteralmente “*muro di fuoco*”, è il principale elemento per la difesa in rete. È un software, installato su sistemi operativi massicciamente ridotti in modo da non fornire all'attaccante alcun strumento di forzatura, che si trova tra la rete esterna, Internet, e la rete interna protetta, LAN. Esistono anche firewall hardware, generalmente installati su apparati di rete (router).

Il compito di questi software è quello di tenere fuori dalla zona protetta tutto ciò che è maligno basandosi su politiche di sicurezza, dettate dall'amministratore, chiamate Access Control Policy. Le tipologie di firewall si differenziano per il livello a cui essi lavorano: ci sono firewall che lavorano a livello IP filtrando i pacchetti per indirizzo, altri che lavorano sulle sessioni TCP, quindi a livello di rete e altri ancora che filtrano direttamente a livello applicativo. Esistono anche firewall ibridi che possono gestire tutti e tre i livelli insieme.

Il **packet filter gateways** è un tipo di firewall che si occupa di controllare gli ip, le porte e il protocollo. È molto efficiente nel bloccare attacchi di tipo DoS o spoofing ma configurare un firewall di questo tipo richiede una grandissima quantità di tempo poichè bisogna mappare tutte le possibili situazioni da bloccare o da permettere; il minimo errore di configurazione potrebbe compromettere l'intero lavoro.

I **circuit gateways** sono ancora più complicati dei packet filter gateways poichè essi controllano tutto il traffico all'interno di un circuito di connessioni TCP, creando una VPN in cui il traffico viene cifrato tra firewall e firewall.

I firewall che lavorano a livello applicativo sono i firewall maggiormente diffusi a livello domestico. Con questa tipologia di firewall si possono bloccare script, macro, codici attivi come ActiveX o java code delle pagine web, proteggendo così i sistemi contro una grossa quantità di malware.

Esistono altri tipi di meccanismi di sicurezza, per esempio gli **IDS** o **intrusion detection system**. Questi sistemi si dividono in Network IDS, se monitorano il traffico di rete, o Host IDS se monitorano le applicazioni.

Un IDS prende in input un file di log, generato dal router nel caso dei Network IDS o dall'applicazione in caso di Host IDS, e cerca di capire se sta avvenendo qualcosa di illecito. Nel caso venga riscontrato un problema l'IDS prende determinate decisioni in base a come è stato impostato come per esempio chiudere le applicazioni, chiudere la connessione o altre operazioni. Il più grosso problema degli IDS è la eterogeneità dei log generati dalle applicazioni.

In conclusione si può dire che la sicurezza informatica è una branca dell'informatica che analizza il rischio, ne valuta l'entità, calcola un costo per la messa in sicurezza basandosi sui diversi meccanismi possibili e opera tutte le opportune operazioni per rendere l'asset più sicuro.

Copyright

Questo opera è derivata da “Sicurezza Informatica” di Matteo Fumagalli (metteus@gmail.com).



Copyright Daniele Monti 2009

Questa opera viene rilasciata sotto licenza [Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/) che permette di modificare e redistribuire il lavoro, mantenendo lo stesso tipo di licenza e citando le fonti. Non è consentito l'uso commerciale.