06_Sicurezza

Cosa vuol dire "sicurezza"?

> Premesse:
>
> - La sicurezza è un processo, che va mantenuto, e non un prodotto.
> - Niente è sicuro al 100%.
> - La sicurezza di un sistema, equivale alla sicurezza del suo componente meno sicuro.
> - Nascondere non funziona mai.
> - La crittografia non basta.
> - Non fare affidamento sull'utente finale.

Gli obiettivi principali della sicurezza sono 3:

1. *Confidenzialità*: Una perdita di confidenzialità è un **accesso** non autorizzato all'informazione.
2. *Integrità*: Una perdita di integrità è una **modifica o eliminazione** non autorizzata dell'informazione.
3. *Disponibilità*: Una perdita di disponibilità è l'**interruzione all'accesso o utilizzo** all'informazione.

Altri 2 concetti sono spesso aggiunti alla base della sicurezza:

4. *Autenticità*: La proprietà di essere verificati e genuini, che un utente sia chi dice di essere.
5. *Responsabilità*: La proprietà che permette di individuare possibili perdite di sicurezza e di tracciare le azioni di una entità.

Examples of Threats

	Availability	Confidentiality	Integrity
Hardware	Equipment is stolen or disabled, thus denying service.		
Software	Programs are deleted, denying access to users.	An unauthorized copy of software is made.	A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task.
Data	Files are deleted, denying access to users.	An unauthorized read of data is performed. An analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication Lines	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated.

Cyber attacchi

- *Passivi*: L'attaccante può **leggere** qualsiasi informazione. Ma non influisce sulle risorse violate.

  Sono difficili da rilevare, poiché non lasciano traccia. Si usa spesso la crittografia. Prevenzione piuttosto che individuazione.

- *Attivi*: L'attaccante, oltre a leggere, può anche **creare, modificare o distruggere** qualsiasi informazione.

  1. *Replay / Riproduzione*: Ottengo un dato e lo utilizzo successivamente per produrre un effetto non autorizzato.
  2. *Masquerade / Mascheramento*: Faccio finta di essere un'altra entità.
  3. *Modifica dei messaggi*: Modifico, ritardo o cancello dei messaggi, per produrre un effetto non autorizzato.
  4. *Negazione di servizio*: Impedisco o inibisco il normale utilizzo delle strutture di comunicazione.

PATTERN DI COMPORTAMENTO

- *Attacchi interni:* Molto difficili da identificare e prevenire, il dipendente ha già accesso alla conoscenza della struttura e ai dati dell'azienda. Potrebbe usare queste possibilità non per i giusti scopi.
- *Organizzazioni criminali:* Sono gruppi organizzati di hackers che fanno pochi errori, agiscono velocemente per evitare di essere rilevati, di solito hanno bersagli ben definiti (es. carte di credito).
- *Minacce persistenti avanzate:* Gruppi organizzati di esperti, che fanno riferimento ad uno stato, mirano alle infrastrutture per prelevare informazioni sensibili, rimangono nascosti all'interno del sistema violato fino al raggiungimento dell'obbiettivo.

Malware

Termine generale per ogni software maligno, concepito per distruggere o sfruttare le risorse di un sistema, spesso mascherato da software legittimo; in alcuni casi si espande da solo attraverso la rete o altri dispositivi.

Terminologia:

Name	Description
Virus	Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.
Logic bomb	A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.
Backdoor (trapdoor)	Any mechanisms that bypasses a normal security check; it may allow unauthorized access to functionality.
Platform independent code	Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Kit (virus generator)	Set of tools for generating new viruses automatically.
Spammer programs	Used to send large volumes of unwanted e-mail.
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.
Spyware	Software that collects information from a computer and transmits it to another system.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

ESEMPI DI ATTACCHI:

Buffer Overflow

Questo è un tipo di attacco molto comune e conosciuto, di cui sono note alcune tattiche di prevenzione. **Causato da un errore di programmazione**, permette di salvare più dati della capacità di un determinato buffer, sovrascrivendo locazioni di memoria adiacenti. **L'attaccante** deve identificare la vulnerabilità all'interno di un determinato programma.

Scrivendo un numero di dati maggiore a quello predisposto nel buffer, si vanno a intaccare le zone di memoria contenenti i dati della funzione, alterando anche gli indirizzi della return; manipolando così la stack delle chiamate a funzione.

Questa è una vulnerabilità soprattutto dei linguaggi a basso livello.

Le difese contro i Buffer Overflow possono essere divise in due categorie:

- *Difese a tempo di compilazione*:
    - Canary: Viene posto una valore di test (canary) tra il return e le variabili locali, viene poi controllato se questo valore è stato modificato.
- *Meccanismi di protezione della stack:*

Denial of Service (Dos)

Un'azione che impedisce o compromette l'uso autorizzato di networks, sistemi e/o applicazioni tramite l'esaurimento di risorse come le CPU, memoria, larghezza di banda e spazio disco.

> SYN Spoofing Attack: L'attaccante lascia aperte connessioni TCP non rispettando l'handshaking a 3 passi, causando congestioni e blocchi per altri utenti.

Distribuited Denial of Service (DDos): Si forma una botnet tramite zombies per creare un maggiore volume di traffico per attaccare (molti sistemi partecipano all'attacco).

Come difendersi dagli attacchi Dos? Prevenzione, rilevazione e filtro, tracciabilità della fonte di attacco.

Tecniche generali di protezione

Strumenti crittografici

Gli algoritmi crittografici sono elementi importanti nella sicurezza dei servizi.

> I numeri casuali non sono facili da ottenere, e usano sorgenti non deterministiche (radiazioni, gas, capacitori), spesso quindi gli algoritmi usano numeri pseudo-randomici.

- *Symmetric Encryption*: La chiave di crittazione è la stessa di decriptazione: $messagio \rightarrow cifratura\ con\ x \rightarrow messaggio\ criptato \rightarrow decifratura\ con\ x \rightarrow messaggio$, per attaccarlo si usa la criptoanalisi per decifrare la chiave o la forza bruta.
- *Schemi di Crittografia*: (Dimensioni dei messaggi e delle chiavi): DES (è lo schema più usato, non molto sicuro), Triple-DES (ripete tre volte lo schema DES), AES (sviluppato per essere l'evoluzione di DES).
- *Public Key Encryption*: La chiave di crittazione è pubblica e la possono conoscere tutti gli utenti, ma la chiave di decriptazione è privata, questo evita di dover condividere la stessa chiave cercando canali di comunicazione sicuri (necessario per la symmetric encryption).

Autenticazione

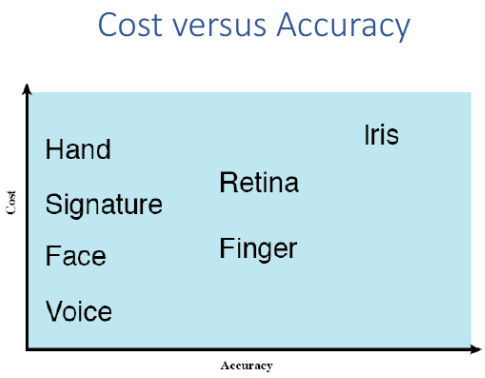
Un sistema molto usato e molto conosciuto per proteggersi dagli intrusi è il sistema delle password.

Il **salt** è una sequenza di bit, spesso un hash, e viene utilizzato insieme alla password. Impedisce l'esistenza di password duplicate, aumenta notevolmente la difficoltà degli attacchi, diventa difficile capire se un utente ha usato la stessa password in sistemi di accesso diversi.

Per il sistema **UNIX** esistono due minacce principali: La possibilità di utilizzare un account guest per poi inserirsi nel sistema. La possibilità di usare un password cracker per scorrere milioni si possibili password in tempo ragionevole.

Autenticazione basata su Token: Memory cards, che conservano dati che non possono essere processati e contengono solo dei codici; Smart cards, che invece hanno al loro interno un processore.

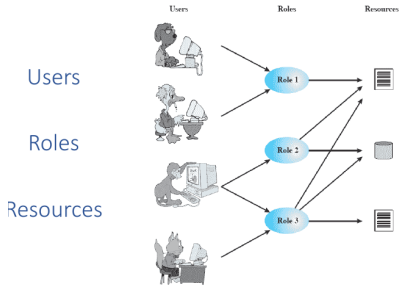
Autenticazione biometrica: Si cerca di autenticare una persona attraverso le proprie uniche caratteristiche fisiche.



Access Control

Si occupa di quali tipi di accesso sono permessi, sotto quali circostanze e per chi. Sono generalmente divisi in tre categorie:

- DAC (discretionary access control): controllo basato sull'identità del richiedente e sulle regole di accesso che definiscono cosa può e non può fare.
- MAC (Mandatory access control): controllo basato sulla comparazione di etichette di sicurezza e rispettive autorizzazioni.
- RBAC (Role-based access control): controllo basato sui ruoli degli utenti e gli accessi che sono possibili con questi ruoli.



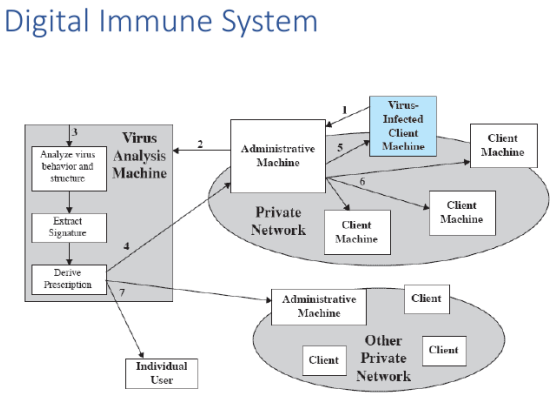
Antivirus

La soluzione ideale contro i virus è la prevenzione, evitare che entrino nel sistema, se questo non è possibile allora:

La **Generic Decryption (GD)** permette di rilevare facilmente anche complessi virus, poiché quando il programma infetto è eseguito, il virus deve auto - decriptarsi per attivarsi. A questo punto si attiva lo scanner GD che contiene: *un controllo di emulazione* (per controllare l'esecuzione del codice in analisi), *scanner delle firme antivirali* (per analizzare l'esecuzione del codice), *emulatore della CPS* (per non intaccare il vero computer).

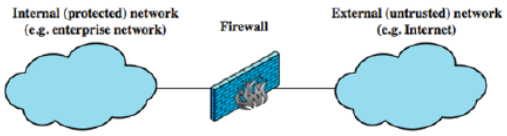
Digital Immune System

Un approccio completo alla protezione virus sviluppato da IBM e perfezionato da Symantec, per contrastare la sempre più elevata presenza di virus. L'obiettivo è quello di eliminare i virus appena vengono introdotti nel sistema. Appena un client viene infettato, si notificano tutte le altre macchine amministratrici per diffondere l'avviso.



Firewall

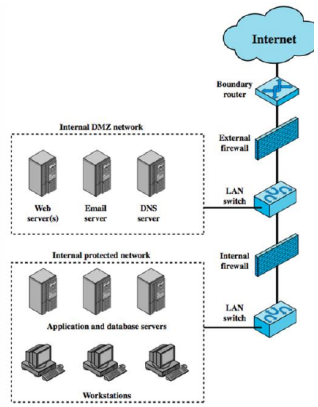
Un sistema di difesa perimetrale di una rete, che funziona come un filtro per proteggere un singolo computer o una rete di computer da possibili minacce in un ambiente esterno che è tipicamente sconosciuto e non attendibile e non sicuro.

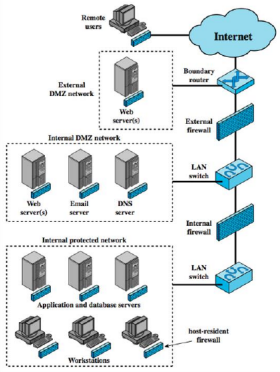


Tipi di firewalls:

- *Packet filter firewall:* Analizza ogni singolo pacchetto che lo attraversa singolarmente senza tenere conto dei pacchetti che lo hanno preceduto, controlla solo l'header del pacchetto e quindi i primi livelli dell'OSI. Il tipo di filtraggio è semplice e leggero ma non garantisce un elevata sicurezza.
- *Stateful Inspection Firewall:* Controlla l'header ma mantiene informazioni anche sulle connessioni TCP, blocco così le connessioni non attive. Mi permette di evitare attacchi di ip spoofing, usando però regole più complesse. Anche questo modello non protegge i livelli superiori dell'OSI, permettendo attacchi di tipo Dos.

Posso avere due tipi di collocamento dei firewall:





Rilevazione degli intrusi (IDS)

La rilevazione degli intrusi si basa sull'assunzione che il comportamento dell'intruso differisce da quello di un utente legittimo, e analizza quindi il comportamento degli utenti nel tempo, se la rilevazione avviene abbastanza in fretta si possono evitare possibili danni e compromissioni, un sistema di questo tipo funziona anche da deterrente.

Può essere **nativo**, quindi senza bisogno di software addizionale, ma potrebbero mancare delle informazioni utili all'identificazione o **specifico**, grazie a un venditore esterno, ma si crea un grosso overhead.

- Rilevamenti anomali: **usato per Dos, scanning e worms**
    - Threshold detection: Controlla un eccessivo verificarsi di eventi nel tempo.
    - Profile based: Caratterizza il passato comportamento di utenti o gruppi e identifica deviazioni significative.
- Rilevamento firma: **usato per app, tran, net layers e violazioni impreviste**
    - Definisce un insieme di regole o di modelli di attacco che possono essere utilizzati per decidere se un dato
      comportamento è quello di un intruso, grazie anche all'aiuto dell'intelligenza artificiale.

> Gli honeypots sono sistemi esca che fungono da diversivo, riempiti di info fasulle, pieni di monitors e log di eventi, possono emulare network interi.

Qualità del software VS Sicurezza

La sicurezza è connessa alla qualità del software, infatti gli attacchi mirano a parti difettose di un codice, magari inserendo input malevoli.

La **Difensive Programming** è una forma difensiva che richiede attenzione in ogni aspetto della progettazione del software, controllando tutte le possibili vulnerabilità.

Bisogna stare attenti soprattutto ai possibili input che può ricevere un programma, per evitare "attacchi ad iniezione".

Per la sicurezza dell'OS è importante la pianificazione dell'istallazione, configurazione, aggiornamento e mantenimento dell'OS stesso e delle sue applicazioni principali.