**SICUREZZA DEI SISTEMI INFORMATIVI**

La sicurezza nei sistemi informative è l’insieme di misure che garantiscono che un utente autorizzato abbia l’accesso a tutti e soli I servizi per cui ha i permessi.   
In una definizione più ampia, è l’insieme degli accorgimenti atti a proteggere i requisiti, detti proprietà di sicurezza, del sistema. Queste proprietà sono:

* *Integrità*: il sistema deve impedire l’alterazione diretta o indiretta delle informazioni
* *Autenticità*: chi riceve le informazioni deve poter verificare l’identità del mittente e l’integrità delle informazioni
* *Riservatezza*: nessun utente deve avere accesso a informazioni per cui non ha l’autorizzazione
* *Disponibilità*: ogni utente deve poter ricevere le informazioni per cui ha le autorizzazioni nei tempi previsti

**Problemi di sicurezza**

Un SI può essere sottoposto a diversi tipi di minacce che rischiano di comprometterne la sicurezza. Le minacce possono essere:

* *Fisiche*: danni a impianti e infrastrutture. Possono essere causate da persone, da eventi accidentali, da disastri naturali, …
* *Logiche*: sottraggono o alterano informazioni
* *Accidentali*: errori nei software e nella rete, errori nei dati, …

Una minaccia logica che “vada a buon fine” costituisce una violazione. Le violazioni vengono messe a segno sfruttando le vulnerabilità del sistema. Una vulnerabilità è costituita dall’esistenza di uno o più punti deboli (exploit).

Un exploit è una tecnica che, facendo leva su una vulnerabilità, consente di forzare un sistema o una parte di esso ad avere un comportamento differente da quello per cui è stato progettato.

Qualunque azione atta a danneggiare o sottrarre risorse a un’organizzazione è detta attacco.   
Gli attacchi a livello di rete sono:

* *Sniffing*: intercettazione di messaggi
* *Spoofing*: invio di pacchetti con indirizzo sorgente “falso” nascondendo la propria identità o “rubandola” a qualcuno
* *Hijacking*: deviazione delle comunicazioni verso un agente malevolo che intercetta e modifica il traffico dei dati
* *Flooding*: intasamento della rete che porta a blocco del servizio (Denial of Service - DoS)

Gli attacchi a livello applicativo sono:

* *Malware*: software malevoli che causano danni. Possono essere: a) *virus*, un frammento di software che “infetta” altri file e si riproduce; b) *worm*, un software che si replica automaticamente, senza bisogno di infettare altri file; c) *Trojan*: programmi apparentemente utili che contengono codice malevolo nascosto; d) *rasomware*, software che limita l’accesso ad alcune funzioni del dispositivo infettato fino al pagamento di un riscatto;
* *Backdoor*: una “porta” che permette di accedere al sistema aggirando le procedure di sicurezza; possono essere create intenzionalmente per la manutenzione ma sfruttate da utenti esterni per manomettere il sistema;
* *Spyware*: software che raccoglie informazioni del sistema su cui sono installati e le trasmettono a qualcun altro.

Attacchi a DBMS: SQL Injection

Il SQL Injection è in tipo di attacco a un DBMS che “forza” l’autenticazione anche quando l’utente non è in possesso di una password.

Questo avviene perché la query usata per l’autenticazione va a buon fine (quindi l’utente è autenticato) solo se restituisce una stringa. Ma è sufficiente aggiungere alla query una tautologia in OR alla richiesta della password perché essa restituisca una stringa, anche senza che la password sia immessa.

**Soluzioni ai problemi di sicurezza: la crittografia**

Una delle tecniche per contrastare gli attacchi è la crittografia. Esse consiste nel codificare l’informazione, trasformando ogni messaggio in un messaggio cifrato, ed è impiegata per garantire le proprietà di riservatezza e autenticità. La crittografia si compone di un algoritmo (funzione crittografica) che codifica il messaggio e da una o più chiavi, che permettono poi di decifrare il messaggio.

Gli algoritmi di crottografia possono essere:

* *algoritmi a chiave simmetrica (o segreta)* : hanno una sola chiave, detta segreta o simmetrica, che deve essere condivisa in modo sicuro. Appartengono a questa classe gli algoritmi a sostituzione e a trasposizione: il primo consiste nel sostituire ogni lettera con altre lettere o altri caratteri e la chiave indica come eseguire la sostituzione, il secondo consiste nel permutare le lettere del testo originale e la chiave indica come eseguire la permutazione.
* *algoritmi a chiave asimmetrica (o pubblica)* : si basano su problemi matematici complesse; ogni soggetto autorizzato possiede una coppia di chiavi (una privata e una pubblica). Tutto ciò che è cifrato con una delle due chiavi può essere decrittato solo con l’altra. Non deve essere possibile, data una chiave, risalire all’altra.

RSA

Come abbiamo detto, gli algoritmi a chiave simmetrica si basano su problemi matematicamente complessi: questo significa scegliere una funzione che sia semplice da realizzare (usata per criptare il testo), ma la cui inversa (usata per decrittare il testo) sia molto complicata. Un esempio di algoritmo a chiave asimmetrica noto è l’RSA. Esso consiste nel:

* scegliere due numeri primi “grandi” P e Q;
* Calcolare N = P \* Q e Z = (Q-1)\*(P-1);
* Scegliere un valore E tale che E sia minore di N e coprimo con Z e un valore D, tale che (E\*D) – 1 sia divisibile per Z.
* Definire la chiave pubblica come (E, N) e quella privata come (D, N).

Eseguire le operazioni dell’algoritmo in modo diretto è semplice, ma eseguire l’inversa senza conoscere le chiavi è molto difficile.

**Soluzioni ai problemi di sicurezza: la funzione di hash**

La funzione di hash è una tecnica impiegata per garantire la proprietà di integrità. Si tratta di una funzione che trasforma un messaggio di una lunghezza arbitraria in un output di lunghezza fissa, detto impronta digitale del messaggio. La funzione deve garantire:

* Coerenza: a messaggi uguali delle associare lo stesso ouput
* Univocità: a messaggi diversi devono corrispondere output diversi
* Non invertibilità: non devo poter risalire al messaggio originale dall’output.

Questa tecnica serve per garantire che il messaggio risulti inalterato dal mittente al destinatario: le l’impronta digitale del messaggio prima di essere inviato coincide con quella del messaggio dopo essere stato ricevuto, allora il messaggio arrivato è integro, altrimenti è stato alterato.

**Soluzioni ai problemi di sicurezza: firma digitale**

La firma digitale garantisce autenticità.

È definita come impronta digitale crittografata del mittente di un messaggio. Al momento dell’invio viene calcolata attraverso la funzione di hash, crittografata con una chiave privata (algoritmo asimmetrico) e allegata al messaggio. All’arrivo viene decriptata con chiave pubblica e confrontata con l’impronta del messaggio arrivato: se coincidono il messaggio è autentico.

**Gestione delle chiavi**

Le chiavi devono essere generate in modo casuale. Migliore è il generatore di chiavi casuali maggiore è il livello di sicurezza.

Devono essere inoltre conservate: possono essere memorizzate in un file criptato protetto da password oppure su dispositivi hardware rimovibili.

Le chiavi poi devono essere scambiate fra agenti autorizzate. Solitamente si usa un canale diverso da quello su cui transitano i dati. Si può anche inviare una chiave privata criptandola con la chiave pubblica del destinatario.

Le chiavi pubbliche non devono essere criptate, ma bisogna assicurarsi che il possessore della chiave sia chi dice di essere. Per questo si usa il Public Key Certificate (PKC), che è una particolare struttura dati.

I sistemi che generano chiavi pubbliche sono detti PKI. Essi possono generare, revocare e distribuire PKC.

Il PKI può anche validare una firma digitale, memorizzando informazioni riguardo ad essa e a chi l’ha apposta.

Un PKI funziona così: un utente chiede un certificato a un’entità di certificazione, la quale ha una propria firma digitale; l’entità gira la richiesta a un’autorità di certificazione, che verifica e garantisce l’identità dell’utente; un’entità di revoca può annullare certificati scaduti. Quando si riceve una chiave pubblica attraverso un certificato, si verifica sempre l’impronta della firma digitale per verificare che sia autentica.

**Controllo degli accessi**

L’autenticazione permette di verificare l’identità di un utente che voglia accedere a determinate informazioni e avere un controllo sulle modalità di accesso.

Per autenticare un utente si può richiedere:

* Something You Know: un’informazione, come una password
* Something You Have: un oggetto, come una chiave magnetica
* Something You Are: una caratteristica fisica, come un’impronta digitale.

Dopo aver verificato con l’autenticazione l’identità di un soggetto, bisogna verificarne i permessi tramite autorizzazione. Questa fase controlla che l’utente sia veramente abilitato a fare ciò che richiede di fare.

Controllo di accesso ai dati

I DBMS devono avere dei meccanismi di sicurezza che proteggano i dati da accessi illegittimi.

I controlli di accesso in un DBMS si basano sulle seguenti regole di accesso:

* *Autorizzatori*: chi crea un dato ne è proprietario e ne diventa autorizzatore, cioè può dare ad altri i permessi per accedere a quel dato;
* *Soggetti*: sono utenti, gruppi, classi, applicazioni o transazioni. Un soggetto può anche essere composto da una tupla di questi attributi (ad esempio per indicare un utente che appartenga a più gruppi o usi più applicazioni).
* *Oggetti*: sono tutti i gli elementi dei BD che vanno protetti, ad esempio tabelle, colonne, schemi, ...
* *Diritti*: sono le azioni che un certo utente ha il permesso di effettuare sul DB.

Un sistema è detto *chiuso* se tutto quello che non è esplicitamente permesso è vietato, *aperto* se tutto quello che non è esplicitamente vietato è permesso.

Un autorizzatore fornisce un permesso come una tupla (*soggetto, oggetto, diritto, vincolo)*, dove il vincolo è un controllo che permette di verificare se per caso l’utente ha qualche legame particolare col dato (ad esempio, un dipendente può accedere a un dato “stipendio” solo se è il suo stipendio). Il controllo sui vincoli viene realizzato tramite una *view*: si verificano i dati dell’utente e si restituiscono i dati in una view “personalizzata” a seconda di “chi è l’utente”.

L’accesso ai dati può essere realizzato secondo due politiche:

* Politiche *discrezionali* (DAC , cioè il proprietario delle risorse può autorizzare chi vuole. Per i DBMS con questa politica si usano ad esempi i comandi SQL di grant \*permesso\* per una certa risorsa con eventuali permessi di propagazione (cioè la possibilità di dare a propria volta permessi per quella risorsa).
* Politiche *mandatorie* (MAC), applicate a dati molto importanti come quelli governativi o militari, in cui dati e utenti sono divisi in gerarchie, in livelli che indicano quanto “sensibile” è una risorsa e quanto “in profondità” un utente può accedere. Il livello di “segretezza” di una risorsa è detto sensitività. In questi sistemi, bisogna garantire che ogni utente possa accedere solo ai dati per cui possiede il “livello” giusto. Un soggetto può accedere in lettura solo ai dati con livello di sicurezza minore o uguale al suo, in scrittura solo a dati con livello di sicurezza maggiore o uguale al suo. Nei DBMS con questa politica si realizzano tabelle multi-livello per dati di “segretezza” diversa. Questo meccanismo di istanziazione di tabelle fisicamente separate a seconda della loro classificazione di sicurezza è detto polinstanziazione. La polinstantazione, al momento di creare un’istanza di una tabella, può assegnare valore “null” agli attributi che hanno sensitività più alta rispetto a quella dell’utente per cui è creata l’istanza (ad esempio un attributo che abbia sensitività *top secret* apparirà con valori nulli in un’istanza di livello *secret*); in alternativa si possono creare righe diverse, con stessa chiave primaria ma livelli di sicurezza diversi, in una sola tabella, e ogni utente visualizzerà solo le righe al livello di sicurezza che li compete.

**Firewall**

Un firewall è un insieme di componenti che controllano e limita il traffico fra una rete da proteggere e l’esterno.

Un firewall si frappone fra una intranet privata e internet. Per funzionare correttamente, deve essere l’unico punto di contatto fra intranet e internet e deve lasciar passare solo pacchetti autorizzati.

I firewall si classificano in due tipologie a seconda di come operano:

* *Packet filtering/inspection*: il firewall decide se instradare o bloccare un pacchetto controllano la sua intestazione (*filtering*) o il suo contenuto (*inspection*). Essi valutano il pacchetto secondo delle regole stabilite e in base ad esse decidono se “lasciarlo passare” o no. Questa tecnica però fa sì che agenti esterni possano modificare la configurazione del firewall con falsi messaggi ICMP: bisogna dunque programmare il firewall affinché non permetta la configurazione remota. Questi firewall operano a livello di rete/trasporto.
* *Application gateway:* forniscono servizi di packet filtering/inspection e in aggiunta ad essi anche servizi di application proxy.L’application proxy è un processo che si pone fra componenti di un’applicazione specifica. Ad esempio, in un’architettura client-server comunica col server simulando il client e viceversa, così i pacchetti invece di viaggiare direttamente fra client e serve sono prima ricevuti e analizzati dal proxy.

Solitamente si utilizza in firewall che riceve il traffico dalla rete esterna e un firewall che filtra il traffico verso la rete interna. La “zona” fra essi è detta DeMilitarized Zone. Questa zona funziona come una sotto-rete isolata posta fra internet e quella interna.

**Intrusion detection system (IDS)**

L’intrusione è un accesso non autorizzato a un sistema o una risorsa volto a compromettere una o più proprietà di sicurezza.

Le componenti funzionali di un IDS sono:

* Un insieme di sensori che raccolgono elementi, monitorando il sistema, che possono tradire le tracce di un’intrusione;
* Algoritmi di analisi, che analizzano i dati raccolti dai sensori e decidono se c’è un attacco in atto oppure che ha già avuto luogo;
* Componenti alerting and response: sono componenti attivi che segnalano un attacco in corso e cercano di bloccarlo.

Gli IDS si classificano in base ai tre componenti sopra elencati.

In base al tipo di sensori, si dividono in:

1. Network based IDS: è un IDS che controlla il traffico di rete per identificare pacchetti malevoli. Questo metodo richiede pochi sensori e non ha grande impatto sulle prestazioni della rete, ma ha lo svantaggio di non riuscire a rilevare se un attacco è già andato a segno né ha la possibilità di analizzare eventuali pacchetti malevoli che siano stati criptati.
2. Host based IDS: è un software che va installato su ogni macchina da proteggere. Può leggere pacchetti criptati e può cercare di fermare un attacco, essendo in grado di identificare quale processo sia vittima dell’attacco e di prendere delle contromisure. Possono tuttavia essere disabilitati da attacchi DoS e rallentano le prestazioni della macchina su cui sono installati.

In base al tipo di analisi, si dividono in:

Misuse detection IDS: analizzano l’attività del sistema e la confrontano con pattern tipici di attacchi conosciuti.

1. Anomaly detection IDS: analizzano l’attività del sistema e rilevano eventuali deviazioni dal suo comportamento normale. Cercano comportamenti anomali negli host o nella rete. Questo viene realizzato tenendo un archivio dei comportamenti considerati normali e a partire da queste informazioni delle “soglie di normalità”, al di fuori delle quali un comportamento viene considerato sospetto.

In base al tipo di reazione, si dividono in:

1. Active response: L’lDS interviene attivamente nel bloccare l’attacco, ad esempio resettando una connessione o riconfigurando un firewall.
2. Passive response: L’IDS avvisa un amministratore del sistema dell’attacco.