

# ESAME - Sicurezza e Affidabilità

Fabio Ferrario

@fefabo

2023/2024

# Indice

<b>1</b>	<b>Crittografia</b>	<b>3</b>
1.0.1	Crittografia Simmetrica . . . . .	3
1.0.2	Crittografia Asimmetrica . . . . .	3
1.0.3	Firme Digitali . . . . .	4
1.0.4	DES e AES . . . . .	4
1.0.5	Domande . . . . .	5
<b>2</b>	<b>Sistemi Operativi</b>	<b>6</b>
2.0.1	DAC . . . . .	6
2.0.2	MAC . . . . .	6
2.0.3	Access Control List . . . . .	6
2.0.4	Principi di Mediazione Completa e Privilegi Minimi . .	7
2.0.5	Bell-LaPadula . . . . .	7
2.0.6	Domande . . . . .	8
<b>3</b>	<b>Sicurezza nelle Applicazioni</b>	<b>9</b>
3.0.1	Attacco SQL-Injection . . . . .	9
3.0.2	Attacco Buffer Overflow . . . . .	9
3.0.3	Attacco XSS . . . . .	10
3.0.4	NULL Pointer Dereference . . . . .	11
3.0.5	Domande Sicurezza Applicazioni . . . . .	11
<b>4</b>	<b>Sicurezza nelle Reti</b>	<b>12</b>
4.0.1	Firewall . . . . .	12
4.0.2	SSL . . . . .	13
4.0.3	IPsec . . . . .	13
4.0.4	Domande Sicurezza Reti . . . . .	15

# Capitolo 1

## Crittografia

### 1.0.1 Crittografia Simmetrica

Nella crittografia simmetrica si utilizza una sola chiave, sia per cifrare che per decifrare un messaggio: Il mittente cifra il messaggio con la chiave  $k$  e lo invia al destinatario, che dovrà decifrare il messaggio con la stessa chiave  $k$

$$\text{decrypt}(\text{encrypt}(\text{message}, k))$$

**Numero di Chiavi** Ogni coppia di utenti ha bisogno di una chiave segreta per potersi scambiare messaggi, quindi il numero di chiavi necessario è  $\frac{N(N-1)}{2}$ , con  $N$  numero di utenti.

### 1.0.2 Crittografia Asimmetrica

Nella crittografia asimmetrica ogni utente genera una coppia di chiavi (legate matematicamente), una pubblica e una privata. La chiave pubblica è accessibile in chiaro a tutti, quella privata invece rimane segreta e conosciuta solo a chi l'ha generata. Queste chiavi sono fatte in modo che se una viene utilizzata per cifrare un messaggio esso potrà essere decifrato solo con l'altra. Quindi se devo inviare un messaggio a un destinatario lo cifrerò con la sua chiave pubblica, di modo che soltanto esso potrà decifrarlo con la sua chiave privata.

**Numero di Chiavi** Nella crittografia asimmetrica è necessaria una coppia di chiavi per ogni utente, quindi  $2N$  chiavi.

### 1.0.3 Firme Digitali

Le Firme Digitali sono utilizzate per verificare la provenienza e l'integrità di un messaggio.

Firmare un messaggio intero è computazionalmente molto costoso, quindi generalmente si effettua prima un digest (MD5) del messaggio e poi si usa RSA per crittografare il digest. Il messaggio viene quindi inviato al destinatario insieme al digest crittato, che verificherà l'identità del mittente decrittando il digest con la chiave pubblica dello stesso.

In questo modo avrà sia conferma del mittente, sia la conferma che il messaggio è integro verificando a sua volta con il digest del messaggio inviato.

Quindi i passaggi sono:

1. Il mittente crea il digest del messaggio da inviare.
2. Cifra poi il digest del messaggio con la sua chiave privata ottenendo così la firma digitale.
3. Infine invia il messaggio insieme alla firma.
4. Il destinatario, ricevuti il messaggio e la firma, decifra la firma con la chiave pubblica del mittente ottenendo così il messaggio e il relativo digest.
5. Per controllare l'autenticità del messaggio genera a sua volta un digest di esso e lo confronta con il digest ricevuto.

### 1.0.4 DES e AES

**DES** il Data Encryption Standard è basato su confusione e diffusione dell'informazione:

- Codifica messaggi in blocchi da 64 bit;
- Applica 16 volte una funzione combinatoria  $F$  ad ogni blocco usando la chiave come uno dei parametri di  $F$ .

DES ha delle problematiche però, perché è stato bucato negli anni 70 e non esiste la prova matematica della sua sicurezza.

È stato quindi sostituito da AES o da TripleDES, che cifra tre volte con 3 chiavi diverse.

**AES** Advanced ES è basato su trasformazioni applicabili a blocchi di 128 bit, utilizza chiavi da 128, 192 o 256 bit (estensibili) e il numero di cicli è fissato tra 10 e 14 (estensibili).

### 1.0.5 Domande

- Si spieghino le differenze tra Crittografia Simmetrica e Asimmetrica
- Si descriva la relazione fra numero di chiavi e utenti per la crittografia simmetrica e asimmetrica
- Si spieghi il funzionamento della crittografia a **chiave pubblica**, indicando in particolare come si può usarla per implementare **firme digitali**
- Si descriva come si possono combinare crittografia a chiave asimmetrica e algoritmi di Message Digest per ottenere meccanismi efficienti di **firma digitale**
- Perché la crittografia DES non é più considerata sicura? Quali modi di suo utilizzo potrebbero migliorare la sicurezza?

# Capitolo 2

## Sistemi Operativi

### 2.0.1 DAC

Nel modello DAC (Discretionary Access Control) il proprietario di una risorsa ne concede l'accesso ad altri utenti a sua discrezione. È quindi il proprietario della risorsa che decide ogni singolo utente che può accedere ad essa.

**esempi** Un esempio è google documenti, in cui il proprietario decide di dare accesso a una sua risorsa a un utente singolarmente.

### 2.0.2 MAC

Nel modello MAC (Mandatory Access Control) il sistema impone un modello che limita e controlla la discrezionalità degli utenti nell'assegnare i diritti di accesso alle risorse.

**esempi** Un esempio è la sicurezza multi livello militare, in cui ad ogni utente e risorsa è assegnato un livello di segretezza.

### 2.0.3 Access Control List

Una Access Control List è una lista dove, data ogni risorsa, possiamo inserire chi ha diritto e che diritti ha (Lettura, scrittura, esecuzione). Una ACL ci permette di risparmiare spazio rispetto alla tabella completa, è facile determinare la lista degli utenti che hanno un diritto specifico su una risorsa condivisa ed è facile revocare/modificare l'accesso di un utente ad una risorsa.

**Funzionamento** Per una risorsa la ACL contiene una lista di utenti o di gruppi e i rispettivi diritti (ownership, read, write, execute).

**ACL in Unix** Unix utilizza un modello semplificato della ACL che utilizza solo 9 bit di protezione (per file). Vengono definiti i permessi soltanto per L'owner, un gruppo definito, e il resto del mondo. Per ognuno di essi abbiamo 3 bit (R,W,X) che definiscono i rispettivi permessi.

Questo tipo di ACL è molto più piccola ma permette una granularità molto inferiore.

## 2.0.4 Principi di Mediazione Completa e Privilegi Minimi

**Mediazione Completa** Le regole di sicurezza vengono applicate ad ogni accesso, quindi ogni tentativo di accesso deve essere controllato.

**Privilegi Minimi** Il principio del privilegio minimo (PoLP) si riferisce a un concetto di sicurezza delle informazioni in base al quale a un utente vengono concessi i livelli - o permessi - minimi di accesso dei quali ha bisogno per svolgere le proprie mansioni. Questo principio può anche essere esteso ad applicazioni, sistemi o dispositivi connessi che richiedano privilegi per eseguire un'attività.

## 2.0.5 Bell-LaPadula

Il modello Bell-LaPadula è un modello MAC multilivello che nasce in ambito militare per garantire la confidenzialità dei dati. In questo tipo di modello gli utenti e le risorse sono classificati secondo dei livelli di sicurezza, in modo che un utente possa accedere ad un dato solo se il suo livello è maggiore o uguale a quello della risorsa.

Bell-LaPadula implementa due proprietà:

- No Read Up: Un soggetto non può leggere oggetti di livello più alto. (Simple Security)
- No Write Down: Un soggetto non può scrivere oggetti di livello più basso (impedendogli di trasferire documenti del suo livello a livelli più bassi). (Confinement)

Bell-LaPadula può essere esteso con **compartimenti**, in cui ad ogni risorsa e utente è assegnato un compartimento, e un utente può accedere ad una risorsa se il suo livello è maggiore o uguale a quello della risorsa e se fanno parte dello stesso compartimento.

## Autenticazione Challenge-Response

Nei protocolli Challenge-Response la Password è una funzione matematica  $F(x)$  che l'utente sa calcolare. Per l'autenticazione il sistema propone un valore  $X$  e l'utente deve rispondere con il valore corretto  $F(X)$ .

**C-R con chiave Simmetrica** Nell'autenticazione C-R con chiave simmetrica, client e server condividono una chiave segreta  $K$ .

In questo caso avremo che:

- Challenge: Il sistema passa all'utente un messaggio.
- Response: L'utente codifica il messaggio con la chiave segreta  $K$ .
- Autenticazione: Se il messaggio può essere decodificato con successo.

In questo caso la Chiave non viene mai mandata sulla rete.

**Nonce** La challenge viene cambiata di volta in volta, altrimenti basterebbe intercettare il messaggio cifrato per poter accedere.

## 2.0.6 Domande

■ Si descriva la differenza tra **DAC** e **MAC**, facendo un esempio di una politica di controllo degli accessi per ognuno dei due tipi

■ Si descriva il funzionamento di una Access Control List

■ Si descriva il funzionamento di una Access Control List basata su 9bit come quella di Unix/Linux

■ Si descrivano i concetti di Mediazione Completa e Principio dei Privilegi Minimi

■ Descrivere il modello Bell-LaPadula completo

■ Si spieghi come la crittografia simmetrica possa essere usata per implementare un algoritmo di autenticazione **challenge-response**

■ Si descriva Challenge-Response con chiave Asimmetrica



# Capitolo 3

## Sicurezza nelle Applicazioni

### 3.0.1 Attacco SQL-Injection

**Funzionamento** L'attacco SQL Injection consiste nell'inserimento di una query SQL attraverso l'input utente al fine di modificare l'esecuzione di comandi SQL parametrici.

Un attacco eseguito con successo può:

- Leggere/Modificare dati sensibili
- Eseguire operazioni di amministrazione del database
- Recuperare il contenuto di un file presente nel filesystem del dbms.

SQL Injection si potrebbe manifestare in tutti i programmi che fanno uso di un DBMS.

**Contromisure** Le contromisure per questo tipo di attacco sono:

- Encoding sull'input: Trasformare i caratteri potenzialmente terminatori in una certa codifica ( $\rightarrow \&\text{lt;}$ ).
- Validazione dell'input: Accettarlo o rifiutarlo in base ad una white/black list per permettere di evitare input potenzialmente dannosi.

### 3.0.2 Attacco Buffer Overflow

**Funzionamento** Un attacco di tipo Buffer-Overflow, dovuto alla mancanza di controlli OutOfBound, consiste nello scrivere in un buffer uscendo intenzionalmente dai limiti dello stesso, andando a scrivere in aree di memoria non assegnategli adiacenti. Questo permette di modificare vari valori, tra cui variabili locali e l'activation record (che contiene il return address) di una funzione.

In genere questo tipo di attacco cerca di sovrascrivere il return address delle funzioni in esecuzione, in modo da farlo puntare ad una funzione malevola inserita dall'attaccante nel buffer stesso.

**ShellCode** Il codice malevolo solitamente è uno shellcode, ovvero un programma che tramite una chiamata di sistema permette di sostituire il processo corrente con uno nuovo (una shell) ed effettuare una syscall per ottenere privilegi Kernel. Spesso è contornato da istruzioni NOP, ovvero istruzioni che non fanno nulla, in modo da poter fare un salto impreciso.

**Difficoltà** Le due difficoltà principali nell'uso di questo exploit sono:

- Modifica del Base Pointer, se esso viene modificato e presenta un valore inconsistente, il programma termina.
- Bisogna indovinare la posizione esatta del return address.

**Heap e Stack** con buffer overflow posso modificare sia Heap che Stack, ma solo modificando lo stack posso modificare gli indirizzi di ritorno.

**Contromisure** Le principali contromisure sono:

- Modifica degli indirizzi di memoria virtuale tra un'esecuzione e l'altra.
- Introduzione di un Canary Value, che cambia ad ogni esecuzione e viene inserito dopo il base pointer.

### 3.0.3 Attacco XSS

L'attacco di tipo Cross Site Scripting è un tipo di code injection.

Si verifica quando un attaccante riesce ad eseguire del codice malevolo all'interno del browser di un utente sfruttando una vulnerabilità di un sito web ignaro dell'attacco.

Il codice malevolo è implementato in un linguaggio di scripting, ad esempio JS.

**Tipologie** Esistono 3 tipologie di XSS:

- Persistent XSS, il codice risiede nel DB del server (inserito tramite un form)
- Reflected XSS, il codice risiede nella richiesta HTTP (inserito in un URL malevolo)
- DOM-Based, la vulnerabilità risiede nel codice eseguito dal client.

### 3.0.4 NULL Pointer Dereference

**Null** Il valore NULL viene tipicamente assegnato alle variabili puntatore dopo la loro dichiarazione per indicare che la variabile è vuota (non punta a niente). Per il sistema operativo rappresenta un valore non valido e un tentativo di accedere alla risorsa null porta a un errore in runtime. Per l'hardware invece é l'indirizzo 0 e rappresenta la prima cella di memoria.

### 3.0.5 Domande Sicurezza Applicazioni

- Si descriva il funzionamento di un attacco SQL injection
- Si descriva il funzionamento di un attacco di tipo Buffer Overflow
- Se opero con codice su rom + heap è possibile avereun buffer overflow?

# Capitolo 4

## Sicurezza nelle Reti

### 4.0.1 Firewall

Il firewall è un sistema HW/SW per controllare il traffico in ingresso/uscita da una rete fidata rispetto a reti non sicure.

#### Tipi di Firewall

- Packet Filter: Agisce sui protocolli di rete e monitora sull'Header IP dei Pacchetti.
- Application Gateway: Agisce sui protocolli applicativi e monitora sui dati applicativi.

#### Packet Filter Firewall

Il Firewall Packet Filter agisce a livello del protocollo di rete, e quindi IP. Esamina in pacchetti in transito ed esegue una selezione in base al loro header IP, quindi l'indirizzo di sorgente/destinazione o la porta utilizzata.

#### Esempi

- Bloccare il traffico verso alcune porte o proveniente da host pericolosi o non fidati.
- Bloccare pacchetti Malformati per difendere da attacchi DoS.
- Non ammettere pacchetti esterni con un indirizzo sorgente interno alla rete, per difendere da IP-Spoofing.

Tutte le decisioni prese da questo firewall si basano su una tabella di filtraggio, la quale identifica tutti quei indirizzi IP le cui comunicazioni devono essere bloccate.

## Application Gateway Firewall

Il firewall Application Gateway, anche chiamato proxy, agisce a livello applicativo monitorando il traffico fra applicazioni mettendosi tra le due e ridistribuendo pacchetti tra le due applicazioni.

Questo tipo di firewall non agisce su pacchetti singoli ma su flussi di pacchetti e permette di controllare i dati applicativi in dettaglio.

### Esempi

- Filtro anti Spam.
- Proxy FTP per bloccare i trasferimenti di file in uscita ma non quelli in entrata.

## 4.0.2 SSL

SSL, ovvero Secure Socket Layer aggiunge privacy e integrità ai pacchetti in transito su una connessione TCP-IP.

SSL Modifica il pacchetto applicativo e aggiunge informazioni per permettere alla macchina di destinazione di riconoscere l'informazione.

**SSL Handshake** L'Handshake SSL é utilizzato per stabilire la connessione sicura fra un SSL Client e un SSL Server. Precede lo scambio di dati relativo alla sessione applicativa (HTTP).

L'Handshake Negozia la versione del protocollo, gli algoritmi crittografici per privacy e integrità.

### Fasi dell'Handshake

1. Hello Phase
2. Server Authentication
3. Client Authentication
4. Finish.

## 4.0.3 IPsec

IPsec, abbreviazione di IP Security, è uno standard per reti a pacchetto che si prefigge di ottenere connessioni sicure su reti IP. è uno standard di sicurezza a livello di rete, implementa privacy e integrità a livello IP:

- Trasparente alle applicazioni e agli utenti;
- Host-to-Host;
- Anche con protocolli di trasporto connection-less;

**Idea di Base** L'idea di base di IPsec è quella di offrire un livello di sicurezza aggiuntivo, lavorando sopra il livello IP, sostanzialmente criptando anche i metadati del livello TCP. Questo permette di nascondere il tipo di comunicazione che sta avvenendo a un potenziale attaccante, il quale (in modalità trasporto) saprà soltanto che la macchina A sta comunicando con la macchina B.

**Modalità** IPsec ha due modalità di funzionamento:

- Modalità Trasporto, che offre una protezione simile a SSL.
- Modalità Tunneling, in cui un ulteriore header incapsula i pacchetti per l'instradamento fra i due host, proteggendo le informazioni di rete originali

**Difese contro attacchi** IPsec permette:

- Maggiore resistenza a sniffing e modifica dei dati,
- Maggiore protezione contro traffic analysis
- Maggiore protezione contro IPspoofing

### Modalità Trasporto

IPsec in modalità trasporto è Host-to-Host, vengono criptati tutti i metadati da TCP compreso in sù. Grazie alla modalità trasposrto di IPsec si vuole offrire un livello di sicurezza in più, lavorando sopra al livello IP e andando a criptare i metadati del livello TCP. Chi intercetta il pacchetto non potrà quindi ne sapere il conenuto del pacchetto e neanche chi sono i veri attori coinvolti.

**Struttura** il pacchetto sarà composto da:

- Informazioni Protette, quindi il pacchetto applicativo e l'header TCP.
- IPsec
- Header IP (in chiaro)

### Modalità Tunneling

Nella modalità Tunneling abbiamo un ulteriore header che incapsula i pacchetti per l'instradamento fra i due host (tunnel end-points), proteggendo le informazioni di rete originali.

## Struttura

- Informazioni Protette, quindi il pacchetto applicativo, l'header TCP e l'header IP originale.
- IPsec
- Nuovo Header IP (in chiaro).

### 4.0.4 Domande Sicurezza Reti

■ Si spieghi come funziona un firewall "Packet-Filter", e come può essere usato per proteggere una sottorete da attacchi IP-Spoofing provenienti dall'esterno della sottorete stessa.

■ Si descrivano le differenze tra firewall di tipo packet filter e firewall di tipo application Gateway.

■ Si descriva la struttura di un pacchetto IP trasmesso usando il IPsec in modalità trasporto/ modalità tunneling, spiegando che tipo di protezione si ottiene