

# Sicurezza dei Sistemi e delle Reti

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Terminologia . . . . .	3
1.2	Sicurezza . . . . .	4
<b>2</b>	<b>Standard e Concetti base</b>	<b>5</b>
2.1	Standard . . . . .	5
2.2	Reference Monitor Model - RMM . . . . .	5
2.3	Definizioni . . . . .	6
2.4	Sfide della sicurezza informatica . . . . .	6
2.5	Principi fondamentali di progettazione della sicurezza . . . . .	7
2.6	Superficie di attacco . . . . .	7
2.6.1	Categorie di attacco . . . . .	8
2.7	Albero di attacco . . . . .	8
2.8	Tipi di attacco . . . . .	8
2.9	Implementazione della sicurezza . . . . .	8
<b>3</b>	<b>Controllo degli accessi</b>	<b>9</b>
3.1	Requisiti di sicurezza . . . . .	10
3.1.1	Requisiti di sicurezza di base . . . . .	10
3.1.2	Requisiti di sicurezza derivati . . . . .	10
3.2	Elementi di Access Control . . . . .	10
3.3	Politiche di controllo degli accessi . . . . .	11
3.3.1	DAC . . . . .	11
3.3.2	MAC . . . . .	12
3.3.3	RBAC . . . . .	13
3.4	Unix security model . . . . .	14
3.4.1	Processi in Linux . . . . .	14
3.4.2	Unix file access control . . . . .	14
3.5	Windows security architecture . . . . .	15
3.5.1	Windows security model . . . . .	15
3.5.2	Security descriptor . . . . .	15
3.6	Set-UID . . . . .	16
3.6.1	Superficie di attacco dei programmi Set-UID . . . . .	16

<b>4</b>	<b>Politiche di sicurezza</b>	<b>18</b>
4.1	Definizioni . . . . .	19
<b>5</b>	<b>Malware</b>	<b>20</b>
5.1	Trojan . . . . .	21
5.2	Virus . . . . .	21
5.2.1	Vettori di infezione . . . . .	22
5.2.2	Classificazione dei virus . . . . .	22
5.3	Worm . . . . .	23
5.4	Drive-by-download . . . . .	23
5.5	Clickjacking . . . . .	23
5.6	Zombie e botnet . . . . .	23
5.7	Rootkit . . . . .	23
5.8	Scareware . . . . .	23
5.9	Ransomware . . . . .	24
5.10	Vulnerabilità zero-day . . . . .	24
5.11	Spear phishing . . . . .	24
5.12	Spyware . . . . .	24
5.13	APT - Advanced Persistent Threats . . . . .	24
5.14	Approcci alle contromisure per i malware . . . . .	25
<b>6</b>	<b>Autenticazione</b>	<b>26</b>
6.1	Principi di autenticazione . . . . .	26
6.2	Requisiti di sicurezza . . . . .	26
6.3	Metodi di autenticazione . . . . .	27
6.4	Password . . . . .	27
6.4.1	Memorizzazione delle password . . . . .	27
6.4.2	Cracking delle password . . . . .	28
6.4.3	Utilizzo delle password su canali non sicuri . . . . .	28
6.5	Altri metodi di autenticazione . . . . .	28
6.6	Problemi di sicurezza dell'autenticazione . . . . .	28

# Capitolo 1

## Introduzione

### 1.1 Terminologia

Un **sistema** può essere visto come (anche una combinazione di):

- *hw*
- *sw*
- persone che lavorano con *hw* e *sw*
- clienti

Un **attore** può essere:

- una *persona* che *interagisce* con il sistema
- un *dispositivo* che *interagisce* con il sistema
- un *ruolo* (cliente)
- un *ruolo complesso* (Alice che finge di essere Bob)

Una rete è una configurazione di individui interconnessi. Una **rete di computer** può essere vista sotto due punti di vista:

- **Fisico:** una infrastruttura *hw* che connette diversi dispositivi
- **Logico:** un sistema che facilita lo scambio di informazioni tra applicazioni che non condividono uno spazio di memoria

## 1.2 Sicurezza

La sicurezza può essere intesa come il **raggiungimento di un obiettivo in presenza di un attacco**; è difficile da assicurare perché l'obiettivo è *negativo*:

- *dimostrare che Alice può accedere ad un file è facile*
- *dimostrare che nessuno oltre ad Alice può accedervi è molto più difficile*

Di norma si raggiunge con un processo **iterativo**:

- si cerca di trovare l'*anello debole* nel sistema
- si adottano delle *contromisure*
- si continua a fare *analisi* in cerca di nuove vulnerabilità

Il concetto di *sicurezza perfetta* non è raggiungibile; per discutere di sicurezza si deve definire:

- **Politica di sicurezza:** definizione di regole di sicurezza che il sistema deve rispettare
- **Modello di minaccia:** assunzioni su cosa possa fare l'avversario per penetrare nel sistema; devo comprendere la potenza dell'avversario
- **Meccanismi:** *sw* o *hw* che cercano di assicurare che la politica sia rispettata, finché l'attaccante segue il modello di minaccia

Le reti di computer sono sistemi insicuri: abbiamo un sistema complesso (*computer*) in un sistema complesso (*rete*) → è difficile prevedere da quale punto arriveranno gli attacchi e quali vettori verranno sfruttati.

Ad oggi, le motivazioni dietro agli attacchi sono principalmente:

- economiche
- politiche / militari
- attivismo

## Capitolo 2

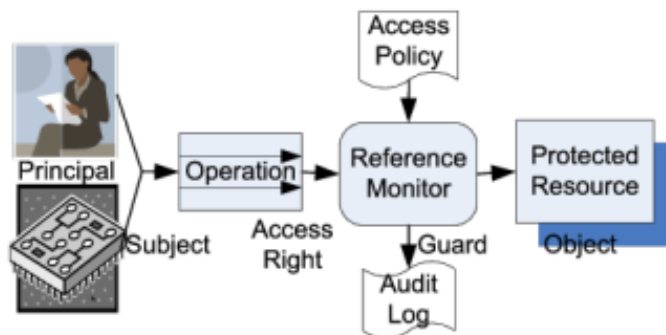
# Standard e Concetti base

### 2.1 Standard

Ci sono diverse organizzazioni che si occupano di standard:

- NIST (*National Institute of Standards and Technology*)
- ISOC (*Internet Society*)
- ITU-T (*International Telecommunication Union*)
- ISO (*International Organization for Standardization*)
  - 27001: documento a cui fare riferimento per costruire un sistema di gestione della sicurezza delle informazioni che possa essere certificato da un ente indipendente
  - 27002: non è certificabile, è una raccolta di *best practices* per soddisfare i requisiti della 27001

### 2.2 Reference Monitor Model - RMM



Il **reference monitor** è un **sistema dotato di una politica di controllo degli accessi**. Si occupa di:

- **autenticare chi vuole accedere**
- **autorizzare** o meno le operazioni richieste in base ai permessi
- fare **audit** → tenere un log delle azioni compiute

## 2.3 Definizioni

- La **sicurezza informatica** è l'insieme di strumenti, politiche, linee guida ... che possono essere utilizzate per proteggere l'ambiente e le risorse dell'organizzazione e degli utenti del cyberspazio
- I **beni dell'organizzazione e degli utenti** comprendono i dispositivi informatici connessi, personale, infrastrutture e la totalità delle informazioni trasmesse e/o archiviate nel cyberspazio
- Gli **obiettivi** generali di sicurezza comprendono *disponibilità, integrità e confidenzialità*
- *Sottoinsiemi* della sicurezza informatica:
  - **Sicurezza delle informazioni:** conservazione della CIA delle informazioni
  - **Sicurezza delle reti:** protezione delle reti e del loro servizio da modifiche non autorizzate e garanzia che la rete svolga correttamente le sue funzioni critiche

## 2.4 Sfide della sicurezza informatica

- Non è semplice; può avere requisiti semplici ma **meccanismi di implementazione complessi**
- Nello sviluppo di un meccanismo di sicurezza, si deve sempre **considerare potenziali attacchi**
- Le procedure utilizzate per fornire particolari servizi possono essere **controintuitive poiché complesse**
- Bisogna decidere **dove utilizzare i meccanismi di sicurezza**, sia a livello logico che a livello fisico
- I meccanismi di sicurezza in genere coinvolgono **più di un algoritmo o protocollo**
- Una battaglia **continua** tra attaccante e difensore

## 2.5 Principi fondamentali di progettazione della sicurezza

- **Fail-safe default:** nel caso in cui il sistema vada in default, deve rimanere in uno *stato protetto*
- **Economia di meccanismo:** i meccanismi devono essere il più semplice possibile
- **Mediazione completa:** *tutti* gli accessi devono essere controllati per assicurarsi che siano consentiti; solitamente accade che solo la prima interazione è controllata
- **Design aperto:** la sicurezza non deve dipendere dalla segretezza della sua progettazione o implementazione
- **Seperazione dei privilegi:** un sistema non dovrebbe concedere l'auto-rizzazione in base a *una singola* condizione
- **Minimi privilegi:** devono essere concessi il minor numero possibile di privilegi ad ogni soggetto; eventuali permessi addizionali devono essere concesso per il tempo minimo possibile
- **Accettabilità psicologica:** i meccanismi di sicurezza non dovrebbero rendere l'accesso ad una risorsa più difficile
- **Isolamento**
- **Incapsulamento**
- **Modularità**
- **Stratificazione (*layering*)**
- **Minima sorpresa:** evitare che l'utente si trovi davanti a situazioni inaspettate che potrebbero portarlo a seguire comportamenti scorretti

## 2.6 Superficie di attacco

Una superficie di attacco è costituita dalle **vulnerabilità raggiungibili e sfruttabili** in un sistema, come ad esempio:

- porte aperte verso l'esterno
- interfacce web
- dipendente con accesso a dati sensibili
- ...

→ è necessario **ridurre al minimo** la superficie di attacco



### 2.6.1 Categorie di attacco

- Superficie di attacco di **rete**: sono incluse vulnerabilità del protocollo di rete, che possono portare a DoS, interruzione dei collegamenti di comunicazioni ed altri attacchi intrusivi
- Superficie di attacco **software**: vulnerabilità nel codice delle applicazioni; un focus particolare è il software per server web
- Superficie di attacco **umano**: vulnerabilità create dal personale o da estranei, come *social engeneering*, errore umano o intrusi

## 2.7 Albero di attacco

Un albero di attacco è un modo di **rappresentare le possibilità di attacco**, e quindi di progettare le **contromisure**.

## 2.8 Tipi di attacco

- **Passivi**: non alterano le informazioni in transito; lo scopo è ottenere informazioni sui messaggi trasmessi
- **Attivi**: modificano il flusso delle informazioni
  - *Attacco di replay*: l'attaccante osserva le informazioni e le riutilizza in un secondo momento per creare una nuova sessione di comunicazione  
→ ci si tutela con *numeri casuali* e *timestamp* per, rispettivamente, controllare che i messaggi non siano già stati scambiati o che siano ancora validi
  - *DoS e DDoS*
  - ...

## 2.9 Implementazione della sicurezza

Quattro linee d'azione complementari:

- **Prevenzione**
- **Rilevamento**
- **Risposta** in modo da fermare un attacco e prevenire ulteriori danni
- **Ripristino** con sistemi di backup in caso l'integrità dei dati sia compromessa

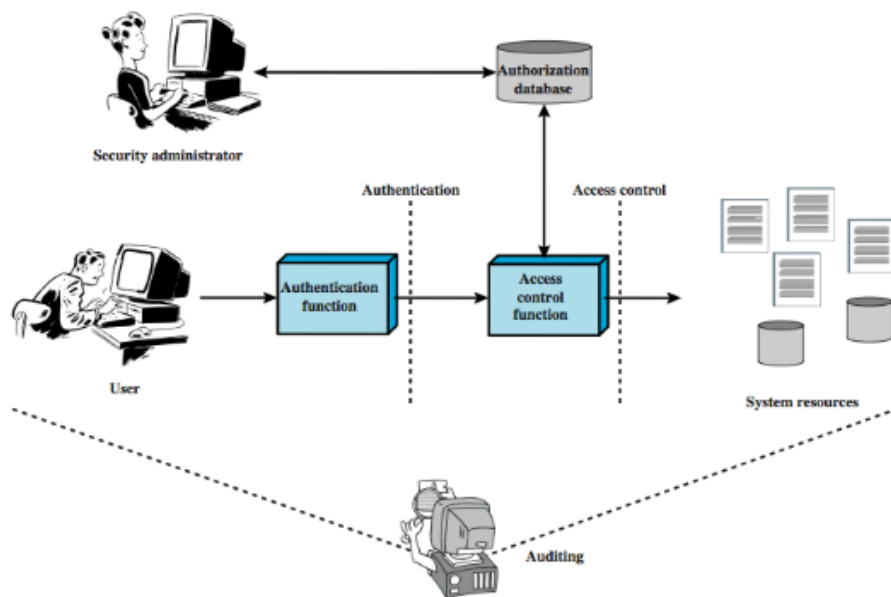
## Capitolo 3

# Controllo degli accessi

Il controllo degli accessi è un elemento centrale nella sicurezza informatica. È il **meccanismo** che definisce una **politica di sicurezza** per la quale si decide quali utenti possano accedere o meno ad una risorsa.

Si basa su tre principi fondamentali:

- **Autenticazione:** verifica che le credenziali fornite siano valide
- **Autorizzazione:** concessione di un permesso ad un'entità affinché possa accedere ad una risorsa del sistema
- **Auditing:** verifica delle attività e dei registri di sistema



## 3.1 Requisiti di sicurezza

### 3.1.1 Requisiti di sicurezza di base

- Limitare l'accesso al sistema informativo agli **utenti autorizzati**, ai processi che agiscono per conto degli utenti autorizzati o ai dispositivi
- Limitare l'accesso al sistema informativo alle tipologie di funzioni che gli utenti autorizzati possono eseguire

### 3.1.2 Requisiti di sicurezza derivati

- **separare i doveri** dei singoli individui per ridurre il rischio di attività malevole
- utilizzare **account non privilegiati** quando si accede a funzioni non di sicurezza
- **impedire agli utenti non privilegiati** di eseguire funzioni privilegiate e controllare l'esecuzione di tali funzioni
- limitare i **tentativi di accesso** non riusciti
- utilizzare il **blocco della sessione** per nascondere l'accesso ai dati dopo un periodo di inattività
- fornire **avvisi di privacy** e sicurezza secondo le norme vigenti
- **terminare automaticamente** la sessione dopo una determinata azione
- **monitorare e controllare** le sessioni di **accesso remoto**
- usare sistemi **crittografici** per garantire la riservatezza delle sessioni per accessi da remoto
- **instradare l'accesso remoto** tramite punti di controllo degli accessi
- **autorizzare** l'esecuzione remota di **comandi privilegiati**
- **autorizzare l'accesso wireless** prima di consentire tali connessioni

## 3.2 Elementi di Access Control

- **Soggetto:** entità che può accedere agli oggetti (ad esempio, un processo che rappresenta l'utente)
- **Oggetto:** risorsa ad accesso controllato (file, directory, ...)
- **Diritto di accesso:** modo in cui un soggetto accede ad un oggetto (lettura, scrittura, ...)

### 3.3 Politiche di controllo degli accessi

- **DAC:** controlla l'accesso in base all'identità del soggetto e alle autorizzazioni che indicano che è/non è consentito fare ai richiedenti
- **MAC:** controlla l'accesso in base al confronto tra delle specifiche etichette di sicurezza applicate agli oggetti e le autorizzazioni di sicurezza
- **RBAC:** controlla l'accesso in base al ruolo che l'utente ha nel sistema e alle regole che stabiliscono quali accessi sono consentiti a quali ruoli
- **ABAC:** controlla l'accesso in base agli attributi dell'utente

#### 3.3.1 DAC

Il controllo dell'accesso viene fatto sull'**identità del soggetto richiedente** e delle **regole di accesso**. Definito *discrezionale* perché un'entità potrebbe avere i privilegi di accessi che le permettono, a sua volta, di concedere l'accesso ad un'altra entità.

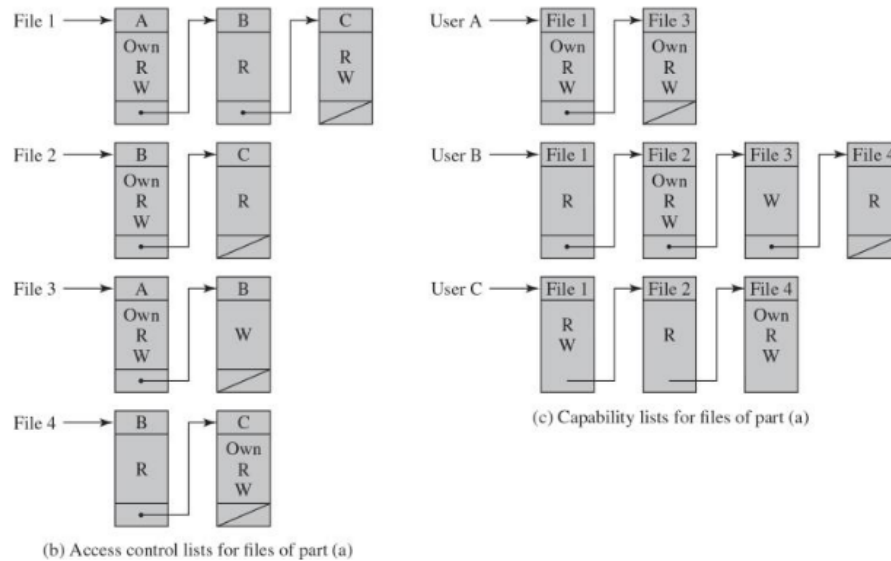
Si può rappresentare mediante una matrice di accesso, dove:

- elenca i soggetti in una dimensione (colonne)
- elenca gli oggetti nell'altra dimensione (righe)
- ogni cella specifica i diritti di accesso di quel soggetto a quel determinato oggetto

		OBJECTS			
		File 1	File 2	File 3	File 4
User A	Own	Read		Write	
	Read				
	Write				
User B	Read		Own	Write	Read
User C	Read		Write		
	Write				Own
			Read		Read
			Write		Write

Questa matrice ha il problema di essere *sparsa*, ovvero molto grande e con molte celle vuote

→ viene trasformata in una serie di liste per risorse ed utenti



### 3.3.2 MAC

Controlla l'accesso in base al confronto tra:

- **Etichette di sicurezza** che indicano quanto sono sensibili le risorse
- **Autorizzazioni di sicurezza** che indicano quali entità del sistema sono idonee ad accedere a quali risorse

Questa politica è definita obbligatoria (*mandatory*) perché un'entità che ha accesso ad una risorsa non può estendere il permesso ad un'altra; può farlo solo l'amministratore di sistema.

I sistemi MAC si dividono in:

- **Multilevel security systems:** consiste in una struttura verticale di livelli di sicurezza; agli utenti viene assegnato un livello e possono accedere solo a risorse con livello uguale o inferiore
- **Multilateral security systems:** l'accesso viene assegnato in base a segmenti che formano gruppi costituiti da livelli di sicurezza e parole in codice  
→ si ottiene una struttura orizzontale, che contiene livelli di sicurezza verticali aggiuntivi

#### Vantaggi e svantaggi

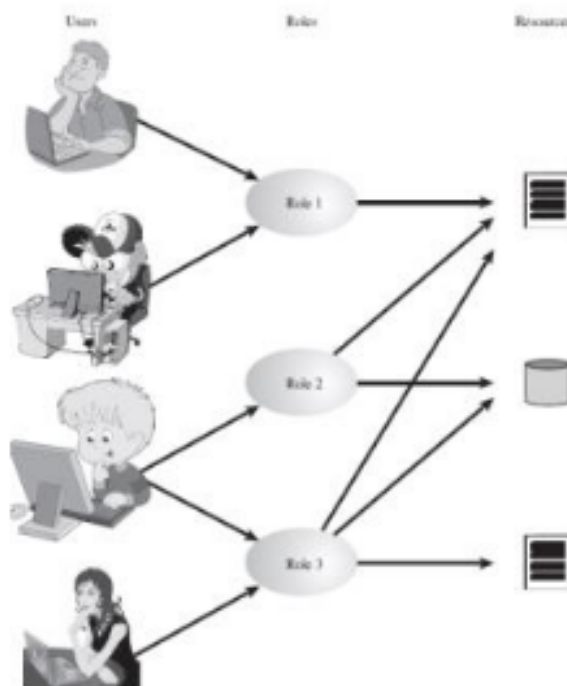
- Il MAC è uno dei sistemi di accesso più sicuri, poiché è praticamente a prova di manomissione

- gli utenti non possono fare modifiche
- il controllo è automatizzato
- i dati non possono essere modificati senza apposita autorizzazione
- ...tuttavia
  - richiede una pianificazione dettagliata e lavoro amministrativo
  - controllare e aggiornare i diritti di accesso
  - manutenzione per aggiunta di nuovi dati o utenti e relative modifiche
  - → elevato carico di lavoro per l'amministratore

### 3.3.3 RBAC

Ci sono quattro tipi di entità:

- **Utente:** una persona che ha accesso al sistema; ogni individuo ha un ID associato
- **Ruolo:** inteso come una funzione lavorativa all'interno dell'organizzazione
- **Autorizzazione:** approvazione di una modalità di accesso ad uno o più oggetti
- **Sessione:** mappatura tra un utente e un sottoinsieme dei ruoli a cui è assegnato

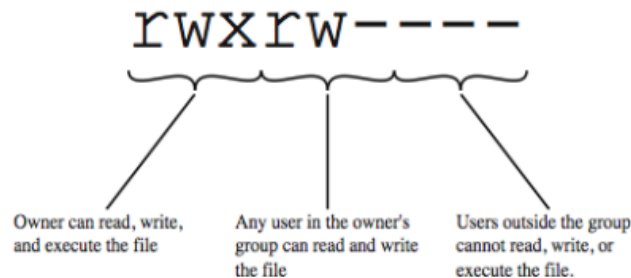


## 3.4 Unix security model

In Linux ci sono tre entità da considerare:

- **Soggetto:** può essere un utente o un processo
- **Oggetto:** file, cartelle, ...
- **Operazioni consentite:** lettura, scrittura, esecuzione

In Unix, ogni utente ha associato un id univoco, detto **UID**; può appartenere a gruppi di utenti, anch'essi identificati da un id univoco detto **GID**. Tutti gli utenti appartenenti ad un gruppo possono condividere tra loro oggetti. Ad ogni file è assegnato un unico utente proprietario e un unico gruppo proprietario. L'autorizzazione viene concessa mediante una ACL che identifica le operazioni che i soggetti possono fare.



### 3.4.1 Processi in Linux

Ogni processo è isolato dagli altri e non possono accedere alla memoria altrui. Ogni processo viene eseguito con le autorizzazioni dell'UID dell'utente che lo sta eseguendo.

Nel momento della creazione, ad ogni processo sono assegnati tre ID (inizialmente tutti uguali all'UID):

- **Effective UID:** determina le autorizzazioni per il processo
- **Real UID:** determina l'utente che ha avviato il processo
- **Saved UID:** EUID prima di eventuali modifiche

L'utente *root* può cambiare EUID/RUID/SUID a valori arbitrari; utenti non privilegiati possono cambiare EUID solo a RUID o SUID

### 3.4.2 Unix file access control

Le modifiche agli ID sono apportate mediante i comandi *setUID* e *setGID*; questa modifica permette ai programmi non privilegiati di accedere a risorse generalmente non accessibili.

Le directory possono aver impostato uno *sticky bit*: specifica che solo il proprietario di un file nella cartella può apportare una modifica a quel file. Il *superuser* è esente dalle consuete restrizioni di controllo degli accessi, ha accesso a tutto il sistema.

## 3.5 Windows security architecture

L'architettura di sicurezza di Windows è basata su più entità:

- **Security Reference Model (SRM)**: componente che in modalità kernel esegue controlli delle autorizzazioni e manipola i privilegi degli utenti
- **Local Security Authority (LSA)**: risiede in un processo utente, è responsabile dell'applicazione della politica di sicurezza locale, tra cui:
  - criteri per le password, come complessità e tempi di scadenza
  - politica di controllo → specifica quali operazioni su quali oggetti vadano controllate
  - impostazioni dei privilegi
- **Security Account Manager (SAM)**: è un database che archivia i dati degli account e le informazioni di sicurezza su entità locali e gruppi
- **Active Directory (AD)**: implementa il protocollo LDAP (*Lightweight Directory Access Protocol*)

### 3.5.1 Windows security model

Windows ha un complesso sistema di controllo dell'accesso; ogni oggetto ha ACL per permettere autorizzazioni granulari ad utenti e/o gruppi di utenti.

### 3.5.2 Security descriptor

Ogni oggetto ha un *security descriptor* che contiene:

- **security identifier (SID)** per il possessore e il gruppo primario dell'oggetto (SID è associato univocamente ad ogni utente)
- **discretionary ACL (DACL)**: diritti di accesso per gli utenti e i gruppi
- **system ACL (SACL)**: tipi di accesso che generano log

Ad ogni processo viene inoltre associato un insieme di **token**, che prende il nome di *security context*. Nel momento in cui un processo vuole accedere ad un oggetto, presenta il suo insieme di token e il sistema controlla se il security context abbia o meno accesso a tale risorsa in base al *security descriptor* dell'oggetto.

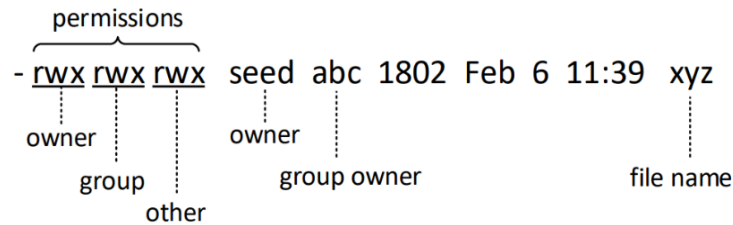


## 3.6 Set-UID

In Linux, ad ogni utente viene assegnato un ID univoco, memorizzato in `/etc/passwd`

Tramite i gruppi si rappresentano i gruppi di utenti che hanno le stesse autorizzazioni.

Le autorizzazioni su file vengono espresse nel seguente modo:



Per **set-UID** si intende **consentire all'utente di eseguire un programma con i privilegi del proprietario del programma**. Quando viene eseguito un normale programma,  $RUID = EUID$ , con entrambi che equivalgono all'id dell'utente che esegue il programma.

Quando viene eseguito un `set-uid`, `RUID` è uguale all'id dell'utente, ma `EUID` è uguale all'id del proprietario del programma

→ se il programma è di proprietà di `root`, viene eseguito con i *privilegi di root*

### 3.6.1 Superficie di attacco dei programmi Set-UID

- **Attacchi tramite input utente:** potrebbe essere inserito del codice malevolo, ad esempio tramite:
  - buffer overflow
  - inserimento di stringhe formattate appositamente
  - comando `chsh` → permette di cambiare shell tra quelli presenti in `/etc/passwd`
- problema: è impossibile disinfettare gli input dell'utente
- **Attacchi tramite input di sistema:** cambiare i riferimenti simbolici a file privilegiati
- **Attacchi tramite variabili d'ambiente:** le variabili d'ambiente possono essere impostate prima dell'esecuzione di un programma e influenzarne il comportamento
- **Perdita di capacità:** i programmi privilegiati potrebbero declassarsi durante l'esecuzione

- **Invocazione di programmi:** invocazione di comandi esterni dall'interno di un programma
  - agli utenti viene chiesto di fornire dati in input al comando; se non vengono fatti gli opportuni controlli, i dati di input dell'utente potrebbero essere trasformati nel nome di un comando
- **Attacchi tramite dynamic linker:** se il programma prevede linking dinamico di librerie, anche il contenuto diventa rilevante; se manomesse, potrebbero portare a comportamenti anomali del codice

## Capitolo 4

# Politiche di sicurezza

La *gestione della sicurezza* è un *processo formale* per rispondere alle domande:

- quali sono i beni da proteggere
- quali sono le possibili minacce
- come si possono contrastare le minacce

Questo processo ha natura iterativa, ed è contenuto nella ISO 31000; in questa norma viene descritto un *modello per la gestione della sicurezza delle informazioni* che comprende le seguenti fasi:

- **Plan:**
  - stabilire politiche, processi e procedure di sicurezza
  - eseguire la valutazione del rischio
  - sviluppare un piano di trattamento del rischio
- **Do:**
  - implementare il piano di trattamento del rischio
- **Check:**
  - monitorare e mantenere il piano di trattamento del rischio
- **Act:**
  - mantenere e migliorare la gestione dei rischi
  - risposta ad incidenti, analisi di vulnerabilità e riprendere il ciclo iterativamente

## 4.1 Definizioni

- **Rischio:** esprime la possibilità che un attacco causi danni ad una organizzazione
- **Risorsa:** tutto ciò che necessita di essere protetto
  - *hw*
  - *sw*
  - *reputazione*

La valutazione di una risorsa viene fatta in base:

- ai costi da sostenere per sostituire la risorsa nel caso non sia più disponibile
- perdita di incassi in caso di attacco
- **Vulnerabilità:** punti deboli che possono essere sfruttati per causare danni al sistema; possono essere classificate come:
  - critico
  - moderato
  - basso

## Capitolo 5

# Malware

Una definizione *informale* può essere quella di programma **malevolo**, solitamente inserito di nascosto in un sistema, che ha lo scopo di compromettere la riservatezza, l'integrità o la disponibilità del sistema stesso.

I malware sono classificati in base a:

- **Propagazione** (sw, rete, social engineering, ...)
- **Azioni sui dati** colpiti (corruzione, furto, crittografia, ...)
- **Attack kit** (strumenti già pronti per attaccare)
- **Attori e/o motivazioni** dell'attacco

Name	Description
Advanced Persistent Threat (APT)	Crimini informatici diretti contro obiettivi aziendali e politici, utilizzando un'ampia varietà di tecnologie di intrusione e malware, applicati in modo persistente ed efficace a obiettivi specifici per un periodo prolungato, spesso attribuiti a organizzazioni sponsorizzate dallo stato.
Adware	Pubblicità integrata nel software. Può provocare annunci pop-up o il reindirizzamento di un browser a un sito commerciale.
Attack kit	Insieme di strumenti per generare automaticamente nuovo malware utilizzando una varietà di meccanismi di propagazione e carico utile forniti.
Auto-rooter	Strumenti di hacker dannosi utilizzati per penetrare in nuove macchine da remoto.
Backdoor (trapdoor)	Qualsiasi meccanismo che eluda un normale controllo di sicurezza; potrebbe consentire l'accesso non autorizzato alle funzionalità di un programma o a un sistema compromesso.
Downloaders	Codice che installa altri elementi su un computer sotto attacco. Normalmente è incluso nel codice malware inserito prima in un sistema compromesso per poi importare un pacchetto malware più grande.
Drive-by-download	Un attacco che utilizza codice su un sito Web compromesso che sfrutta una vulnerabilità del browser per attaccare un sistema client quando viene visualizzato il sito.
Exploits	Codice specifico per una singola vulnerabilità o un insieme di vulnerabilità

Name	Description
Flooders (DoS client)	Utilizzato per generare un grande volume di dati per attaccare i sistemi informatici in rete, eseguendo una qualche forma di negazione del servizio (DoS) attacco.
Keyloggers	Cattura le sequenze di tasti su un sistema compromesso.
Logic bomb	Codice inserito nel malware da un intruso. Una bomba logica rimane dormiente finché non viene soddisfatta una condizione predefinita; il codice quindi attiva un carico utile.
Macro virus	Un tipo di virus che utilizza codice macro o script, generalmente incorporato in un documento o modello di documento e attivato quando il documento viene visualizzato o modificato, per essere eseguito e replicarsi in altri documenti simili.
Mobile code	Software (ad esempio, script e macro) che può essere distribuito senza modifiche a un insieme eterogeneo di piattaforme ed eseguito con la stessa semantica.
Rootkit	Insieme di strumenti hacker utilizzati dopo che l'aggressore è entrato in un sistema informatico e ha ottenuto l'accesso a livello di root.
Spammer programs	Utilizzato per inviare grandi volumi di posta elettronica indesiderata.
Spyware	Software che raccoglie informazioni da un computer e le trasmette a un altro sistema monitorando sequenze di tasti, dati sullo schermo e/o traffico di rete; o eseguendo la scansione dei file sul sistema alla ricerca di informazioni sensibili.

Name	Description
Trojan horse	Un programma per computer che sembra avere una funzione utile, ma ha anche una funzione nascosta e potenzialmente dannosa che elude i meccanismi di sicurezza, a volte sfruttando le autorizzazioni legittime di un'entità di sistema che lo invoca.
Virus	Malware che, una volta eseguito, tenta di replicarsi in un altro codice macchina o script eseguibile; quando ha successo, si dice che il codice è infetto. Quando viene eseguito il codice infetto, viene eseguito anche il virus.
Worm	Un programma per computer che può essere eseguito in modo indipendente e può propagare una versione completa e funzionante di se stesso su altri host di una rete, sfruttando le vulnerabilità del software nel sistema di destinazione o utilizzando credenziali di autorizzazione acquisite.
Zombie, bot	Programma installato su un computer infetto che viene attivato per lanciare attacchi su altri computer.

## 5.1 Trojan

È un programma che ha un effetto evidente e atteso dall'utente, che ha però anche un effetto **nascosto** che viola le politiche di sicurezza e che viene condotto senza l'autorizzazione dell'utente

## 5.2 Virus

È un codice che può replicarsi modificando altri file o programmi per inserire codice in grado di replicarsi a sua volta; questa **proprietà di replicazione** è ciò che distingue i virus dagli altri tipi di malware. Non svolge nessuna azione evidente, ma cerca di rimanere nell'ombra.

La replica richiede un certo tipo di assistenza da parte dell'utente, come ad esempio cliccare su un allegato.

Un virus è composto da tre parti:

- **Meccanismo di infezione**
- **Trigger:** evento che determina quando il payload viene attivato
- **Payload:** cosa fa il virus (oltre a diffondersi)

I virus attraversano quattro fasi:

1. **Dormiente:** il virus è inattivo in attesa di essere attivato
2. **Scatenante:** il virus viene attivato
3. **Propagazione:** il virus inserisce una copia di sé stesso in certe parti del sistema; ogni programma infetto conterrà ora un altro virus che entrerà a sua volta in fase di propagazione
4. **Esecutiva:** la funzione viene eseguita

### 5.2.1 Vettori di infezione

I principali vettori di infezione sono:

- **Boot sector** di dispositivi esterni; il codice è inserito nel boot sector e viene eseguito in fase di avvio
- **Eseguibili**
- **File macro:** il virus si attacca ai documenti per propagarsi

### 5.2.2 Classificazione dei virus

È possibile classificare i virus in base alle **tecniche usate per superare i controlli di sistema:**

- **Cifratura del virus:** crea una chiave per "*crittografarsi*"; quando viene chiamato un programma infetto, con tale chiave viene decifrato il virus. Per evitare pattern di bit, durante la propagazione la chiave viene cambiata
- **Stealth virus:** si nasconde dal rilevamento da parte dell'antivirus, tramite mutazione o compressione del codice
- **Polymorphic virus:** durante la replica crea copie che svolgono la stessa funzione ma che hanno pattern di bit diversi
- **Metamorphic virus:** si riscrive completamente ad ogni iterazione per aumentare la difficoltà di rilevamento
- **Compression virus:** comprimono il file eseguibile in modo che la versione infetta abbia la stessa dimensione di quella originale

## 5.3 Worm

I worm sono programmi *stand alone* (a differenza dei virus che devono essere attivati da un qualche evento) in grado di replicarsi.

Le fasi di esecuzione sono:

- **Probing:** cerca informazioni sulla macchina
- **Exploitation:** sfrutta le informazioni raccolte per trovare vulnerabilità
- **Replicazione**
- **Attacco** (payload)

## 5.4 Drive-by-download

Sfruttano **vulnerabilità del browser** per installare codice malevolo ad insaputa dell'utente nel momento in cui visita la pagina web dell'attaccante.

## 5.5 Clickjacking

L'attaccante intercetta un *click* dell'utente per costringerlo a fare delle cose contro la sua volontà.

## 5.6 Zombie e botnet

Lo *zombie* è una singola macchina, mentre la *botnet* è un insieme di macchine zombie controllate da una singola entità; vengono usate per fare DDoS, phishing, spamming, ...

## 5.7 Rootkit

È un insieme di programmi installati su un sistema per mantenere l'accesso ad un sistema, ad esempio, con privilegi di amministratore, nascondendo le prove della sua presenza e aggirando i meccanismi di controllo.

Permettono di fare attacchi anche con scarse conoscenze tecniche.

## 5.8 Scareware

Software che hanno lo scopo di diffondere shock, ansia e/o la percezione di una minaccia; sono un attacco di *social engineering*.



## 5.9 Ransomware

Software che tiene in ostaggio il sistema per richiedere un riscatto all'utente, spesso tramite cifratura.

## 5.10 Vulnerabilità zero-day

Si intende un'exploit non ancora nota e che non ha quindi una contromisura.

## 5.11 Spear phishing

Viene **studiato nel dettaglio il bersaglio**, in modo tale da fare del phishing più mirato ed efficace.

## 5.12 Spyware

Malware che raccoglie piccole informazioni alla volta sugli utenti a loro insaputa, come ad esempio un *keylogger*.

## 5.13 APT - Advanced Persistent Threats

- **Advanced:** è un'applicazione con un'ampia varietà di tecnologie di intrusione e malware
- **Persistent:** attacchi per un periodo di tempo prolungato verso il target
- **Target:** target selezionati in modo accurato

Le fasi principali di un attacco tramite APT sono:

- **Ricognizione:** si sceglie una vittima e la si studia
- **Weaponization:** si mette un trojan che permette accesso remoto in un payload consegnabile (email, USB, web)
- **Sfruttamento:** il codice malevolo viene attivato per portare a termine il suo scopo

## 5.14 Approcci alle contromisure per i malware

Le principali contromisure da adottare sono:

- assicurare di **aggiornare** tutti i sistemi
- ridurre al **minimo le vulnerabilità**
- impostare adeguati **controlli** di accesso alle applicazioni e ai dati
- **limitare** il numero di file a cui un utente può accedere, e che quindi può potenzialmente infettare

Le azioni da prendere nel caso il sistema venga attaccato sono:

- **Rilevamento:** accertarsi della presenza del virus; può essere fatto con:
  - programmi anti-virus, 4 generazioni:
    - \* **scanner semplici** → ricercano malware noti
    - \* **scanner euristici** → utilizzano regole euristiche
    - \* **trap di attività:** viene cercato il malware in base alle sue azioni piuttosto che in base alla sua struttura → l'analisi è dinamica
    - \* **protezione completa:** più tecniche usate insieme
  - meccanismi di protezione perimetrale nei firewall
- **Identificazione:** individuare lo specifico malware
- **Rimozione** di tutte le tracce dal sistema

## Capitolo 6

# Autenticazione

Processo per determinare se un utente, un'applicazione o un processo che agisce per conto di un utente è **effettivamente chi o cosa dichiara di essere**; si fa un confronto tra le credenziali inserite e quelle memorizzate nel sistema.

### 6.1 Principi di autenticazione

- **Identità digitale:** è la rappresentazione unica di un soggetto, consiste in un attributo o un insieme di attributi che descrivono univocamente un soggetto
- **Prova dell'identità:** stabilisce che un soggetto è chi afferma di essere ad un determinato livello di certezza
- **Autenticazione digitale:** il processo di determinazione della validità di uno o più autenticator utilizzati per rivendicare un'identità digitale

### 6.2 Requisiti di sicurezza

- **Di base:**
  - **identificare** gli utenti del sistema informativo e i processi che agiscono per loro conto
  - autenticare (o **verificare**) le identità di tali utenti
- **Derivati:**
  - utilizzare l'**autenticazione a più fattori**
  - impiegare meccanismi resistenti ad **attacchi di replay**
  - impedire il riutilizzo degli *id* per un certo intervallo di tempo
  - disabilitare gli *id* dopo un periodo di inattività

- applicare una **complessità minima** della password
- proibire il **riutilizzo** delle password
- memorizzare e trasmettere solo password **criptate**
- **oscurare il feedback** delle informazioni di autenticazione

## 6.3 Metodi di autenticazione

- **qualcosa che uno sa:** password, ...
- **qualcosa che uno ha:** token → smartcard, chiave fisica, ...
- **qualcosa che l'individuo è:** tratto biometrico
- **qualcosa che l'individuo fa:** biometria dinamica

## 6.4 Password

Lo schema standard *id-password* è soggetto a diverse vulnerabilità:

- canale tra client e server
- compromissione del client
- compromissione del server
- social engeneering
- password deboli

Esistono diversi tipi di attacco alle password:

- attacchi a **dizionario**
- **forza bruta**
- uso della **stessa password** per più servizi

### 6.4.1 Memorizzazione delle password

Nello vecchio sistema UNIX veniva memorizzato nel file `/etc/passwd` l'hash delle password insieme al suo id.

Oggi, la password viene salvata come **hash** nel file `/etc/shadow`, che è **leggibile solo da root**.

Per proteggersi da attacchi di forza bruta, è possibile usare un **salt**: invece che memorizzare l'hash della password, ad esso viene concatenato all'inizio e/o alla fine un *numero casuale* di lunghezza arbitraria; vengono usati salt fino a 48 bit.

### 6.4.2 Cracking delle password

- attacchi di dizionario
- tabelle *rainbow*: contengono hash precalcolati di diversi dizionari, con già inseriti dei salt

La maggioranza degli attacchi **sfrutta password deboli**; la soluzione è costringere gli utenti ad usare **password complesse**.

### 6.4.3 Utilizzo delle password su canali non sicuri

Esistono tre approcci:

- **One-time-password**
- **Challenge response**: chi si autentica dimostra di conoscere un segreto (challenge); di norma vengono usati anche timestamp per evitare attacchi di replay
- **Zero knowledge proof**: chi si autentica fornisce una prova di conoscere la password senza fornirla esplicitamente

## 6.5 Altri metodi di autenticazione

- **Smart token**: dispositivi dotati di **capacità computazionale**, possono fare diversi tipi di controlli
- **Smart card**: forniscono informazioni una volta inserite nel sensore, non hanno sistemi di calcolo
- **Autenticazione biometrica**
- **Autenticazione reciproca**: permettono ad entrambe le parti di accertarsi reciprocamente della propria identità

## 6.6 Problemi di sicurezza dell'autenticazione

- **Intercettazione**, ad esempio con un attacco che implica la vicinanza fisica di utente ed attaccante
- **Attacchi host**: diretti al file dell'utente in cui sono contenute le password/token/...
- **Replay**: l'avversario ripete una risposta dell'utente acquisita in precedenza, senza che il server sia in grado di distinguerla; si può contrastare con tre approcci:
  - allegare un **numero di sequenza** a ciascun messaggio

- **timestamp** per verificare la validità
- **challenge-response**: quando una parte invia un messaggio, deve contenere il *nonce* del messaggio precedente
- **Attacchi client**: l'avversario cerca di autenticarsi senza accedere all'host remoto
- **DoS**: tenta di disabilitare un servizio