Sicurezza e protezione

Programma – Sistemi Operativi

- Introduzione ai sistemi operativi
- Gestione dei processi
- Sincronizzazione dei processi
- Gestione della memoria centrale
- Gestione della memoria di massa
- File system
- Sicurezza e protezione

Sicurezza e protezione

- La sicurezza misura la fiducia nel fatto che l'integrità di un sistema e dei suoi dati siano preservati

- La protezione è l'insieme di meccanismi che controllano l'accesso di processi e utenti alle risorse di un sistema informatico

Sicurezza

La sicurezza si occupa di preservare le risorse del sistema da:

- ✔ accessi non autorizzati
- ✔ distruzione o alterazione dolosa
- ✔ involontaria introduzione di elementi di incoerenza

Risorse da preservare

Le risorse da preservare includono:

- ✔ informazione memorizzata nel sistema sotto forma di dati e programmi
- ✔ CPU
- ✔ memoria
- ✔ dischi
- ✔ connessioni di rete

Il problema della sicurezza

Le violazioni della sicurezza del sistema si possono classificare come *intenzionali* (*dolose*) o *accidentali*. Nell'elenco che segue sono comprese sia le *intrusioni accidentali* sia le *violazioni dolose*.

- Violazione della riservatezza
- Compromissione dell'integrità
- Violazione della disponibilità
- Appropriazione del servizio
- Rifiuto del servizio
  DOS
  (*Denial-Of-Service*)

Il problema della sicurezza

Violazione della riservatezza

- Lettura non autorizzata di dati
- Furto di informazioni

Compromissione dell'integrità

- Modifica non autorizzata di dati
- Modifica codice sorgente

Violazione della disponibilità

- Distruzione non autorizzata di dati
- Sabotaggio di siti web

Il problema della sicurezza

Appropriazione del servizio

- Uso non autorizzato delle risorse

Rifiuto del servizio

- Blocco dell'utilizzo legittimo del sistema
- Attacchi DOS (Denial-Of-Service)

Sicurezza del sistema

Per proteggere il sistema è necessario prendere misure di sicurezza a quattro livelli:

Sicurezza del sistema

Fisico

- Edifici
- Macchine
- Stazioni di lavoro
- Terminali

Rete

- Linee di comunicazione private
- Linee condivise
- Connessioni Wi-Fi

Sicurezza del sistema

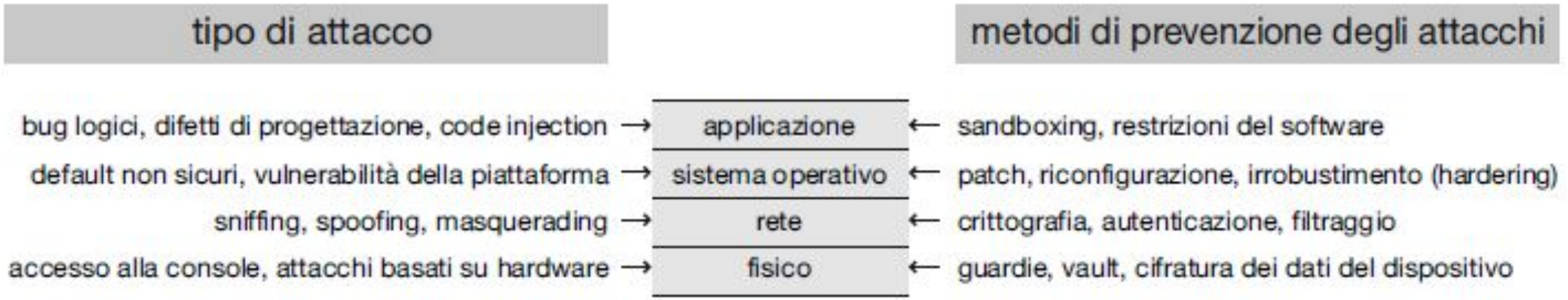
SO

- Impostazioni predefinite
- Parametri di configurazione
- Bug di sicurezza

Applicazione

- Programmi di terze parti
- Bug di sicurezza

Modello di sicurezza a quattro livelli



Il modello di sicurezza a quattro livelli è come una catena formata da anelli collegati: una vulnerabilità in uno qualsiasi dei suoi livelli può compromettere l'intero sistema.

Fattore umano

- L’autorizzazione degli utenti richiede cautela per garantire l’accesso al sistema solo agli utenti che ne abbiano diritto.

- Anche gli utenti autorizzati potrebbero essere malintenzionati oppure incoraggiati a cedere le loro credenziali ad altri volontariamente o mediante tecniche di ingegneria sociale (social engineering).

Phishing

consiste nel contraffare e-mail o pagine web rendendole simili a quelle autentiche per spingere gli utenti tratti in inganno a comunicare informazioni confidenziali

Esempio di Phishing

Italia Attualità

Temi Caldi Libia Usa 2020 Piano Autostrade Taglio cuneo fiscale Debito Italia

24+ ABBONATI Accedi

ITALIA Reddito di cittadinanza, ecco che lavori faranno i percettori

MONDO L'abbattimento del Boeing 737-800 nei cieli di Teheran

IL MILANESE IMBRUTTITO L'economia spiegata dal Nano: il mutuo

23 dicembre 2019

Natale
Iban
Ing Bank
NoiPA
CONSOB

Salva
Commenta

CYBER SICUREZZA

Truffa di Natale: hacker contro NoiPA, rubati stipendi e tredicesime a dipendenti pubblici

Operazione basata su tecniche di phishing, che riguarderebbe un numero non definito di dipendenti pubblici. Un furto che lascia spazio a molti interrogativi e al pesante dubbio di non poter recuperare il maltolto

di Biagio Simonetta

Il meglio di 24+

1. OCCUPAZIONE
Lavoro, richiesta record di laureati. Quali sono i titoli più gettonati

2. 24PLUS
Auto elettriche, perché è urgente l'alternativa al cobalto nelle batterie

3. REDDITI
Negli ultimi 20 anni le pensioni italiane sono cresciute più degli stipendi

4. L'INCHIESTA DELLA DOMENICA
Quali sono le scuole che fanno trovare più velocemente il lavoro

5. INDUSTRIA SOSTENIBILE
La sfida difficile di Volkswagen, Daimler e Bmw verso l'auto elettrica

https://www.ilsole24ore.com/art/truffa-natale-hacker-contro-noipa-rubati-stipendi-e-tredicesime-dipendenti-pubblici-ACtqa77

Minacce legate ai programmi

- Malware
  - Trojan
  - Spyware
  - Ransomware
  - Trap door
  - Logic bomb
- Code injection
  - SQL Injection
- Virus e Worm

Malware

- Il **malware** è un software progettato per *sfruttare*, *disabilitare* o *danneggiare* i sistemi informatici.

- Il termine deriva dall'abbreviazione dell'inglese *malicious software*

- Esempi di malware sono:
  - Trojan
  - Spyware
  - Ransomware
  - Trap door e Logic bomb

Trojan

- Un programma che agisce in modo clandestino o malevolo, anziché eseguire semplicemente la sua funzione dichiarata, è chiamato **cavallo di Troia**.

- Una variante è un programma (detto “**trojan mule**”) che emula una procedura di login: l’ignaro utente, nella fase di accesso a un terminale, crede di aver scritto erroneamente la propria password; prova ancora e, questa volta, ha successo. Questo porta alla sottrazione del nome utente e password.

Spyware

Mira a

- Visualizzare annunci pubblicitari sullo schermo dell'utente
- Creare finestre a comparsa nel browser quando si visitano alcuni siti
- Prelevare informazioni dal sistema dell'utente per trasmetterle ad un sito di raccolta senza che l'utente ne sia a conoscenza
- Talvolta accompagna un programma che l'utente ha scelto di installare

Esempio Spyware

Google Chrome vittima dello spyware: scoperte migliaia di estensioni fasulle

Home > Cyber Security

Condividi questo articolo

Dal Web Store effettuati 32 milioni di download malevoli che hanno permesso di catturare cronologia e credenziali degli utenti. Per i ricercatori questi attacchi rappresentano un nuovo strumento di spionaggio politico e industriale

18 Giu 2020

https://www.corrierecomunicazioni.it/cyber-security/google-chrome-vittima-dello-spyware-scoperte-migliaia-di-estensioni-fasulle/

Ransomware

- Sono malware che non rubano informazioni, bensì sono in grado di cifrare (parzialmente o totalmente) le informazioni presenti sul computer che attaccano, rendendole inaccessibili al legittimo proprietario

- Il fine è quello di chiedere al proprietario un riscatto (in inglese ransom) per avere la chiave necessaria a decifrare i dati.

- Si noti che sebbene l'informazione che si va a cifrare abbia di solito poco valore per l'attaccante, essa può essere estremamente importante per la vittima. Per tale motivo, molto spesso l'attaccato cede alle richieste dell'attaccante.

Esempio Ransomware

MENU | CERCA la Repubblica R+ | Rep: | ABBONATI | ACCEDI

Tecnologia

HOME NEWS SPECIALI MOBILE SOCIAL NETWORK SICUREZZA PRODOTTI INTERATTIVI VIDEO

Il ransomware "terrorista": chiede il riscatto in bitcoin e minaccia un attacco bomba

Il messaggio che compare sul computer bloccato intima di pagare 20.000 dollari in bitcoin, altrimenti "una persona reclutata" appositamente farà esplodere la bomba in quell'edificio. Ma i dubbi sono tanti

Trap door e Logic Bomb

- Una trap door è un tipo di malware in cui il progettista di un programma o di un sistema può lasciare nel programma un buco segreto che solo lui è in grado di utilizzare

- Logic bomb: trap door che si attiva solo al verificarsi di uno specifico insieme di condizioni logiche

Esempio Trap door

Trap door nei compilatori: Malware XCodeGhost

MENU TOP NEWS

LA STAMPA

TECNOLOGIA

NEWS GIOCHI IDEE PROVE TUTORIAL

ANDREA NEPORI

PUBBLICATO IL
20 Ottobre 2015

ULTIMA MODIFICA
24 Giugno 2019
ora: 10:06

Attacco hacker contro l'App Store di Apple, a rischio anche WeChat

Il malware cinese XcodeGhost è riuscito a infettare applicazioni per iOS disponibili sull'App store. Colpite anche alcune app diffuse in Europa e negli Stati Uniti

'/www.lastampa.it/promozioni/lettori/top-news/presentazione?ref=lastampa.abbonati.tntop_off

Difendersi dai Malware

I malware ottengono successi se riescono a violare il principio del minimo privilegio

IL PRINCIPIO DEL MINIMO PRIVILEGIO

"Il principio del minimo privilegio: in un sistema, ogni programma e ogni utente dotato di privilegi dovrebbero operare con il minimo privilegio necessario per completare il proprio lavoro, allo scopo di ridurre il numero di potenziali interazioni tra programmi privilegiati al minimo necessario per poter operare correttamente, in modo che si possa essere ragionevolmente fiduciosi del non verificarsi di usi non intenzionali, indesiderati o impropri del privilegio." Jerome H. Saltzer, nella descrizione di un principio di progettazione del sistema operativo Multics nel 1974:
https://pdfs.semanticscholar.org/1c8d/06510ad449ad24fbdd164f8008cc730cab47.pdf.

Code injection

Overflow di un buffer: il più semplice vettore di code injection

```
#include <stdio.h>
#include <string.h>
#define BUFFER_SIZE 0

int main(int argc, char *argv[])
{
    int j = 0;
    char buffer[BUFFER_SIZE];
    int k = 0;
    if (argc < 2) {return -1;}

    strcpy(buffer,argv[1]);
    printf("K is %d, J is %d, buffer is %s\n",j,k,buffer);
    return 0;
}
```

Figura 16.2 Programma C che esemplifica il buffer overflow.

Buffer overflow

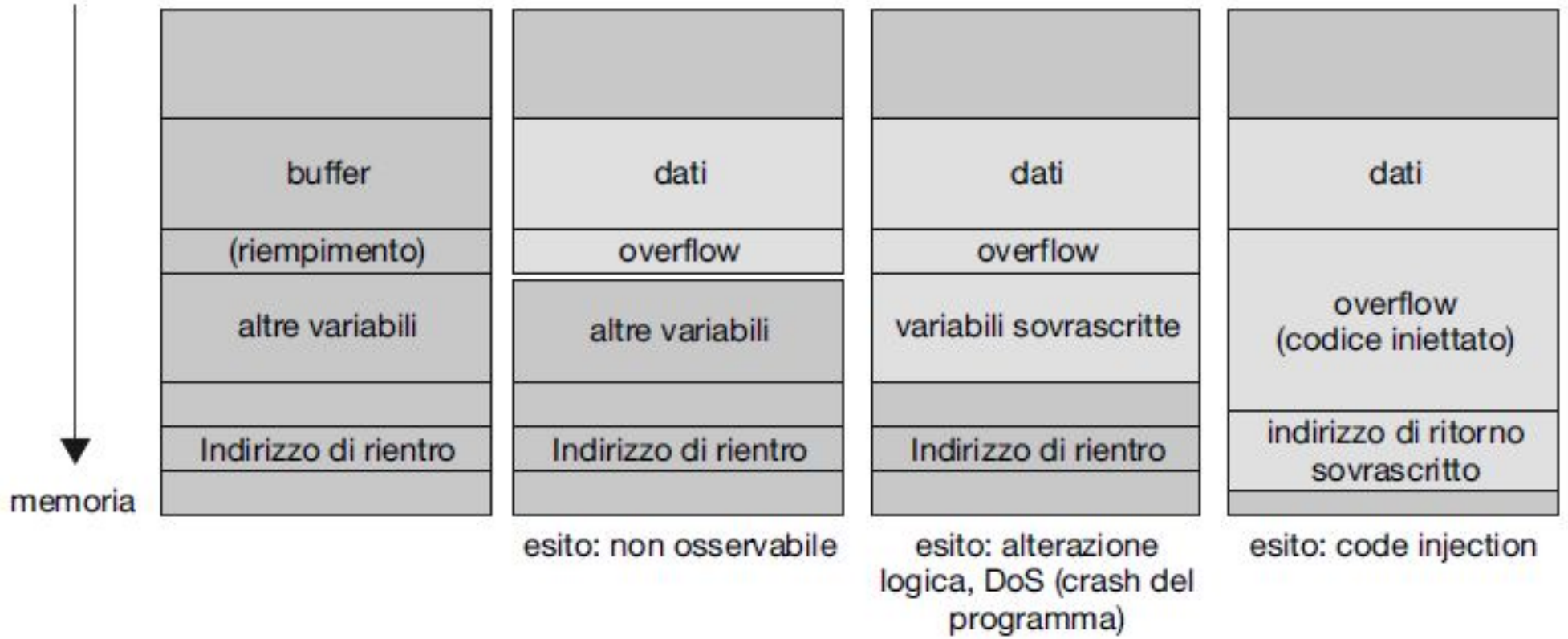


Figura 16.3 Possibili esiti di un buffer overflow.

Exploit shellcode

Un **exploit shellcode** è mostrato nella Figura 16.4.

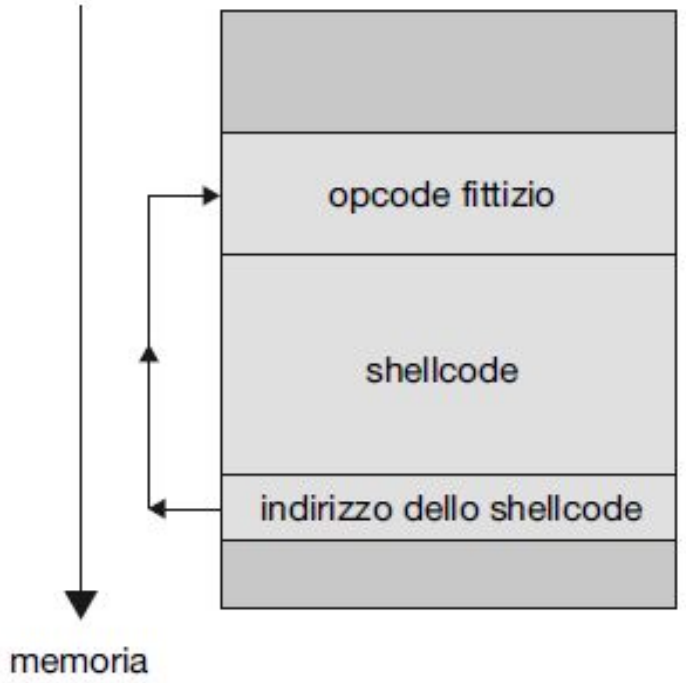


Figura 16.4 "Trampolino" per l'esecuzione di codice sfruttando un buffer overflow.

SQL injection

- SQL injection is the placement of malicious code in SQL statements, via web page input.
- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.

SQL in web pages

SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will **unknowingly** run on your database.

Look at the following example which creates a SELECT statement by adding a variable (txtUserId) to a select string. The variable is fetched from user input (getRequestString):

Example

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

SQL injection based on 1 = 1

Look at the previous example again. The original purpose of the code was to create an SQL statement to select a user, with a given `UserId`

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input like this:

UserId: `105 OR 1=1`

Then, the SQL statement will look like this:

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

The SQL above is valid and will return ALL rows from the "Users" table, since **OR 1=1** is always TRUE.

SQL injection based on 1 = 1

Does the discussed example look dangerous? What if the "Users" table contains names and passwords?

The SQL statement above is much the same as this:

```
SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1;
```

A hacker might get access to all the user names and passwords in a database, by simply inserting `105 OR 1=1` into the input field.

Virus

Virus: frammento di codice inserito in un programma legittimo.

I virus si autoriproducono e sono concepiti in modo da “contagiare” altri programmi. Il contagio può portare al *crash del sistema*

Normalmente i **virus** si trasmettono per posta elettronica, tramite posta indesiderata (*SPAM*), e utilizzando tecniche di *phishing.*

Dopo che un virus raggiunge la macchina presa di mira, un programma chiamato **portatore di virus** (*virus dropper*) inserisce il virus nel sistema.

Worm

Si può fare una distinzione tra i virus, che richiedono attività da parte dell'uomo, e i **worm**, che usano una rete per replicarsi, senza l'aiuto dell'uomo.

Categoria di Virus

Esistono migliaia di virus, che tuttavia possono essere ricondotti ad alcune categorie principali:

Virus del boot sector

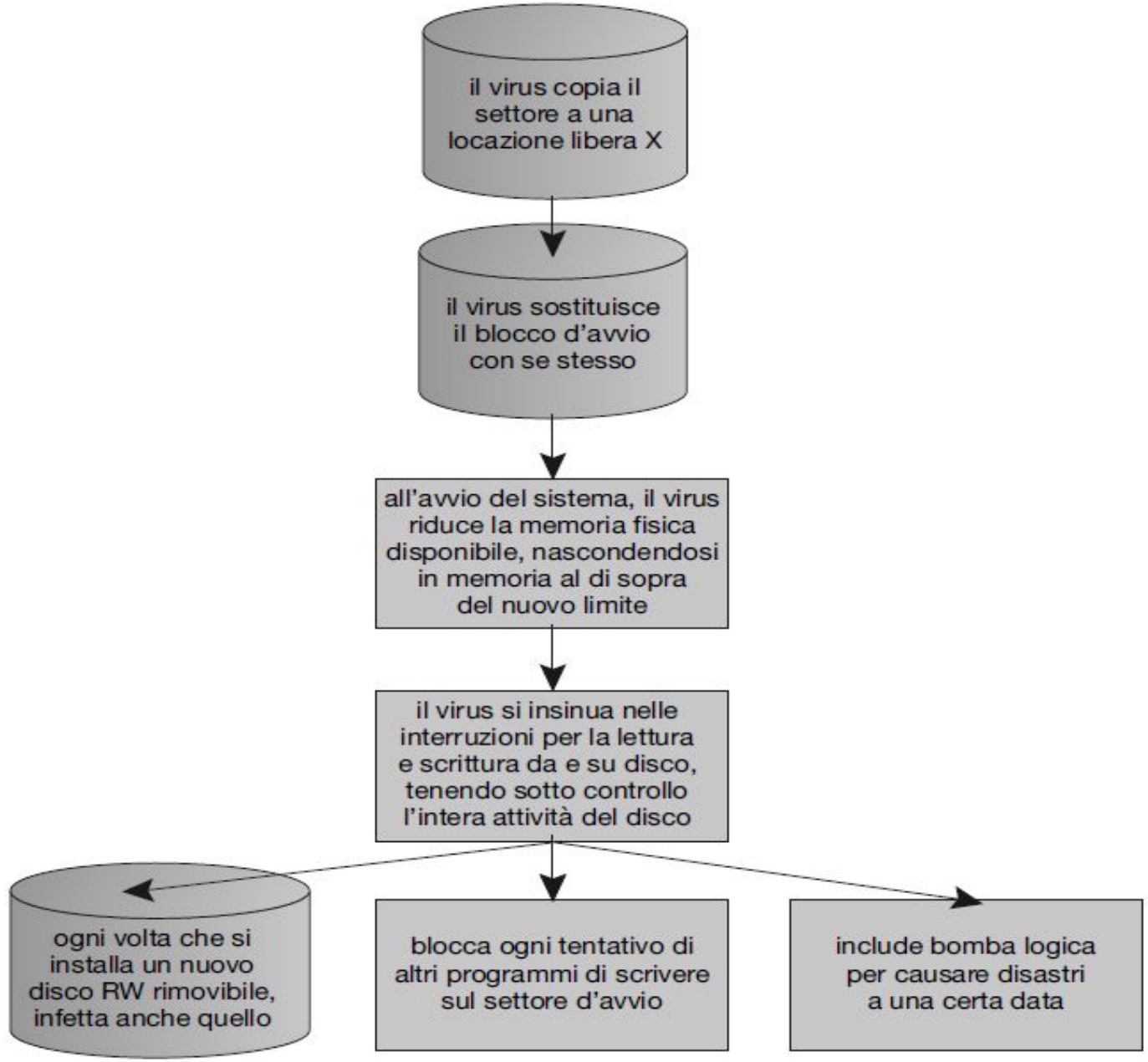
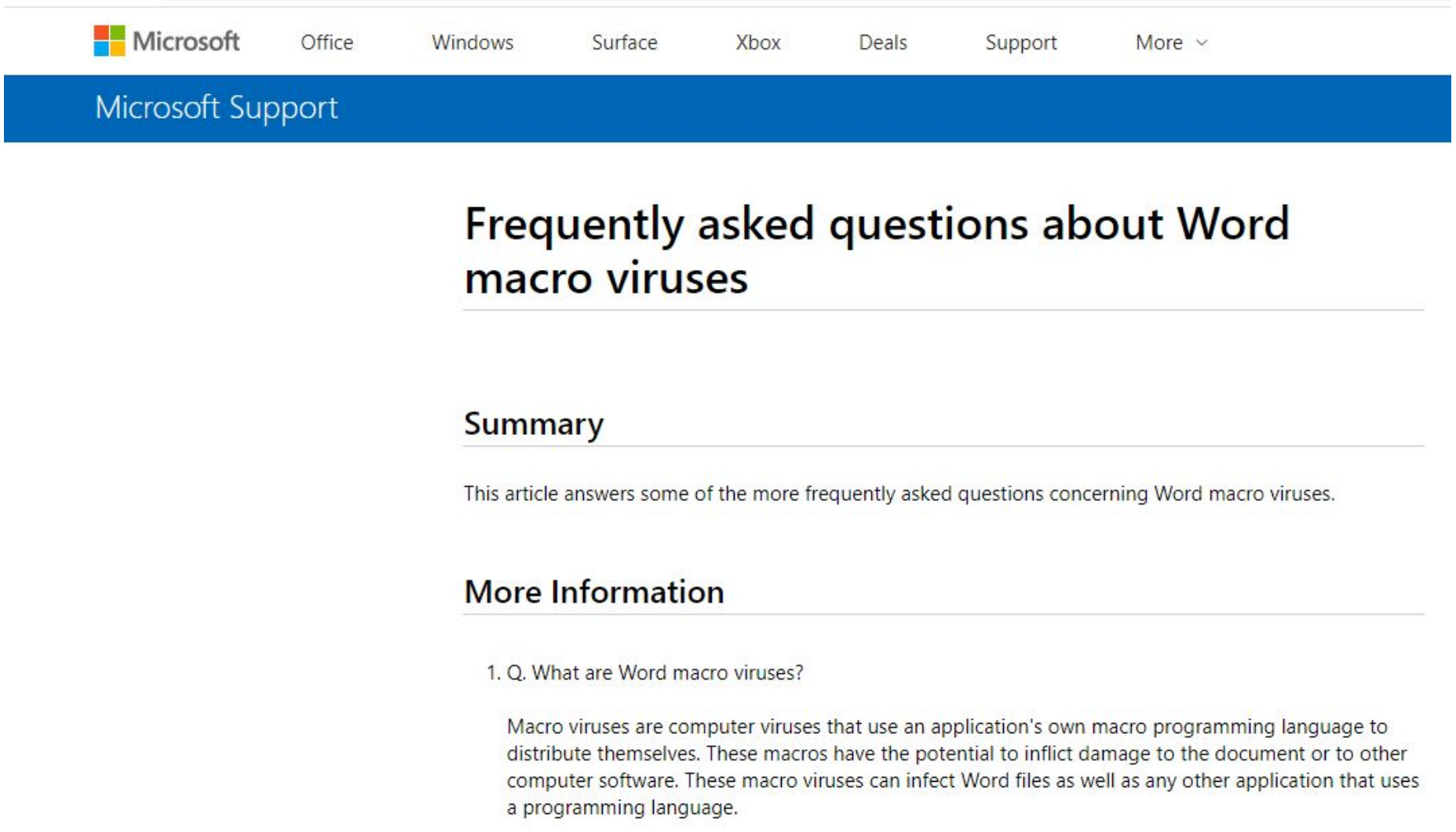


Figura 16.5 Virus del boot sector di un computer.

Macrovirus



Microsoft Office Windows Surface Xbox Deals Support More

Microsoft Support

Frequently asked questions about Word macro viruses

Summary

This article answers some of the more frequently asked questions concerning Word macro viruses.

More Information

1. Q. What are Word macro viruses?

   Macro viruses are computer viruses that use an application's own macro programming language to distribute themselves. These macros have the potential to inflict damage to the document or to other computer software. These macro viruses can infect Word files as well as any other application that uses a programming language.

Minacce relative al sistema e alla rete

Attaccare il traffico di rete

Un utente malintenzionato può scegliere:

1. di rimanere passivo e intercettare il traffico di rete (questo attacco è comunemente indicato come **sniffing**);

1. di assumere un ruolo più attivo, mascherandosi come una delle parti (**spoofing**);

1. di diventare un **man-in-the-middle** (*uomo nel mezzo*) completamente attivo, che intercetta ed eventualmente modifica le transazioni tra due soggetti comunicanti.

Attacchi alla sicurezza

Metodi standard per infrangere la sicurezza:

- attacco mimetico (masquerading)
- attacco replay (replay attack)
- attacco di interposizione (man-in-the-middle attack)

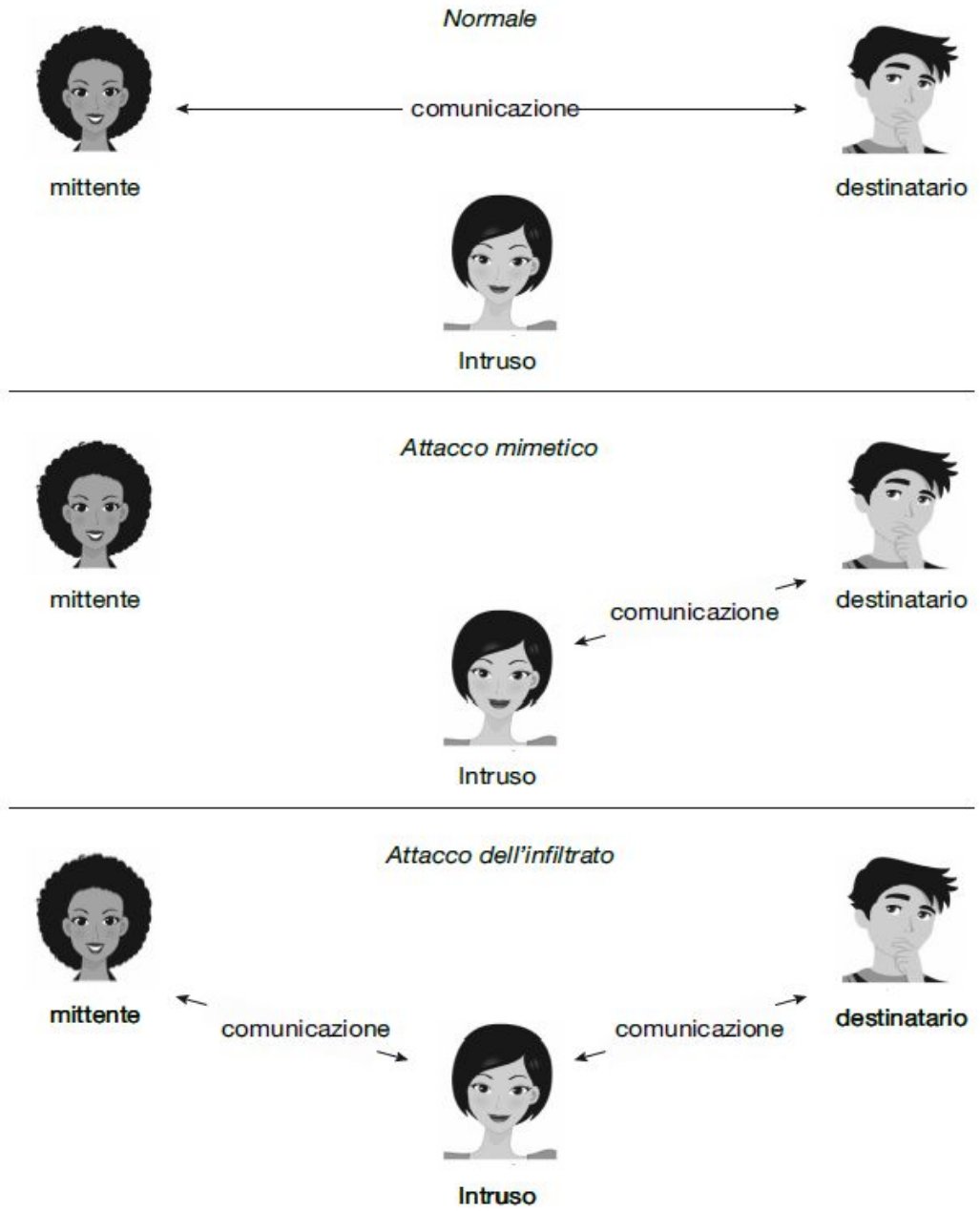


Figura 16.6 Attacchi comuni alla sicurezza.[3]

Attacchi denial of service

Attacco **denial-of-service** (**DoS**): non mirano a ottenere informazioni o a sottrarre risorse, bensì a impedire l'uso corretto di un sistema o di una funzionalità.

Due categorie:

1. l'aggressore occupa un numero così alto di risorse di un servizio da bloccarne completamente la funzionalità;

1. il sabotaggio di una rete che ospita un servizio.

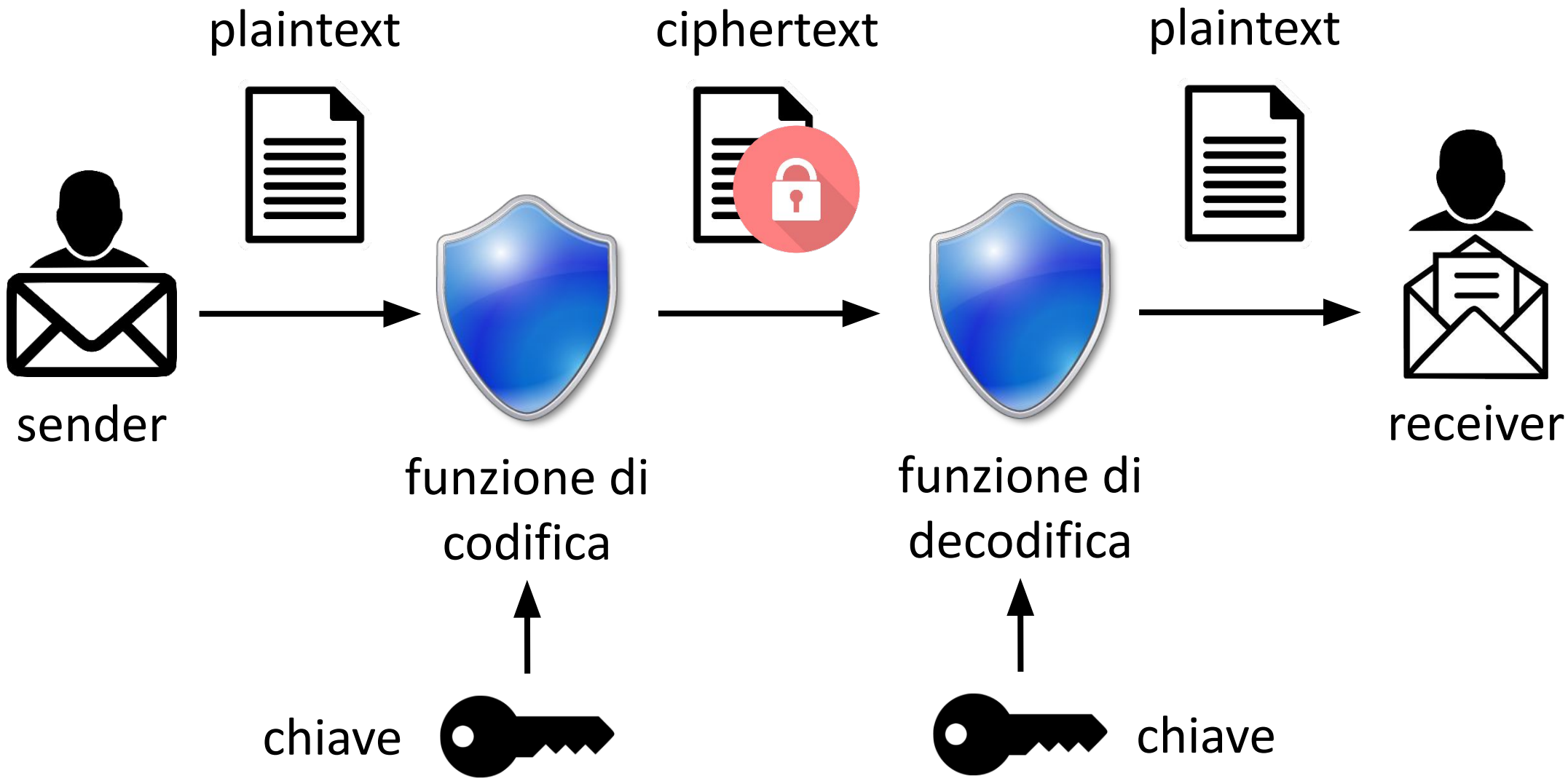
È impossibile impedire gli attacchi denial-of-service, poiché essi sfruttano gli stessi meccanismi del funzionamento normale.
Ancor peggio: **attacchi denial-of-service distribuiti** (*distributed denial-of-service,* **DDOS***)*

Crittografia

La crittografia è la scienza che studia le tecniche e le metodologie per cifrare (codificare) un testo in chiaro *(plaintext)*, al fine di produrre un testo cifrato *(ciphertext)* comprensibile solo ad un destinatario legittimo *(receiver)*

Il receiver deve possedere l'informazione sufficiente (*chiave*) per decifrare il testo cifrato, recuperando così il testo in chiaro

Sistema Crittografico



Chiavi

La **crittografia** moderna si fonda su codici segreti, chiamati **chiavi**, che si distribuiscono selettivamente ai calcolatori di una rete e si usano per elaborare i messaggi.

La crittografia permette al destinatario di un messaggio di verificare che il messaggio sia stato creato da un calcolatore che possiede una certa chiave.

- Cifratura
- Cifratura simmetrica
- Cifratura asimmetrica
- Autenticazione

Cifratura

La **cifratura dei messaggi**, come si sa, è una pratica antica; alcuni algoritmi di cifratura risalgono all'antichità.

Un **algoritmo di cifratura** permette al mittente di un messaggio di imporre che solo un calcolatore che possiede una certa chiave possa leggere il messaggio.

La **cifratura** delimita l'insieme di coloro i quali ricevono informazioni, mentre l'**autenticazione** circoscrive il dominio di chi le trasmette.

La **cifratura simmetrica** richiede una chiave condivisa, mentre la **cifratura asimmetrica** è effettuata con una chiave pubblica e una chiave privata.

L'uso combinato dell'**autenticazione** e delle **funzioni hash** permette di verificare che i dati non abbiano subìto modifiche.

Cifratura simmetrica

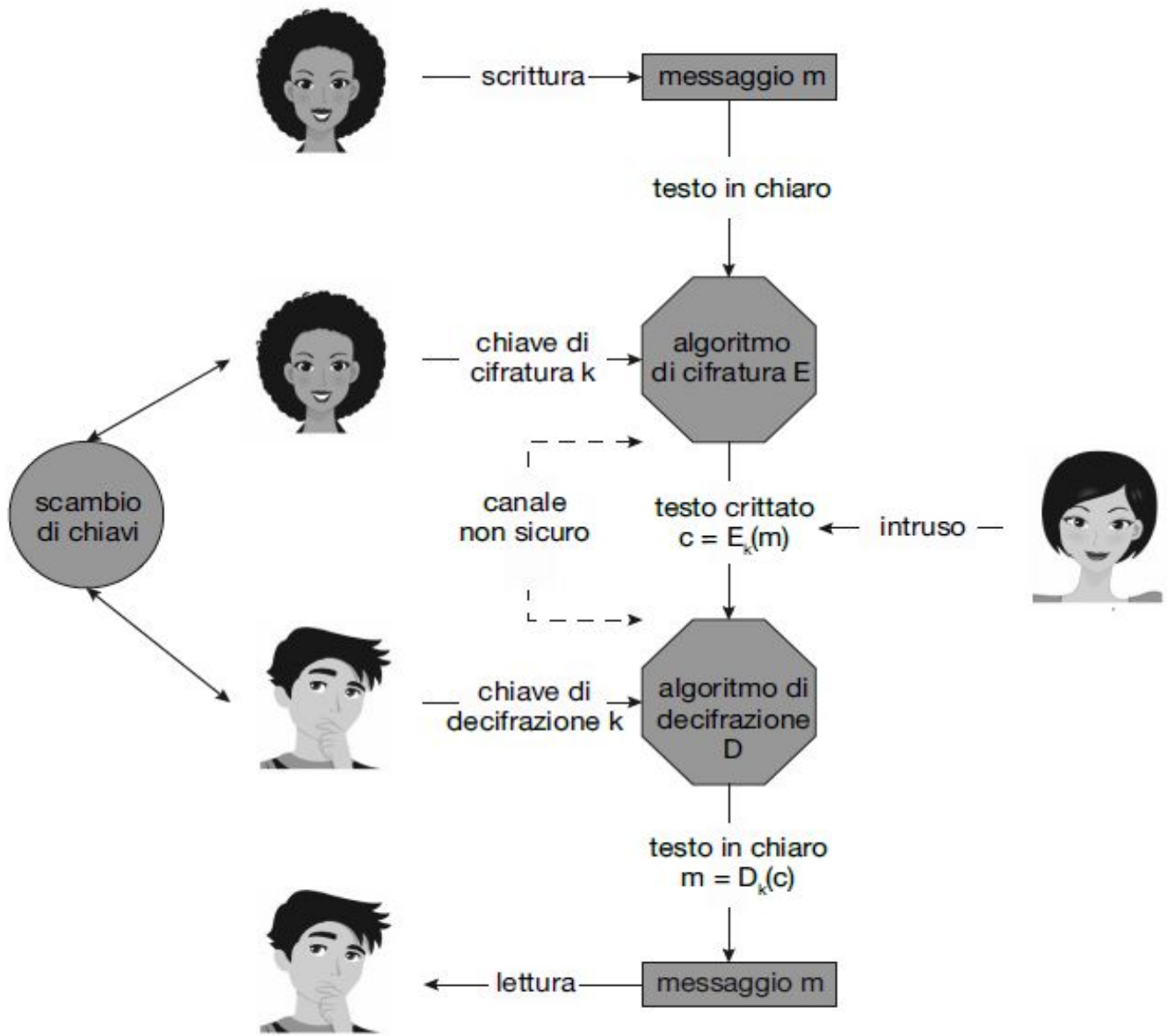


Figura 16.7 Comunicazione sicura su un canale non sicuro.[4]

Cifratura a chiave pubblica

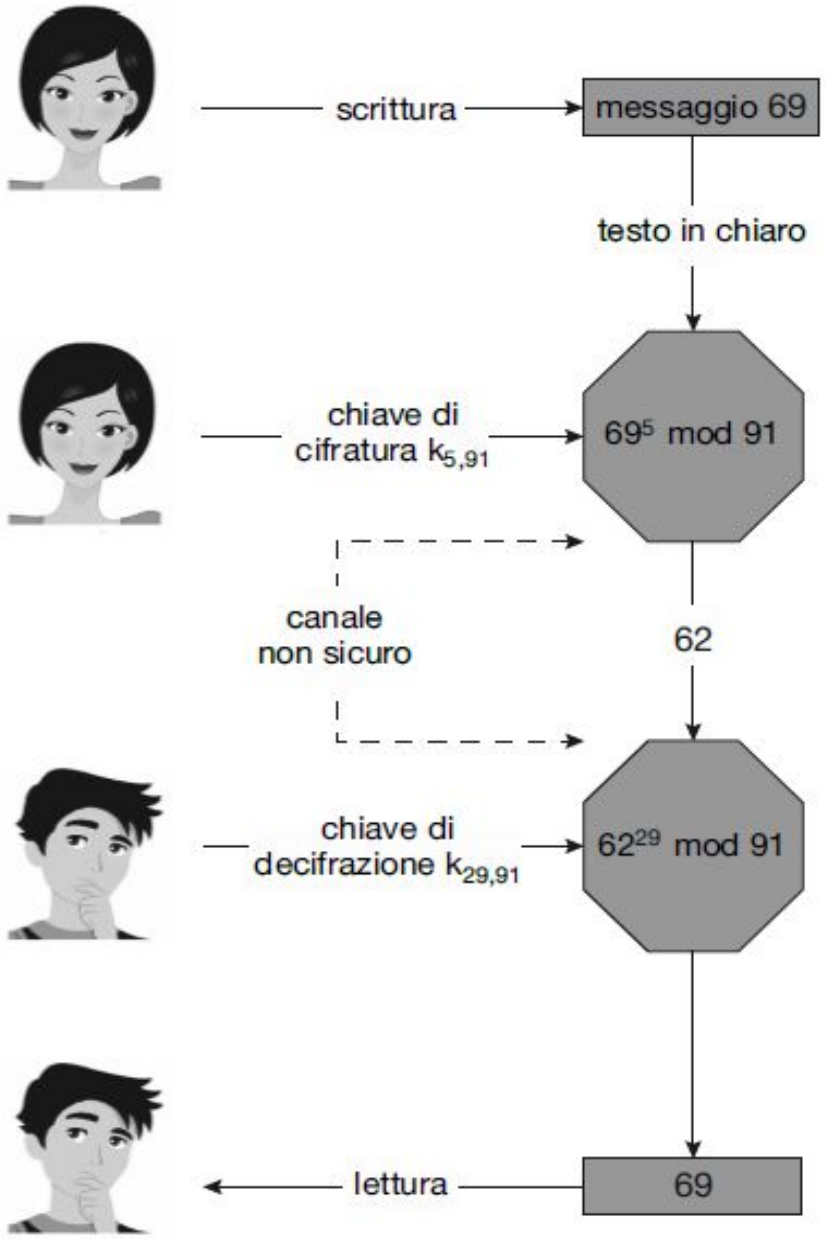


Figura 16.8 Cifratura e decifrazione per mezzo della crittografia asimmetrica RSA.[5]

Attacco alla cifratura a chiave pubblica

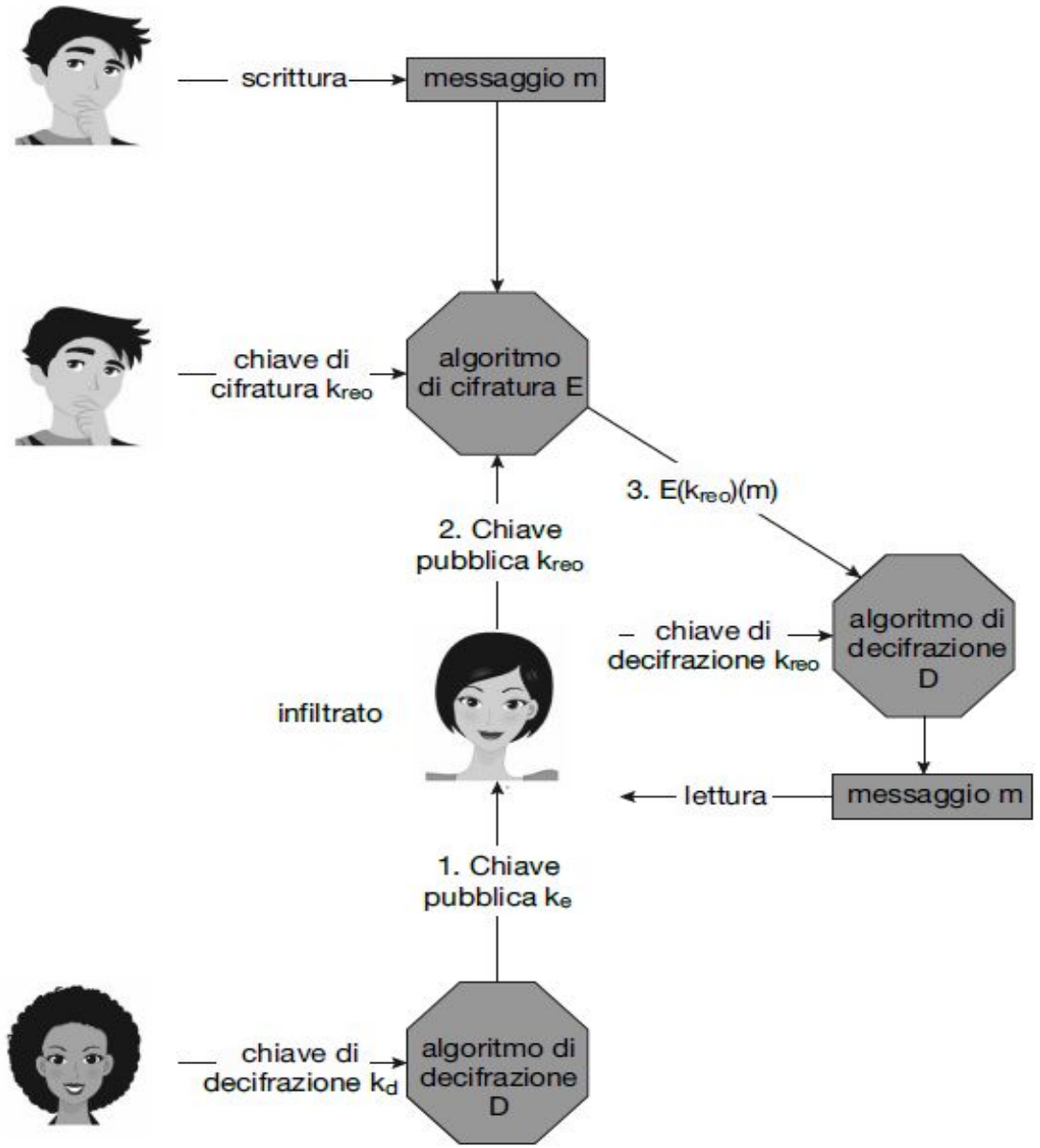


Figura 16.9 Attacco di interposizione alla cifratura asimmetrica.[6]

Autenticazione degli utenti

Normalmente un utente identifica se stesso. Come si può capire se l'identità dichiarata sia autentica?

Autenticazione basata su 3 elementi:

1. oggetti (qualcosa che l'utente ha)
2. conoscenze (qualcosa che l'utente sa)
3. attributo fisico (una caratteristica dell'utente)

Token

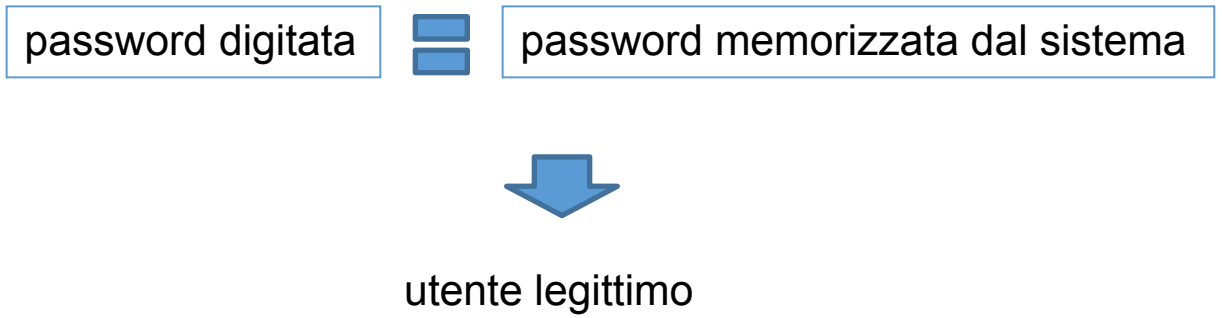
Un token è un oggetto fisico necessario per l'autenticazione

- Spesso sotto forma di dispositivo elettronico portatile di piccole dimensioni, alimentato a batteria (autonomia di qualche anno).
- Alcuni token possono essere collegati ad un PC tramite una porta USB per facilitare lo scambio di dati
- Anche di tipo software, ove le informazioni necessarie risiedono direttamente nel PC dell'utente, o in una app per telefoni

Password

Le **password** si possono considerare un caso particolare di *chiavi* o di *abilitazioni*.

Autenticazione: user ID + password



Tecniche biometriche

- Impronta del palmo e impronta della mano
  - Ingombranti e/o costosi per autenticazione al calcolatore

- Lettori di impronte digitali: disegno formato dalle increspature della pelle sulle dita e convertito in una sequenza di numeri
  - Di solito memorizzano un insieme di sequenze per "adattarsi" alla posizione del dito sulla tavoletta e ad altri fattori

Altri metodi

Ai tradizionali metodi di protezione basati su nome-utente e password, se ne possono affiancare altri:

Password monouso. Esse cambiano da sessione a sessione per evitare attacchi replay.

Autenticazione a due fattori richiede due elementi di autenticazione, per esempio un dispositivo hardware insieme a un PIN di attivazione.

Autenticazione multifattoriale impiega tre o più elementi. I metodi citati riducono fortemente le possibilità di falsificare l'autenticazione.

Misure di sicurezza

I metodi per prevenire o rilevare le violazioni alla sicurezza comprendono:

- Politica di sicurezza
- Sistemi di rilevamento delle intrusioni
- Programmi antivirus
- Auditing e il log delle attività
- Monitoraggio delle chiamate di sistema
- Firma digitale del codice
- Sandbox
- Firewall

Firewall

Un **firewall** può separare una rete in più *domìni*. Uno schema comune considera la rete Internet come *dominio non fidato*; prevede una rete parzialmente fidata, la cosiddetta **zona smilitarizzata** (*demilitarized zone,* **DMZ***),* come *secondo dominio;* e un *terzo dominio* che comprende i calcolatori aziendali.

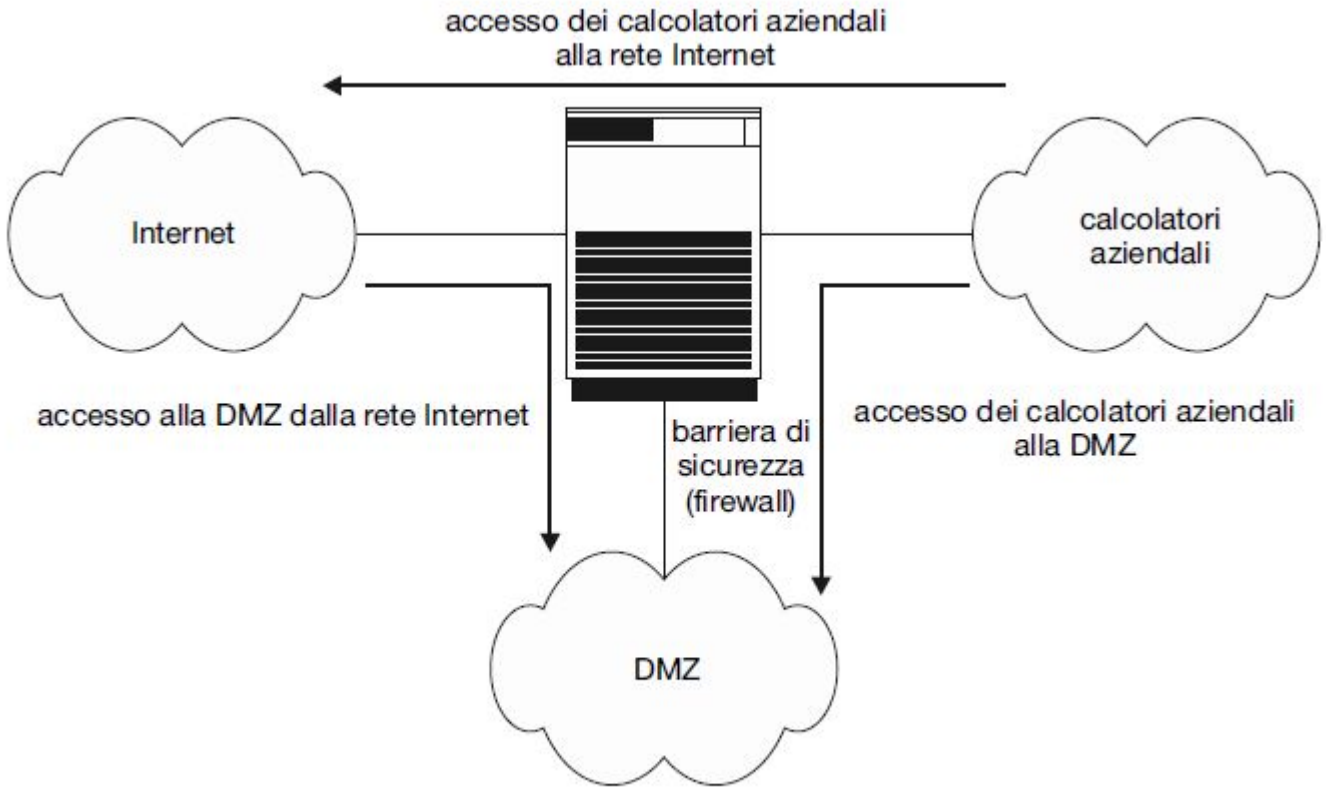


Figura 16.10 Sicurezza di rete con separazione in domìni tramite firewall.

Sicurezza e protezione

La sicurezza misura la fiducia nel fatto che l'integrità di un sistema e dei suoi dati siano preservati

La protezione è l'insieme di meccanismi che controllano l'accesso di processi e utenti alle risorse di un sistema informatico

Protezione

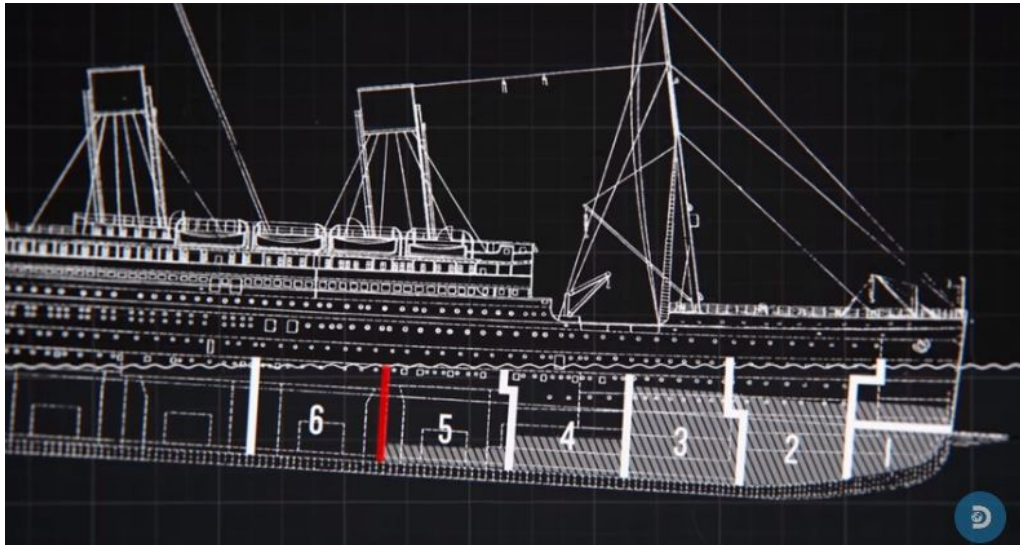
Il ruolo della protezione è quello di offrire un meccanismo d'imposizione di criteri che controllino l'uso delle risorse.

I criteri vanno distinti dai meccanismi.
I meccanismi determinano come qualcosa si debba eseguire; i criteri decidono che cosa si debba fare.

Principi della protezione

minimo privilegio

compartimentazione



https://youtu.be/45B6soUWBoI

Principio del minimo privilegio

principio del minimo privilegio → i programmi, gli utenti (e anche i sistemi) devono ricevere solo i privilegi strettamente necessari per l'esecuzione dei rispettivi compiti

È un principio guida, che nel tempo ha confermato la sua importanza per la protezione

Compartimentazione

compartimentazione → processo di protezione di ogni singolo componente del sistema attraverso l'uso di autorizzazioni specifiche e restrizioni di accesso

È implementata in molte forme, dalle *zone demilitarizzate* (DMZ) a livello di rete, alla *virtualizzazione*

Anelli di protezione

Un modello di separazione dei privilegi utilizzato di frequente è quello degli **anelli di protezione**

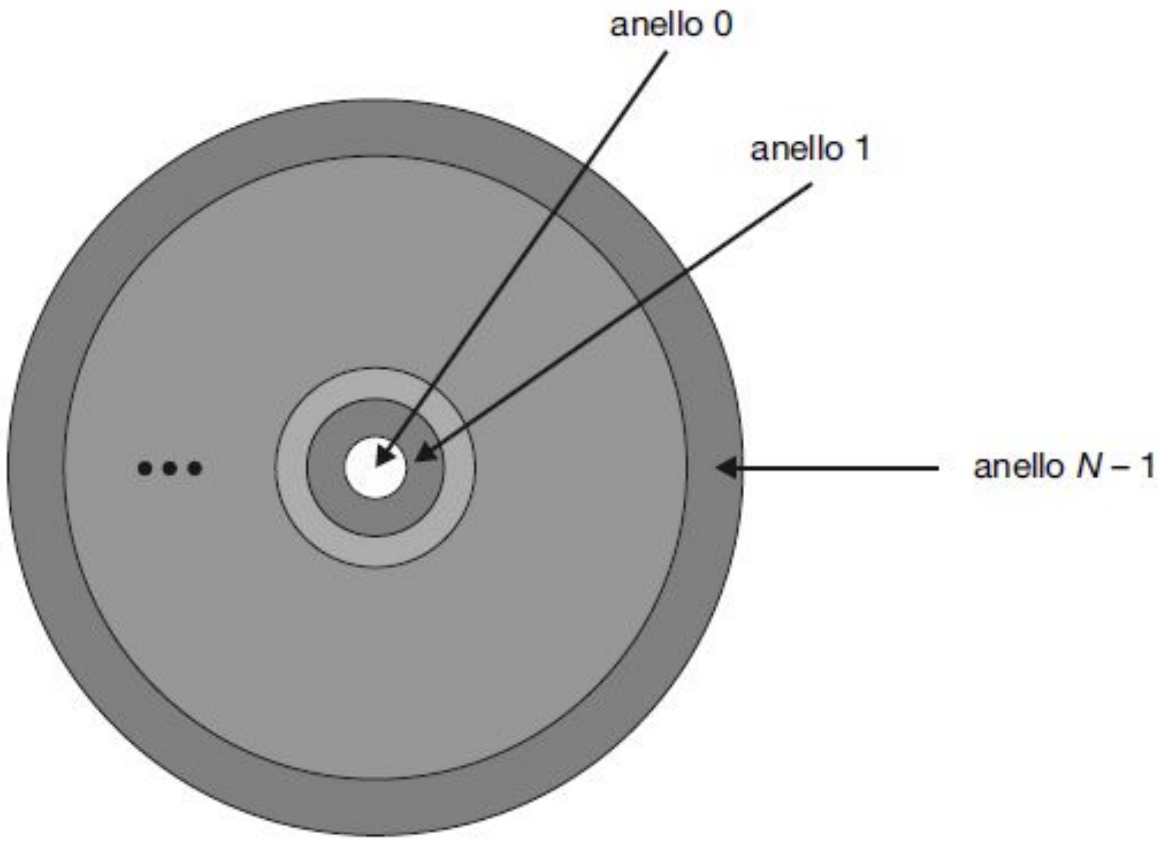


Figura 17.1 Struttura di protezione ad anelli.

TrustZone

La TrustZone (TZ) ha fornito un anello aggiuntivo

Android utilizza la TrustZone in maniera estesa a partire dalla sua versione 5.0

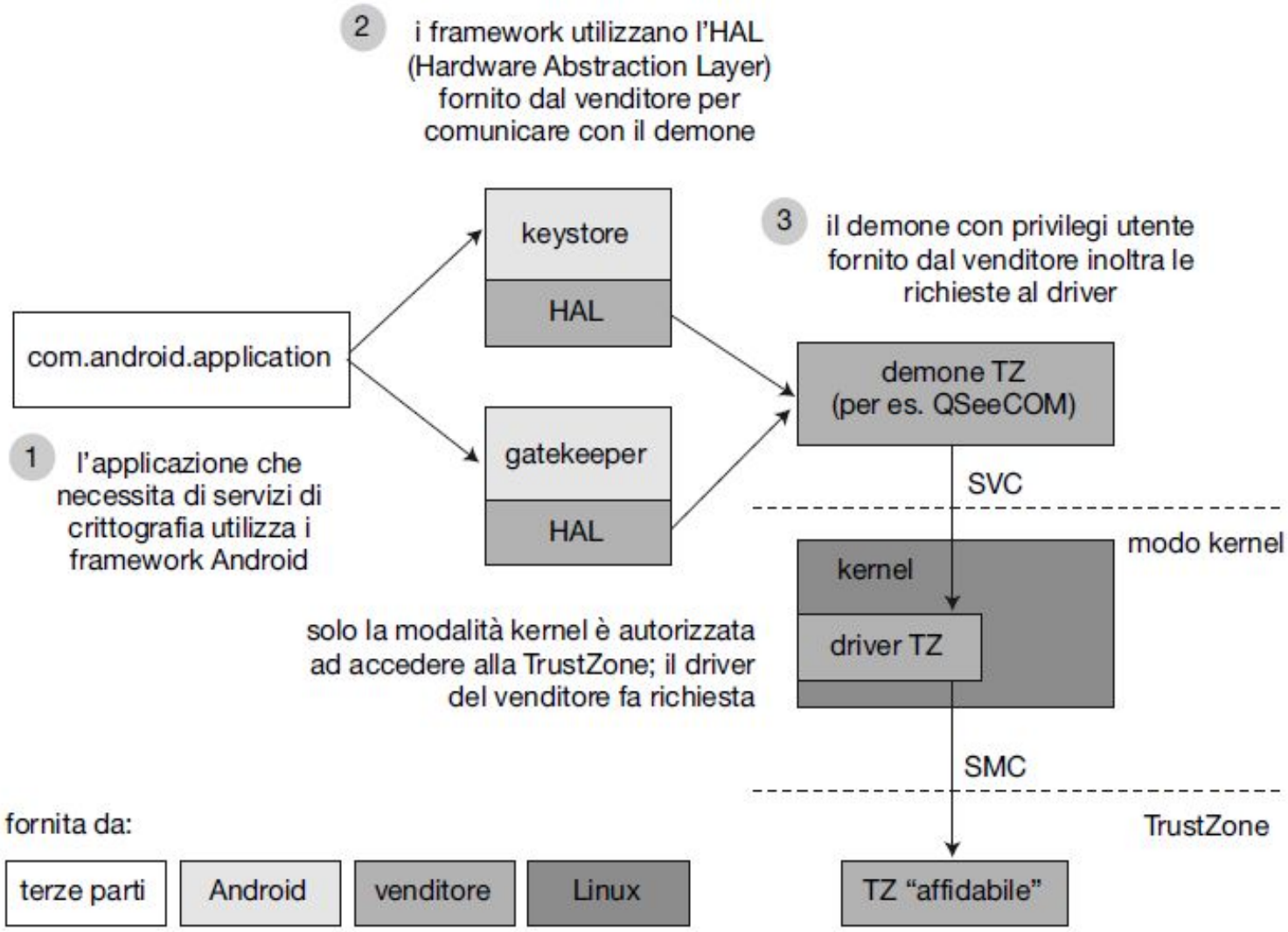


Figura 17.2 Utilizzo della TrustZone in Android.

Livelli di eccezione

Nell'architettura ARMv8 a 64 bit, ARM ha esteso il suo modello per supportare quattro livelli, denominati "**livelli di eccezione**" e numerati da EL0 a EL3

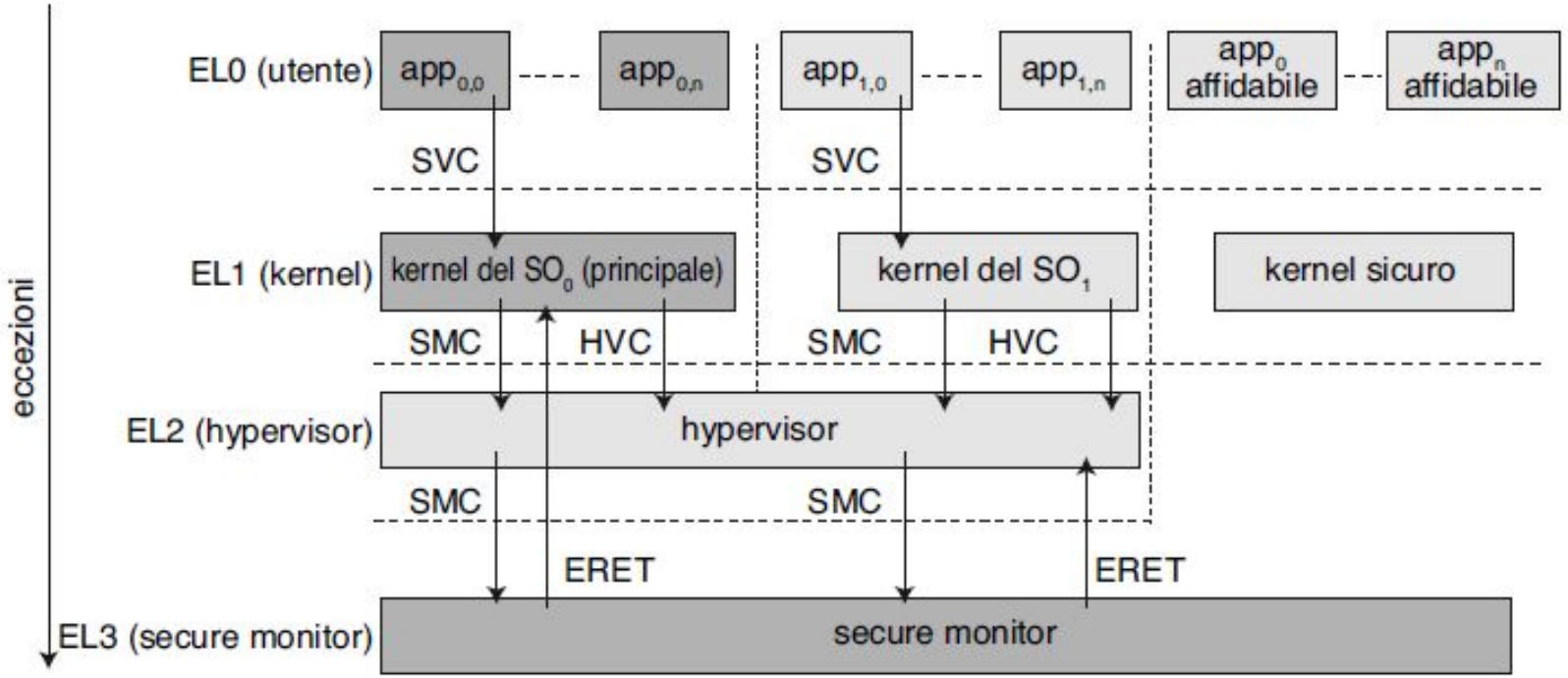


Figura 17.3 Architettura ARM.

Necessità di sapere

Principio della necessità di sapere (*need-to-know-principle)*
utile per limitare i danni che possono essere causati al sistema da un processo difettoso.

Confrontando la politica della **necessità di sapere** con quella del **privilegio minimo** si riscontra che la prima è volta alla politica adottata, mentre la seconda al meccanismo per ottenere questa politica.

Diritti di accesso

- Un **diritto d’accesso** è un permesso per eseguire un’operazione su un *oggetto*.

- Un **dominio** è un insieme di diritti d’accesso.

Domini di protezione

I processi vengono eseguiti in domìni e possono usare tutti i diritti d'accesso del dominio per accedere agli *oggetti* e manipolarli.

Durante il suo ciclo di vita un processo può essere vincolato a un dominio di protezione o può essergli consentito **di passare da un dominio a un altro**.

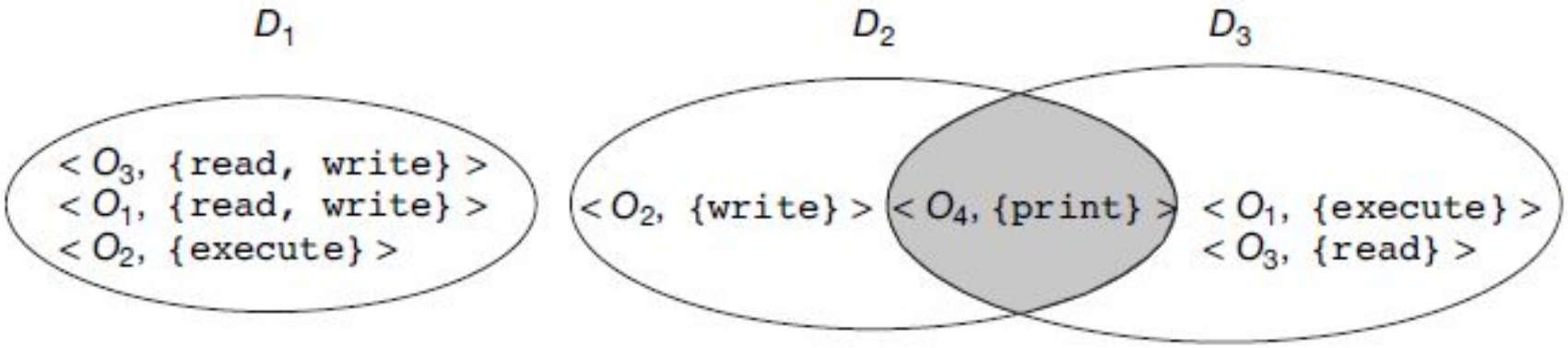


Figura 17.4 Sistema con tre domìni di protezione.

Domini di protezione

Un dominio si può realizzare in diversi modi:

Ogni *utente* può essere un dominio

Ogni *processo* può essere un dominio

Ogni *procedura* può essere un dominio

Matrice d’accesso

· Le righe della matrice rappresentano i *domìni*, e le colonne gli *oggetti*.

· Ciascun elemento della matrice consiste di un insieme di diritti d’accesso.

oggetto / dominio	F_1	F_2	F_3	stampante
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Figura 17.5 Matrice d’accesso.

Matrice d’accesso

Un processo in esecuzione nel dominio D_2 può passare al dominio D_3 oppure al dominio D_4. Un processo del dominio D_4 può passare al dominio D_1, e uno del dominio D_1 può passare al dominio D_2.

oggetto / dominio	F_1	F_2	F_3	stampante	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Figura 17.6 Matrice d’accesso della Figura 17.5 con domini come oggetti.

Matrice d’accesso

Un processo in esecuzione nel dominio D_2 può copiare l’operazione *read* in un elemento qualsiasi associato al file F_2.

Quindi, la matrice d’accesso della Figura 17.7(a) si può modificare nella matrice d’accesso illustrata nella Figura 17.7(b).

dominio \ oggetto	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)

dominio \ oggetto	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

Figura 17.7 Matrice d’accesso con diritti copy.

Matrice d’accesso

Il dominio D_1 *è il proprietario di* F_1 e quindi può aggiungere e cancellare qualsiasi diritto valido nella colonna di F_1.

Così, la matrice d’accesso della Figura 17.8(a) si può modificare nella matrice d’accesso illustrata nella Figura 17.8(b).

oggetto / dominio	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write
D_3	execute		

(a)

oggetto / dominio	F_1	F_2	F_3
D_1	owner execute		
D_2		owner read* write*	read* owner write
D_3		write	write

(b)

Figura 17.8 Matrice d’accesso con diritti owner.

Matrice d’accesso

Confronto tra le due matrici d’accesso delle Figure 17.6 e 17.9.

oggetto / dominio	F_1	F_2	F_3	stampante	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

Figura 17.6 Matrice d’accesso della Figura 17.5 con domìni come oggetti.

oggetto / dominio	F_1	F_2	F_3	stampante	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch control
D_3		read	execute					
D_4	write		write		switch			

Figura 17.9 Matrice d’accesso della Figura 17.6 modificata.

Realizzazione della matrice d’accesso

La matrice d’accesso è sparsa, ossia la maggior parte dei suoi elementi è vuota.

Normalmente si realizza per mezzo di liste d’accesso associate a ciascun *oggetto*, oppure per mezzo di liste di abilitazioni associate a ciascun dominio.

Si può inserire la **protezione dinamica** nel modello della matrice d’accesso considerando i domìni e la stessa matrice d’accesso come *oggetti.*

Lo **schema chiave-serratura** *(lock-key scheme*) rappresenta un compromesso tra le liste d’accesso e le liste di abilitazioni.

Revoca dei diritti di accesso

La **revoca dei diritti d'accesso** in un modello di protezione dinamico è di solito più facile da realizzare con lo schema delle liste d'accesso che con le liste di abilitazioni.

Tra gli schemi che realizzano la revoca delle abilitazioni ci sono i seguenti:

- Riacquisizione
- Puntatori all'indietro
- Riferimento indiretto
- Chiavi

Controllo dell’accesso basato sui ruoli

controllo dell’accesso basato sui ruoli (*role-based access control, RBAC)* è una funzionalità che si basa sui privilegi, cioè il diritto di eseguire una chiamata di sistema o di sfruttare un’opzione di tale chiamata.

Solaris, dalla versione 10, realizza il principio del privilegio minimo attraverso il controllo dell’accesso basato sul ruolo, una forma di matrice d’accesso.

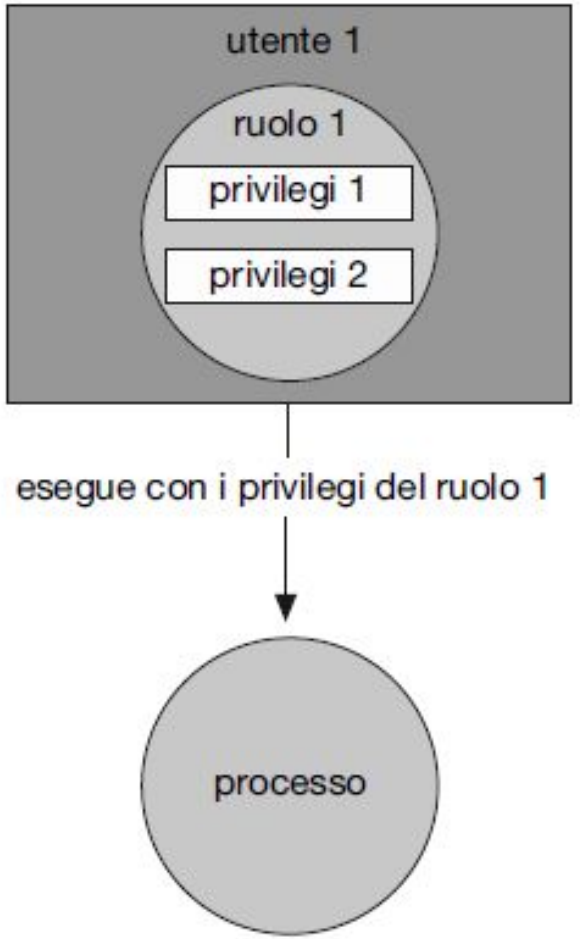


Figura 17.10 Controllo dell’accesso basato sui ruoli in Solaris 10.

Controllo obbligatorio dell'accesso

I sistemi operativi hanno tradizionalmente utilizzato il **controllo discrezionale di accesso** (DAC) come mezzo per limitare l'accesso ai file e agli altri oggetti del sistema.

Un'altra estensione di protezione è il **controllo obbligatorio dell'accesso** (MAC), una forma di imposizione delle politiche di sistema. Il MAC viene applicato come una politica di sistema che nemmeno l'utente root può modificare.

Il cuore del MAC è il concetto di **etichette** → identificatori (di solito una stringa) assegnati a un oggetto (file, dispositivi e altro).

Sistemi basati su abilitazioni

Abilitazioni di Linux

Le abilitazioni di Linux “spezzettano” i poteri della root in aree distinte, ciascuna rappresentata da un bit in una maschera di bit

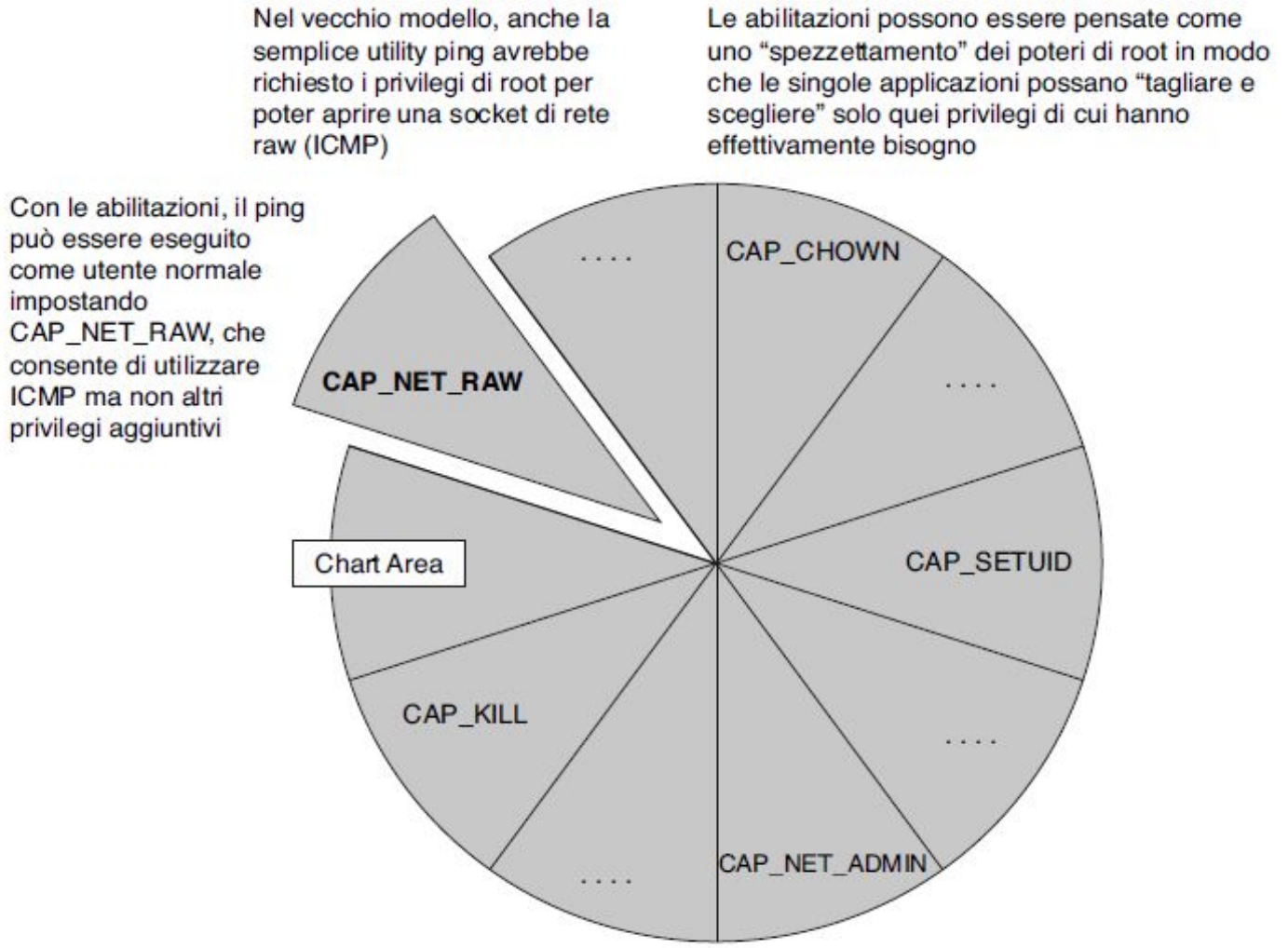


Figura 17.11 Abilitazioni in POSIX.1e.

Autorizzazioni di Darwin

La protezione del sistema di Apple si basa sulle **autorizzazioni** (*entitlement).* Le autorizzazioni sono permessi dichiarativi e consistono in un elenco XML di proprietà che indica quali autorizzazioni sono dichiarate come necessarie dal programma.

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>com.apple.private.kernel.get-kext-info
   <true/>
   <key>com.apple.rootless.kext-management
   <true/>
</dict>
</plist>
```

Figura 17.12 Autorizzazioni di Apple Darwin.

Altri metodi per il miglioramento della protezione

- Protezione dell'integrità del sistema
- Filtraggio delle chiamate di sistema
- Sandboxing
- Firma del codice

```
(version 1)
(deny default)
(allow file-chroot)
(allow file-read-metadata (literal "/var"))
(allow sysctl-read)
(allow mach-per-user-lookup)
(allow mach-lookup)
   (global-name "com.apple.system.logger")
```

Figura 17.13 Un profilo sandbox di un demone MacOS che nega la maggior parte delle operazioni.

Protezione basata sul linguaggio

La **protezione basata sul linguaggio** offre un controllo delle richieste e dei privilegi più selettivo di quello ottenibile con il sistema operativo.

Per esempio, una singola JVM può eseguire molti thread, ognuno in un diverso dominio di protezione. La JVM controlla le richieste di risorse attraverso un raffinato meccanismo di ispezione dello stack e attraverso la sicurezza dei tipi offerta dal linguaggio.

dominio di protezione:	*applet* non fidata	caricatore di URL	interconnessione
permesso della socket:	nessuno	`*.lucent.com:80, connect`	qualsiasi

classe:	`gui:`
	`. . .`
	`get(url);`
	`open(addr);`
	`. . .`
		`get(URL u):`
		`. . .`
		`doPrivileged {`
		`  open('proxy.lucent.com:80');`
		`}`
		`<request u from proxy>`
		`. . .`
			`open(Addr a):`
			`. . .`
			`checkPermission (a, connect);`
			`connect (a);`
			`. . .`

Figura 17.14 Ispezione dello stack.