Sicurezza nei Sistemi Operativi

Sicurezza

- Problema della sicurezza
- Convalida
- Pericoli per i programmi
- Pericoli per il sistema
- Difendere i sistemi
- Scoperta di intrusioni
- Esempio: Windows NT

Il problema della sicurezza

- La sicurezza di un sistema deve considerare l'ambiente esterno che interagisce con il sistema e deve proteggere da:
  - Accessi non autorizzati.
  - Modifiche o distruzioni non autorizzate.
  - Introduzione accidentale di inconsistenze ed errori.

- **Non si può avere un sistema totalmente sicuro** ma occorre fare in modo di limitare al massimo le operazioni non ammesse.

- E' più facile proteggersi da abusi accidentali che da abusi volontari.

Convalida

- Occorre identificare utenti e processi che usano un sistema.
- L'identità degli utenti generalmente prevede l'uso di *password*.

- Le password devono essere tenute segrete.
  - Cambio frequente di password.
  - Uso di password non facili da indovinare.
  - Password non alfabetiche e molto lunghe.
  - Log di tutti i tentativi di accesso.

- Le password spesso devono essere cifrate o possono essere usate solo una volta (monouso) o poche volte e poi cambiate.

- Si usano anche controlli biometrici (impronte, iride, volto, palmo della mano).

Pericoli per i programmi

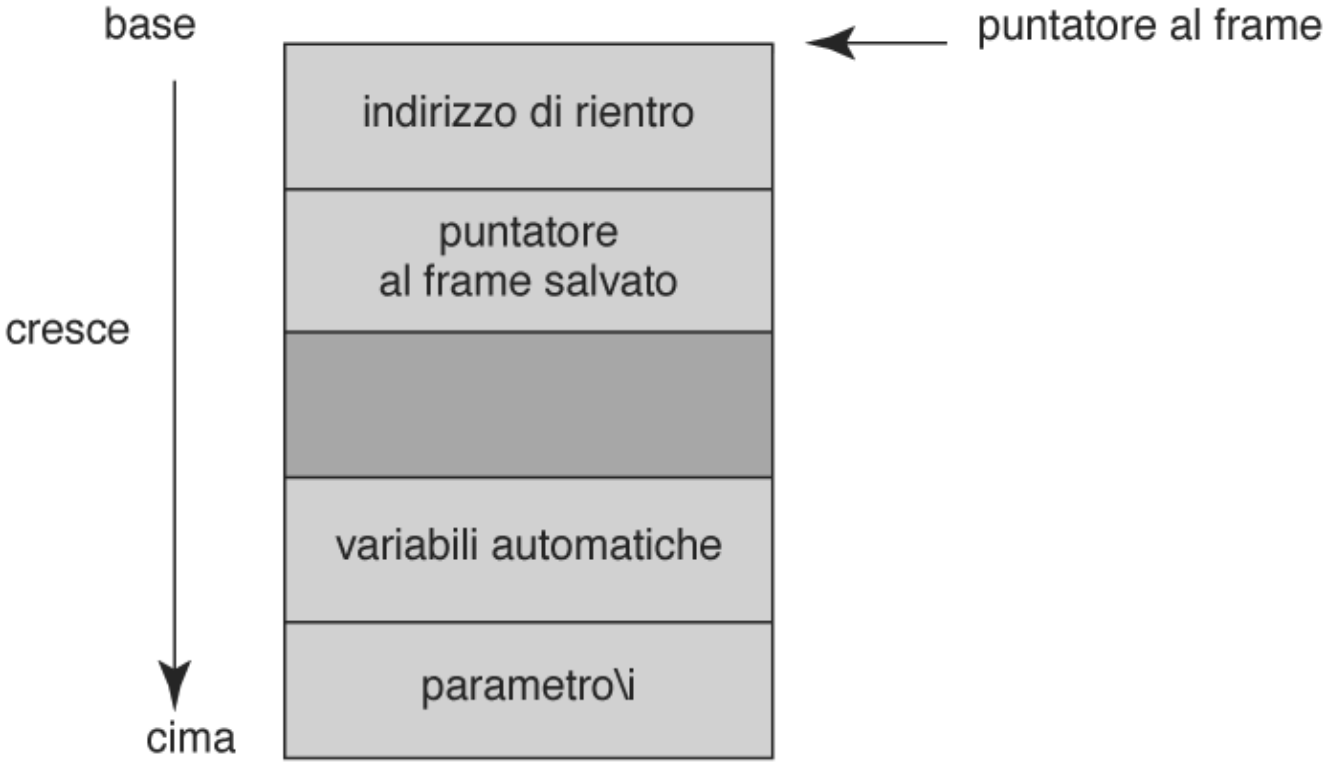
- **Cavallo di Troia (Trojan Horse) e Spyware**
  - Segmenti di codice che abusano dell'ambiente in cui vengono eseguiti.
  - Sfruttano meccanismi che permettono ad un utente di eseguire programmi scritti da altri utenti.
  - Esempio: *simulazione di una sessione di login*.
- **Trabocchetto (Trap Door o Back Door)**
  - Specifica di una user-id o password che supera i normali controlli di sicurezza.
  - Può essere incluso in un compilatore.
- **Stack** e **Buffer Overflow**
  - Sfrutta un errore in un programma che provoca overflow nello stack o nei buffer di memoria.

Programma C potenzialmente soggetto a buffer overflow

```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
   char buffer[BUFFER SIZE];
   if (argc < 2)
        return -1;
   else {
        strcpy(buffer,argv[1]);
        return 0;
   }
}
```

Struttura di una tipica pila (stack frame)

Un attacco di buffer overflow deve cercare di **sostituire l'indirizzo di rientro** nello Stack assegnandolo al programma intruso



STACK

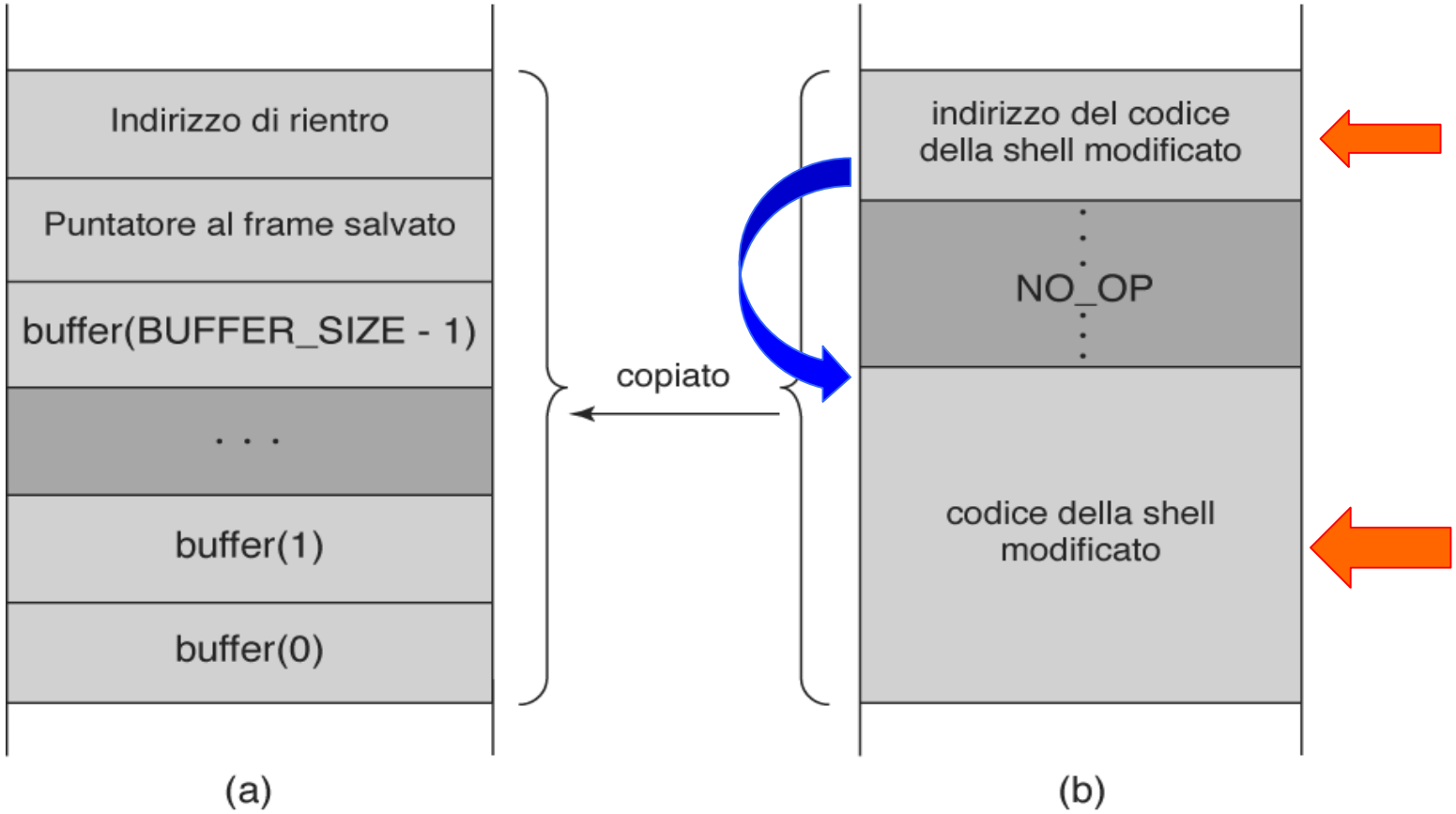
Attacco tramite buffer overflow

Il programma intruso potrebbe contenere questo codice:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
   execvp(“\bin\sh”,“\bin \sh”, NULL);
   return 0;
}
```

Che se eseguito offrirebbe un interprete comandi con i privilegi del programma attaccato.

Attacco tramite buffer overflow



Stack prima dell'attacco

Stack dopo l'attacco

Attacco tramite buffer overflow

- Una soluzione all'attacco con buffer overflow si realizza impedendo che possa essere eseguito codice presente all'interno dello spazio di memoria dello stack.
- Rilevando l'attacco si segnala un'eccezione e il programma termina.
- Si usa nei processori SPARC, AMD e Intel x86 e lo usano sistemi operativi come Solaris, Linux e Windows.
- Si implementa associando un bit ad ogni pagina che indica se la pagina sia eseguibile o meno.

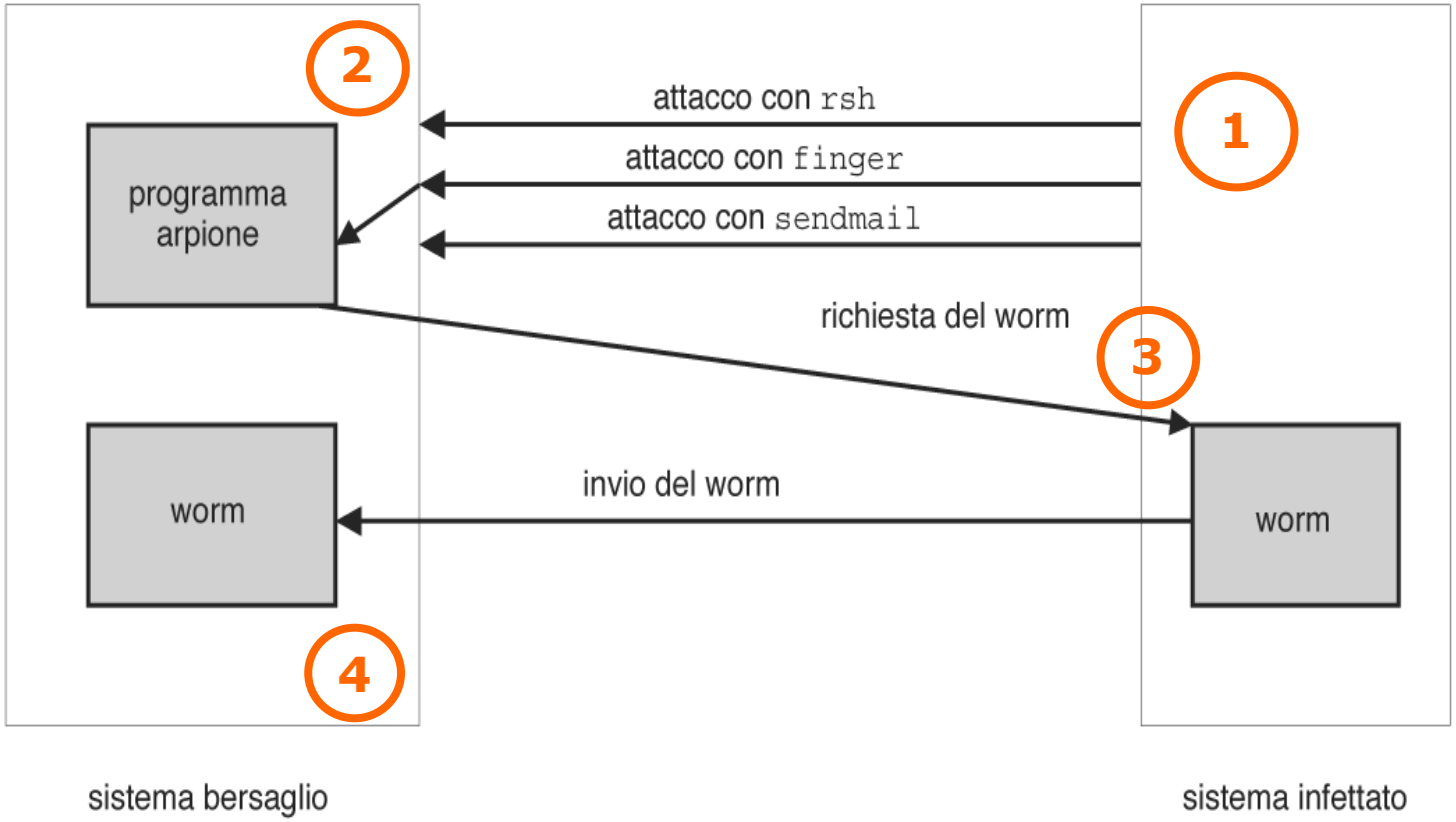
Pericoli per il sistema: Worm

- **Worm**: processo completo che usa il meccanismo di creazione per rigenerarsi e diffondersi nel sistema tramite la rete.

- ***Internet worm***
  - Sfrutta le funzioni di networking di UNIX (accesso remoto) e errori in *finger* e *sendmail*.

- **Caso storico**: Cornell University, 2/11/1988
  - Robert T. Morris lanciò un worm su macchine UNIX connesse in rete.
  - Arpione + programma principale.
  - ***rsh*** per spostarsi su altre macchine senza richiesta di password, ***finger*** e ***sendmail***.

Internet worm di Robert Morris



Pericoli per il sistema: Virus e DOS

- **Virus**: frammento di codice inserito in un programma legittimo.
  - Vengono tipicamente scaricati con programmi pubblici.
  - Diffusione tramite posta elettronica.
  - *Safe computing* e uso di antivirus.

- **Rifiuto di servizio (denial-of-service o DOS)**
  - Sovraccarico del sistema obiettivo in modo tale da impedire che questo possa essere usato utilmente.
  - *Distributed denial-of-service* (DDOS) viene attuato da più siti remoti contemporaneamente.

Monitoraggio dei pericoli

- ***Controllo di sequenze di operazioni sospette*** – ad es., numerosi tentativi di accesso usando password scorrette.

- ***Audit log*** – memorizzazione del tempo dell'utente e del tipo di accesso ad un oggetto; usato per il ripristino e per la definizione di misure di sicurezza.

- ***Scan*** - si controlla periodicamente la presenza di "buchi".
  - Vedi esempi seguenti.

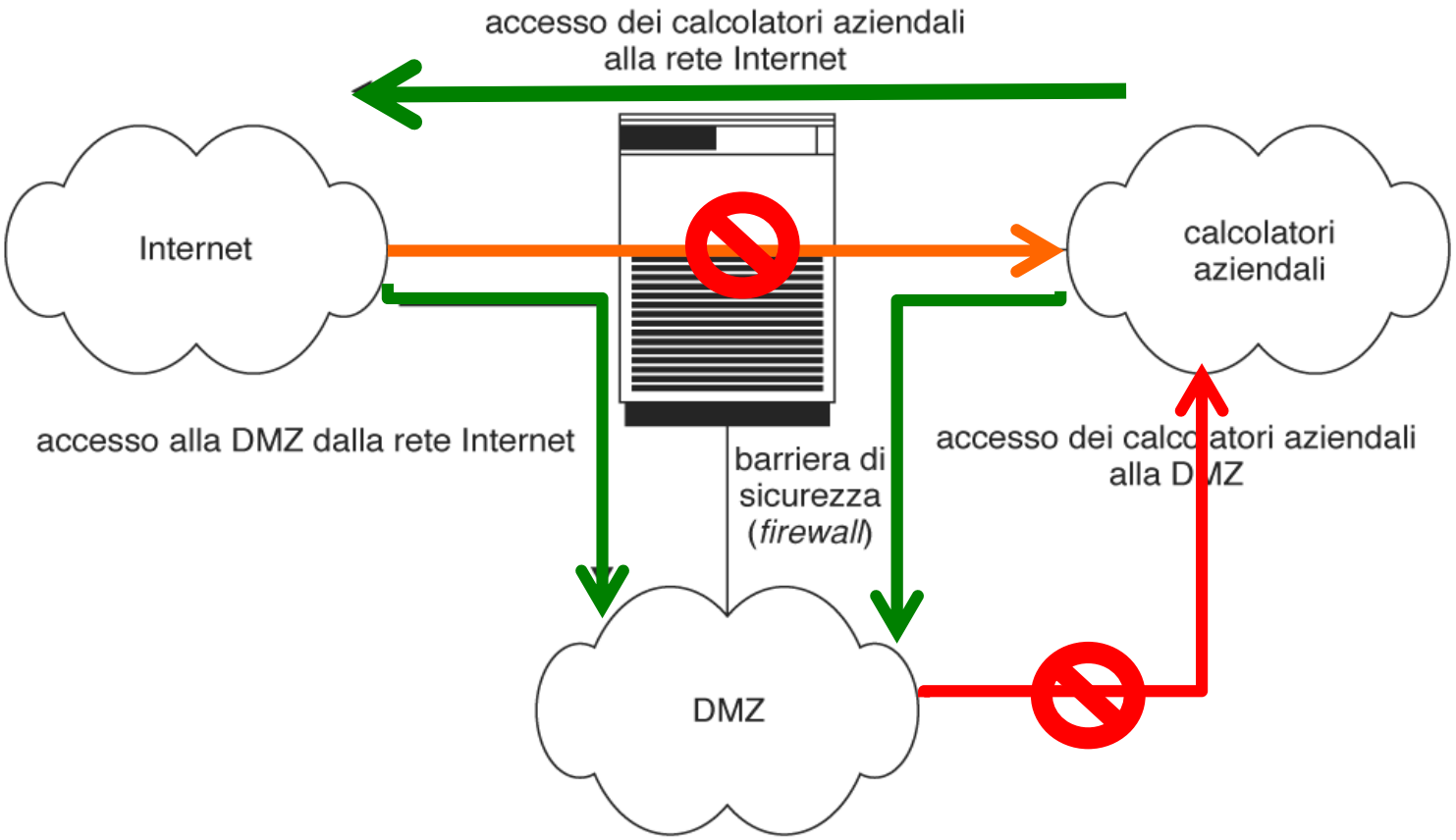
Monitoraggio dei pericoli

- Controllo e monitoraggio di:
  - Password facili da indovinare
  - Programmi con *setuid* non autorizzati
  - Programmi non autorizzati in directory di sistema
  - Processi di durata molto lunga
  - Protezioni di directory improprie
  - Protezioni di file di sistema improprie
  - Elementi pericolosi sui percorsi di ricerca dei file
  - Modifiche ai programmi di sistema (*checksum*).

Firewall

- Un **firewall** è uno strumento di controllo degli accessi che viene posto tra un sistema affidabile ed uno inaffidabile.
- Il firewall limita e/o controlla gli accessi tra questi due tipi di sistemi.
- Si possono controllare accessi legati ad un particolare protocollo; es: finger.
- Firewall hardware e firewall software.

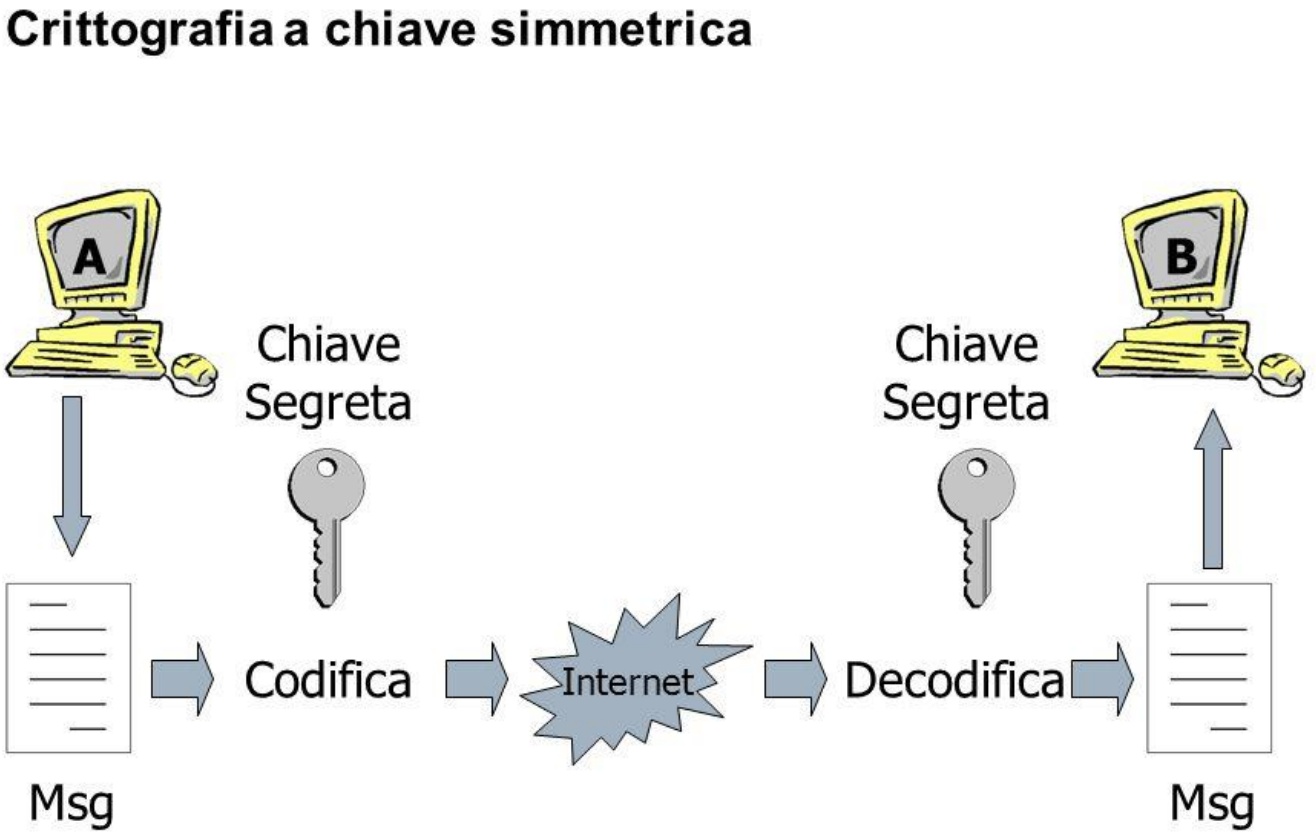
Firewall e DMZ



Scoperta di intrusioni

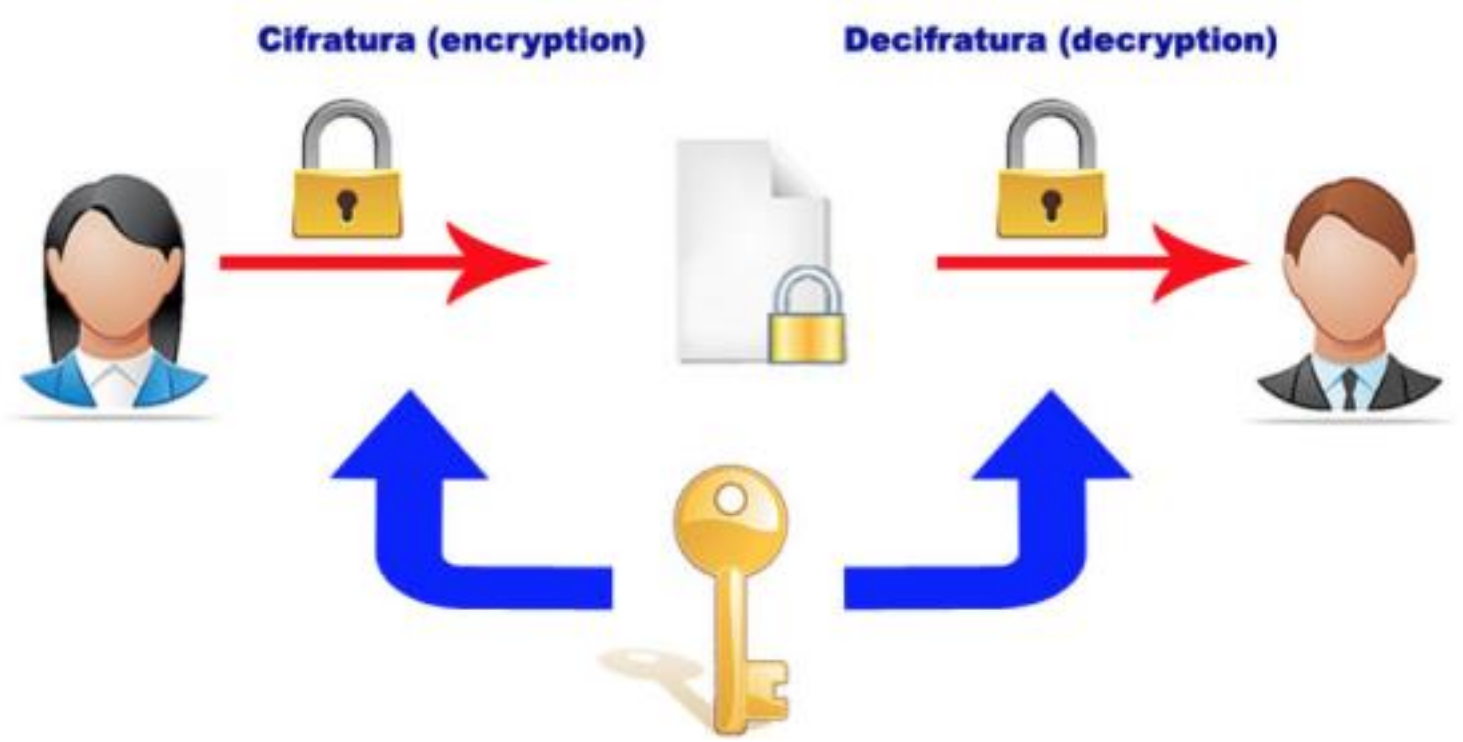
- La scoperta di intrusioni cerca di identificare richieste anomale.
- Metodi di identificazione:
  - Verifica e *logging*.
  - *Tripwire*: software UNIX che controlla se certi file o directory sono stati alterati; ad es. file delle password.
- Monitoraggio delle system call
  - per identificare sequenze anomale di chiamate al sistema.

Crittografia



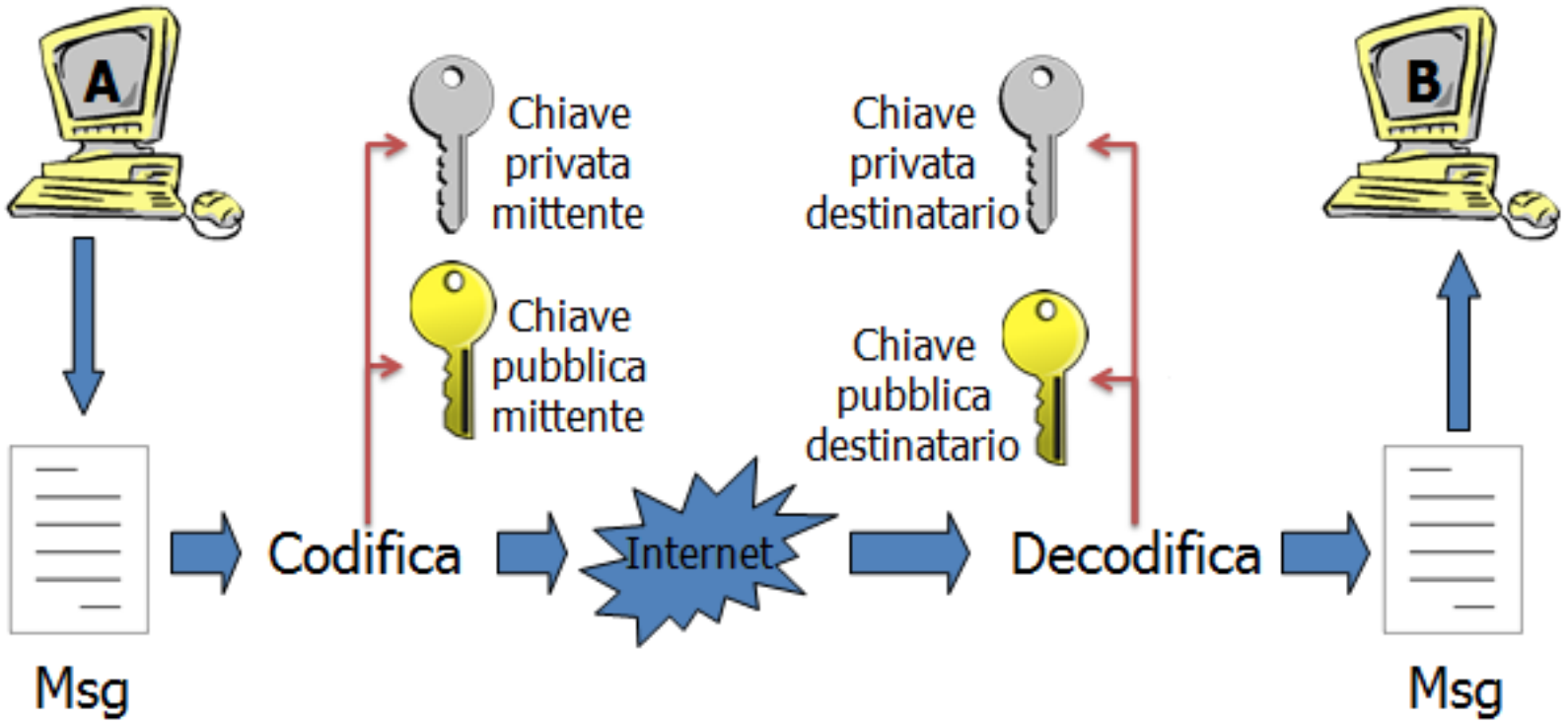
Crittografia

Crittografia a chiave simmetrica – Algoritmo DES



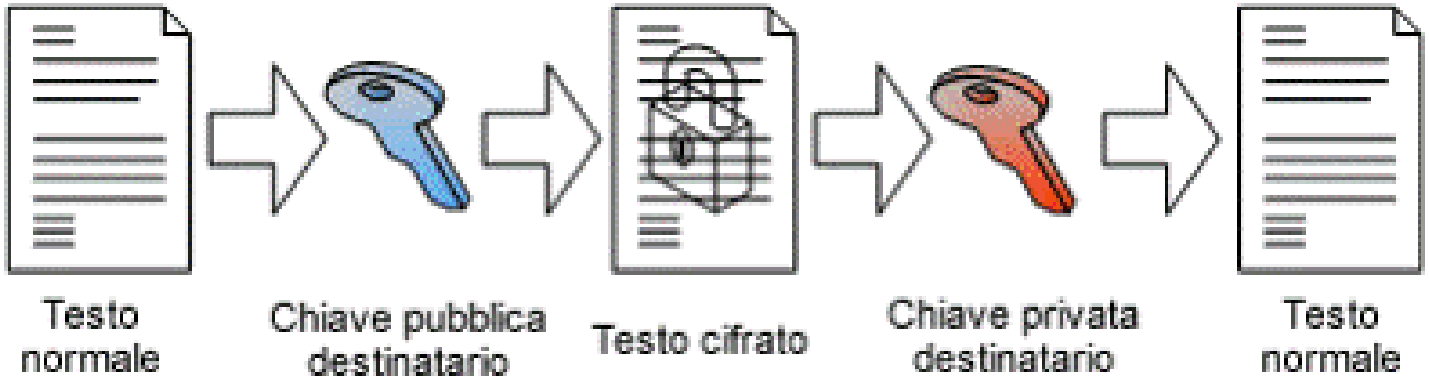
Crittografia

Crittografia a chiave asimmetrica



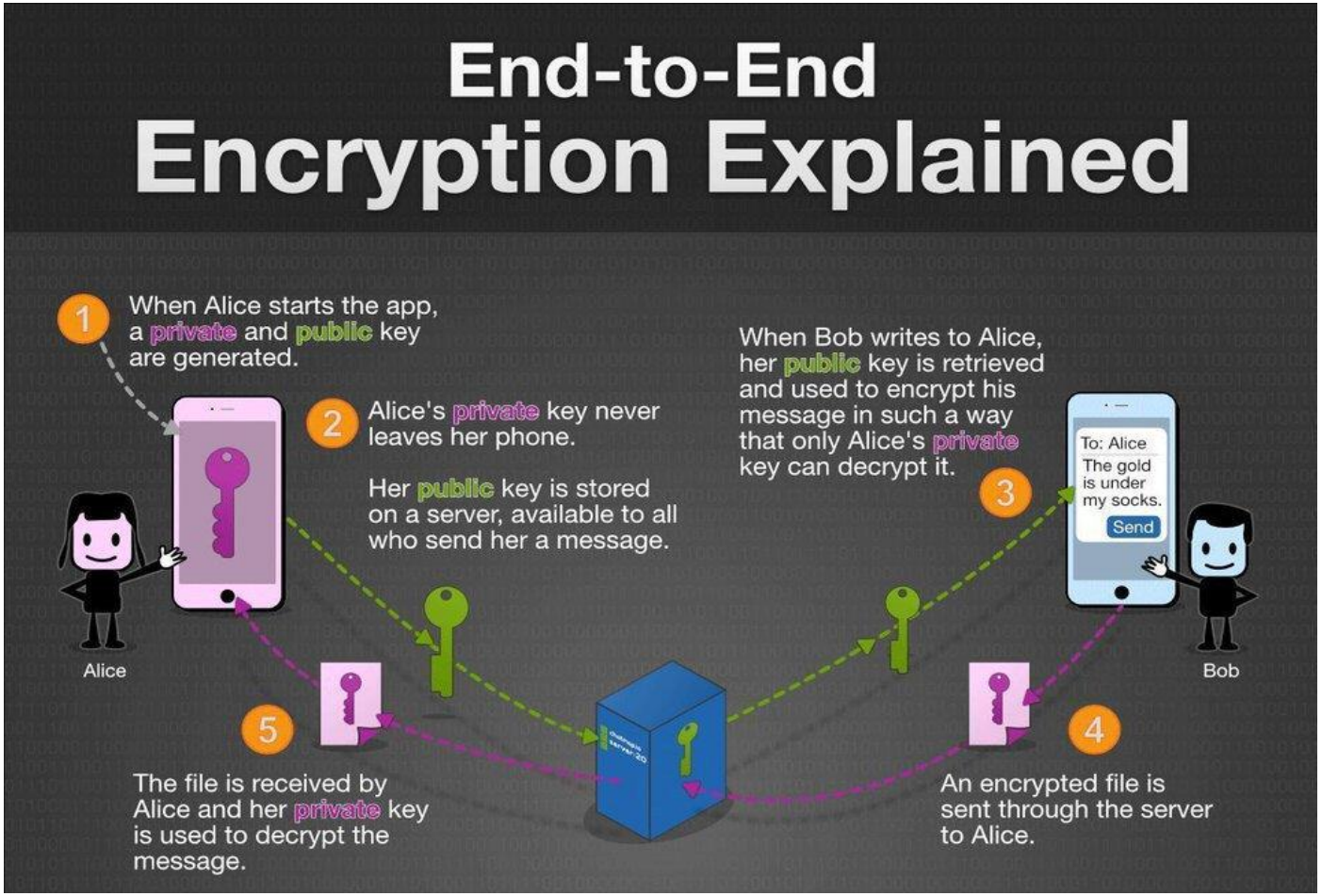
Crittografia

Crittografia a chiave asimmetrica – Algoritmo RSA



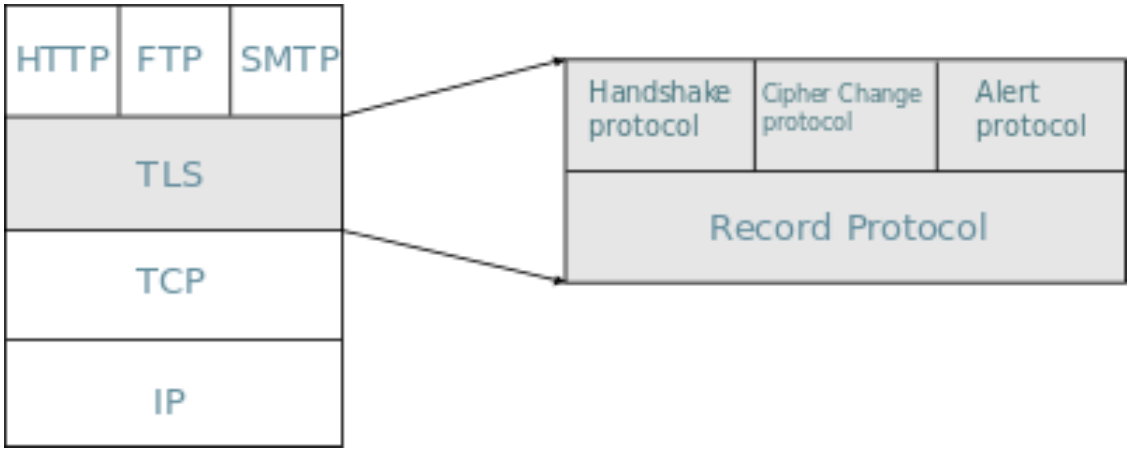
Crittografia

Crittografia di un sistema di messaggistica (es. Whatsapp).



Sicurezza in SSL e TSL

- SSL (Secure Socket Layer) eTLS (Transport Layer Security) sono protocolli sicuri basati su crittografia per il livello di trasporto di Internet.
- Il loro compito è criptare i flussi di comunicazione di dati tra client e server sulla rete.
- TLS è il protocollo sviluppato come successore di SSL. È stato introdotto nel 1999 come una versione estesa e migliorata di SSL 3.0.



Sicurezza in TSL

- Il funzionamento del protocollo TLS può essere suddiviso in tre fasi principali:
  1. Negoziazione fra le parti dell'algoritmo da utilizzare,
  2. Scambio delle chiavi e autenticazione,
  3. Cifratura simmetrica e autenticazione dei messaggi.

- Nelle prime due fasi, il client e il server negoziano il protocollo di cifratura che sarà utilizzato nella comunicazione sicura, il protocollo per lo scambio delle chiavi e l'algoritmo di autenticazione.
- L'algoritmo per lo scambio delle chiavi e quello per l'autenticazione normalmente sono algoritmi a chiave pubblica.
- La comunicazione avviene con chiave simmetrica.

Sicurezza in TSL

- Quando un computer client tramite un browser si mette in contatto con un computer server, quest'ultimo gli invia prima di tutto il suo certificato TSL.
- Questo Certificato TSL dimostra che si tratta di un server autentico e non di una falsa identità.
- Il client verifica la validità del certificato e invia al server un numero casuale, crittografato con la **chiave pubblica** del server.
- Da questo numero casuale il server crea una **chiave di sessione** (*Session Key*) simmetrica che servirà a cifrare la comunicazione.
- Il server invia la **chiave di sessione** al client in forma crittografata, ad esempio usando la chiave pubblica del client.
- Adesso entrambe le parti possono inviare i loro dati in sicurezza usando la chiave di sessione.

Sicurezza in HTTPS

- HTTPS è la sigla di “Hypertext Transfer Protocol Secure”, cioè “Protocollo di trasferimento per ipertesti sicuro”.
- Il protocollo di trasferimento di dati/pagine Web è usato da client web, di solito tramite il browser, e server web per comunicare in modalità cifrata.
- Due Obiettivi:
  - La comunicazione tra client web e server web è **crittografata**, per impedire a terzi non autorizzati di intercettare la comunicazione.
  - Il server web viene **autenticato** inviando, all’inizio della comunicazione, un certificato al client web, che certifica l’affidabilità del dominio. Questa misura è utile per combattere la frode da parte di siti web falsi.

Sicurezza in HTTPS

- Il protocollo HTTP regola come devono essere strutturati i contenuti scambiati tra il client web e il server web.

- Il protocollo di trasporto (TLS) invece indica come i flussi di dati vengono trasferiti tra i computer.

- In particolare, la cifratura del protocollo HTTPS include:
  - la URL richiesta (*la pagina web che è stata richiesta*),
  - i parametri di query,
  - le intestazioni della connessione (*headers*),
  - i cookies (*i quali spesso contengono le informazioni sull'identità dell'utente*).

Classificazione della sicurezza dei computer

- Il Dipartimento della Difesa degli USA definisce 4 categorie di sicurezza per i calcolatori: **A**, **B**, **C**, e **D**.
- **D** – Sicurezza minima (es.: DOS, prime versioni di Windows)
- **C** – Protezione discrezionale con identificazione degli utenti. (UNIX e versioni più recenti di Windows)
  - **C1** protezioni sui singoli o su gruppi di utenti.
  - **C2** controllo di accesso a livello individuale.
- **B** – Tutte le proprietà di **C** e ogni oggetto può contenere il proprio livello di riservatezza. Diviso in **B1**, **B2**, e **B3**.
- **A** – Come **B3** ma con uso di tecniche formali di specifica e verifica del sistema per garantire la sicurezza.

Esempio: Windows XP / Windows 7

- Politiche di sicurezza riconfigurabili tra D e C2.
- La sicurezza è basata sugli account degli utenti ognuno dei quali ha un *security ID*.
- Usa un *security access token* (*uat*) e un *subject* per gestire gli accessi dei programmi eseguiti dagli utenti. Tramite il *subject* (*uat* + *codice eseguibile*) si gestiscono le modalità di accesso ai processi.
- Ogni oggetto in Windows XP ha un *descrittore di sicurezza* (ID del proprietario, lista di accessi e lista di controllo).
- Ad esempio, un file ha un descrittore di sicurezza che indica i permessi di accesso per ogni utente.

Sicurezza nei Sistemi Operativi

- La sicurezza del sistema operativo può essere affrontata e gestita in molti modi:
  - Creazione di account protetti da password/biometria con i privilegi richiesti (gestione degli utenti).
  - Esecuzione di aggiornamenti regolari delle *patch* del sistema operativo.
  - Installazione di controlli software e di antivirus aggiornati.
  - Controllo di tutto il traffico di rete in entrata e in uscita attraverso un firewall.
  - Uso di sistemi di intrusion detection (IDS).
  - Monitoraggio del log del sistema operativo.