

## **SICUREZZA**

# Sicurezza

---

- Il Problema della Sicurezza
- Convalida
- Pericoli per i Programmi
- Pericoli per il Sistema
- Difendere i Sistemi
- Scoperta di Intrusioni
- Cifratura
- 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.
- Non si può avere un sistema sicuro con certezza ma occorre fare in maniera 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 (monouso) o poche volte e poi cambiate.

# Pericoli per i Programmi

---

## ■ Cavallo di Troia

- Segmenti di codice che abusano dell'ambiente in cui vengono eseguiti.
- Sfruttano i meccanismi che permettono ad un utente di eseguire programmi scritti da altri utenti.
- Variante: simulazione di una sessione di login.

## ■ Trabocchetto

- Specifica di una user-id o password che supera i normali controlli di sicurezza.
- Può essere incluso in un compilatore. Es: uova di Word.

## ■ Stack e Buffer Overflow

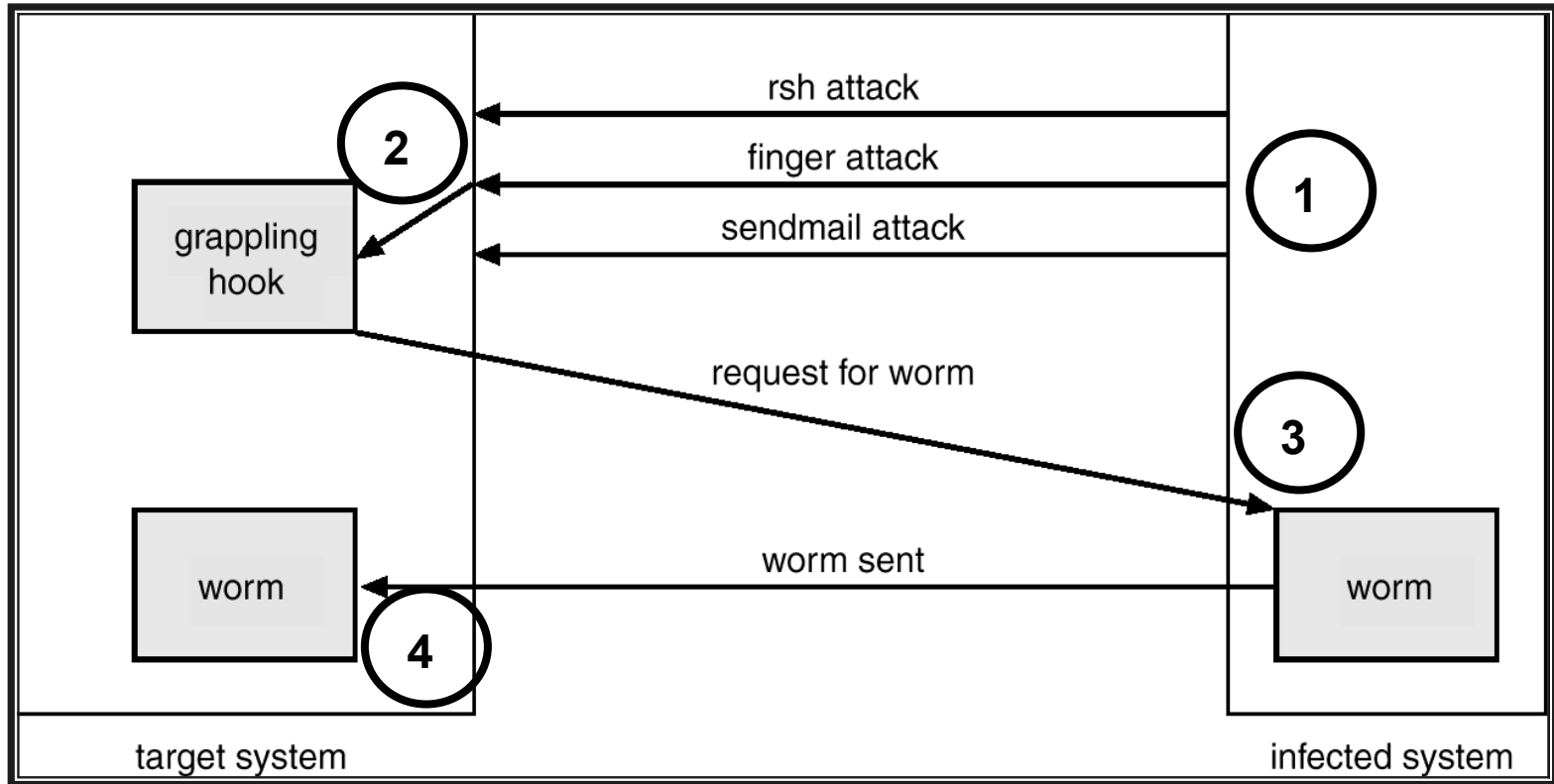
- Sfrutta un errore in un programma che provoca overflow nello stack o nei buffer di memoria. Es: finger di Unix.

# Pericoli per il Sistema : Worm

---

- **Worm:** processo completo che usa il meccanismo di creazione per rigenerarsi e diffondersi nel sistema tramite la rete.
- ***Worm Internet***
  - Sfrutta le funzioni di networking di UNIX (accesso remoto) e errori in *finger* e *sendmail*.
- Cornell University, 2/11/1998
  - R. T. Morris lanciò un worm su macchine UNIX connesse in rete.
  - Rampino + programma principale.
  - *rsh* per spostarsi su altre macchine senza richiesta di password, *finger* e *sendmail*.

# Internet Worm di Morris



# Pericoli per il Sistema : Virus

---

- **Virus** : frammento di codice inserito in un programma legittimo.
  - Presente nei sistemi basati su PC.
  - Vengono scaricati con programmi pubblici o tramite floppy disk.
  - Diffusione tramite posta elettronica.
  - *Safe computing* e uso di antivirus.
  
- **Rifiuto di servizio**
  - Sovraccarico del sistema obiettivo in modo tale da vietare che questo possa essere usato utilmente.

# 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

---

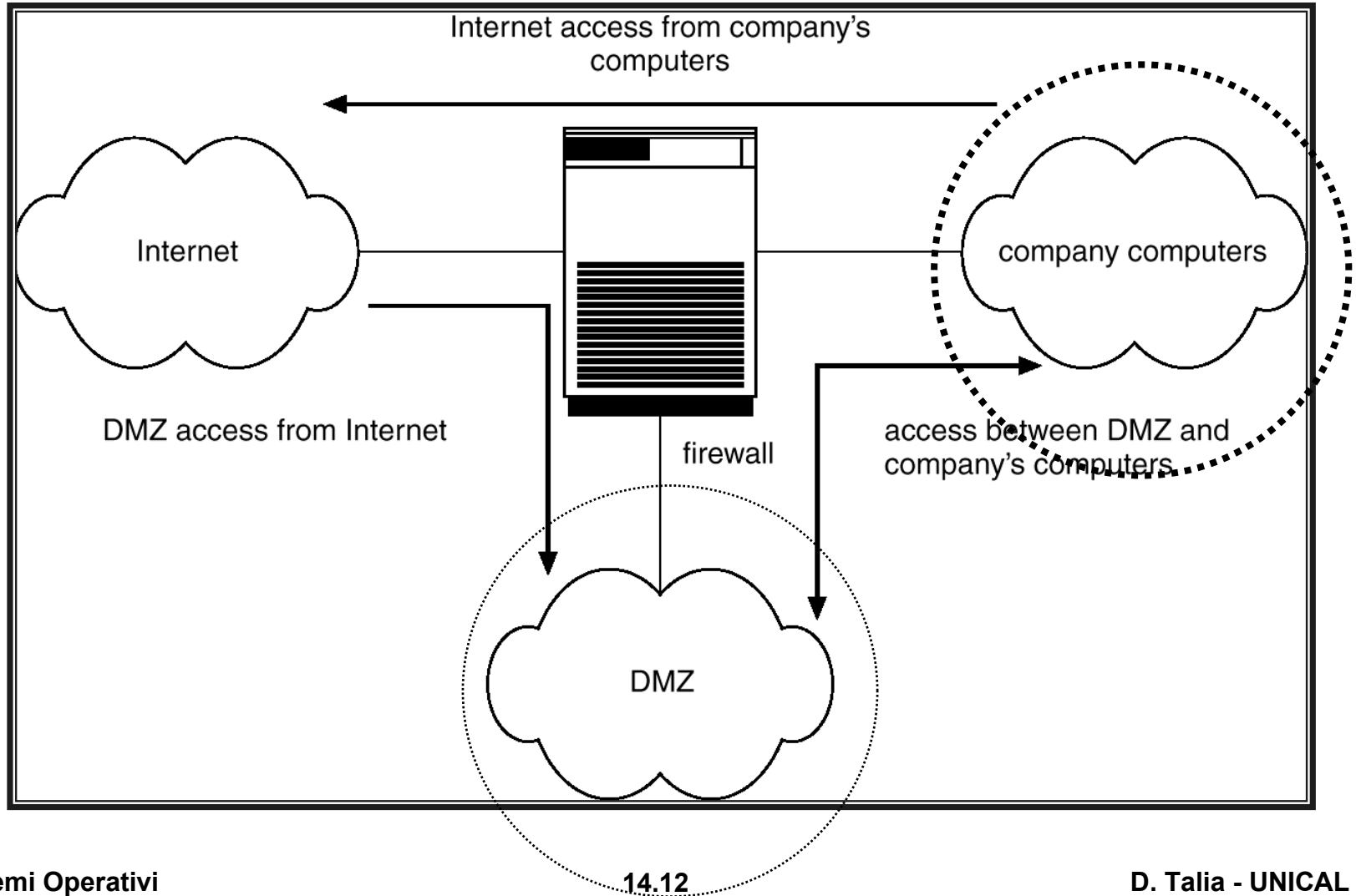
- Controllo e monitoraggio di:
  - Password facili da indovinare
  - Programmi con set-uid non autorizzati
  - Programmi non autorizzati in directory di sistema
  - Processi di durata molto lunga
  - Protezioni di directory improprie
  - Protezioni di file di sistema improprie
  - Elementi pericolosi 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.

# Separazione dei Domini Tramite Firewall



# 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.

# Strutture Dati Derivate da Sequenze di System Call

system call	distance = 1	distance = 2	distance = 3
open	read getrlimit	mmap	mmap close
read	mmap	mmap	open
mmap	mmap open close	open getrlimit	getrlimit mmap
getrlimit	mmap	close	
close			

# Cifratura

---

- La cifratura o crittografia trasforma un testo in chiaro in un testo cifrato.
- Proprietà:
  - Semplice da usare per gli utenti che devono cifrare e decifrare dati.
  - Lo schema non deve dipendere dall'algoritmo di cifratura ma dalla chiave.
  - Estremamente difficile per un intruso decifrare la chiave.
- Il *Data Encryption Standard* sostituisce i caratteri e li scambia di posizione in base ad un algoritmo di cifratura ed uno di decifratura e ad una chiave.
- La chiave è l'elemento critico.

# Cifratura

---

- Cifratura a chiave pubblica basata sul fatto che un utente abbia:
  - Una **chiave pubblica** – usata per cifrare i dati.
  - Una **chiave privata** – usata per decifrare i dati.
- Il sistema di cifratura può essere pubblico senza che questo permetta di decifrare i dati se non si dispone della chiave privata.
  - Schema basato sui fattori primi di un numero (che non sono facili da identificare)

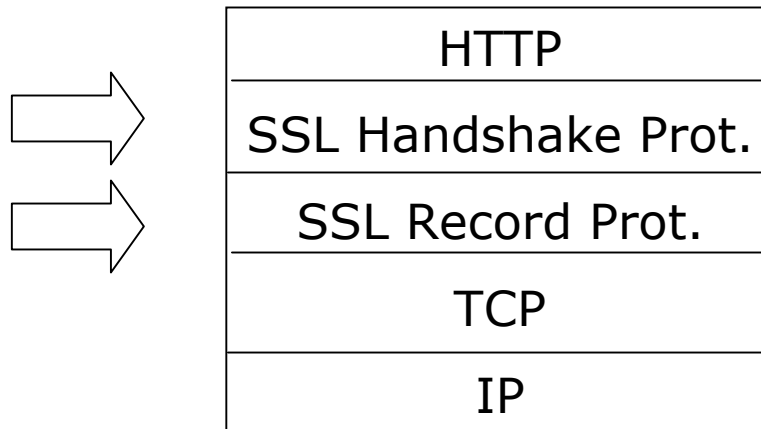
# Esempio di Cifratura - SSL

---

- **SSL** – *Secure Socket Layer*.
- Protocollo basato su cifratura che viene usato nelle comunicazioni client/server per evitare intrusioni, manomissioni e falsificazione dei messaggi.
- Usato tra web server e browser per comunicazioni sicure (es: commercio elettronico con carte di credito)
- Il server viene verificato con un **certificato**.
- Le comunicazioni usano uno schema di cifratura a chiave simmetrica o segreta dopo un handshake per definire una chiave simmetrica.

# Esempio di Cifratura - SSL

- **SSL** è un protocollo non proprietario proposto da Netscape basato su due livelli:
  - *SSL Handshake protocol* : per l'autenticazione
  - *SSL Record protocol* : per lo scambio dei pacchetti cifrati.



# Classificazione della Sicurezza dei Computer

---

- Il Dipartimento della Difesa degli USA 4 categorie di sicurezza per i calcolatori: **A**, **B**, **C**, e **D**.
- **D** – Sicurezza minima. (DOS, Windows)
- **C** – Protezione discrezionale con identificazione degli utenti. (UNIX)
  - **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.

# Esemio: Windows NT

---

- 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 *subject* per gestire gli accessi dei programmi eseguiti dagli utenti. Tramite il *subject* si danno modalità di accesso ai processi.
- Ogni oggetto in Windows NT 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.

# Domande

---

- Quali metodi si possono usare per evitare che le password degli utenti di un sistema non siano scoperte ?
- Discutere le differenze principali tra worm e virus.
- Descrivere il funzionamento del protocollo SSL nel caso venga usato in una sessione verso un server di posta elettronica.
- Quale livello di sicurezza tra quelli definiti dal Dipartimento della Difesa sarebbe opportuno in un ufficio pubblico che contiene dati riservati ?