# Attacchi

H1 # Introduzione

H2 > Un **attacco** a un sistema è ogni atto *malevolo* contro un sistema o un complesso di sistemi

In un attacco si distinguono le seguenti componenti:

- **goal**
- **sotto-goal**
- **attività**
- **eventi**
- **conseguenze**
- **impatto**

## Goal

> Il danno causato al sistema

H4

> Esempio:
>
> rubare denaro, rubare servizi, etc.

## Sotto-goal

> *Obiettivi intermedi* necessari a raggiungere il **goal**

H4

Esempio:

ottenere privilegi, autorizzazioni etc.

## Attività

Le **attività** che l'attaccante deve svolgere per raggiungere i **sottogoal**

H4

Esempio:

usare credenziali rubate, invadere la rete con pacchetti malformati, etc.

## Eventi

L'occorrenza di attacchi causati dalle **attività**

H4

Esempio:

ottenere un accesso improprio, fermare un sistema etc.

## Conseguenze

Le **conseguenze** del verificarsi dell'**evento**

H4

Esempio:

incorrettezza estratti conto, etc.

### Impatto

> Effetti di business

H4

> [Esempio:]
>
> offuscare la reputazione di un'azienda, perdite di denaro, etc.

# Tipi di attacco

Gli attacchi possono avvenire in fasi diverse del ciclo di vita di un SW.

H2

- **Progettazione**: mentre si sta progettando il SW
- **Implementazione**: mentre si sta scrivendo il SW
- **Operazione**: dopo che l'applicazione è in produzione

# Attacchi in fase di progettazione

## H3 Man-in-the-middle attack

H2

> Si verifica quando un attaccante intercetta una trasmissione di rete tra due host e si spaccia per una delle due parti coinvolte nella transazione.

### Difesa

- Usare *intensivamente* ***tecniche crittografiche***
- Usare *session checksum*

H4

## H3 Race condition attack

> Si verifica quando una operazione di controllo apre una finestra temporale durante cui l'attaccante può compromettere la sicurezza. La finestra temporale viene aperta perché l'output di un processo è in attesa del terminare di altri eventi (**race condition**)

#### Difesa

- Evitare operazioni **non atomiche**

## Replay attack

> Si verifica quando un attaccante riesce a catturare o ottenere il record di un'intera transazione tra un programma client ed un server; egli ha infatti la possibilità di riprodurre la transazione a scopo sovversivo

#### Difesa

- Crittografia
- Introduzione di *sequence identifier* per differire ogni transazione

## Sniffer attack

> Si verifica quando un programma malevolo silenziosamente memorizza i dati trasmessi tra gli host della rete, che possono essere anche dati sensibili

#### Difesa

- Come sistemista di rete
  - attenta configurazione della rete
  - uso di switch
- Come programmatore
  - crittografia

### Session hijacking attack

> Si verifica quando un attaccante prende il controllo di una connessione già stabilita sfruttando le debolezze del protocollo TCP/IP

#### Difesa

- È difficile difendersi dal punto di vista del SW
- La crittografia aiuta
- Un sistema di login ben fatto aiuta a riconoscere l'attacco dopo averlo subito

### Session killing attack

> Si verifica quando un attaccante è in grado di inviare ad una delle due parti della comunicazione un messaggio TCP *packet reset*, che chiude prematuramente la comunicazione

#### Difesa

- È difficile difendersi dal punto di vista del SW

# Attacchi in fase di implementazione

### Buffer overflow attack

> Si verifica quando un attaccante causa un overflow del buffer e forza il programma ad eseguire comandi non autorizzati

Per poter subire questo attacco, l'applicazione non deve essere in grado di effettuare un adeguato *bounds checking* e accetta più caratteri di quanto sia possibile memorizzarne nel buffer.

### Difesa

- Uso di linguaggi che effettuano *bouds checking* (Java)
- Evitare di leggere stringhe di testo di lunghezza indeterminata in buffer di lunghezza fissa

H4

## Back door attack

H3

> Si verifica quando uno sviluppatore introduce volutamente all'interno del programma del codice che permetterà di bypassare il controllo d'accesso ad altri. Tale codice è detto **back door**

### Difesa

- Controllare l'assenza di **back door** nel codice

## Parsing error attack

H4
H3

> Si verifica quando un programma accetta input da utenti remoti senza fare *parsing* opportuno dell'input

### Difesa

- Eseguire test dei dati in input

# Attacchi in fase di operazione

H4

## Denial of service attack

H3

H2

> Si verifica quando un'applicazione o un sistema è reso inutilizzabile per via di una cascata di richieste di servizio, o un continuo flusso di input, e quindi un utente legittimo si vede negato il servizio

#### Difesa

- progettare l'applicazione in modo che ottimizzi le risorse per ciascun utente
- prevedere un meccanismo per la gestione dell'eccessivo numero di richieste

H4

## H3 Default accounts attack

> Si verifica quando username e password di *default* (che molti sistemi usano), vengono usate da un attaccante per accedere in modo illecito

#### Difesa

- Rimuovere gli account di default

H4
## H3 Password cracking attack

> Si verifica quando l'attaccante riesce a ottenere tramite programmi di *cracking* la password di un sistema

#### Difesa

- come utente
  - scegliere password intelligenti

H4

- come programmatore
  - richiedere password robuste
  - usare metodi alternativi di autenticazione (dati biometrici, smart card...)