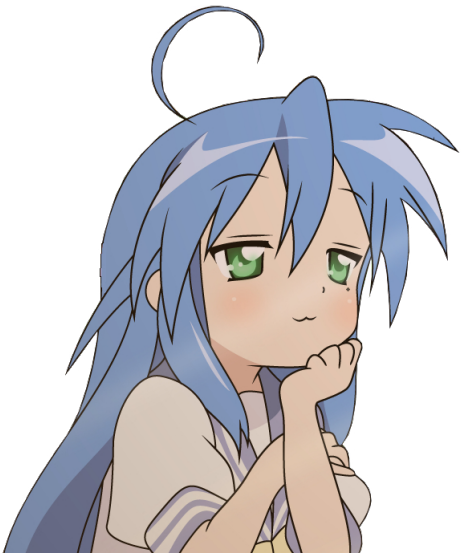
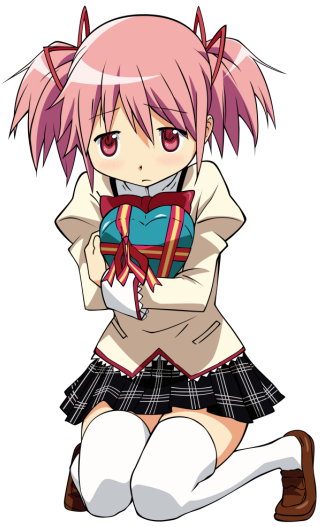
ANNO ACCADEMICO 2025/2026

Sicurezza dei Sistemi e dei Software

Teoria

Altair's Notes





UNIVERSITÀ DI TORINO

DIPARTIMENTO DI INFORMATICA

Premessa

Licenza

Formato utilizzato

Box di "Concetto sbagliato":

Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contente il concetto giusto.

Box di "Corollario":

Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 0.0.1: Nome delle definizione

Testo della definizione.

Box di "Domanda":

Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Osservazioni":

Osservazioni 0.0.1

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.

1 Introduzione

1.1 Tassonomie e Problemi di Sicurezza Software

Categorie di problemi di sicurezza dei software:

- *Input Validation Failure:*
  - Injection attacks (SQL, command, code).
  - Cross-site scripting (XSS).
- *Memory Safety Issue:*
  - Buffer overflows, use-after-free, double-free[1].
  - Causa principale del 70% delle vulnerabilità critiche (Microsoft, google).
- *Authentication & Authorization Flaws:*
  - Broken access control, problemi con i privilegi.
  - Session management issue.
- *Logic & Design Flaws:*
  - Race conditions, time-of-check-time-of-use (TOCTOU).
  - Business logic bypass.

1.1.1 I Problemi in Dettaglio

Definizione 1.1.1: Input Validation Failure

Gli Input Validation Failure avvengono quando un applicativo "si fida" eccessivamente degli utenti. Sono causati da una non sufficiente sanificazione e validazione di un input.

Osservazioni 1.1.1 Tipi

- Injection vulnerability:
  - SQL Injection: codice SQL malevolo messo come input.

[1] Un approfondimento interessante riguarda l'Arbitrary Code Execution (ACE) nelle console.

  - Command Injection: comandi OS messi come input.
  - Code Injection: codice eseguibile messo come input.
- Cross-site scripting (XSS):
  - Reflected: esecuzione immediata di scripts malevoli.
  - Stored: scripts malevoli persistenti in un database.
  - DOM-Based: manipolazione di scripts client-side.
- Path traversal:
  - Accesso a file al di fuori della directory corretta (../../../etc/passwd).

Definizione 1.1.2: Memory Safety Issue

Errori causati da una scorretta manipolazione della memoria.

Osservazioni 1.1.2 Tipi

- Buffer Overflows:
  - Stack-Based: viene sovrascritto il return address (RA).
  - Heap-Based:corruzione della memoria che gestisce le strutture dati.
- Use-After-Free (UAF):
  - Accedere alla memoria liberata consente Arbitrary Code Execution (ACE).
  - Comune in applicazioni C/C++.
- Double-Free:
  - Liberare due volte la stessa memoria.
  - Corruzione dei metadati dell'heap.
- Integer overflows:
  - Possono causare buffer overflows quando usati per il calcolo di indirizzi.

Definizione 1.1.3: Authentication & Authorization Flaws

Problemi legati al "chi sei" e al "che cosa puoi fare".
Accesso non autorizzato a dati sensibili e funzionalità.

Osservazioni 1.1.3

- Broken Authentication:
  - Password deboli.
  - Session fixation, Session hijacking.
  - Attacchi alle credenziali.
- Broken Access Control:
  - Problemi con i privilegi verticali (user $\rightarrow$ admin).
  - Problemi con i privilegi orizzontali (user A $\rightarrow$ user B).

- Session Management Issue:
  - Session ID predicibile.
  - Sessione non invalidata dopo il logout.
  - Cross-site request forgery (CSRF).

Definizione 1.1.4: Logic & Design Flaws

Problemi con la logica con cui è stato ragionato un applicativo.

Osservazioni 1.1.4

- Race Conditions:
  - TOCTOU.
  - Multipli threads che accedono a risorse condivise.
- Business Logic Bypass:
  - Violazioni dello stato di una macchina.
  - Workflow circumvention.
- Cryptographic failures:
  - Algoritmi deboli, key management scarso.
  - Validazione di certificati impropria.
  - Side-Channel attacks.
- Configuration Issue:
  - Credenziali di default.
  - Error handling improprio.

Principali cause:

- *Linguaggio scelto:*
  - C/C++: efficiente, ma richiede gestione manuale della memoria.
  - Linguaggi interpretati: soggetti a Injection vulnerability.
- *Developer training:*
  - Mancanza di consapevolezza per i rischi sulla sicurezza e pressione per avere delivery costante (TAASS fa schifo).
- *Complessità:*
  - Codebases enormi e con multiple dipendenze.
  - Interazione di componenti.
- *Legacy systems:*
  - Vecchio codice con pratiche di sicurezza superate.
  - Difficile da aggiornare o rimpiazzare.
- *Testing non adeguato:*
  - Che si focalizza sulla funzionalità e non sulla sicurezza.
  - Numero di tools limitato.

1.1.2 Frameworks

Standard dell'industria:

- *CWE* (Common Weakness Enumeration):
  - Categorizza i tipi di debolezze software.
  - Si focalizza sulle cause e sui patterns.
- *OWASP Top 10*:
  - Riguardano le applicazioni web più critiche.
  - Viene aggiornata ogni 3-4 anni basandosi sui dati dell'industria.

Definizione 1.1.5: Open Web Application Security Project

Fondato nel 2001 è un fondazione no profit che si occupa di cybersecurity. Progetti chiave:

- OWASP Top 10.
- OWASP Testing Guide: guida alle metodologie di testing.
- OWASP ZAP: tool per testing di sicurezza.
- ASVS: Application Security Verification Standard.

Category	CWE Examples	OWASP Top 10
Input Validation	CWE-79 (XSS), CWE-89 (SQL Injection)	A03:2021-Injection
Memory Safety	CWE-787 (Out-of-bounds Write)	-
Authentication	CWE-862 (Missing Authorization)	A01:2021-Broken Access Control
Logic Flaws	CWE-362 (Race Conditions)	A04:2021-Insecure Design

Figure 1.1: Esempi di CWE e OWASP.

Definizione 1.1.6: MITRE ATT&CK

Framework usato per categorizzare e descrivere cyberattacchi su tattiche, tecniche e common knowledge.

Idee di MITRE ATT&CK:

- Linguaggio comune per descrivere i comportamenti *avversari*.
- Organizzare metodi di attacco conosciuti.
- Strumenti per la sicurezza.

Overview:

- Tattiche (colonne): il "perché", i goal degli avversari.
  - Accesso iniziale, esecuzione, persistenza, privilegi.
  - Accesso alle credenziali, scoperte, movimenti laterali.
  - Collezioni, Comandi e controlli, impatto.

- Tecniche (righe): il "come", i metodi per ottenere i goal.
  - T1566 - Phishing.
- Sotto-tecniche: variazioni specifiche di tecniche.
  - T1566.001 - Spearphishing Attachment.
  - T1566.002 - Spearphishing Link.
- Procedure: specifiche implementazioni.

Definizione 1.1.7: Vulnerability Classification

Framework per descrivere, categorizzare e prioritizzare problemi di sicurezza.

Sistemi chiave:

- CVE: identifica specifiche vulnerabilità.
- CWE: categorizza tipi di debolezze.
- CVSS: dà un punteggio alla severità delle vulnerabilità.
- CPE: identifica prodotti e sistemi affetti da vulnerabilità.

Note:-

Questa è terminologia comune in ambito di cybersecurity.

1.1.3 Classificazione delle Vulnerabilità

Definizione 1.1.8: CVE

Definizione 1.1.9: CWE

Definizione 1.1.10: CVSS

Definizione 1.1.11: CPE

2 Approfondimenti

In questa sezione metto qualche approfondimento che ho ritenuto interessante.

2.1 Arbitrary Code Execution in Animal Crossing

Source: video.

2.1.1 Spiegazione di ACE

2.1.2 ACE in Animal Crossing

2.1.3 ACE Achievements