



DISCLAIMER

Il materiale contenuto nel drive è stato raccolto e richiesto tramite autorizzazione ai ragazzi frequentanti il corso di studi di Informatica dell'Università degli Studi di Salerno. Gli appunti e gli esercizi nascono da un uso e consumo degli autori che li hanno creati e risistemati per tanto non ci assumiamo la responsabilità di eventuali mancanze o difetti all'interno del materiale pubblicato.

Il materiale sarà modificato aggiungendo il logo dell'associazione, in tal caso questo possa recare problemi ad alcuni autori di materiale pubblicato, tale persona può contattarci in privato ed elimineremo o modificheremo il materiale in base alle sue preferenze.

Ringraziamo eventuali segnalazioni di errori così da poter modificare e fornire il miglior materiale possibile a supporto degli studenti.



CoScienze
Associazione

CATALOGAZIONE DELLE DEBOLEZZE

Oltre al catalogo delle vulnerabilità col sistema CVE e come attribuire loro dei punteggi basati su metriche col sistema CVSS, esiste un *catalogo delle debolezze* col sistema **CWE** e un sistema **CWSS** per l'attribuzione di punteggi agli oggetti del catalogo.




Il **Common Weaknesses Enumeration (CWE)** è un sistema per catalogare in modo uniforme le debolezze software. Un esempio è: *CWE-276: Incorrect Default Permissions*, *CWE-272: Least Privilege Violation*, *CWE-426: Untrusted Search Path*.

Il catalogo CWE è un insieme di oggetti, ciascuno dotato di un **identificatore** e di un numero di **attributi**. Ecco una breve spiegazione degli attributi che vengono utilizzati per la catalogazione:

- **Abstraction**: questo attributo fornisce un'astrazione generica della debolezza, consentendo di comprendere il suo concetto di base senza considerare implementazioni specifiche.
- **Description**: descrive la debolezza in modo dettagliato, spiegando cosa sia e come possa essere sfruttata da un attaccante.
- **Applicable platforms**: specifica le piattaforme o gli ambienti in cui la debolezza è rilevante, ad esempio determinati sistemi operativi, linguaggi di programmazione o applicazioni.
- **Common consequences**: indica le conseguenze comuni che possono verificarsi se la debolezza viene sfruttata con successo, come il furto di dati, la compromissione della sicurezza o l'interruzione dei servizi.
- **Likelihood of exploit**: valuta la probabilità che un aggressore sfrutti la debolezza, aiutando a determinare il livello di rischio associato.
- **Demonstrative examples**: fornisce esempi concreti o scenari in cui la debolezza potrebbe verificarsi, aiutando a comprendere meglio come identificarla e mitigarla.
- **Potential mitigations**: suggerisce possibili misure o strategie per ridurre o eliminare la debolezza, come l'implementazione di contromisure tecniche, l'adozione di best practice o l'aggiornamento del software.
- **Relationships**: descrive le relazioni con altre debolezze, consentendo di comprendere come una debolezza possa essere correlata ad altre e come affrontarle in modo integrato.

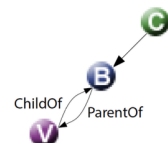
Un oggetto del catalogo, a differenza del catalogo delle vulnerabilità, può essere la descrizione di una singola debolezza o un elenco di identificatori a singole debolezze in relazione tra loro.

L'attributo **Abstraction** specifica il tipo di debolezza e può essere di tre diversi tipi:

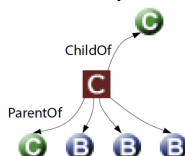
- **Class - C**: debolezza descritta in termini generali, senza riferimenti a linguaggi o tecnologie specifiche.  Oggetto CWE Class
- **Base - B**: debolezza descritta in modo più dettagliato, in modo da poter intuire tecniche di rilevazione e prevenzione.  Oggetto CWE Base
- **Variant - V**: debolezza descritta nei minimi dettagli, nell'ambito di uno specifico linguaggio e tecnologia.  Oggetto CWE Variant

L'attributo **Relationships** specifica il tipo di relazioni che l'oggetto ha con altri oggetti del catalogo. Esempio di relazioni:

- **ChildOf**: l'oggetto è figlio di un altro oggetto.
- **ParentOf**: l'oggetto è padre di un altro oggetto.

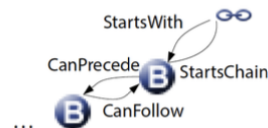
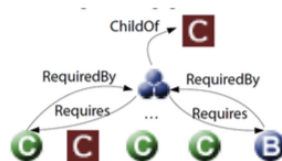
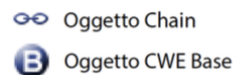
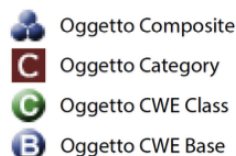


Un oggetto **Category** punta ad un insieme di oggetti che condividono uno specifico attributo. Essendo un oggetto raggruppatore, di solito ha più relazioni ParentOf che ChildOf:



Un oggetto **Compound** mette in relazione tra loro diverse debolezze implicate in una vulnerabilità. Due tipologie:

- **Composite**: aggrega tutte le debolezze che sfruttate insieme, provocano una vulnerabilità.
- **Chain**: aggrega tutte le debolezze che, sfruttate in cascata, provocano una vulnerabilità.



Common Weaknesses Scoring System (CWSS)

Così come per il catalogo CVE esiste il sistema di punteggi CVSS, anche per il catalogo CWE esiste un sistema analogo. Il **Common Weaknesses Scoring System (CWSS)** è molto simile al CVSS. Ad ogni CWE id è assegnato un punteggio da 0 a 100, dove 0 indica un impatto nullo mentre 100 indica conseguenze catastrofiche.

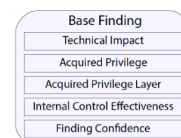
Il punteggio CWSS è dato dal prodotto di tre sottopunteggi: **BaseFinding** (tra 0 e 100), **Attack Surface** (tra 0 e 1) ed **Environmental** (tra 0 e 1). Ad ogni metrica è associata una domanda a risposta multipla, ciascuna risposta fornisce un peso numerico e i singoli pesi sono poi aggregati in un risultato finale tramite una serie di formule.

Le risposte relative alle domande del questionario sono presentate sotto forma di stringa di testo. Tale stringa, detta **vector string**, è formata da terne di abbreviazioni domanda:risposta,peso separate dal carattere /.

Esempio: TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0

BASE FINDING:

Stimano il rischio della debolezza in sé, l'accuratezza della scoperta, la robustezza dei meccanismi di protezione.



La metrica **Technical Impact** risponde alla domanda:

- Nell'ipotesi che la debolezza possa essere sfruttata con successo, qual è la principale conseguenza tecnica?

| Valore | Descrizione | Punt. |
|--------------|---|-------|
| Critical (C) | Controllo completo; interruzione delle operazioni. | 1.0 |
| High (H) | Controllo di molte operazioni; accesso ad informazioni critiche. | 0.9 |
| Medium (M) | Controllo di alcune operazioni; accesso ad informazioni importanti. | 0.6 |
| Low (L) | Controllo minimo; accesso ad informazioni irrilevanti. | 0.3 |
| None (N) | La debolezza non porta ad una vulnerabilità. | 0.0 |

La metrica **Acquired Privilege** risponde alla domanda:

- Nell'ipotesi che la debolezza possa essere sfruttata con successo, che tipi di privilegi si ottengono?

| Valore | Descrizione | Punt. |
|-------------------------------|--|-------|
| Administrator (A) | L'attaccante diventa amministratore (root in UNIX, SYSTEM in Windows, admin su un router). | 1.0 |
| Partially Privileged User (P) | L'attaccante diventa un utente con alcuni privilegi, ma non tutti quelli di un amministratore. | 0.9 |
| Regular User (RU) | L'attaccante diventa un utente normale, senza privilegi particolari. | 0.7 |
| Limited or Guest (L) | L'attaccante diventa un utente con privilegi ristretti (ad esempio, nobody su UNIX). | 0.6 |
| None (N) | L'attaccante non riesce a diventare un utente. | 0.1 |

La metrica **Acquired Privilege Layer** risponde alla domanda:

- Nell'ipotesi che la debolezza possa essere sfruttata con successo, a che livello operazionale si ottengono i privilegi?

| Valore | Descrizione | Punt. |
|-------------------------------|--|-------|
| Application (A) | L'attaccante acquisisce privilegi a livello di utente di una applicazione software. | 1.0 |
| System (S) | L'attaccante acquisisce privilegi a livello di utente di un sistema operativo. | 0.9 |
| Network (N) | L'attaccante acquisisce il privilegio di accesso alla rete. | 0.7 |
| Enterprise Infrastructure (E) | L'attaccante acquisisce l'accesso ad una porzione dell'infrastruttura (router, switch, DNS, controller di dominio, firewall, ...). | 1.0 |

La metrica **Internal Control Effectiveness** risponde alla domanda:

- Qual è l'efficacia delle contromisure, a livello di codice?

| Valore | Descrizione | Punt. |
|--------------------|--|-------|
| None (N) | Non esistono contromisure. | 1.0 |
| Limited (L) | Esiste un meccanismo semplice o fortuito, in grado di rintuzzare un attaccante occasionale. | 0.9 |
| Moderate (M) | Esiste un meccanismo standard con dei limiti, aggirabile con un po' di impegno da un esperto. | 0.7 |
| Indirect (I) | Un meccanismo non specifico per la debolezza ne riduce l'impatto in maniera indiretta. | 0.5 |
| Best-Available (B) | È implementato il meccanismo migliore noto. Un attaccante esperto e determinato potrebbe aggirarlo con l'aiuto di altre debolezze. | 0.3 |
| Complete (C) | Il meccanismo impedisce lo sfruttamento. | 0.0 |

La metrica **Finding Confidence** risponde alla domanda:

- Quanto si è sicuri che il difetto individuato sia una debolezza e possa essere usato da un attaccante?

| Valore | Descrizione | Punt. |
|--------------------------|--|-------|
| Proven True (T) | La debolezza esiste ed è raggiungibile da un attaccante. | 1.0 |
| Proven Locally True (LT) | La debolezza esiste, ma non è chiaro se sia o meno sfruttabile da un attaccante. | 0.8 |
| Proven False (F) | Il difetto/bug non costituisce una debolezza e/o non è sfruttabile da un attaccante. | 0.0 |

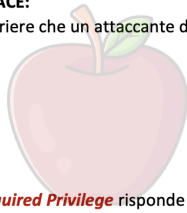
Il Punteggio **Base Findings** è un valore tra 0 e 100, calcolato nel modo seguente:

$$f(TI) = \begin{cases} 0 & \text{if } TI = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$BaseFindingScore = [(10 * TI + 5 * (AP + AL) + 5 * FC) * f(TI) * IC] * 4.0$$

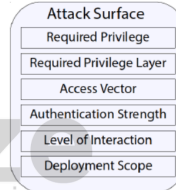
ATTACK SURFACE:

Stimano le barriere che un attaccante deve superare per sfruttare la debolezza.



CoScienze

Associazione



La metrica **Required Privilege** risponde alla domanda:

- Quali privilegi deve già possedere l'utente per sfruttare la debolezza?

| Valore | Descrizione | Punt. |
|-------------------------------|--|-------|
| None (N) | Non sono richiesti privilegi particolari. | 1.0 |
| Limited / Guest (L) | L'attaccante deve già avere i privilegi di un utente ristretto. | 0.9 |
| Regular User (RU) | L'attaccante deve già avere i privilegi di un utente normale. | 0.7 |
| Partially Privileged User (P) | L'attaccante deve già avere i privilegi di un utente speciale (con alcuni privilegi in più rispetto ad uno normale, ma non tutti quelli di un amministratore). | 0.6 |
| Administrator (A) | L'attaccante deve già avere i privilegi di un utente amministratore. | 0.1 |

La metrica **Access Vector** risponde alla domanda:

- Attraverso quale canale deve comunicare l'attaccante per poter sfruttare la debolezza?

| Valore | Descrizione | Punt. |
|----------------------|---|-------|
| Internet (I) | L'attaccante deve avere accesso ad Internet. | 1.0 |
| Intranet (R) | L'attaccante deve avere accesso ad una Intranet schermata da un proxy Web. | 0.8 |
| Private Network (V) | L'attaccante deve avere accesso ad una rete privata disponibile solo ad alcuni utenti fidati. | 0.8 |
| Adjacent Network (A) | L'attaccante deve avere accesso fisico al dominio di broadcast o di collisione della rete. | 0.7 |
| Local (L) | L'attaccante deve avere accesso locale ad una shell. | 0.5 |
| Physical (P) | L'attaccante deve avere accesso fisico all'asset. | 0.2 |

La metrica **Level of Interaction** risponde alla domanda:

- Quali azioni deve compiere la vittima per consentire all'attaccante di svolgere l'attacco con successo?

| Valore | Descrizione | Punt. |
|-----------------------|---|-------|
| Automated (A) | Non è richiesta interazione umana. | 1.0 |
| Typical / Limited (T) | L'attaccante deve convincere l'utente a svolgere una azione normale nel contesto del software. | 0.9 |
| Moderate (M) | L'attaccante deve convincere l'utente a svolgere una azione sospetta per un conoscente della sicurezza. | 0.8 |
| Opportunistic (O) | L'attaccante non può controllare direttamente la vittima; può solo capitalizzare errori altrui. | 0.3 |
| High (H) | L'attaccante deve usare il social engineering. | 0.1 |
| No Interaction (NI) | Non è possibile alcuna interazione. | 0.0 |

Il Punteggio **Attack Surface** è un valore tra 0 e 1, calcolato nel modo seguente:

$$\text{AttackSurfaceScore} = \frac{20 * (RP + RL + AV) + 20 * SC + 15 * IN + 5 * AS}{100.0}$$

ENVIRONMENTAL:

Stimano le specificità legate ad uno specifico contesto operativo.



La metrica **Business Impact** risponde alla domanda:

- Qual è l'impatto ambientale di uno sfruttamento della sicurezza?

| Valore | Descrizione | Punt. |
|--------------|---|-------|
| Critical (C) | L'azienda può fallire. | 1.0 |
| High (H) | Le operazioni aziendali sono colpite gravemente. | 0.9 |
| Medium (M) | Alcune operazioni aziendali sono colpite, ma non quelle più comuni. | 0.6 |
| Low (L) | L'impatto aziendale è minimo. | 0.3 |
| None (N) | Non vi è impatto aziendale alcuno. | 0.0 |

La metrica **Likelihood of Exploit** risponde alla domanda:

- Qual è la probabilità che, una volta scoperta la debolezza, un attaccante con il giusto privilegio sia in grado di sfruttarla?

| Valore | Descrizione | Punt. |
|------------|--|-------|
| High (H) | È molto probabile che un attaccante riesca a sfruttare la debolezza tramite un exploit di facile implementazione. | 1.0 |
| Medium (M) | Un attaccante potrebbe riuscire a sfruttare la debolezza. Le probabilità di successo variano; potrebbero essere necessari più tentativi. | 0.6 |
| Low (L) | È improbabile che un attaccante riesca a sfruttare la debolezza. | 0.2 |
| None (N) | L'attaccante non ha alcuna chance di successo. | 0.0 |

La metrica **Required Privilege Layer** risponde alla domanda:

- A quale livello operativo l'attaccante deve avere privilegi per poter sfruttare la debolezza?

| Valore | Descrizione | Punt. |
|-------------------------------|--|-------|
| Application (A) | L'attaccante deve già avere privilegi applicativi. | 1.0 |
| System (S) | L'attaccante deve già avere privilegi a livello di sistema operativo. | 0.9 |
| Network (N) | L'attaccante deve già avere i privilegi di accesso alla rete. | 0.7 |
| Enterprise Infrastructure (E) | L'attaccante deve già avere i privilegi a livello di infrastruttura (router, switch, DNS, controller di dominio, firewall, ...). | 1.0 |

La metrica **Authentication Strength** risponde alla domanda:

- Quanto la procedura di autenticazione protegge la debolezza?

| Valore | Descrizione | Punt. |
|--------------|---|-------|
| None (N) | Non è prevista alcuna forma di autenticazione. | 1.0 |
| Weak (W) | È prevista una autenticazione debole (username e password). | 0.9 |
| Moderate (M) | È prevista una autenticazione moderatamente forte (uso di certificati, autenticazione basata su conoscenza, one-time password). | 0.8 |
| Strong (S) | È prevista una autenticazione forte (token hardware, multi-fattore). | 0.7 |

La metrica **Deployment Scope** risponde alla domanda:

- In quali piattaforma e/o configurazioni si presenta la debolezza?

| Valore | Descrizione | Punt. |
|---------------------------|---|-------|
| All (A) | La debolezza si manifesta in tutte le piattaforme ed in tutte le configurazioni. | 1.0 |
| Moderate (M) | La debolezza si manifesta nelle piattaforme e/o nelle configurazioni più comuni. | 0.9 |
| Rare (R) | La debolezza si manifesta solo in piattaforme e/o nelle configurazioni più rare. | 0.5 |
| Potentially Reachable (P) | La debolezza è potenzialmente sfruttabile. In questo specifico istante tutti i percorsi di codice sembrano sicuri e/o la debolezza è codice "morto" (non raggiungibile in pratica). | 0.1 |

La metrica **Likelihood of Discovery** risponde alla domanda:

- Qual è la probabilità che un attaccante scopra la debolezza?

| Valore | Descrizione | Punt. |
|------------|---|-------|
| High (H) | È molto probabile che un attaccante riesca a scoprire la debolezza usando tecniche semplici e senza accesso al codice sorgente del software. | 1.0 |
| Medium (M) | Un attaccante potrebbe riuscire a scoprire la debolezza, ma solo con accesso al codice sorgente del software e tanto tempo a disposizione. | 0.6 |
| Low (L) | È improbabile che un attaccante riesca a scoprire la debolezza senza avere capacità particolari, accesso al codice sorgente e tanto tempo a disposizione. | 0.2 |

La metrica **External Control Effectiveness** risponde a:

- Qual è l'efficacia delle contromisure esterne (NON a livello di codice)?

| Valore | Descrizione | Punt. |
|---------------------|--|-------|
| None (N) | Non esistono contromisure. | 1.0 |
| Limited (L) | Esiste un meccanismo semplice o fortuito, in grado di rintuzzare un attaccante occasionale. | 0.9 |
| Moderate (M) | Esiste un meccanismo standard con dei limiti, aggirabile con un po' di impegno da un esperto. | 0.7 |
| Indirect (I) | Un meccanismo non specifico per la debolezza ne riduce l'impatto in maniera indiretta. | 0.5 |
| Best- Available (B) | È implementato il meccanismo migliore noto. Un attaccante esperto e determinato potrebbe aggirarlo con l'aiuto di altre debolezze. | 0.3 |
| Complete (C) | Il meccanismo impedisce lo sfruttamento. | 0.1 |

La metrica **Prevalence** risponde alla domanda:

- Qual è la frequenza di occorrenza della debolezza nel software in generale?

| Valore | Descrizione | Punt. |
|----------------|---|-------|
| Widespread (W) | La debolezza è presente nella maggioranza (se non la totalità) dei software in esecuzione nella infrastruttura considerata. | 1.0 |
| High (H) | La debolezza si incontra spesso, ma non è diffusa su ampio spettro. | 0.9 |
| Common (C) | La debolezza si incontra di tanto in tanto. | 0.8 |
| Limited (L) | La debolezza si incontra raramente (oppure, mai). | 0.7 |

Il Punteggio **Environmental** è un valore tra 0 e 1, calcolato nel modo seguente:

$$f(BI) = \begin{cases} 0 & \text{if } BI=0 \\ 1 & \text{otherwise} \end{cases}$$

$$EnvironmentalScore = [(10 * BI + 3 * DI + 4 * EX + 3 * P) * f(BI) * EC] / 20.0$$

CVSS vs CWSS

CVSS e **CWSS** sono molto simili, ma ci sono alcune differenze:

- CVSS assume che una vulnerabilità sia già stata scoperta e verificata, mentre CWSS può essere utilizzata prima che ciò accada. CVSS cataloga gli errori fatti, mentre CWSS cataloga gli errori fattibili.
- In CVSS alcuni aspetti combinano caratteristiche multiple, che sono invece separate in CWSS. Ad esempio, Access Complexity (AC) si suddivide in Required Privilege Level e Level of Interaction.

