

Programmazione Sicura



Catalogazione
delle debolezze



Barbara Masucci
UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Punto della situazione

- Nella lezione scorsa abbiamo visto come catalogare le **vulnerabilità software** con il sistema **CVE** e come attribuire loro dei punteggi basati su metriche con il sistema **CVSS**



- Scopo della lezione di oggi:
 - Descrivere un sistema di catalogazione delle **debolezze software**: il **CWE**
 - Descrivere un sistema per l'attribuzione di punteggi agli oggetti del catalogo CWE: il **CWSS**



Common Weaknesses Enumeration

- I sistemi **CVE** e **CVSS** rappresentano un buon passo verso la creazione di un catalogo uniforme delle vulnerabilità
- Un approccio simile è stato utilizzato anche per l'**enumerazione** e la **valutazione** delle **debolezze**
- Il **Common Weaknesses Enumeration (CWE)** è un sistema per catalogare in modo uniforme le debolezze software
 - Home page: <https://cwe.mitre.org>
 - Ultima versione: 4.14



Common Weaknesses Enumeration

- Nelle sfide CTF che risolveremo durante il corso, incontreremo diversi oggetti del catalogo CWE, tra cui
 - CWE-276: Incorrect Default Permissions
 - CWE-272: Least Privilege Violation
 - CWE-426: Untrusted Search Path
 - CWE-77: Command Injection
 - CWE-90: Authentication Bypass by Spoofing
 - CWE-250: Execution with Unnecessary Privileges
 - CWE-61: Symlink Following
 - CWE-367: Time-Of-Check Time-Of-Use (TOCTOU) Race Condition
 - CWE-78: OS Command Injection
 - CWE-89: SQL Injection
 - CWE-79: Cross-site Scripting (XSS)
 - CWE-352: Cross-Site Request Forgery (CSRF)
 - CWE-121: Stack-based Buffer Overflow



Common Weaknesses Enumeration

- Nelle sfide CTF che risolverete nell'ambito dei progetti potreste incontrare altri oggetti del catalogo CWE, tra cui
 - CWE-732: Incorrect Permission Assignment for Critical Resource
 - CWE-59: Improper Link Resolution Before File Access
 - CWE-261: Weak Encoding for Password
 - CWE-257: Storing Passwords in a Recoverable Format
 - CWE-319 : Cleartext Transmission of Sensitive Information
 - CWE-624: Executable Regular Expression Error
 - CWE-88: Argument Injection / Modification
 - CWE-912: Hidden Functionality
 - CWE-502: Deserialization of Untrusted Data



Common Weaknesses Enumeration

- Il catalogo CWE è un insieme di oggetti, ciascuno dotato di un **identificatore** e di un numero di **attributi**
- Attributi:
 - Abstraction, Description, Applicable platforms, Common consequences, Likelihood of exploit, Demonstrative examples, Potential mitigations, Relationships,...
- Un oggetto può essere
 - La descrizione di una singola debolezza
 - Un elenco di identificatori a singole debolezze in relazione tra loro



Common Weaknesses Enumeration

- L'attributo **Abstraction** specifica il tipo di debolezza
- Può essere di tre tipi diversi
 - **Class**  Debolezza descritta in termini generali, senza riferimenti a linguaggi o tecnologie specifiche
 - **Base**  Debolezza descritta in modo più dettagliato, in modo da poter intuire tecniche di rilevazione e prevenzione
 - **Variant**  Debolezza descritta nei minimi dettagli, nell'ambito di uno specifico linguaggio e tecnologia



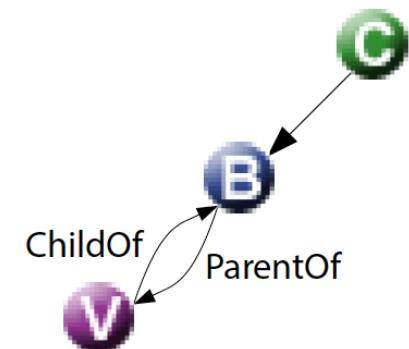
Common Weaknesses Enumeration

- 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

 Oggetto CWE Class

 Oggetto CWE Base

 Oggetto CWE Variant



Esempio di oggetto CWE

- L'oggetto **CWE:121**, illustrato al link
<https://cwe.mitre.org/data/definitions/121.html>
descrive lo **Stack-based Buffer Overflow**

CWE Common Weakness Enumeration
A Community-Developed List of Software Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (3.0)

ID Lookup: Go

Home | About | CWE List | Scoring | Community | News | Search

CWE-121: Stack-based Buffer Overflow

Weakness ID: 121
Abstraction: Variant
Structure: Simple

Status: Draft

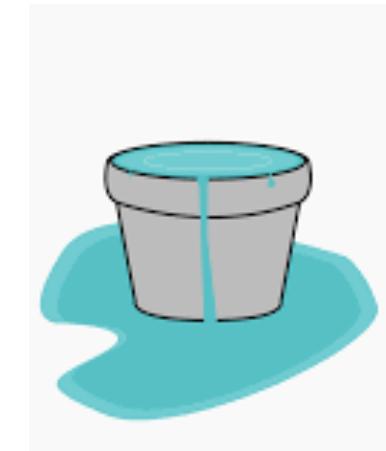
Presentation Filter: Basic

Description
A stack-based buffer overflow condition is a condition where the buffer being overwritten is allocated on the stack (i.e., is a local variable or, rarely, a parameter to a function).

Relationships
The table(s) below shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as PeerOf and CanAlsoBe are defined to show similar weaknesses that the user may want to explore.

- Relevant to the view "Research Concepts" (CWE-1000)
- Relevant to the view "Development Concepts" (CWE-699)

Modes Of Introduction
The different Modes of Introduction provide information about how and when this weakness



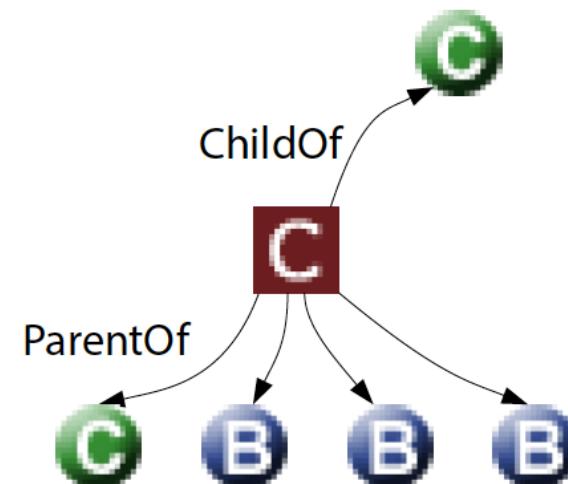
Common Weaknesses Enumeration

- 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

C Oggetto Category

C Oggetto CWE Class

B Oggetto CWE Base



Esempio di oggetto Category

- L'oggetto **Category CWE:21**, illustrato al link <https://cwe.mitre.org/data/definitions/21.html> descrive una diffusa categoria di debolezze: il **Pathname-Traversal and Equivalence Errors**

CWE Common Weakness Enumeration
A Community-Developed List of Software Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (3.0)

ID Lookup: Go

CWE CATEGORY: Pathname Traversal and Equivalence Errors

Category ID: 21 Status: Incomplete

Summary
Weaknesses in this category can be used to access files outside of a restricted directory (path traversal) or to perform operations on files that would otherwise be restricted (path equivalence). Files, directories, and folders are so central to information technology that many different weaknesses and variants have been discovered. The manipulations generally involve special characters or sequences in filenames, or the use of alternate references or channels.

Membership

Nature	Type	ID	Name
MemberOf	C	18	Source Code
MemberOf	V	699	Development Concepts
HasMember	C	22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
HasMember	C	22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
HasMember	B	41	Improper Resolution of Path Equivalence
HasMember	B	59	Improper Link Resolution Before File Access ('Link Following')
HasMember	B	59	Improper Link Resolution Before File Access ('Link Following')
HasMember	B	66	Improper Handling of File Names that Identify Virtual Resources

Content History



Common Weaknesses Enumeration

➤ Un oggetto **Compound** mette in relazione tra loro diverse debolezze implicate in una vulnerabilità

➤ Due tipologie

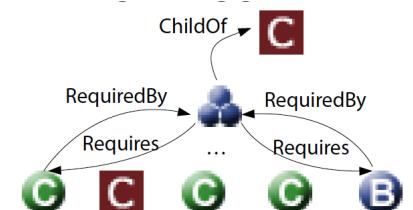
➤ **Composite**

aggrega tutte le debolezze che,
sfruttate insieme, provocano
una vulnerabilità

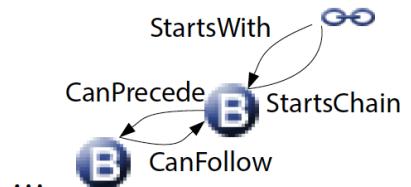
➤ **Chain**

aggrega tutte le debolezze che,
sfruttate in cascata, provocano
una vulnerabilità

- Oggetto Composite
- C Oggetto Category
- C Oggetto CWE Class
- B Oggetto CWE Base



- ∞ Oggetto Chain
- B Oggetto CWE Base



Esempio di oggetto Composite

- L'oggetto **Composite CWE:61**, illustrato al link <https://cwe.mitre.org/data/definitions/61.html> descrive una debolezza complessa: il **Symbolic Link Following**

CWE Common Weakness Enumeration
A Community-Developed List of Software Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (3.0)

ID Lookup: Go

CWE-61: UNIX Symbolic Link (Symlink) Following

Weakness ID: 61 Status: Incomplete

Abstraction: Compound

Structure: Composite

Presentation Filter: Basic

Description
The software, when opening a file or directory, does not sufficiently account for when the file is a symbolic link that resolves to a target outside of the intended control sphere. This could allow an attacker to cause the software to operate on unauthorized files.

Composite Components

Nature	Type	ID	Name
Requires	C	362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
Requires	C	340	Predictability Problems
Requires	C	216	Containment Errors (Container Errors)
Requires	B	386	Symbolic Name not Mapping to Correct Object
Requires	C	732	Incorrect Permission Assignment for Critical Resource

Extended Description
A software system that allows UNIX symbolic links (symlink) as part of paths whether in internal code or through user input can allow an attacker to spoof the symbolic link and traverse the file system to unintended locations or access arbitrary files. The symbolic link can permit an attacker to read/write/corrupt a file that they originally did not have permissions to access.



Esempio di oggetto Chain

- L'oggetto **Chain CWE:692**, illustrato al link <https://cwe.mitre.org/data/definitions/692.html> descrive una tipica catena di debolezze: da una **Blacklist Incompleta** a un **Cross-Site Scripting**

CWE Common Weakness Enumeration
A Community-Developed List of Software Weakness Types

Home > CWE List > CWE- Individual Dictionary Definition (3.0)

ID Lookup: Go

Home | About | CWE List | Scoring | Community | News | Search

CWE-692: Incomplete Blacklist to Cross-Site Scripting

Weakness ID: 692
Abstraction: Compound
Structure: Chain

Status: Draft

Presentation Filter: Basic

Description
The product uses a blacklist-based protection mechanism to defend against XSS attacks, but the blacklist is incomplete, allowing XSS variants to succeed.

Chain Components

Nature	Type	ID	Name
StartsWith	B	184	Incomplete Blacklist
FollowedBy	B	79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Extended Description
While XSS might seem simple to prevent, web browsers vary so widely in how they parse web pages, that a blacklist cannot keep track of all the variations. The "XSS Cheat Sheet" [REF-564] contains a large number of attacks that are intended to bypass incomplete blacklists.

Relationships
The table(s) below shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as PeerOf and CanAlsoBe are defined to show similar weaknesses that the user may want to explore.

XSS
Cross Site Scripting



Top 25 CWE

- Ogni anno viene fornita la lista delle debolezze software più comuni e facili da utilizzare degli ultimi due anni

https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html



2023 CWE Top 25 Most Dangerous
Software Weaknesses

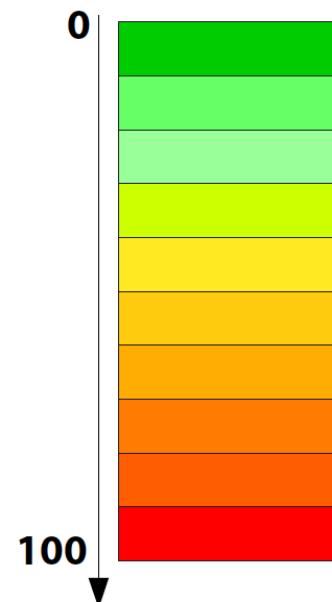
Rank	ID	Name	Score	CVEs in KEV	Rank Change
1	CWE-787	Out-of-bounds Write	63.72	70	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.54	4	0
3	CWE-89	Improper Neutralization of Special Elements Used in an SQL Command ('SQL Injection')	34.27	6	0
4	CWE-416	Use After Free	16.71	44	3
5	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	15.65	23	1
6	CWE-20	Improper Input Validation	15.50	35	-2
7	CWE-125	Out-of-bounds Read	14.60	2	-2
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.11	16	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	11.73	0	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	10.41	5	0
11	CWE-862	Missing Authorization	6.90	0	5
12	CWE-476	NULL Pointer Dereference	6.59	0	-1
13	CWE-287	Improper Authentication	6.39	10	1
14	CWE-190	Integer Overflow or Wraparound	5.89	4	-1
15	CWE-502	Deserialization of Untrusted Data	5.56	14	-3
16	CWE-77	Improper Neutralization of Special Elements Used in a Command ('Command Injection')	4.95	4	1
17	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	4.75	7	2
18	CWE-798	Use of Hard-coded Credentials	4.57	2	-3
19	CWE-918	Server-Side Request Forgery (SSRF)	4.56	16	2
20	CWE-306	Missing Authentication for Critical Function	3.78	8	-2
21	CWE-362	Concurrent Execution using Shared Resources with Improper Synchronization ('Race Condition')	3.53	8	1
22	CWE-269	Improper Privilege Management	3.31	5	7
23	CWE-94	Improper Control of Generation of Code ('Code Injection')	3.30	6	2
24	CWE-863	Incorrect Authorization	3.16	0	4
25	CWE-276	Incorrect Default Permissions	3.16	0	-5



Top 25 CWE 2023

Common Weaknesses Scoring System

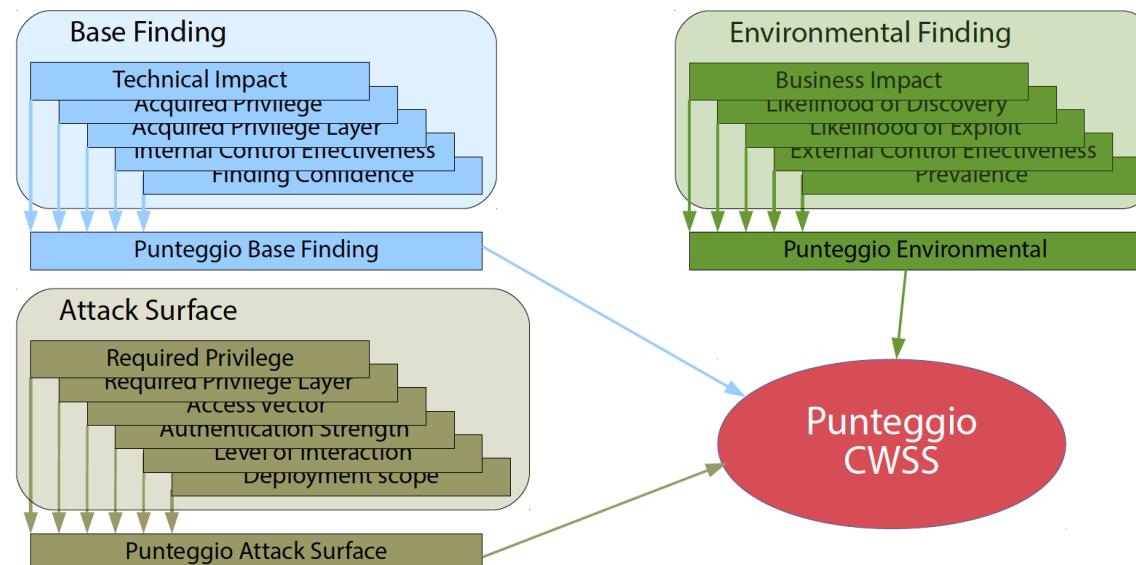
- Così come per il catalogo CVE esiste il sistema di punteggi CVSS, anche per il catalogo CWE esiste un sistema analogo
- Il **Common Weaknesses Scoring System (CWSS)** è molto simile al CVSS
 - Home page: <https://cwe.mitre.org/cwss/>
- Ad ogni CWE id è assegnato un punteggio da 0 a 100
 - 0: impatto nullo
 - 100: conseguenze catastrofiche



Common Weaknesses Scoring System

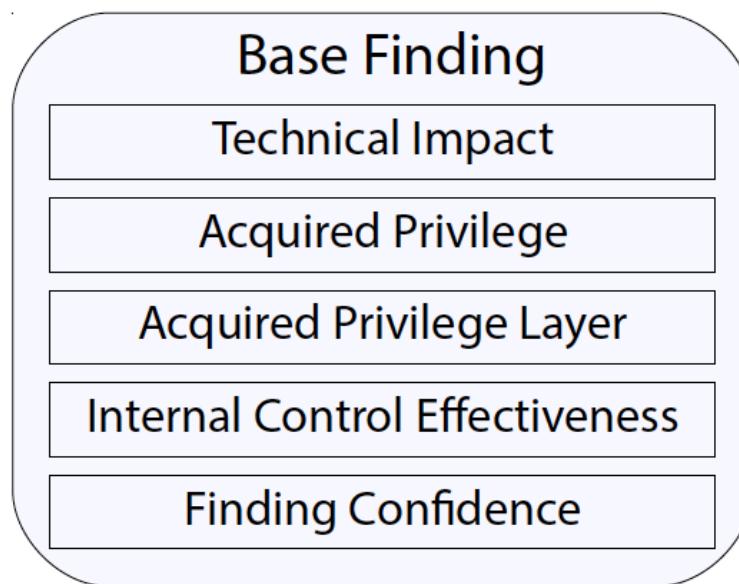
Il punteggio CWSS è dato dal prodotto di tre sottopunteggi:

- **BaseFinding** (tra 0 e 100)
- **Attack Surface** (tra 0 e 1)
- **Environmental** (tra 0 e 1)



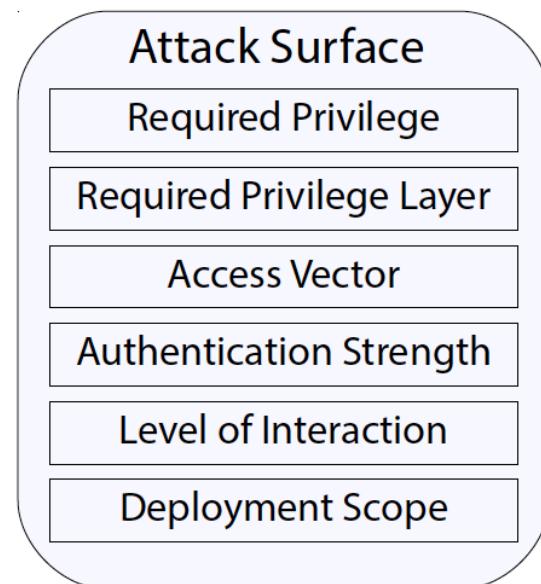
Metriche Base Finding

- Stimano il rischio della debolezza in sè, l'accuratezza della scoperta, la robustezza dei meccanismi di protezione
 - E' veramente presente?
 - E' grave?
 - E' protetta bene?



Metriche Attack Surface

- Stimano le barriere che un attaccante deve superare per sfruttare la debolezza
 - E' necessaria l'autenticazione?
 - Servono privilegi particolari?



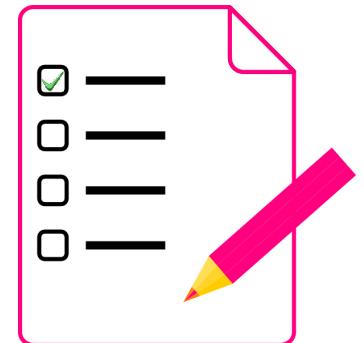
Metriche Environmental

- Stimano le specificità legate ad uno specifico contesto operativo
 - Che impatto ha?
 - Quanto è probabile scoprirla e usarla?



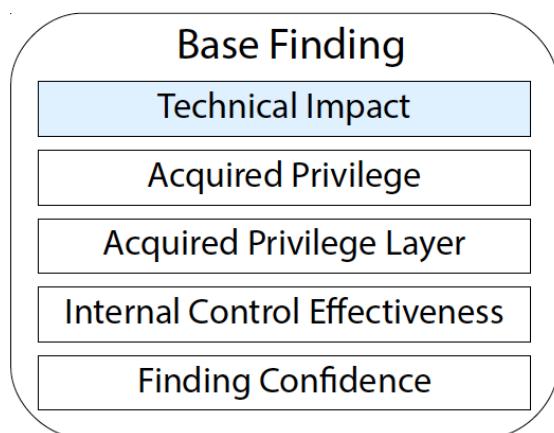
Calcolo del punteggio

- Ad ogni metrica è associata una **domanda a risposta multipla**
 - Ciascuna risposta fornisce un peso numerico
 - I singoli pesi sono poi aggregati in un risultato finale tramite una serie di formule



Metriche Base Finding

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

NOTA:

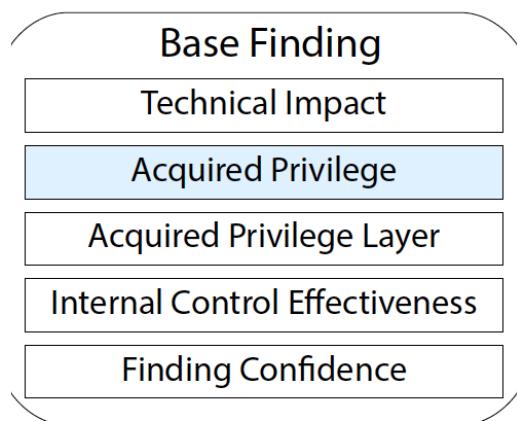
Alla metrica **TI** (Technical Impact) può essere assegnato uno dei cinque valori

C (Critical), H (High), M (Medium), L (Low), N (None)



Metriche Base Finding

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

NOTA:

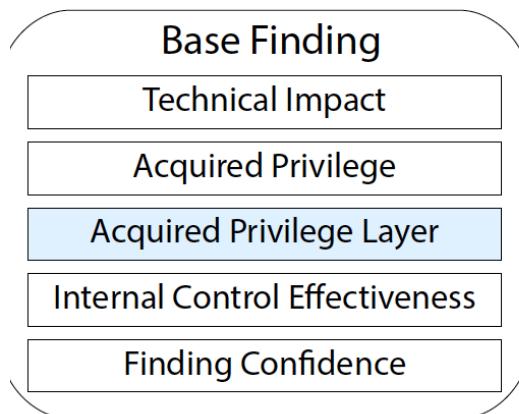
Alla metrica AP (Acquired Privilege) può essere assegnato uno dei cinque valori

A (Administrator), P (Partially Privileged User), RU (Regular User), L (Limited or Guest), N (None)



Metriche Base Finding

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

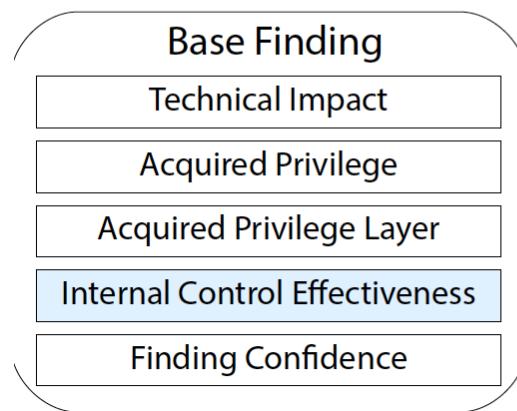
NOTA:

Alla metrica **AL** (Acquired Privilege Layer) può essere assegnato uno dei quattro valori
A (Application), **S** (System),
N (Network), **E** (Enterprise Infrastructure),



Metriche Base Finding

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

NOTA:

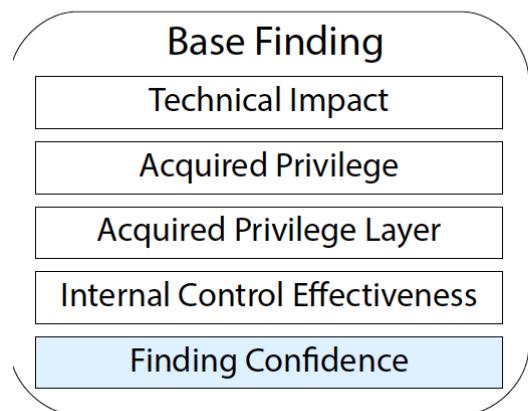
Alla metrica **IC** (Internal Control Effectiveness) può essere assegnato uno dei sei valori

N (None), **L** (Limited), **M** (Moderate),
I (Indirect), **B** (Best-Available), **C** (Complete)



Metriche Base Finding

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

NOTA:

Alla metrica **FC** (Finding Confidence) può essere assegnato uno dei tre valori

T (Proven True), **LT** (Proven Locally True), **F** (Proven False)



Calcolo del Punteggio Base Findings

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

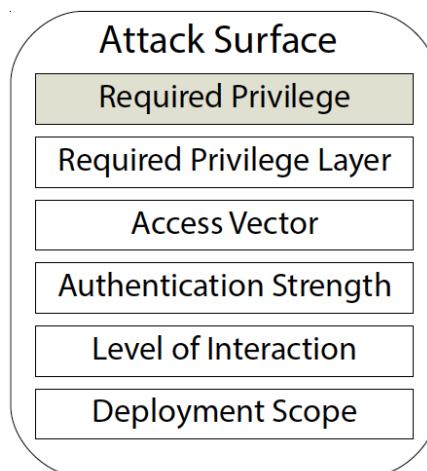
$$f(TI) = \begin{cases} 0 & \text{if } TI = 0 \\ 1 & \text{otherwise} \end{cases}$$

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



Metriche Attack Surface

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

NOTA:

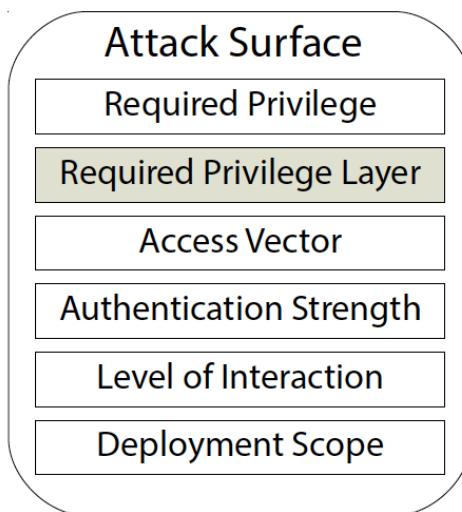
Alla metrica RP (Required Privilege) può essere assegnato uno dei cinque valori

N (None), L (Limited or Guest), RU (Regular User), P (Partially Privileged User), A (Administrator)



Metriche Attack Surface

A quale livello operazionale 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

NOTA:

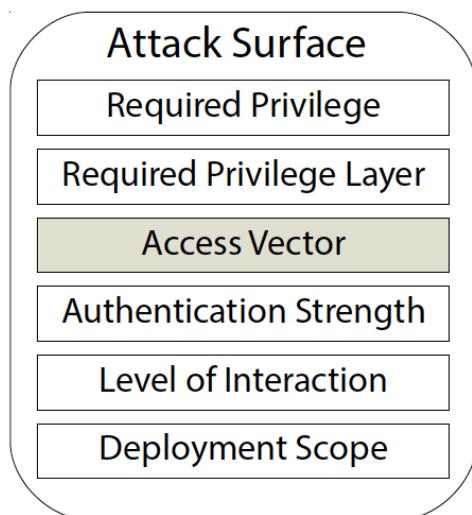
Alla metrica **RL** (Required Privilege Layer) può essere assegnato uno dei quattro valori

A (Application), **S** (System), **N** (Network), **E** (Enterprise Infrastructure)



Metriche Attack Surface

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

NOTA:

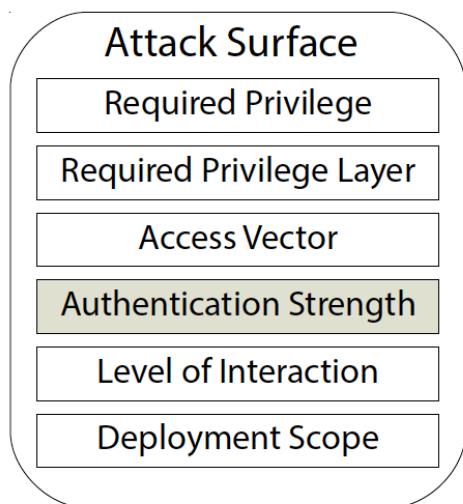
Alla metrica **AV (Access Vector)** può essere assegnato uno dei sei valori

I (Internet), R (Intranet), V (Private Network), A (Adjacent Network), L (Local), P (Physical)



Metriche Attack Surface

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

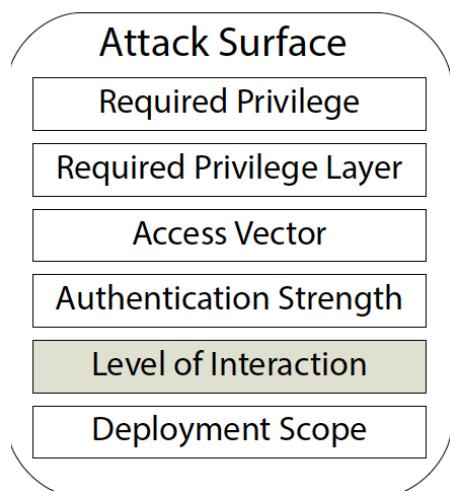
NOTA:

Alla metrica **AS** (Authentication Strength) può essere assegnato uno dei quattro valori **N (None)**, **W (Weak)**, **M (Moderate)**, **S (Strong)**



Metriche Attack Surface

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 (N)	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

NOTA:

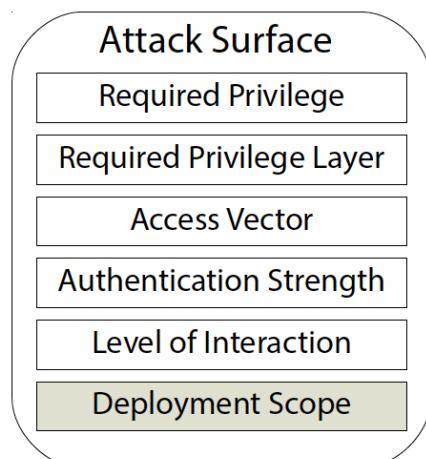
Alla metrica **LI** (Level of Interaction) può essere assegnato uno dei sei valori

A (Automated), **T** (Typical/Limited), **M** (Moderate),
N (Opportunistic), **H** (High), **NI** (No Interaction)



Metriche Attack Surface

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

NOTA:

Alla metrica **DS** (Deployment Scope) può essere
assegnato uno dei quattro valori
A (All), M (Moderate), R (Rare), P (Potentially Reachable)



Calcolo del Punteggio Attack Surface

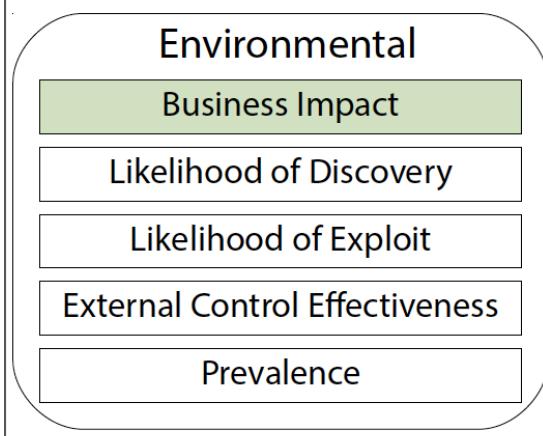
Il Punteggio Attack Surface è un valore tra 0 e 1,
calcolato nel modo seguente

$$AttackSurfaceScore = [20 * (RP + RL + AV) + 20 * SC + 15 * IN * 5 * AS] / 100.0$$



Metriche Environmental

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

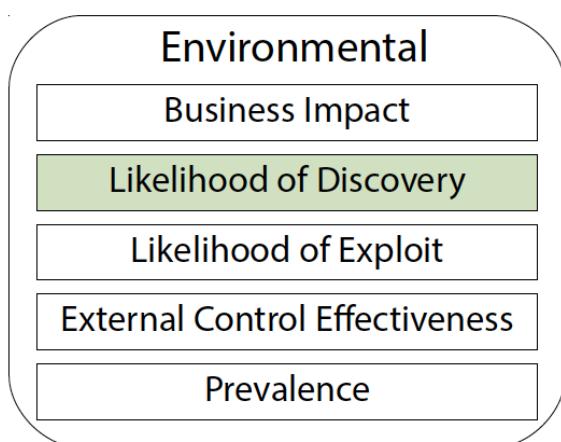
NOTA:

Alla metrica BI (Business Impact) può essere assegnato uno dei cinque valori C (Critical), H (High), M (Medium), L (Low), N (None)



Metriche Environmental

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

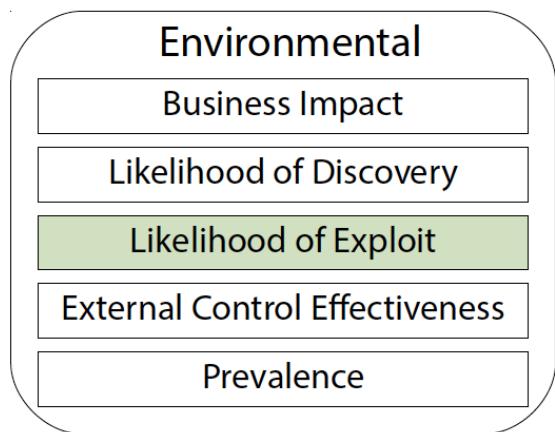
NOTA:

Alla metrica DI (Likelihood of DIcovery)
può essere assegnato uno dei tre valori
H (High), M (Medium), L (Low)



Metriche Environmental

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

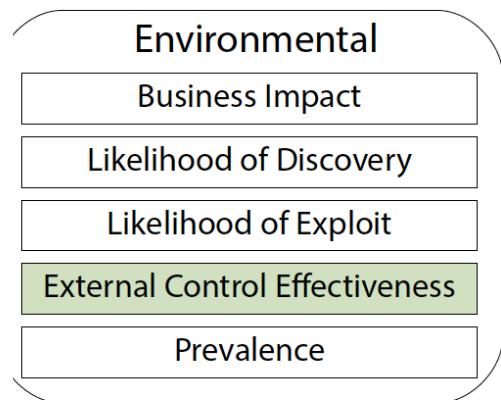
NOTA:

Alla metrica EX (Likelyhood of EXPloit) può essere assegnato uno dei quattro valori H (High), M (Medium), L (Low), N (None)



Metriche Environmental

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, aggribile 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

NOTA:

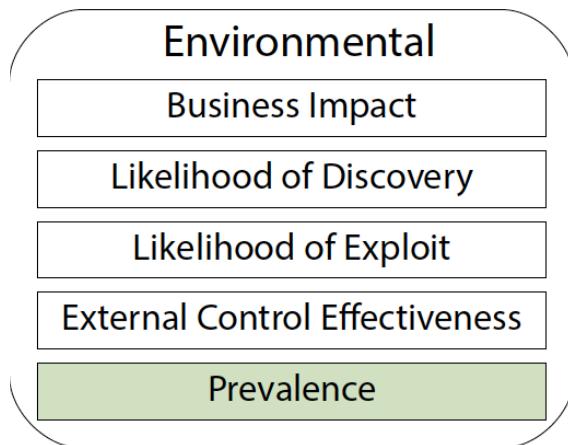
Alla metrica EC (External Control Effectiveness) può essere assegnato uno dei sei valori

N (None), L (Limited) M (Moderate),
I (Indirect), B (Best-Available), C (Complete)



Metriche Environmental

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

NOTA:

Alla metrica P (Prevalence) può essere assegnato uno dei quattro valori

W (Widespread), H (High), C (Common), L (Limited)



Calcolo del Punteggio Environmental

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

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

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



Calcolo del Punteggio Totale

Il Punteggio Totale è un valore tra 0 e 100,
dato dal prodotto dei tre punteggi parziali



Common Weaknesses Scoring System

- 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



Foglio di calcolo CWSS

Al seguente URL

<https://github.com/g4xyk00/CWSS-Calculator>

è presente un foglio di calcolo (non ufficiale) che consente di calcolare i punteggi CWSS con pochi click

The screenshot shows an Excel spreadsheet titled "Axcelsec - CWSS Calculator v1.0". The spreadsheet includes a ribbon bar with tabs like Home, Inserisci, Disegno, etc., and various toolbars for formatting and data manipulation. The main content consists of several tables and a summary section.

Tables:

- Base Finding:** A table with columns "Code", "Value", and "Weight".

	Code	Value	Weight
Technical Impact	TI	m	0,60
Acquired Privilege	AP	P	0,90
Acquired Privilege Layer	AL	A	1,00
Internal Control Effectiveness	IC	i	0,50
Finding Confidence	FC	T	1,00
Required Privilege	RP	n	1,00
Required Privilege Layer	RL	A	1,00
- Attack Surface:** A table with columns "Code", "Value", and "Weight".

	Code	Value	Weight
Access Vector	AV	I	1,00
Authentication Strength	AS	M	0,80
Level of Interaction	IN	A	1,00
Deployment Scope	SC	A	1,00
Business Impact	BI	h	0,90
Likelihood of Discovery	DI	h	1,00
Likelihood of Exploit	EX	l	0,20
- Environmental:** A table with columns "Code", "Value", and "Weight".

	Code	Value	Weight
External Control Effectiveness	EC	m	0,70
Prevalence	P	c	0,80

CWSS Vector: A formula box containing:
=(TI:m,0,6/AP:P,0,9/AL:A,1/IC:i,0,5/FC:T,1/
RP:n,1/RL:A,1/AV:I/AS:M,0,8/IN:A,1/SC:A,1/
BI:h,0,9/DI:h,1/EX:l,0,2/EC:m,0,7/P:c,0,8)

CWSS Score:

	Rating
Base Finding Subscore	41
Attack Surface Subscore	0,99
Environmental Subscore	0,53
Final Score	21,6

Note: Scoring Not Provided by CWE

None: 0
Low: 0,1 – 54,9
Medium: 55,0 – 64,9
High: 65,0 – 74,9
Critical: 75,0 – 100,0



Un esempio concreto

- Si consideri un'azienda che offre i propri prodotti al pubblico mediante un server Web che fornisce
 - Un catalogo dei prodotti
 - Un negozio elettronico
- Una delle applicazioni utilizzate ha una **debolezza**:
 - Permette ad un utente di registrare un account usando solo un indirizzo di posta elettronica
 - Sfruttando la debolezza, un utente può ottenere privilegi da amministratore per l'applicazione
- Si vogliono calcolare i **Punteggi CWSS** relativi a tale debolezza



Un esempio concreto

➤ Otteniamo maggiori dettagli sulla debolezza:

- Un attaccante non può avere controllo completo dell'applicazione ma può cancellare dati o utenti
- L'attacco non ha successo fino a quando l'amministratore non controlla le richieste di registrazione
- L'applicazione non ha meccanismi di protezione della debolezza
- La debolezza può essere evitata con poche righe di codice



Un esempio concreto

- Iniziamo a considerare le **Metriche Base Finding**

Metrica	Valore	Fattore
Technical Impact (TI)	High	0.9
Acquired Privilege (AP)	Administrator	1.0
Acquired Privilege Layer (AL)	Application	1.0
Internal Control Effectiveness (IC)	None	1.0
Finding Confidence (FC)	Proven True	1.0

- Vettore: TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0
- **Punteggio Base Finding: 96.0**
 - Il rischio legato alla debolezza è molto elevato
 - La debolezza esiste davvero
 - La debolezza non è protetta a livello di codice



Un esempio concreto

- Ora consideriamo le **Metriche Attack Surface**

Metrica	Valore	Fattore
Required Privilege (RP)	Guest	0.9
Required Privilege Layer (RL)	Application	1.0
Access Vector (AV)	Internet	1.0
Authentication Strength (AS)	None	1.0
Level of Interaction (IN)	Typical/Limited	1.0
Deployment Scope (SC)	All	1.0

- Vettore: RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,1.0/SC:A,1.0
- **Punteggio Attack Surface: 0.965 (vicino a 1.0)**
 - Le barriere poste ad un attaccante sono poche
 - Non sono richieste particolari credenziali
 - Attacco eseguibile da remoto



Un esempio concreto

- Infine consideriamo le **Metriche Environmental**

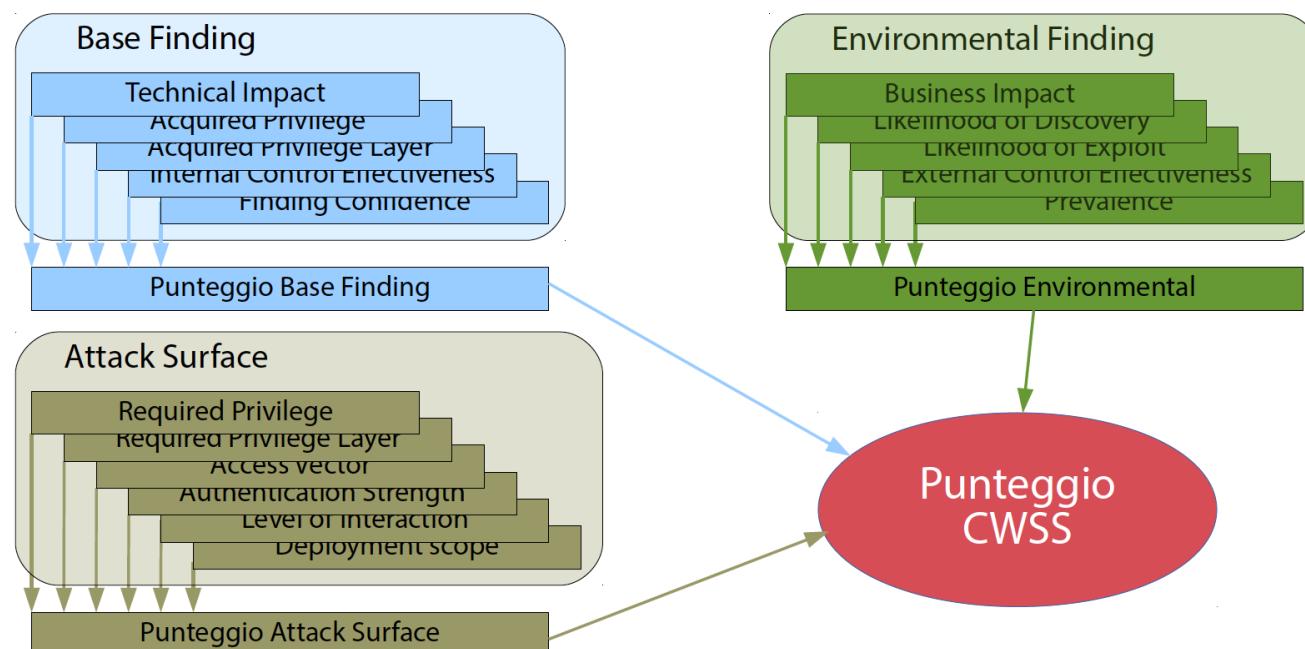
Metrica	Valore	Fattore
Business Impact (BI)	Critical	0.9
Likelihood of Discovery (DI)	High	1.0
Likelihood of Exploit (EX)	High	1.0
External Control Effectiveness (EC)	None	1.0
Prevalence (P)	Not Applicable	1.0

- Vettore: BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0
- **Punteggio Environmental: 1.0**
 - Le condizioni ambientali sono favorevoli ad un attaccante
 - Debolezza facile da scoprire e da sfruttare
 - Debolezza non protetta a livello operazionale (firewall, etc.)



Un esempio concreto

Il punteggio *CWSS* è $96.0 \times 0.9625 \times 1.0 = 92.6$



CVSS versus 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

