FORMAZIONE AGID – FORMEZ SULLA TRANSIZIONE DIGITALE DELLA PA

Progetto Informazione e formazione per la transizione digitale della PA nell'ambito del progetto «Italia Login – la casa del cittadino»

(A valere sul PON Governance e Capacità Istituzionale 2014-2020)

La sicurezza informatica nella PA

Webinar: Vulnerabilità Software

18 Novembre 2021

Dario Basti
CERT-AgID

…alcune Definizioni

Difetto: Una qualunque variazione/deviazione dalle specifiche.

Bug: Errore d'implementazione di una funzionalità/specifica.

Debolezza: Difetto o bug che potrebbe, in determinate circostanze, rendere reale una minaccia di sicurezza.

Vulnerabilità: Una debolezza presente, comprensibile e sfruttabile da un attaccante.

Minaccia: E' lo sfruttamento della Vulnerabilità per arrecare danno.

Vulnerabilità…

MITRE

Secondo la terminologia CVE elaborata dal MITRE:

Un **difetto** in un software, firmware, hardware o componente di servizio derivante da una **debolezza** che **può essere sfruttata da un attaccante**, causando un impatto negativo sulla **riservatezza, integrità o disponibilità** di uno o più componenti interessati.

Una vulnerabilità è la particolare condizione di un sistema informatico (o di alcuni sistemi) che consente ad un **attaccante** di realizzare potenzialmente quanto segue:

- **eseguire comandi** a nome di un altro utente;
- **ottenere l'accesso a dati** ai quali tale utente non può accedere;
- farsi passare **per un'altra entità**;
- **condurre un attacco** di tipo DoS (Denial of Service).

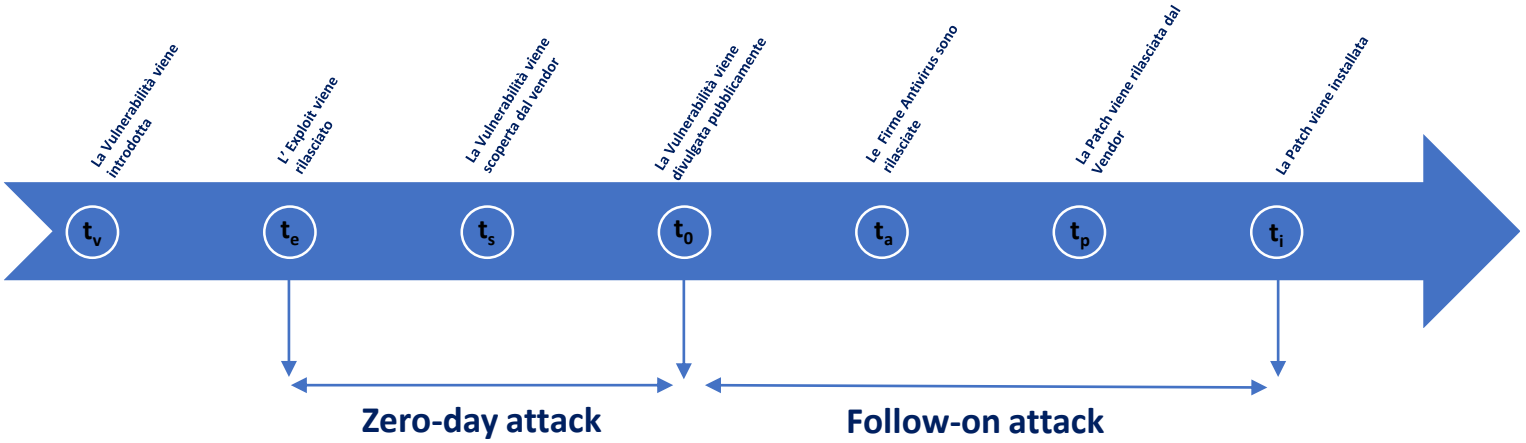
…Tipologie Vulnerabilità

All'interno di un **sistema informatico** possiamo classificare le seguenti macro tipologie di vulnerabilità che causano i 3/4 degli attacchi informatici odierni.

- **Errori di configurazione (maggiormente diffusi)**
  - *Utilizzo di password deboli*
  - *Porte aperte (FTP, Telnet)*
  - *Servizi non necessari pubblicati/esposti su internet*
  - *Dispositivi accessibili in rete (Stampanti, IoT)*
- **Vulnerabilità Software**
  - *Introduzione nuove funzionalità*
  - *Integrazione tra diverse tipologie di SW*
- **Aggiornamenti/Patch mancanti**
  - *Dispositivi non aggiornati*
  - *Patch non installate*

Ciclo di vita di una Vulnerabilità Software

Un fornitore rilascia una nuova versione di un software con una vulnerabilità(t_v)
Un attaccante rilascia un exploit senza notificare il vendor (t_e)
Il Vendor si accorge dell'exploit in autonomia o tramite segnalazioni (t_s)
La Vulnerabilità viene divulgata pubblicamente (t_0)
Gli Antivirus sono in grado di rilevare l'exploit (t_a)
Il Vendor rilascia la patch pubblicamente (t_p)
La patch viene installata sui sistemi (t_i)



Nell'intero periodo [t_e, t_0] l'attacco avviene in assenza di una sua pubblica conoscenza

Nell'intero periodo [t_0, t_i] l'attacco avviene in presenza di una sua pubblica conoscenza. La vulnerabilità è sfruttata pubblicamente.

Vulnerability Disclosure

Il processo di **Vulnerability disclosure** consiste nella condivisione di informazioni sulle vulnerabilità del software in modo che i suoi effetti negativi vengano ridotti o completamente eliminati.

ISO Riferimento Vulnerability disclosure **ISO/IEC 29147:2018**

Già nel 2017, la Commissione Europea nel suo Cybersecurity Strategy nomina esplicitamente, tra le iniziative per aumentare la resilienza europea agli attacchi informatici, la necessità di creare un processo di **Software Coordinated Vulnerability Disclosure** tra gli stati membri.

Tipologie/Modelli di **Vulnerability Disclosure**:

- **No Disclosure**: Le informazioni sulla vulnerabilità non vengono divulgate ma trattenute dei governi per scopi di sicurezza nazionale.
- **Responsible Disclosure**: la vulnerabilità viene segnalata lasciando al ricevente lasciando al ricevente/vendor del tempo per poter individuare ed applicare le opportune contromisure, prima di divulgarla pubblicamente.
- **Full Disclosure**: Vengono diffuse tutte le informazioni sulla vulnerabilità senza che sia stato individuato rimedio/patch.

Bug Bounty Program

- I **Bug Bounty Program** sono delle iniziative rivolte a programmatori ed ethical hacker che hanno la voglia e la capacità di trovare bug e problemi di sicurezza in tutte le tipologie di software(web, applicazioni, app, firmware).
- E' a tutti gli effetti un contratto tra l'azienda che propone il suo bug bounty program e chiunque si presti a segnalare bug, in particolar modo, forme di vulnerabilità che possono mettere in pericolo la sicurezza del sistema oggetto dell'accordo.
- Segnalare un bug/vulnerabilità consente ai partecipanti di ricevere dei riconoscimenti in denaro oltre che essere elencati nel **Hall of Fame** dell'azienda che ha proposto il programma.

Catalogo Vulnerabilità

Molti team/ricercatori di sicurezza scoprono e divulgano le vulnerabilità di sicurezza.

La creazione di un catalogo delle Vulnerabilità è quindi un'attività necessaria e molto importante in quanto permette:

- **Identificazione/Enumerazione**: Costruzione di una tupla univoca per ogni vulnerabilità es: (*id, tipo, vettore attacco, minaccia, exploit*).
- **Memorizzazione/Catalogazione**: inserimento della tupla in un apposito database.

Team Indipendenti

→ Catalogazione Indipendente

→ Archivi Eterogenei

→ Formati diversi

→ Non interoperabili

Le Vulnerabilità non possono essere catalogate in modo diverso!!!

Common Vulnerabilities and Exposures (CVE)

CVE- Catalogo uniforme di vulnerabilità introdotto dal MITRE: ente no-profit americano
https://cve.mitre.org/

Ogni Vulnerabilità è rappresentata da un identificatore (**CVE-id**)

ANNO (4 digit): anno in cui è stata scoperta la vulnerabilità

Il formato dell'identificatore è **CVE-ANNO-NUMERO**

NUMERO: numero intero progressivo

Database singole vulnerabilità

..Common Vulnerabilities and Exposures (CVE)

CVE-2021-2034

CVE-id

Descrizione vulnerabilità

MITRE ..Common Vulnerabilities and Exposures (CVE)

Scheda dettaglio CVE-2021-2034

CVE-ID	
CVE-2021-2034	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Vulnerability in the Oracle Common Applications Calendar product of Oracle E-Business Suite (component: Tasks). Supported versions that are affected are 12.1.1-12.1.3. Easily exploitable vulnerability allows unauthenticated attacker with network access via HTTP to compromise Oracle Common Applications Calendar. Successful attacks require human interaction from a person other than the attacker and while the vulnerability is in Oracle Common Applications Calendar, attacks may significantly impact additional products. Successful attacks of this vulnerability can result in unauthorized access to critical data or complete access to all Oracle Common Applications Calendar accessible data as well as unauthorized update, insert or delete access to some of Oracle Common Applications Calendar accessible data. CVSS 3.1 Base Score 8.2 (Confidentiality and Integrity impacts). CVSS Vector: (CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N).

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://www.oracle.com/security-alerts/cpujan2021.html
- URL:https://www.oracle.com/security-alerts/cpujan2021.html

Assigning CNA

Oracle

Date Record Created

20201209	Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Phase (Legacy)

Assigned (20201209)

Votes (Legacy)

Comments (Legacy)

Proposed (Legacy)

N/A

This is a record on the CVE List, which provides common identifiers for publicly known cybersecurity vulnerabilities.

SEARCH CVE USING KEYWORDS: Submit

You can also search by reference using the CVE Reference Maps.

For More Information: CVE Request Web Form (select "Other" from dropdown)

Common Weakness Enumeration (CWE)

https://cwe.mitre.org/

CWE: sistema di classificazione delle **debolezze** software e hardware

Top 25 – Software Weakness

Rank	ID	Name	Score	2020 Rank Change
[1]	CWE-787	Out-of-bounds Write	65.93	+1
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	46.84	-1
[3]	CWE-125	Out-of-bounds Read	24.9	+1
[4]	CWE-20	Improper Input Validation	20.47	-1
[5]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	19.55	+5
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	19.54	0
[7]	CWE-416	Use After Free	16.83	+1
[8]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.69	+4
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	14.46	0
[10]	CWE-434	Unrestricted Upload of File with Dangerous Type	8.45	+5
[11]	CWE-306	Missing Authentication for Critical Function	7.93	+13
[12]	CWE-190	Integer Overflow or Wraparound	7.12	-1
[13]	CWE-502	Deserialization of Untrusted Data	6.71	+8
[14]	CWE-287	Improper Authentication	6.58	0
[15]	CWE-476	NULL Pointer Dereference	6.54	-2
[16]	CWE-798	Use of Hard-coded Credentials	6.27	+4
[17]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	5.84	-12
[18]	CWE-862	Missing Authorization	5.47	+7
[19]	CWE-276	Incorrect Default Permissions	5.09	+22
[20]	CWE-200	Exposure of Sensitive Information to an Unauthorized Actor	4.74	-13
[21]	CWE-522	Insufficiently Protected Credentials	4.21	-3
[22]	CWE-732	Incorrect Permission Assignment for Critical Resource	4.2	-6
[23]	CWE-611	Improper Restriction of XML External Entity Reference	4.02	-4
[24]	CWE-918	Server-Side Request Forgery (SSRF)	3.78	+3
[25]	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	3.58	+6

Most Important – Hardware Weakness

CWE-1189	Improper Isolation of Shared Resources on System-on-a-Chip (SoC)
CWE-1191	On-Chip Debug and Test Interface With Improper Access Control
CWE-1231	Improper Prevention of Lock Bit Modification
CWE-1233	Security-Sensitive Hardware Controls with Missing Lock Bit Protection
CWE-1240	Use of a Cryptographic Primitive with a Risky Implementation
CWE-1244	Internal Asset Exposed to Unsafe Debug Access Level or State
CWE-1256	Improper Restriction of Software Interfaces to Hardware Features
CWE-1260	Improper Handling of Overlap Between Protected Memory Ranges
CWE-1272	Sensitive Information Uncleared Before Debug/Power State Transition
CWE-1274	Improper Access Control for Volatile Memory Containing Boot Code
CWE-1277	Firmware Not Updateable
CWE-1300	Improper Protection of Physical Side Channels

MITRE

Common Attack Pattern Enumeration and Classification (CAPEC)

https://capec.mitre.org/

CAPEC: dizionario completo di modelli noti di attacco utilizzati dagli avversari per sfruttare le debolezze note

3000 - Domains of Attack

- Software - *(513)*
- Hardware - *(515)*
- Communications - *(512)*
- Supply Chain - *(437)*
- Social Engineering - *(403)*
- Physical Security - *(514)*

3000 - Domains of Attack

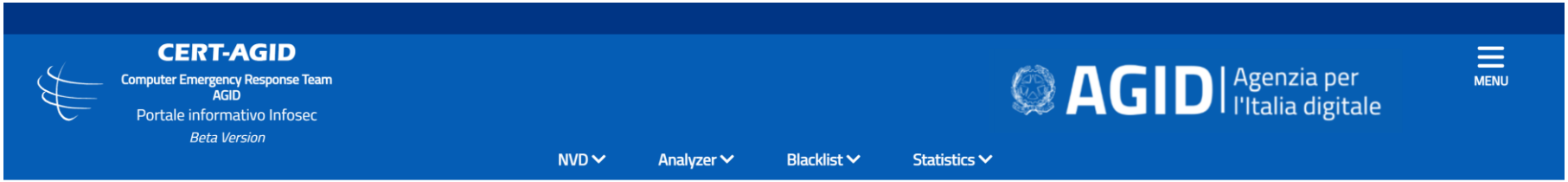
- Software - *(513)*
  - Exploitation of Trusted Identifiers - *(21)*
  - Exploiting Trust in Client - *(22)*
  - Forced Deadlock - *(25)*
  - Leveraging Race Conditions - *(26)*
  - Fuzzing - *(28)*
  - Manipulating State - *(74)*
  - Adversary in the Middle (AiTM) - *(94)*
  - Brute Force - *(112)*
    - Encryption Brute Forcing - *(20)*
    - Password Brute Forcing - *(49)*
      - Dictionary-based Password Attack - *(16)*
      - Rainbow Table Password Cracking - *(55)*
      - Password Spraying - *(565)*
      - Try Common or Default Usernames and Passwords - *(70)*

Related Weaknesses

CWE-ID	Weakness Name
521	Weak Password Requirements
262	Not Using Password Aging
263	Password Aging with Long Expiration
257	Storing Passwords in a Recoverable Format
654	Reliance on a Single Factor in a Security Decision
307	Improper Restriction of Excessive Authentication Attempts
308	Use of Single-factor Authentication
309	Use of Password System for Primary Authentication

CERT-AgID Infosec

https://infosec.cert-agid.gov.it/



Infosec

2021 Top CVE - Vendors

#	Vendor	CVE(s)
#1	Qualcomm	51417
#2	Microsoft	1466
#3	Cisco	1104
#4	Netgear	399
#5	Ibm	362
#6	Siemens	341
#7	Fedoraproject	292
#8	Juniper	254
#9	Google	213
#10	Debian	195

2021 Top CVE - Products

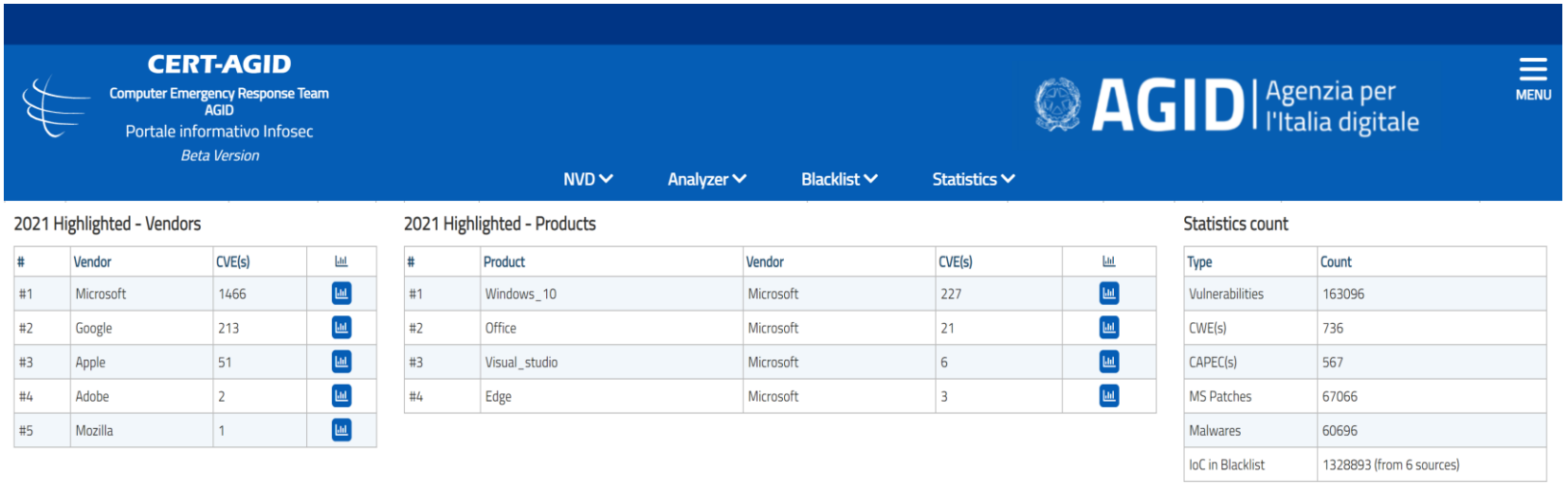
#	Product	Vendor	CVE(s)
#1	Fedora	Fedoraproject	292
#2	Windows_10	Microsoft	227
#3	Windows	Microsoft	225
#4	Windows_server_2016	Microsoft	225
#5	Windows_server_2019	Microsoft	208
#6	Android	Google	204
#7	Debian_linux	Debian	195
#8	Windows_server_2012	Microsoft	141
#9	Windows_rt_8.1	Microsoft	133
#10	Windows_server_2008	Microsoft	127

Top 10 Malware type

#	Malware type	Count
#1	Linux	8
#2	MSIL	7
#3	Razy	2
#4	Downloader	2
#5	Kryptik	2
#6	a	2
#7	Donut	1
#8	PowerShell	1
#9	Gafgyt	1
#10	Trj	1

...CERT-AgID Infosec

https://infosec.cert-agid.gov.it/



MITRE

Limiti del sistema (CVE)

Il sistema CVE enumera esclusivamente le vulnerabilità

Dati due CVE è difficile capire quale sia quello più urgente da gestire

Il sistema CVE non misura l'impatto di una vulnerabilità(non è nato per questo scopo)

Un determinato CVE può avere un **impatto diverso nel tempo** e un **impatto diverso su sistemi diversi**

Common Vulnerability Score System (CVSS)

https://www.first.org/cvss/
https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

Il Common Vulnerability Score System (CVSS) stima la gravità di un vulnerabilità

Ratings	CVSS score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 – 10.0

L'ultima versione del CVSS è la 3.1 utilizzata dai principali vendor.

Ad ogni CVE viene assegnato un punteggio (score) da 0 a 10

0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10

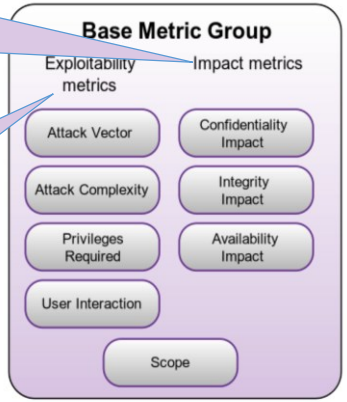
Il punteggio CVSS è composto da 3 gruppi di metriche (**Base, Temporal, Environmental**)

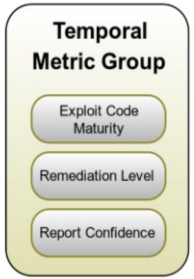
…Common Vulnerability Score System (CVSS)

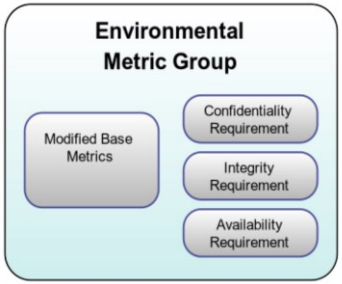
https://www.first.org/cvss/
https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf

Le metriche «**Impacts**» riflettono la diretta conseguenza di un exploit di successo.. ***Cosa permette di ottenere?***

Le metriche «**Exploitability**» riflettono la facilità e i mezzi tecnici con cui la vulnerabilità può essere sfruttata.. ***Da dove si riesce a sfruttare? Quanto è semplice metterla in atto?***







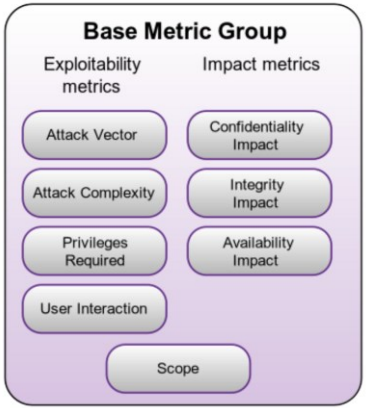
Le metriche «**Base**» stimano la gravità della vulnerabilità in sé basandosi sulle sue caratteristiche intrinseche a prescindere da fattori temporali e ambientali. È composta da due insiemi di metriche: **l'Exploitability Metrics e Impact Metrics**.

Le metriche «**temporali**» stimano la gravità di una vulnerabilità dal punto di vista temporale. ***E' disponibile un exploit? Sono disponibili patch ufficali?*** Ad esempio, un exploit di facile utilizzo **aumenta** il punteggio CVSS, mentre la presenza di una patch ufficiale lo **diminuisce**.

Le metriche «**ambientali**» stimano la vulnerabilità dal punto di vista ambientale. ***Qual è la conseguenza dell'exploit su cose/persone? Quali sistemi della mia infrastruttura sono vulnerabili?*** La metrica prende quindi in considerazione la presenza o meno di **controlli di sicurezza** che possono **mitigare** alcune o tutte le conseguenze di un attacco riuscito. Prende inoltre in considerazione l'importanza del sistema vulnerabile all'interno di un'infrastruttura tecnologica.

…Common Vulnerability Score System (CVSS)

Attack Vector		
Metric Value	**Description**	**Numerical Value**
Network (N)	L'attaccante può sfruttare la vulnerabilità da remoto.	0,85
Adjacent (A)	L'attaccante deve avere accesso al dominio di broadcast o di collisione del sistema. Es. (VPN, stessa subnet).	0,62
Local (L)	L'attaccante deve avere accesso ad un account sul sistema. (Tastiera, Documenti malevoli, Social Engineering).	0,55
Physical (P)	L'attacco richiede che l'attaccante tocchi o manipoli fisicamente il componente vulnerabile. (Es. Inserimento Pendrive USB).	0,2
Attack Complexity		
Low (L)	Nessun condizione/configurazione di accesso speciale. Attacco facilmente ripetibile.	0,77
High (H)	L'attacco richiede particolari condizioni e conoscenza del target.	0,44
Privileges Required		
None(N)	L'attacco non richiede nessun tipo di privilegio (autenticazione/autorizzazione).	0,85
Low (L)	L'attacco richiede privilegi di un utente base del sistema.	0,62 (0,68 se Scope è C)
High (H)	L'attacco richiede privilegi amministrativi sul sistema.	0,27 (0,5 se Scope è C)
User Interaction		
None(L)	La vulnerabilità può essere sfruttata senza l'intervento di alcun utente.	0,85
Required (R)	La vulnerabilità per essere sfruttata richiede l'intervento utente.	0,62



Scope	
Metric Value	**Description**
Unchanged(U)	Non si verifica una modifica d'ambito. La vulnerabilità non influisce su componenti fuori l'ambito di sicurezza del componente vulnerabile.
Changed (C)	Si verifica una modifica d'ambito. La vulnerabilità ha impatto su componenti fuori l'ambito di sicurezza del componente vulnerabile.

Confidentiality Impact		
Metric Value	**Description**	**Numerical Value**
High(H)	É possibile estrarre e divulgare tutti i dati presenti nel sistema.	0,56
Low (L)	É possibile estrarre e divulgare solo un sotto-insieme dei dati presenti nel sistema.	0,22
None (N)	Nessun impatto.	0

Integrity Impact		
Metric Value	**Description**	**Numerical Value**
High(H)	É possibile modificare tutti i dati presenti nel sistema.	0,56
Low (L)	É possibile modificare solo alcuni dati presenti nel sistema.	0,22
None (N)	Nessun impatto.	0

Availability Impact		
Metric Value	**Description**	**Numerical Value**
High(H)	É possibile ridurre completamente le prestazioni e/o i servizi erogati dal sistema.	0,56
Low (L)	É possibile ridurre parzialmente le prestazioni e/o i servizi erogati dal sistema.	0,22
None (N)	Nessun Impatto	0

CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N

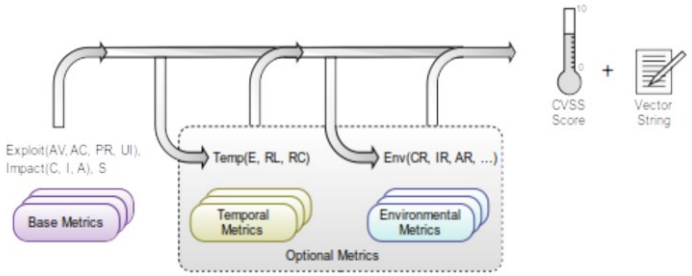
…Common Vulnerability Score System (CVSS)

Exploit Code Maturity		
Metric Value	**Description**	**Numerical Value**
Not Defined (X)	Si assegna questo valore quando non si hanno sufficienti informazioni per scegliere altri valori. In pratica ha lo stesso valore di High(H).	1
High (H)	Esiste un codice exploit, documentato, affidabile, funzionale, automatizzabile e facilmente utilizzabile. L'exploit funziona sempre in ogni situazione.	1
Functional (F)	Esiste un codice exploit funzionale utilizzabile nella maggior parte dei casi dove esiste la vulnerabilità.	0,97
Proof-of-concept (P)	Esiste un codice exploit funzionale Proof-of-concept, una demo. L'exploit non funziona in tutte le situazione e skill per essere utilizzato/adattato.	0,94
Unproven(U)	Non è disponibile nessuno codice exploit oppure l'exploit è solo teorico.	0,91

Remediation Level		
Metric Value	**Description**	**Numerical Value**
Not Defined (X)	Si assegna questo valore quando non si hanno sufficienti informazioni per scegliere altri valori. In pratica ha lo stesso valore di Unavailable(U).	1
Unavailable (U)	Non ci sono soluzioni/remediation disponibili o sono impossibili da applicare.	1
Workaround (W)	E' disponibile una soluzione/remediation non ufficiale non fornita dal fornitore/vendor. Insieme di step che l'utente può applicare per cercare di mitigare la vulnerabilità.	0,97
Temporary Fix (T)	E' disponibile una soluzione/remediation ufficiale fornita dal fornitore/vendor ma temporanea (hotfix).	0,96
Official Fix (O)	E' disponibile una soluzione/remediation ufficiale fornita dal fornitore scaricabile tramite aggiornamento ufficiale.	0,95

Qual è lo stato attuale delle tecniche di sfruttamento della vulnerabilità?

Report Confidence		
Metric Value	**Description**	**Numerical Value**
Not Defined (X)	Si assegna questo valore quando non si hanno sufficienti informazioni per scegliere altri valori. In pratica ha lo stesso valore di Confirmed(C).	1
Confirmed (C)	La Vulnerabilità è confermata dal vendor.	1
Reasonable (R)	Dettagli sulla vulnerabilità sono pubblicati, ma i ricercatori hanno ancora qualche dubbio sulla causa della vulnerabilità o non hanno accesso al codice completo.	0,96
Unknown (U)	Ci sono segnalazioni di impatti che indicano la presenza della vulnerabilità. Il rapporti indicano che la causa della vulnerabilità è sconosciuta. I rapporti sono discordanti sulla causa della vulnerabilità e sui suoi impatti.	0,92

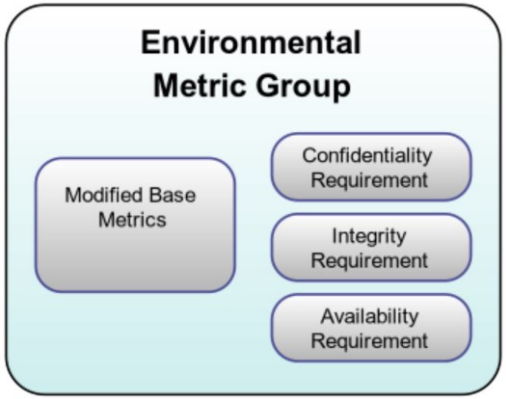


$$TemporalScore = roundTo1Decimal\left(BaseScore * Exploitab * RemedLvl * ReportConf\right)$$

…Common Vulnerability Score System (CVSS)

Le metriche «environmental» consentono all'analista di personalizzare il punteggio CVSS a seconda dell'importanza, per l'organizzazione, del risorsa IT interessata dalla vulnerabilità

Le metriche «**Modified Base Metric**» consentono all'analista di sovrascrivere le singole «metriche di base» in relazione alle specifiche caratteristiche dell'ambiente dell'organizzazione (architettura, configurazioni, controlli sicurezza).



Le Metriche «**Security Requirements**» consentono all'analista di personalizzare il punteggio CVSS in termini di riservatezza, integrità e disponibilità.

Modified Base Metric	
Metric Value	**Numerical Value**
Modified Attack Vector (MAV)	Gli stessi valori della metrica di base corrispondente più il valore Not Defined che è il default
Modified Attack Complexity (MAC)	
Modified Privileges Required (MPR)	
Modified User Interaction (MUI)	
Modified Scope (MS)	
Modified Confidentiality (MC)	
Modified Integrity (MI)	
Modified Availability (MA)	

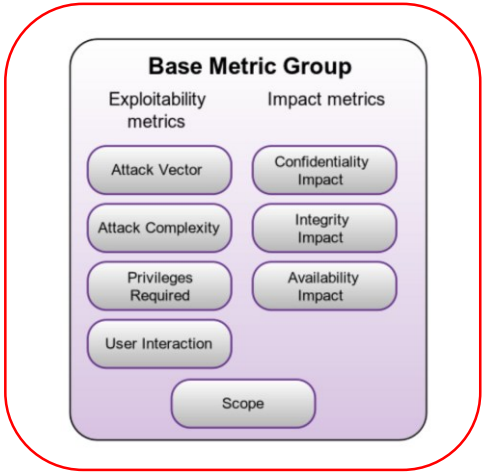
Per chi si vuole divertire di seguito il link al calcolatore.
https://www.first.org/cvss/calculator/3.1

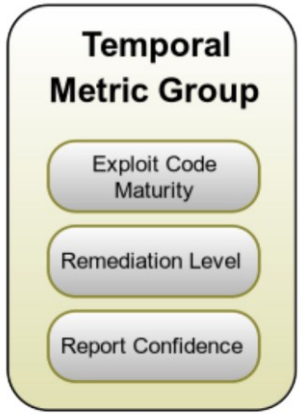
Security Requirements (CR,IR,AR)		
Metric Value	**Description**	**Numerical Value**
Not Defined (X)	Si assegna questo valore quando non si hanno sufficienti informazioni per scegliere altri valori. In pratica ha lo stesso valore di Medium(M).	1
High (H)	Perdita di [Riservatezza | Integrità | Disponibilità] è probabile che abbia un **effetto negativo catastrofico** sull'organizzazione o sugli individui dell'organizzazione (ad es. dipendenti, clienti).	1,5
Medium (M)	Perdita di [Riservatezza | Integrità | Disponibilità] rischia di avere un **serio effetto negativo** sull'organizzazione o sugli individui dell'organizzazione (ad es. dipendenti, clienti).	1
Low (L)	Perdita di [Riservatezza | Integrità | Disponibilità] è probabile che abbia solo **un limitato effetto negativo** sull'organizzazione o sugli individui dell' organizzazione (ad es. dipendenti, clienti).	0,5

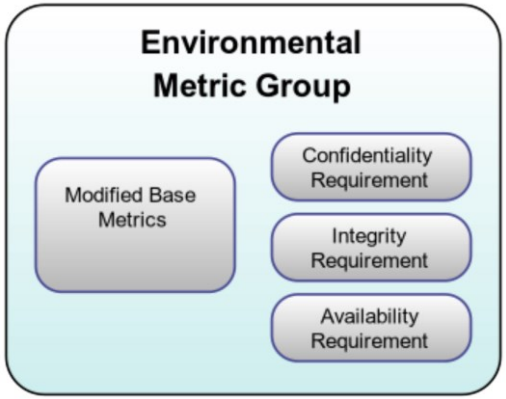
Chi calcola il punteggio CVSS?

Per calcolare il punteggio CVSS sono obbligatorie solo le metriche base

Dipende dalla tipologia di punteggio..



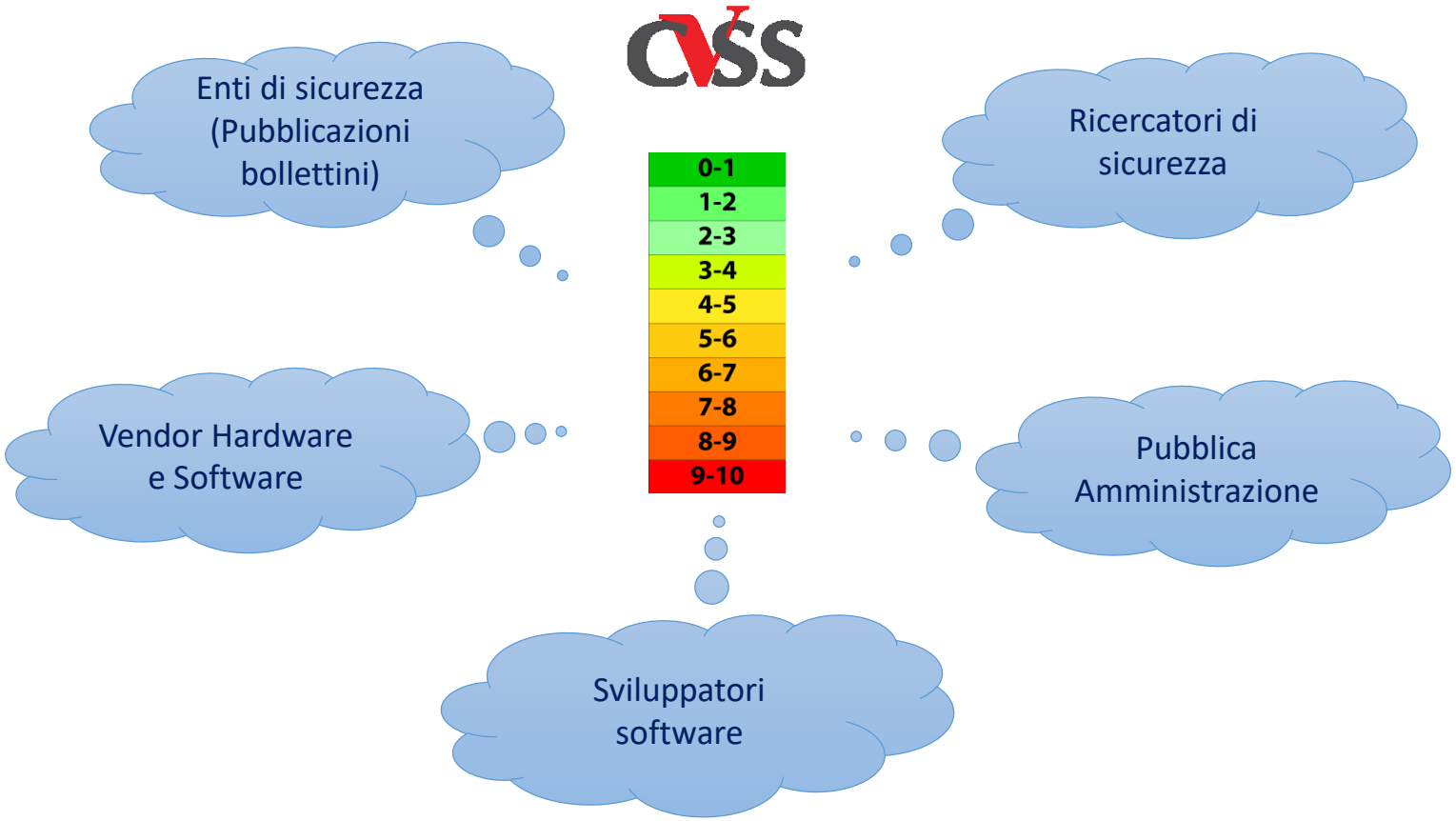




Vendor Hardware e Software i quali conoscono molto bene i dettagli dei loro prodotti e le dinamiche delle vulnerabilità

Amministratori ed utenti delle organizzazioni i quali conoscono molto bene l'ambiente in cui è installato il prodotto sia hardware che software vulnerabile.

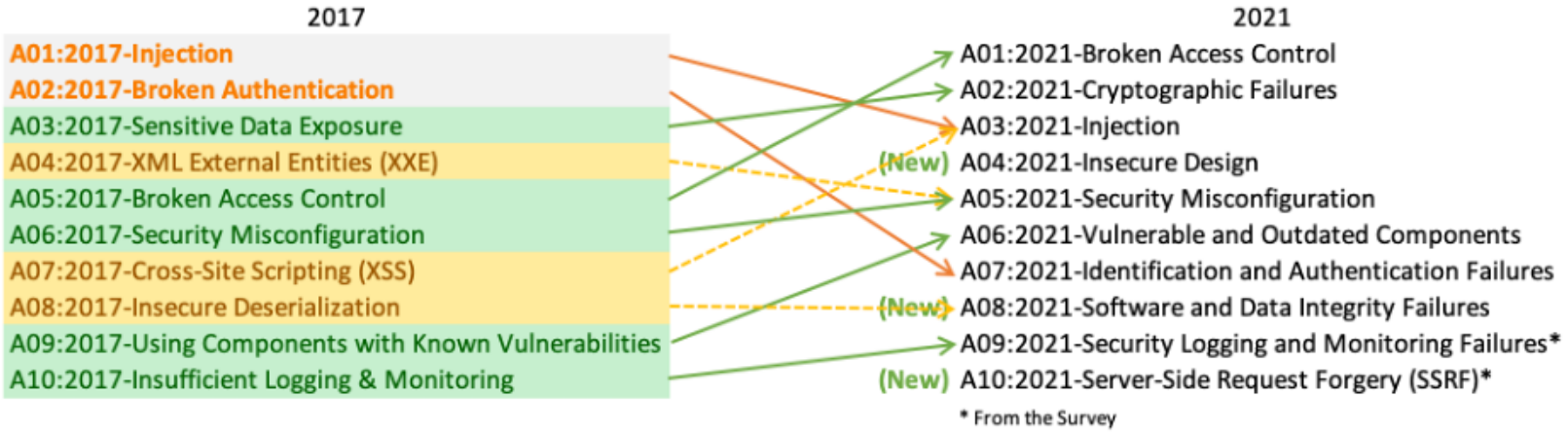
Chi usa il punteggio CVSS?



Top 10 - OWASP

L'Open Web Application Security Project® (OWASP) è una fondazione senza scopo di lucro che lavora per migliorare la sicurezza del software.

- OWASP è la principale fonte per gli sviluppatori per proteggere le applicazioni web.



A2:2017 – Broken Authentication

A2. Broken Authentication: Le procedure applicative relative all'autenticazione e alla gestione della sessione sono spesso implementate in modo non corretto, permettendo agli attaccanti di compromettere password, chiavi, token di sessione o sfruttare debolezze implementative per assumere l'identità di altri utenti.

A2:2017-Broken Authentication

Languages: [en] de

← A1:2017-Injection

OWASP Top Ten 2017
PDF version

A3:2017-Sensitive Data Exposure →

Threat Agents / Attack Vectors		Security Weakness		Impacts	
App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Attackers have access to hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.		The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.		Attackers have to gain access to only a few accounts, or just one admin account to compromise the system. Depending on the domain of the application, this may allow money laundering, social security fraud, and identity theft, or disclose legally protected highly sensitive information.	

..A2:2017 – Broken Authentication

Is the Application Vulnerable?	How to Prevent
Confirmation of the user's identity, authentication, and session management are critical to protect against authentication-related attacks. There may be authentication weaknesses if the application: * Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords. * Permits brute force or other automated attacks. * Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin". * Uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers", which cannot be made safe. * Uses plain text, encrypted, or weakly hashed passwords (see A3:2017-Sensitive Data Exposure). * Has missing or ineffective multi-factor authentication. * Exposes Session IDs in the URL (e.g., URL rewriting). * Does not rotate Session IDs after successful login. * Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.	* Where possible, implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks. * Do not ship or deploy with any default credentials, particularly for admin users. * Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords. * Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies. * Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes. * Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected. * Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.

..A2:2017 – Broken Authentication

Is the Application Vulnerable?	How to Prevent
Confirmation of the user's identity, authentication, and session management are critical to protect against authentication-related attacks. There may be authentication weaknesses if the application: * Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords. * Permits brute force or other automated attacks. * Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin". * Uses weak or ineffective credential recovery and forgot-password processes, such as "knowledge-based answers", which cannot be made safe. * Uses plain text, encrypted, or weakly hashed passwords (see A3:2017-Sensitive Data Exposure). * Has missing or ineffective multi-factor authentication. * Exposes Session IDs in the URL (e.g., URL rewriting). * Does not rotate Session IDs after successful login. * Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.	* Where possible, implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks. * Do not ship or deploy with any default credentials, particularly for admin users. * Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords. * Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies. * Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes. * Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected. * Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.

..A2:2017 – Broken Authentication

OWASP Open Web Application Security Project

OWASP TOP10

Example Attack Scenarios

Scenario #1: Credential stuffing, the use of lists of known passwords, is a common attack. If an application does not implement automated threat or credential stuffing protections, the application can be used as a password oracle to determine if the credentials are valid.

Scenario #2: Most authentication attacks occur due to the continued use of passwords as a sole factor. Once considered best practices, password rotation and complexity requirements are viewed as encouraging users to use, and reuse, weak passwords. Organizations are recommended to stop these practices per NIST 800-63 and use multi-factor authentication.

Scenario #3: Application session timeouts aren't set properly. A user uses a public computer to access an application. Instead of selecting "logout" the user simply closes the browser tab and walks away. An attacker uses the same browser an hour later, and the user is still authenticated.

References

OWASP

* OWASP Proactive Controls: Implement Digital Identity
* OWASP Application Security Verification Standard: V2 Authentication
* OWASP Application Security Verification Standard: V3 Session Management
* OWASP Testing Guide: Identity, Authentication
* OWASP Cheat Sheet: Authentication
* OWASP Cheat Sheet: Credential Stuffing
* OWASP Cheat Sheet: Forgot Password
* OWASP Cheat Sheet: Session Management
* OWASP Automated Threats Handbook

External

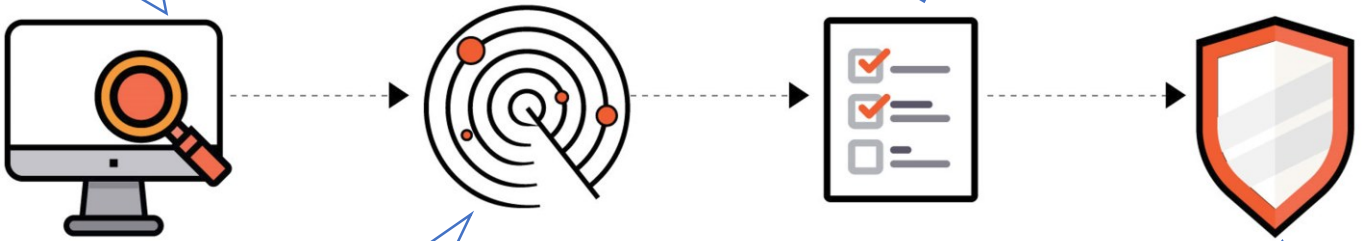
* NIST 800-63b: 5.1.1 Memorized Secrets
* CWE-287: Improper Authentication
* CWE-384: Session Fixation

Vulnerability Assessment

Asset Discovery
Identificazione delle risorse

Assessment/Report
Report sulle vulnerabilità rilavate (CVSS, Assett coinvolti)

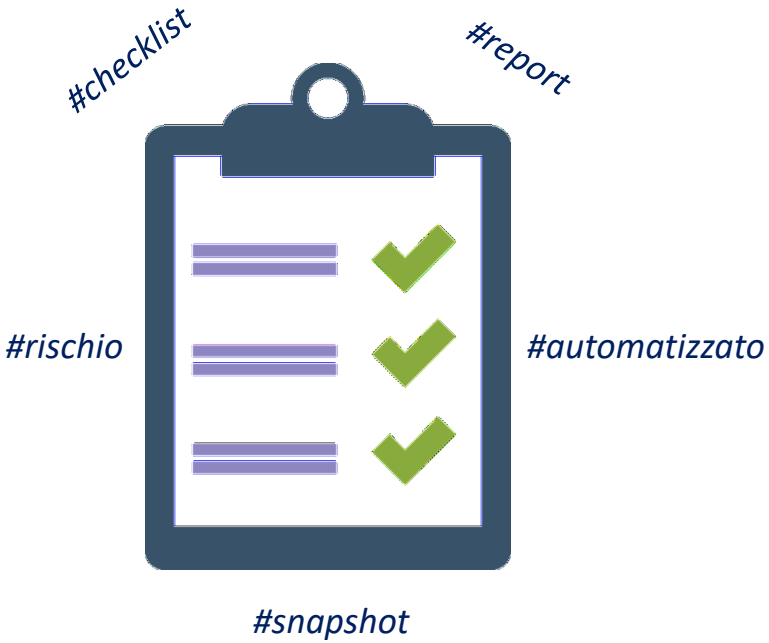
Tools VA
Open Source:
OpenVAS
Commerciali:
Nessus, Acunetix



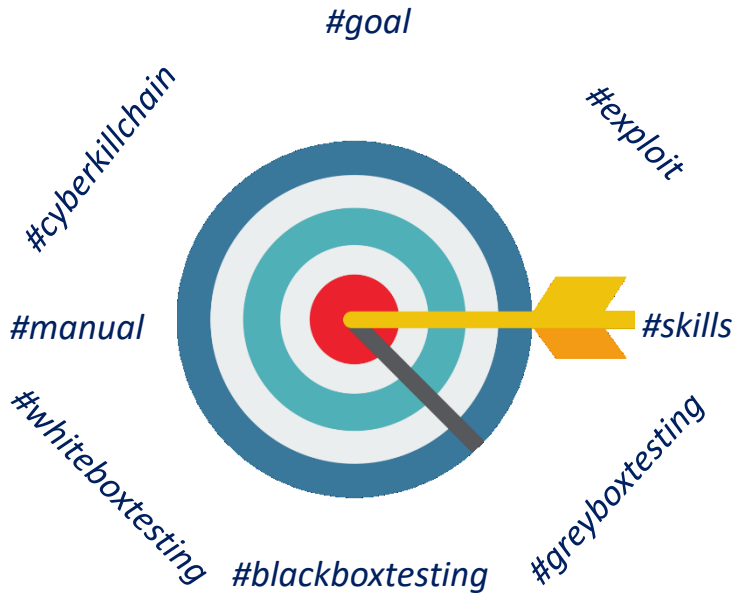
Scanning
Ricerca delle vulnerabilità nei SO, CMS, Plug-in, Server, Database

Remediation
Installazione Patch/Controlli Sicurezza/Azioni Mitigazione

Vulnerability Assessment VS Penetration Testing



«Hanno obiettivi diversi»!!!



Un Vulnerability Assessment è «**Document Oriented**» l'output è un Report con tutte le vulnerabilità rilevate e la loro gravità.

Un Penetration Test è «**Goal Oriented**» è la simulazione di un attacco informatico che cerca di sfruttare le vulnerabilità rilevate. Verifica in tempo reale la validità dei miei sistemi/controlli di sicurezza.

CERT-AGID

Uno sguardo ai server della PA..

SHODAN

- 22.553 • Elenco domini censiti IPA
- 401.532 • Elenco sottodomini ottenuti da fonti OSINT
- 62.749 • Indirizzi IP (IPv4) ottenuti

#dispositivi

#motorericerca

#internet

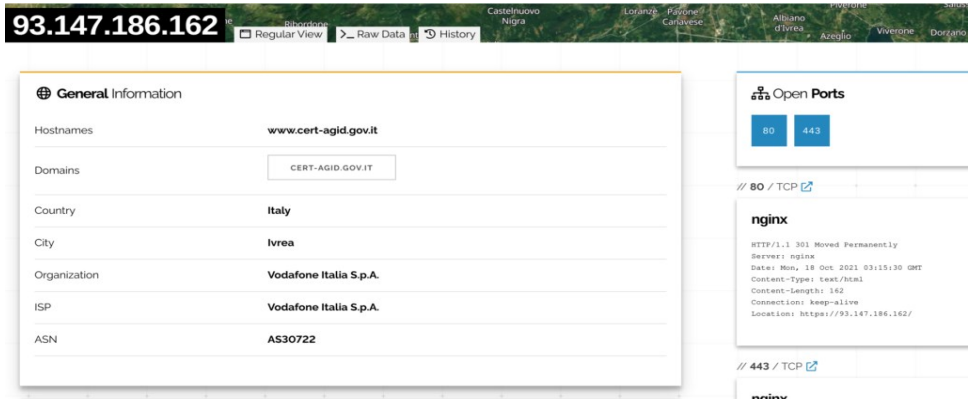
SHODAN Computer Search Engine

#metadati

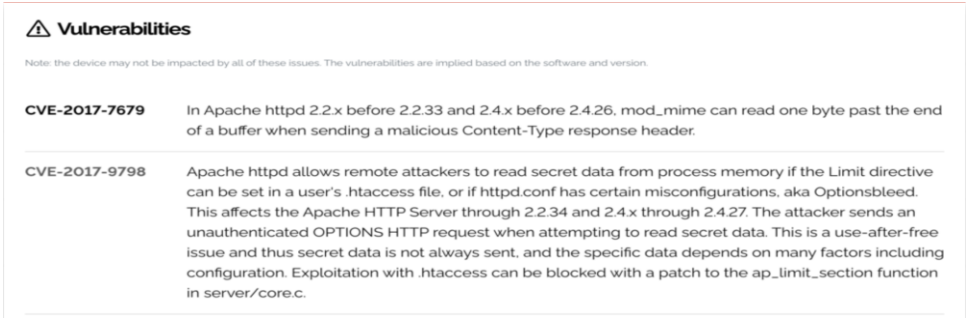
#porteaperte (21,8080)

#vulnerabilità potenziali

Esempio delle informazioni sul nostro sito web ottenibili da Shodan



Esempio di elenco di vulnerabilità trovate da Shodan su una macchina.

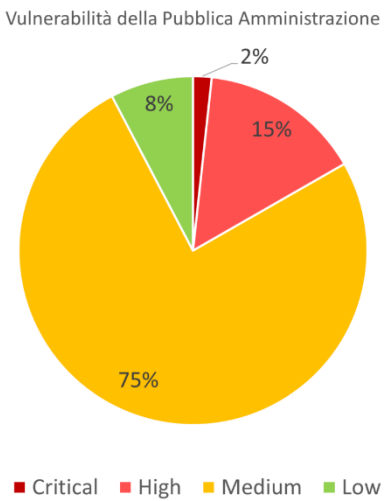


..Uno sguardo ai server della PA

Report CERT-AgID

La scansione ha rilevato **495.762 vulnerabilità puntuali che affliggono i sistemi della PA**

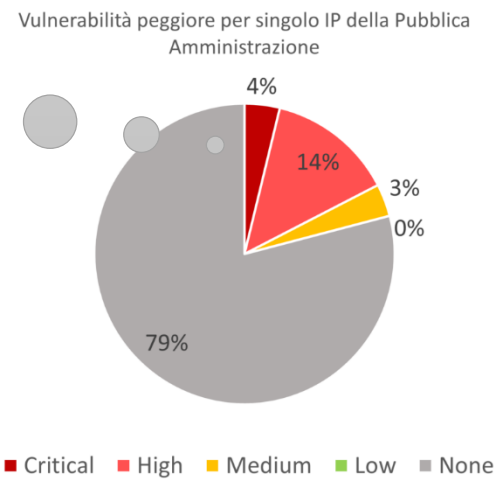
Analisi Qualitativa



Il grafico sopra mostra come sono distribuite, in base alla loro gravità (CVSS), le vulnerabilità trovate ma non dice nulla riguardo la diffusione di tali vulnerabilità sui server della PA.

Il grafico mostra che il **21%** dei server della PA è **potenzialmente vulnerabile**

Analisi Quantitativa



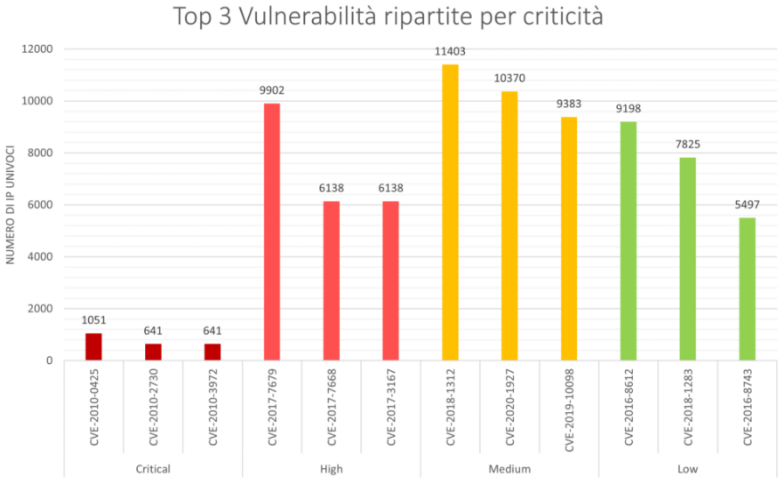
Il grafico sopra mostra quanto sono diffuse, tra i vari server, le classi di vulnerabilità. Utilizzando i dati aggregati, in particolare il punteggio CVSS peggiore (leggi: più alto) si può risalire al numero di IP che hanno almeno una vulnerabilità di tipo **Low, Medium, High, Critical e None**

..Uno sguardo ai server della PA

CERT-AGID

SHODAN

Pericolosità Vulnerabilità

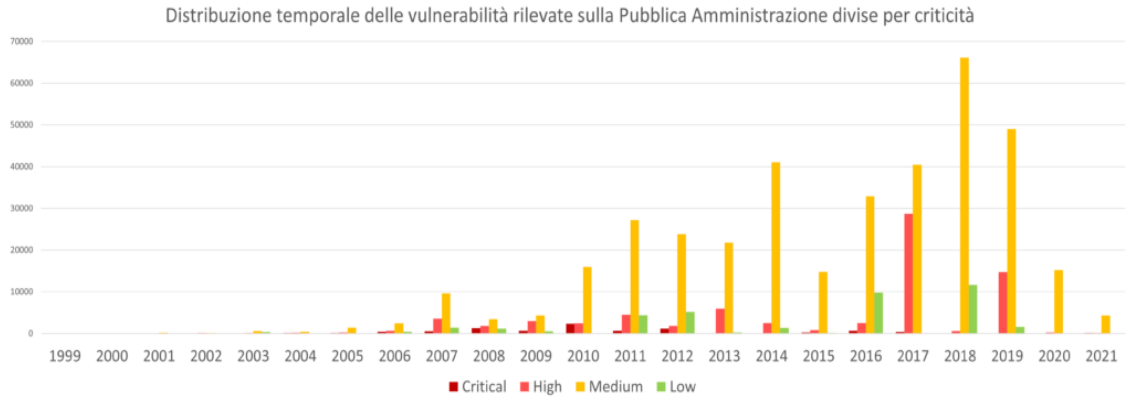


Tutte le vulnerabilità di grado *High, Medium e Low* sono riconducibili al servizio web server **Apache**. Quelle Critical invece sono ripartite su tre servizi: **Microsoft FTP Server, Microsoft IIS** e di nuovo Apache **HTTP server**.

Obsolescenza software

Età Vulnerabilità

Mancata pianificazione dei processi di aggiornamento



Dal grafico notiamo che le vulnerabilità di tipo critico sono state scoperte principalmente negli anni **2008, 2010 e 2012**. Per quanto concerne le CVE categorizzate come High, quelle più comuni fanno riferimento agli anni 2007, 2011, 2013, 2017 (con un picco facilmente distinguibile) e 2019 (con un secondo picco).

AGID | Agenzia per l'Italia Digitale

FormezPA

GRAZIE PER L'ATTENZIONE

UNIONE EUROPEA
Fondo sociale europeo
Fondo Europeo di Sviluppo Regionale

Agenzia per la Coesione Territoriale

Presidenza del Consiglio dei Ministri
Dipartimento della Funzione Pubblica

PON GOVERNANCE E CAPACITÀ ISTITUZIONALE 2014-2020

AGID | Agenzia per l'Italia Digitale

Contatti utili del CERT-AGID:

e-mail : info@cert-agid.gov.it
web : https://cert-agid.gov.it
twitter : @agidcert
telegram : @certagid

Per segnalarci nuove campagne **malware / phishing / scam** da analizzare basta allegare l'email originale sospetta e inviarla all'indirizzo:

malware@cert-agid.gov.it