CyberSecurity Ovidius Camp

OWASP-21 TOP 10

Lector dr. Dorin IORDACHE

30.06-04.07.2025
Facultatea de Matematică și Informatică

Open Web Application Security Project (OWASP)

OWASP Top 10 - 2021

dorin.iordache@365.univ-ovidius.ro

Iulie, 2025

Agenda

- OWASP and the OWASP Top 10
- Understanding the Top 10
- Data Factors
- The OWASP Top 10: 2021
- The OWASP Top 10 as a Standard

Non-Tehnic: Managerial, strategic și la nivel înalt (public general)

Tehnic: Tactic/IOC-uri; cunoștințe aprofundate (administratori de sistem, IRT)

IOC – Incident Operation Centre
IRT – Incident Response Team

OWASP and the OWASP Top 10

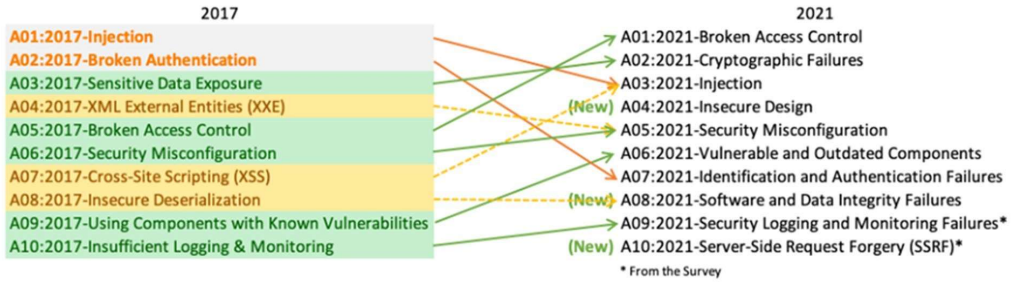
- Open Web Application Security Project (OWASP)
  - ONG dedicată îmbunătățirii securității software
  - Funcționează sub forma de „comunitate deschisă”;
- OWASPOWASP Top 10
  - Un document standard de conștientizare pentru dezvoltatori și securitatea aplicațiilor web
  - Reprezintă un consens larg cu privire la cele mai critice riscuri de securitate pentru aplicațiile web

Sursa: OWASP

https://owasp.org/Top10/

Understanding the Top 10

- Cele 10 Categoriii
  - 8 din 10 categorii sunt bazate pe date
  - 2 din categorii provin din studiul Top 10 community
    - Aceste categorii reflectă ceea ce cercetătorii AppSec consideră a fi cele mai mari riscuri care pot să nu fie incluse în date (și care s-ar putea să nu fie niciodată exprimate în date).
- Common Weakness Enumerations (CWEs)
  - O listă dezvoltată de comunitate cu tipuri de vulnerabilități software și hardware
  - Servește ca limbaj comun, punct de măsurare pentru instrumentele de securitate și ca punct de referință pentru identificarea, atenuarea și eforturile de prevenire a vulnerabilităților

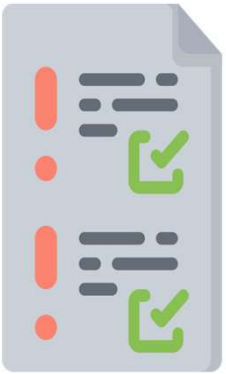


Sursa: OWASP

Factorii date

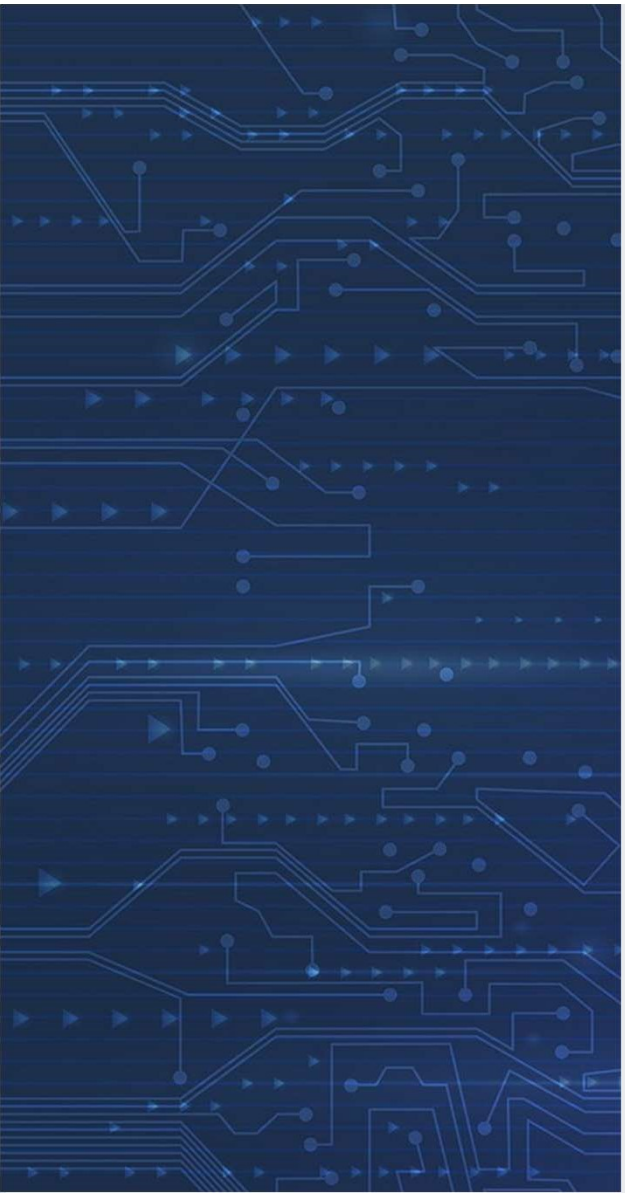
Factorii de date pentru fiecare dintre primele 10 categorii:

- **CWE-uri indicate:** Numărul de CWE-uri mapate la o categorie de către echipa Top 10
- **Rata de incidență:** Procentul de aplicații vulnerabile la acel CWE din populația testată de organizația respectivă pentru anul respectiv
- **Exploatare ponderată:** Sub-scorul extras din CVSSv2 și CVSSv3 atribuite CVE-urilor mapate la CWE-uri
- **Impact ponderat:** Sub-scorul Impact din CVSSv2 și CVSSv3 atribuite CVE-urilor mapate la CWE-uri
- **Total apariții:** Numărul total de aplicații găsite cu CWE-uri mapate la o categorie
- **Total CVE-uri:** Numărul total de CVE-uri din Baza de date națională a vulnerabilităților (NVD DB) care au fost mapate la CWE-urile mapate la o categorie



NVD DB https://nvd.nist.gov/
CVSS https://nvd.nist.gov/vuln-metrics/cvss

CVE = Common Vulnerabilities and Exposures
CWE = Common Weakness Enumeration
CVSS = Common Vulnerability Scoring System



OWASP Top 10: 2021

OWASP Top 10: 2021

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

A01:2021 – Broken Access Control

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
34	55.97%	3.81%	6.92	5.93	318,487	19,013

Exemplu de scenariu:

O aplicație folosește date neverificate într-un apel SQL (limbaj de interogare structurată) care accesează informații despre cont:

pstmt.setString(1, request.getParameter("acct"));

ResultSet results = pstmt.executeQuery( );

Un atacator modifică parametrul „***account***” din browser pentru a trimite orice număr de cont dorește.
Dacă nu este verificat corect, atacatorul poate accesa contul oricărui utilizator.

https://example.com/app/accountInfo?acct=notmyacct

Notable Common Weakness Enumerations (CWEs):

- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor
- CWE-201: Insertion of Sensitive Information Into Sent Data
- CWE-352: Cross-Site Request Forgery

Broken Access Control Vulnerabilities

- Încălcarea principiului privilegiilor minime sau a refuzului implicit;
- Ocolirea verificărilor de control al accesului prin modificarea URL-ului (manipulare parametri sau navigare forțată), a stării interne a aplicației sau a paginii HTML sau prin utilizarea unui instrument de atac care modifică cererile API;
- Permiterea vizualizării sau editării contului altcuiva, prin furnizarea identificatorului său unic (referințe directe nesigure la obiecte);
- Accesarea API-ului cu controale de acces lipsă pentru POST, PUT și DELETE;
- Creșterea privilegiilor - acționarea ca utilizator fără a fi autentificat sau acționarea ca administrator atunci când este autentificat ca Utilizator;
- Manipularea metadatelor - cum ar fi redarea sau modificarea unui token de control al accesului JSON Web Token (JWT), a unui cookie sau a unui câmp ascuns manipulat pentru a ridica privilegii sau abuzul de invalidare JWT;
- Creșterea forțată a paginilor autentificate ca utilizator neautentificat sau a paginilor privilegiate ca utilizator standard.

Broken Access Control - Prevenire

- Controlul accesului în codul server-side de încredere sau în interfața de programare a aplicațiilor (API) fără server;
- Implementarea de mecanisme de control al accesului o singură dată și reutilizate în întreaga aplicație, inclusiv minimizarea utilizării partajării resurselor între origini (Cross-Origin Resource Sharing - CORS);
- Controalele de acces să impună proprietatea asupra înregistrărilor, decât să accepte că utilizatorul poate crea, citi, actualiza sau șterge orice înregistrare;
- Dezactivarea listării structurii directoarelor serverului web, iar metadatele fișierelor (de exemplu, .git) și fișierele de rezervă nu sunt stocate în rădăcinile web;
- Înregistrarea erorilor de control al accesului, alertarea administratorilor atunci când este cazul (de exemplu, erori repetate).
- Limitarea ratei de acces la API și controler pentru a minimiza daunele cauzate de instrumentele automate de atac;
- Identificatorii de sesiune cu stare trebuie invalidați pe server după deconectare.

A02:2021 – Cryptographic Failures

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
29	46.44%	4.49%	7.29	6.81	233,788	3,075

Exemplu de scenariu:

O aplicație criptează numerele de carduri de credit dintr-o bază de date, utilizând criptarea implicită a bazei de date. Datele sunt decriptate automat la preluare, permițând unei erori de injectare a limbajului de interogare structurat (SQL) să preia numerele de carduri de credit în text clar.

Notable Common Weakness Enumerations (CWEs):

- CWE-259: Use of Hard-coded Password
- CWE-327: Broken or Risky Crypto Algorithm
- CWE-331 Insufficient Entropy

Cryptographic Failure Vulnerabilities

- Datele sunt transmise în text clar;
- Utilizare de Algoritmi sau protocoale criptografice vechi sau slabe ;
- Utilizare de chei criptografice implicite, slabe sau lipsește o gestionare sau o rotație adecvată;
- Neutilizare criptare, eg. lipsesc directivele de securitate sau antetele HTTP (browser);
- Nevalidare corectă a Certificatelor serverului și a lanțului de încredere;
- Utilizarea Paroleloe pe rol de chei criptografice;
- Utilizare de Funcții hash depreciate (eg. MD5,SHA1), sau non-criptografice;
- Folosire Metode de umplere *(padding)* criptografice depreciate (PKCS 1 v1.5);
- Exploatarea Mesajelor de eroare criptografice sau informațiile din canalul lateral.

Cryptographic Failure – Prevenție

- Clasificarea datele procesate, stocate sau transmise de o aplicație, conform nevoilor și legislației;
- Nu stocați date sensibile în mod inutil;
- Criptarea datele sensibile aflate în repaus;
- Criptarea tuturor datelor în tranzit cu protocoale securizate(eg. TLS, cu cifruri forward secrecy (FS), folosind directive precum HTTP Strict Transport Security - HSTS);
- Aplicarea de controale de securitate necesare conform clasificării datelor;
- Eliminarea protocoalelor vechi (eg. FTP, SMTP), pentru transportul datelor sensibile;
- Stocarea parolelor folosind funcții de hashing adaptive puternice;
- Generarea aleatoare a Cheilor folosind alg. criptografici și stocate în memorie ca vector de octeți;
- Utilizarea algoritmilor aleatori în criptografie, după caz, cu entropie scăzută;
- Evitarea utilizării funcțiile criptografice depreciate (eg. MD5, SHA1, PKCS 1 v1.5);
- Verificarea independentă a eficienței configurației și setărilor.

A03:2021-Injection

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
33	19.09%	3.37%	7.25	7.15	274,228	32,078

Exemplu de scenariu:
O aplicație folosește date nesigure în construcția următorului apel vulnerabil în limbaj de interogare structurat (SQL):

Notable Common Weakness Enumerations (CWEs):

- **CWE-79: Cross-site Scripting**
- **CWE-89: SQL Injection**
- **CWE-73: External Control of File Name or Path**

*String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";*

http://example.com/app/accountView?id=' UNION SELECT SLEEP(10);--

interogare pentru a returna toate înregistrările din tabela ***accounts***

Injection Vulnerabilities

- Ne – validarea, filtrarea, igienizarea Datelor utilizator;
- Utilizarea directă a Interogărilor dinamice sau apeluriore neparametrizate fără analiză;
- Utilizarea Datelor ostile în cadrul parametrilor de căutare de mapare relațională obiectuală (ORM) pentru a extrage înregistrări suplimentare, sensibile;
- Utilizarea directă sau concatenată a Datelor ostile;
- Inserarea de conținut sau structură malițioasă în interogarea SQL dinamice.

Injection - Prevenție

- Utilizarea API sigure, oferă interfață parametrizată sau migrează către Instrumente de mapare relațională a obiectelor (ORM);
- Validarea pozitivă a intrărilor pe partea serverului;
- Eliminarea caracterelor speciale din interogări dinamice reziduale, folosind sintaxe de escape;
- LIMIT și alte controale SQL interogări, pentru a preveni divulgarea în masă a înregistrărilor în cazul injecției SQL;
- Revizuirea codului sursă al aplicațiilor;
- Testarea automată a tuturor parametrilor, anteturilor, URL-urilor, cookie-urilor, intrărilor de date JSON, SOAP și XML.

A04:2021 – Insecure Design

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
40	24.19%	3.00%	6.46	6.78	262,407	2,691

Exemplu de scenariu:

Un lanț de cinematografe care permite reduceri la rezervările de grup necesită un avans pentru grupuri de peste cincisprezece persoane. Atacatorii modelează acest flux pentru a vedea dacă pot rezerva sute de locuri în diverse cinematografe din lanț, provocând astfel pierderi de venituri.

Notable Common Weakness Enumerations (CWEs):

- CWE-209: Generation of Error Message Containing Sensitive Information
- CWE-256: Unprotected Storage of Credentials
- CWE-501: Trust Boundary Violation
- CWE-522: Insufficiently Protected Credentials

Insecure Design – Prevenție

- Stabilirea și utilizarea unui ciclu de viață de dezvoltare securizat, pentru securitatea aplicațiilor.
- Stabilirea și utilizarea unei biblioteci de modele de design securizat;
- Modelarea amenințărilor pentru autentificare critică, controlul accesului, și fluxurile cheie;
- Verificări de plauzibilitate la fiecare nivel al aplicației (de la frontend la backend);
- Realizarea de teste unitare și de integrare pentru a valida că fluxurile critice sunt rezistente la modelul de amenințare;
- Segregarea straturilor de nivel: de sistem și de rețea, conform nevoilor de protecție;
- Segregarea entităților prin proiectare pe toate nivelurile;
- Limitarea consumului de resurse de către utilizator/serviciu.

A05:2021 – Security Misconfiguration

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
20	19.84%	4.51%	8.12	6.56	208,387	789

Exemplu de scenariu:

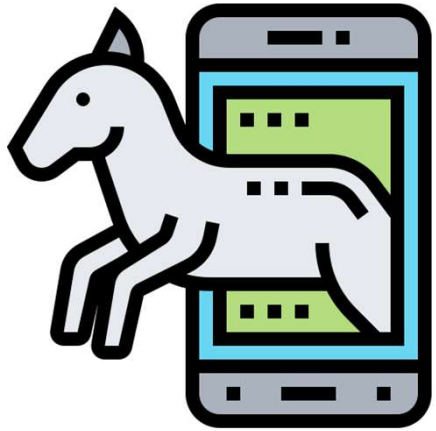
Serverul de aplicații vine cu aplicații de test care nu au fost eliminate de pe serverul de producție. Aceste aplicații prezintă vulnerabilități de securitate cunoscute pe care atacatorii le folosesc pentru a compromite serverul. Presupunem că una dintre aceste aplicații este consola de administrare, iar conturile implicite nu au fost modificate/blocate. În acest caz, atacatorul se conectează cu parolele implicite și preia controlul.

Notable Common Weakness Enumerations (CWEs):

- CWE-16 Configuration
- CWE-611 Improper Restriction of XML External Entity Reference

Secure Misconfiguration Vulnerabilities

- Absența unei securități adecvate în orice parte a stocării aplicațiilor sau permisiuni configurate incorect pentru serviciile cloud;
- Funcționalități inutile active (eg. porturi, servicii, pagini, conturi sau privilegii);
- Conturi implicite și parolele acestora active sau neschimbate;
- Urme ale erorilor în stiva de mesaje, cu prea multe detalii;
- Dezactivarea sau neconfigurarea celor mai recente funcții de Securitate, pentru sistemele actualizate;
- Setările de securitate ale: serverelor de aplicații, biblioteci, baze de date etc. nu sunt setate la valori securizate;
- Serverul nu transmite antete sau directive de securitate, ori nu sunt setate la valori securizate;
- Software-ul este învechit sau vulnerabil.



Security Misconfiguration – Prevenție

- Mediile de dezvoltare, QA și producție să fie configurate identic, cu credențiale diferite în fiecare mediu de lucru;
- O platformă minimalist: fără caracteristici necesare, componente, documentație și mostre inutile;
- Revizuirea și actualizarea configurațiilor corespunzătoare tuturor cerințelor de securitate, actualizărilor și patch-urilor. Revizuirea permisiunilor de stocare în cloud (eg. permisiunile pentru bucket-ul S3 - AWS).
- Segmentarea arhitecturii de aplicații: separare eficientă și sigură între componente sau entități găzduite, cu segmentare, containerizare sau grupuri de securitate în cloud;
- Trimiterea de directive de securitate către clienți (eg. Security Headers - http);
- Verificarea automată a eficienței configurațiilor și setărilor în toate mediile.

A06:2021 – Vulnerable and Outdated Components

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
3	27.96%	8.77%	51.78	22.47	30,457	0

Exemplu de scenariu:

Din cauza volumului de componente utilizate în dezvoltare, este posibil ca o echipă de dezvoltare să nu cunoască sau să nu înțeleagă toate componentele utilizate în aplicația lor, iar unele dintre aceste componente ar putea fi învechite și, prin urmare, vulnerabile la atacuri.

Notable Common Weakness Enumerations (CWEs):

- CWE-1104: Use of Unmaintained Third-Party Components
- CWE-937 OWASP Top 10 2013: Using Components with Known Vulnerabilities
- CWE-1035 2017 Top 10 A9: Using Components with Known Vulnerabilities

Vulnerable and Outdated Components Vulnerabilities

- Versiunile componentelor utilizate (client sau server) sunt necunoscute;
- Software-ul este vulnerabil, neacceptat sau învechit; SO, Web server / aplicații, SGBD, aplicațiile, API, medii de execuție și biblioteci;
- Vulnerabilitățile nu sunt scanate în mod regulat;
- Remedierile sau actualizările nu sunt implementate pe platforma, framework-urile și dependențele bazat pe riscuri și în timp util;
- Dezvoltatorii de software nu testează compatibilitatea bibliotecilor (update, upgrade sau patch-uri);
- Configurațiile componentelor nu sunt sigure.

Vulnerable and Outdated Components – Prevenție

- Eliminarea dependențelor neutilizate, caracteristicile, componentele, fișierele și documentația inutile;
- Inventarierea continuă a versiunior componentelor (eg. framework-uri, biblioteci) și dependențele acestora folosind instrumente ca: versions, OWASP Dependency Check, retire.js etc.
- Monitorizarea continuă Common Vulnerability and Exposures (CVE) și National Vulnerability Database (NVD);
- Utilizare de componente doar din surse oficiale prin link-uri securizate;
- Monitorizarea bibliotecilor și componentelor care nu sunt la zi sau nu dețin patch-uri de securitate pentru versiunile mai vechi;
- Internetul lucrurilor (IoT) adesea dificil sau imposibil de aplicat patch-uri, importanța aplicării patch-urilor poate fi mare (de exemplu, dispozitivele biomedicale);
- Asigurarea unui plan de monitorizare, triere și aplicare a actualizărilor sau a modificărilor de configurare, pe durata de viață a aplicației sau portofoliului, la nivel de organizație.

A07:2021 – Identification and Authentication Failures

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
22	14.84%	2.55%	7.40	6.50	132,195	3,897

Exemplu de scenariu:

Majoritatea atacurilor de autentificare apar din cauza utilizării continue a parolelor ca factor unic. Considerate cândva cele mai bune practici, cerințele de rotație a parolelor și de complexitate încurajează utilizatorii să utilizeze și să reutilizeze parole slabe. Organizațiilor li se recomandă să oprească aceste practici, conform NIST 800-63, și să utilizeze autentificarea multi-factor.

Notable Common Weakness Enumerations (CWEs):

- CWE-297: Improper Validation of Certificate with Host Mismatch
- CWE-287: Improper Authentication
- CWE-384: Session Fixation

Identification and Authentication Failures and Vulnerabilities

- Facilitarea atacurilor automate: inundarea credențialelor, unde atacatorul are o listă de nume de utilizator și parole valide;
- Facilitarea atacurilor de tip forță brută sau alte atacuri automate;
- Utilizarea de parole implicite, slabe sau bine cunoscute (eg. „Password1” sau „admin/admin”);
- Utilizare de pocese slabe / ineficiente de recuperare a credențialelor (ed. „Răspunsuri bazate pe cunoștințe”, care nu pot fi securizate);
- Stocarea parolei în clar, criptate sau cu hash slab (vezi A02:2021 - Erori criptografice);
- Fără mecanisme MFA sau ineficiente;
- Expunerea identificatorului de sesiune în URL;
- Reutilizearea identificatorului de sesiune după o conectare reușită;
- Nu invalidează corect ID-urile de sesiune.

Identification and Authentication Failures – Prevenție

- Implementare MFA, acolo unde este posibil;
- Nu se trasferă sau implementează conturi cu credențiale implicite, mai ales pentru administrator;
- Verificarea parolelor slabe, testarea parolelor noi sau modificate, raportat la lista celor mai slabe 10.000 de parole;
- Politicile privind: lungimea, complexitatea și rotația parolelor conform ghidurilor NIST 800-63b/5.1.1;
- Înregistrarea, recuperarea credențialelor și mijloacele API să fie protejate împotriva atacurilor;
- Limitarea sau întârzierea încercărilor de conectare eșuate, evitând crearea unei situații denial-of-service;
- Înregistrarea tuturor erorilor și alertare administratorii în caz de atac;
- Utilizați un manager de sesiune încorporat, securizat, pe server, care generează un nou ID de sesiune aleatoriu cu entropie ridicată după conectare.

A08:2021 – Software and Data Integrity Failures

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
10	16.67%	2.05%	6.94	7.94	47,972	1,152

Exemplu de scenariu:

Actualizarea eronată a SolarWinds.

Compania care dezvoltă SolarWindws avea procese securizate de integritate a aplicației și a actualizărilor. Cu toate acestea, mecanismele au putut fi subminate și, timp de câteva luni, firma a distribuit o actualizare rău intenționată extrem de precis direcționată către peste 18.000 de organizații, dintre care aproximativ 100 - inclusiv un spital - au fost afectate. Aceasta este una dintre cele mai ample și mai semnificative încălcări de acest gen din istorie.

Notable Common Weakness Enumerations (CWEs):

- CWE-829: Inclusion of Functionality from Untrusted Control Sphere
- CWE-494: Download of Code Without Integrity Check
- CWE-502: Deserialization of Untrusted Data

Software and Data Integrity Failures – Prevenție

- Utilizare semnături digitale sau mecanisme similare pentru a verifica dacă software-ul sau datele provin din sursa sigure și nu au fost modificate;
- Bibliotecile și dependențele, cum ar fi npm sau Maven, provin din depozite de încredere;
- Utilizarea unui instrument de securitate a lanțului de aprovizionare software: OWASP Dependency Check sau OWASP CycloneDX;
- Existența unui proces de revizuire a modificărilor de cod și configurație pentru a minimiza șansa ca un cod sau o configurație rău intenționată să fie introdusă în fluxul software;
- Fluxul de integrare și implementare continuă (CI/CD) a software are segregare, configurație și control al accesului adecvate pentru a asigura integritatea acestuia;
- Datele serializate nesemnate sau necriptate nu sunt trimise către clienți necunoscuți/nu sunt de încredere, fără o formă de verificare a integrității sau semnătură digitală pentru a detecta modificarea sau reluarea datelor serializate.

A09:2021 – Security Logging and Monitoring Failures

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
4	19.23%	6.51%	6.87	4.99	53,615	242

Exemplu de scenariu:

Operatorul site-ului web al unui furnizor de planuri de sănătate pentru copii nu a putut detecta o breșă de securitate din cauza lipsei de monitorizare și înregistrare. O parte externă a informat furnizorul de planuri de sănătate că un atacator a accesat și modificat mii de dosare medicale sensibile a peste 3,5 milioane de copii. O analiză ulterioară incidentului a constatat că dezvoltatorii site-ului web nu au remediat vulnerabilități semnificative. Deoarece nu a existat nicio înregistrare sau monitorizare a sistemului, breșa de securitate ar fi putut fi în desfășurare o perioadă mare de timp.

Notable Common Weakness Enumerations (CWES):

- CWE-778 Insufficient Logging
- CWE-117 Improper Output Neutralization for Logs
- CWE-223 Omission of Security-relevant Information
- CWE-532 Insertion of Sensitive Information into Log File

Security Logging and Monitoring Failures Vulnerabilities

- Înregistrarea, detectarea, monitorizarea și răspunsul activ insuficiente apar oricând:
  - Evenimentele de interes (eg. conectările, cu succes sau eșuate, tranzacțiile cu valoare mare) nu sunt înregistrate;
  - Avertismentele și erorile generează mesaje de jurnal inexistente, inadecvate sau neclare;
  - Jurnalele aplicațiilor și API-urilor nu sunt monitorizate pentru activități suspecte;
  - Jurnalele sunt stocate doar local;
  - Nu există sau nu sunt eficiente praguri de alertă adecvate și procese de escaladare a răspunsului;
  - Testarea de penetrare și scanările prin instrumente de testare dinamică a securității aplicațiilor (DAST) (cum ar fi OWASP ZAP) nu declanșează alerte;
  - Aplicația nu poate detecta, escalada sau alerta pentru atacuri active în timp real sau aproape în timp real.

Security Logging and Monitoring Failures – Prevenție

- Toate erorile de conectare, control al accesului și validare a intrărilor pe server să fie înregistrate cu un context de utilizator suficient pentru a identifica conturile suspecte sau rău intenționate și păstrate suficient de mult timp pentru a permite o analiză criminalistică întârziată;
- Jurnalele sunt generate într-un format pe care soluțiile de gestionare a acestora îl pot procesa cu ușurință;
- Datele din jurnal sunt codificate corect pentru a preveni injectările sau atacurile asupra sistemelor de înregistrare sau monitorizare;
- Tranzacțiile cu valoare mare au o pistă de audit cu controale de integritate pentru a preveni manipularea sau ștergerea: tabelele de baze de date cu funcție de adăugare sau similar;
- Echipele DevSecOps ar trebui să stabilească o monitorizare și o alertare eficiente, astfel încât activitățile suspecte să fie detectate și să se răspundă rapid;
- Existența unui plan de răspuns la incidente și de recuperare, cum ar fi standardul Institutului Național de Standarde și Tehnologie (NIST) 800-61r2 sau o versiune ulterioară.

A10:2021 – Server-Side Request Forgery (SSRF)

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Total Occurrences	Total CVEs
1	2.72%	2.72%	8.28	6.72	9,503	385

Exemplu de scenariu:

Dacă o arhitectură de rețea nu este segmentată, atacatorii pot utiliza rezultatele conexiunilor sau timpul scurs pentru a conecta sau respinge conexiuni de tip server-side request forgery (SSRF) pentru a mapa rețelele interne și a determina dacă porturile sunt deschise sau închise pe serverele interne.

Notable Common Weakness Enumerations (CWEs):

As new entries are likely to be a single or small cluster of CWEs for attention and awareness, the hope is that they are subject to focus and can be rolled into a larger category in a future edition.

Server-Side Request Forgery (SSRF) – Prevenție

- La nivel de Rețea:
  - Segmentarea funcționalității de acces la resurse la distanță în rețele separate pentru a reduce impactul SSRF;
  - Aplicarea politicilor de firewall de tip „*refuzare implicită*” sau reguli de control al accesului la rețea pentru a bloca tot traficul intranet, cu excepția traficului esențial;
- La nivel de Aplicație:
  - Sanitizarea și validarea tuturor datelor de intrare furnizate de client;
  - Aplicarea schemei URL, port și destinație cu o listă de permisiuni pozitivă;
  - Nu se trimit răspunsuri brute clienților;
  - Dezactivarea redirecționărilor HTTP;
  - Atenție la consistența URL-ului pentru a evita atacuri precum *relegarea* DNS și condițiile de concurență de tip „*momentul verificării, momentul utilizării*” (TOCTOU);
  - Nu se atenuează SSRF prin utilizarea unei liste de refuz sau a unei expresii regulate;
- Măsuri suplimentare în atenție:
  - Nu se implementează alte servicii relevante pentru securitate pe sistemele front-end (de exemplu, OpenID). Controlarea traficului local pe aceste sisteme (de exemplu, localhost);
  - Pentru frontend-urile cu grupuri de utilizatori dedicate și ușor de gestionat, se utilizează criptarea rețelei (de exemplu, VPN-uri) pe sisteme independente pentru a lua în considerare nevoile foarte ridicate de protecție.

The OWASP Top 10 as a Standard

Also use:

- OWASP Application Security Verification Standard

Use Case	OWASP Top 10 2021	OWASP Application Security Verification Standard
Awareness	Yes	-
Training	Entry level	Comprehensive
Design and architecture	Occasionally	Yes
Coding standard	Bare minimum	Yes
Secure Code review	Bare minimum	Yes
Peer review checklist	Bare minimum	Yes
Unit testing	Occasionally	Yes
Integration testing	Occasionally	Yes
Penetration testing	Bare minimum	Yes
Tool support	Bare minimum	Yes
Secure Supply Chain	Occasionally	Yes

Referințe bibliografice

Referințe

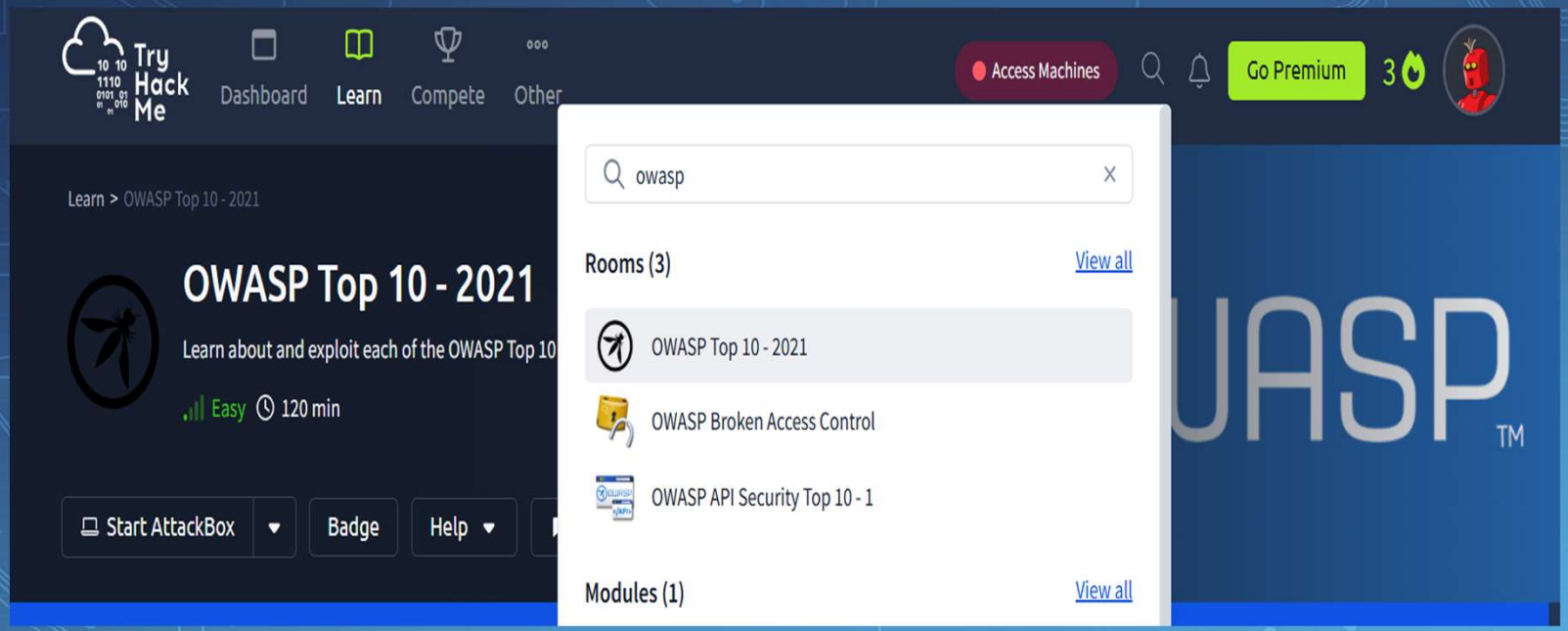
- “Broken Access Control,” OWASP. N.d. https://owasp.org/Top10/A01_2021-Broken_Access_Control/
- “Cryptographic Failures,” OWASP. N.d. https://owasp.org/Top10/A02_2021-Cryptographic_Failures/
- “How to use the OWASP Top 10 as a standard,” OWASP. N.d. https://owasp.org/Top10/A00_2021_How_to_use_the_OWASP_Top_10_as_a_standard/
- “Identification and Authentication Failures,” OWASP. N.d. https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/
- “Injection,” OWASP. N.d. https://owasp.org/Top10/A03_2021-Injection/
- “Insecure Design,” OWASP. N.d. https://owasp.org/Top10/A04_2021-Insecure_Design/
- “OWASP Top 10,” OWASP. N.d. https://owasp.org/www-project-top-ten/
- “OWASP Top 10 2021,” Synopsys. N.d. https://www.synopsys.com/glossary/what-is-owasp-top-10.html
- “OWASP Top Ten 2021 : Related Cheat Sheets,” OWASP. N.d. https://cheatsheetseries.owasp.org/IndexTopTen.html

Referențe

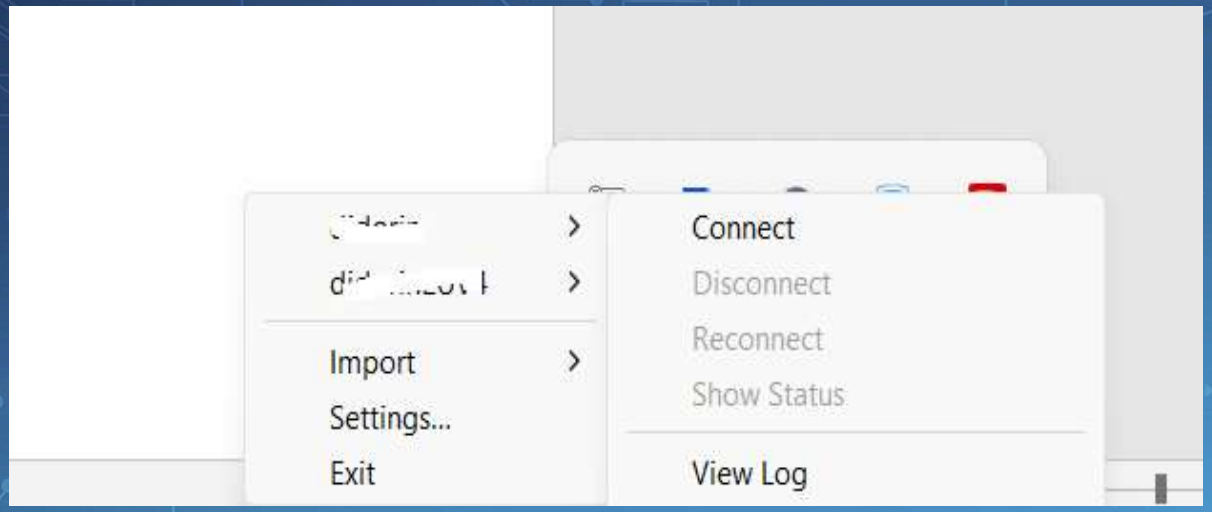
- "Security Logging and Monitoring Failures," OWASP. N.d. https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/
- "Security Misconfiguration," OWASP. N.d. https://owasp.org/Top10/A05_2021-Security_Misconfiguration/
- "Server Side Request Forgery," OWASP. N.d. https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/
- "Software and Data Integrity Failures," OWASP. N.d. https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/
- "Vulnerable and Outdated Components," OWASP. N.d. https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
- "Welcome to the OWASP Top 10 - 2021," OWASP. N.d. https://owasp.org/Top10/A00_2021_Introduction/

Întrebări

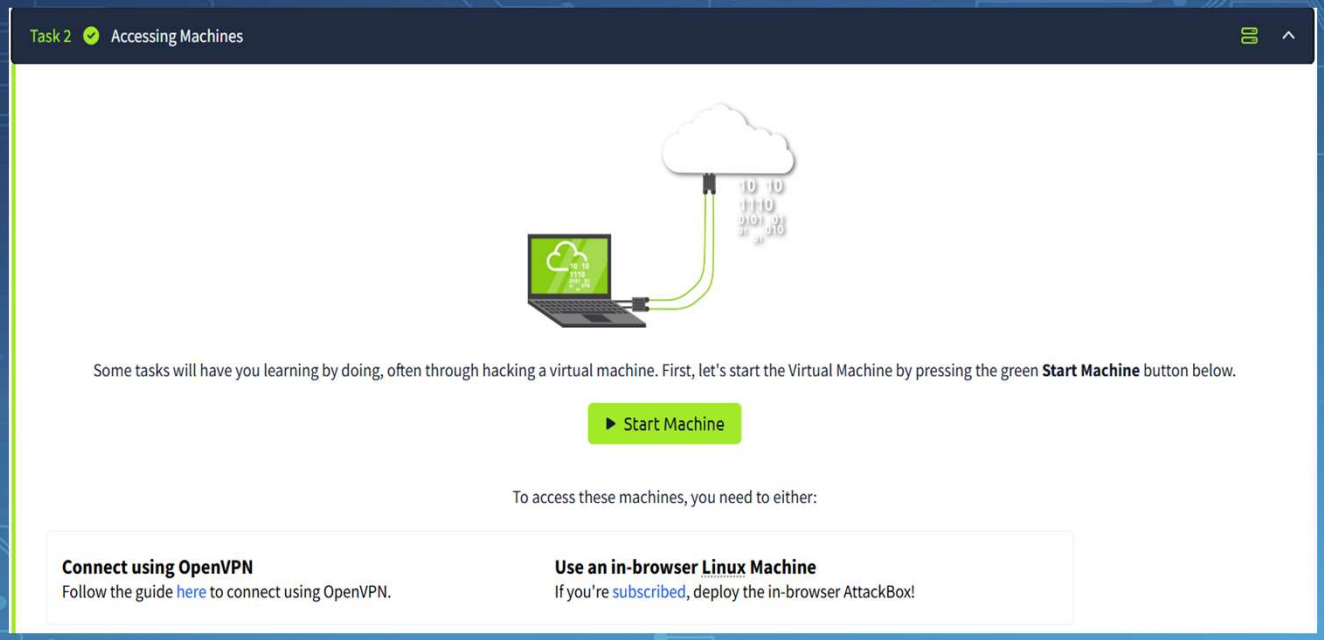
Start mașina de test TryHackMe



După ce se realizează conexiune ala openvpn



Selectați Task 2
Și apoi Start Machine



Așteptați alocarea adresei IP

În acest moment arhitectura este gata de utilizare.

SPOR!