

Securizarea aplicaţiilor web.

Universitatea Tehnica din Moldova, [Facultatea Calculatoare, Informatică și Microelectronică](https://www.facebook.com/UTM.FacultateaCIM/)

Chirita Stanisalv SI-212

**Rezumat:**

O vulnerabilitate este o gaură sau o slăbiciune a aplicației, care poate fi un defect de proiectare sau un bug de implementare, care permite unui atacator să provoace prejudicii părților interesate ale unei aplicații. Părțile interesate includ proprietarul aplicației, utilizatorii aplicației și alte entități care se bazează pe aplicație. Vă rugăm să nu publicați vulnerabilități reale în produse, servicii sau aplicații web. Aceste rapoarte de dezvăluire ar trebui postate pe bugtraq sau pe listele de corespondență cu divulgare completă.

**Cuvinte cheie:**

OWASP-Open Web Application Security Project (https://ru.wikipedia.org/wiki/OWASP).

Injecție SQL- este o tehnică de injectare a codului.

XXE- Atacul unei entitati externe.

**Introducere:**

Utilizarea recomandărilor Open Web Application Security Project (OWASP) a fost mult timp standardul pentru securitatea aplicațiilor web. Să vedem care sunt 10 vulnerabilități pe care organizația le consideră cele mai periculoase în acest moment.

1. **Injecții**

Atacurile de injectare apar atunci când datele neîncrezătoare sunt transmise interpretorului de cod prin introducerea formularului sau printr-o altă modalitate de trimitere a informațiilor către o aplicație web. De exemplu, un atacator ar putea introduce cod SQL într-un formular care așteaptă un nume de utilizator. Dacă intrarea nu este protejată corespunzător, va duce la executarea codului - astfel de atacuri sunt cunoscute sub numele de injecții SQL.

Atacurile prin injecție pot fi prevenite prin validarea și/sau igienizarea datelor trimise de utilizator. În primul caz, datele suspecte sunt respinse complet, iar în al doilea caz, doar partea suspectă a acestora este ștearsă. În plus, DBA poate seta controale speciale pentru a minimiza cantitatea de informații pe care o poate dezvălui un atac de injecție SQL.

1. **Autentificare ruptă.**

Vulnerabilitățile de autentificare pot permite atacatorilor să obțină acces la conturile de utilizator, inclusiv la cele privilegiate, care pot fi apoi utilizate pentru a obține controlul asupra sistemelor de informații corporative.

Site-urile web au adesea probleme cu mecanismele de autentificare. În primul rând, acestea sunt asociate cu gestionarea incorectă a sesiunii, care poate fi folosită de atacatori pentru a obține acces la conturile de utilizator și acreditările de conectare.

OWASP Top 10 conține o listă întreagă de astfel de vulnerabilități:

* Posibilitatea de a completa automat acreditările și/sau de a selecta o parolă.
* Abilitatea de a utiliza parole standard, slabe și binecunoscute.
* Procese ineficiente de recuperare a acreditărilor.
* Lipsa autentificării cu mai mulți factori (MFA) sau implementare slabă.
* Furnizarea ID-urilor de sesiune într-un URL (Uniform Resource Locator), fără intercalarea ID-urilor de sesiune și revocarea incorectă a ID-urilor de sesiune și a simbolurilor de autentificare atunci când un utilizator se deconectează sau după o perioadă de inactivitate.

Astfel de vulnerabilități apar adesea din lipsa de experiență a dezvoltatorilor, probleme de testare și lansări de software prea grăbite.

Numărul de vulnerabilități poate fi redus prin implementarea autentificării cu mai mulți factori, precum și prin introducerea de restricții care fac imposibile atacurile automate cu forță brută (de exemplu, prin forță brută). La fel de importantă este lipsa de grabă și timpul necesar dezvoltatorilor pentru a testa modificările înainte de lansarea în producție, precum și implementarea procedurilor de testare a codului pentru securitate. De asemenea, trebuie să implementați o politică puternică de parole și să utilizați manageri de sesiune securizat.

1. **Dezvăluirea datelor critice**

Principalul motiv al riscului dezvăluirii datelor critice se datorează lipsei de criptare sau utilizării unor metode nesigure de generare și gestionare a cheilor, algoritmi slabi de criptare, metode nesigure de stocare a parolelor etc. În plus, dezvoltatorii de aplicații web stochează adesea date sensibile chiar și atunci când nu sunt necesare.

1. **Entități XML externe (XXE)**

Atacurile XXE vizează aplicațiile web care analizează Extensible Markup Language (XML). Acestea apar atunci când intrarea care conține un cod XML care face referire la un obiect extern este procesată de un parser cu o configurație slabă. Analizatoarele XML sunt adesea vulnerabile la XXE în mod implicit, ceea ce înseamnă că dezvoltatorii trebuie să elimine manual vulnerabilitatea.

Atacurile XXE pot fi evitate dacă aplicațiile web acceptă forme mai puțin complexe de date (cum ar fi jetoanele web JavaScript Object Notation (JSON)), repară analizatoarele XML sau dezactivează utilizarea entităților externe. Vă puteți proteja împotriva atacurilor XXE prin implementarea gateway-urilor de securitate pentru interfața de programare a aplicațiilor (API), patch-uri virtuale și firewall-uri pentru aplicații web (WAF).

1. **Controlul accesului întrerupt**

Problemele de control al accesului permit atacatorilor să ocolească restricțiile și să obțină acces neautorizat la sisteme și date sensibile și, potențial, să obțină acces la conturile de administrator și utilizatori privilegiați.

Riscul de încălcare a controlului accesului poate fi redus prin implementarea conceptului de acces cel mai puțin privilegiat, verificarea regulată a serverelor și site-urilor web, aplicarea MFA și eliminarea utilizatorilor inactivi și a serviciilor inutile de pe servere. De asemenea, puteți securiza controalele de acces utilizând jetoane de autorizare atunci când utilizatorii se conectează la o aplicație web și le invalidați atunci când se deconectează. Alte recomandări includ înregistrarea și raportarea erorilor de acces și utilizarea limitării ratei pentru a minimiza daunele cauzate de atacurile automate.

1. **Configurare de securitate incorectă**

Configurațiile greșite de securitate sunt considerate cea mai comună vulnerabilitate din Top 10 OWASP și sunt cel mai frecvent asociate cu utilizarea setărilor implicite ale site-ului web sau a unui sistem de management al conținutului (CMS). Erorile obișnuite de configurare includ, de asemenea, neremedierea defectelor software, pagini web neutilizate, directoare și fișiere neprotejate, permisiuni implicite de partajare pentru serviciile de stocare în cloud și servicii neutilizate sau care nu sunt necesare.

Configurarea greșită a securității poate fi oriunde: în aplicații și servere web, baze de date, servicii de rețea, cod personalizat, cadre, mașini virtuale preinstalate și containere.

Configurația de securitate poate fi corectată prin modificarea setărilor implicite ale serverului web sau CMS, eliminând caracteristicile de cod neutilizate și controlând datele utilizatorului și vizibilitatea informațiilor utilizatorului. De asemenea, dezvoltatorii ar trebui să elimine documentația, caracteristicile, cadrele și mostrele inutile, să segmenteze arhitectura aplicației și să automatizeze validarea eficienței configurațiilor și setărilor mediului web.

1. **Cross-Site Scripting (XSS)**

Vulnerabilitățile XSS permit infractorilor cibernetici să injecteze scripturi într-un site web și să-l folosească pentru a distribui cod rău intenționat care se execută în browserul unui utilizator. De regulă, acest lucru este necesar pentru a intercepta sesiunile utilizatorilor, pentru a fura date confidențiale sau pentru a redirecționa utilizatorul către site-uri rău intenționate.

Puteți preveni exploatarea vulnerabilităților XSS utilizând paravanele de protecție pentru aplicații web (WAF), în timp ce dezvoltatorii pot reduce șansa de atacuri XSS prin separarea datelor nesigure de browserele active. Aceasta include utilizarea cadrelor care evită XSS prin proiectare, utilizarea igienizării și validării datelor, evitarea datelor de solicitare HTTP (Hypertext Transfer Protocol) nesigure și implementarea Politicii de securitate a conținutului (CSP).

1. **Deserializare nesigură**

În termeni de stocare a datelor și informatică, serializarea înseamnă convertirea obiectelor sau structurilor de date în șiruri de octeți. Deserializarea înseamnă convertirea acestor șiruri de octeți în obiecte. Deserializarea nesigură implică atacatorii care modifică datele înainte ca acestea să fie deserializate.

Avizul de securitate OWASP pentru deserializarea nesigură se ocupă de supercookie-urile care conțin informații serializate despre utilizatori. Dacă atacatorii pot deserializa cu succes obiectul, ei își pot acorda un rol de administrator, se pot serializa datele și pot compromite toate aplicațiile web.

1. **Utilizarea componentelor cu vulnerabilități cunoscute**

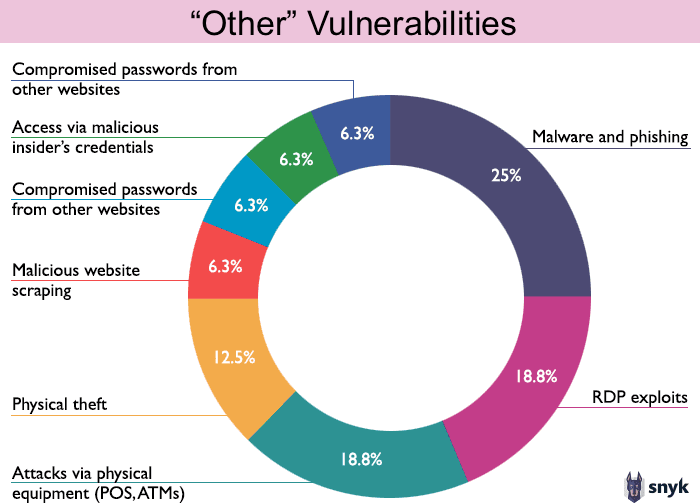
Componentele software, cum ar fi cadrele și bibliotecile sunt adesea folosite în aplicațiile web pentru a oferi anumite funcționalități. Cu toate acestea, aceste componente pot conține vulnerabilități care permit unui atacator să lanseze un atac cibernetic.

Adesea, dezvoltatorii nu actualizează componentele de la terți, deoarece codul lor învechit nu funcționează cu cele mai recente versiuni ale software-ului, iar webmasterii fie sunt îngrijorați că actualizările întrerup site-urile, fie nu au experiența necesară pentru a aplica actualizări. În plus, atacatorii caută în permanență potențiale vulnerabilități pe care dezvoltatorii nu le-au descoperit încă (cunoscute sub numele de vulnerabilități zero-day) și pe care le pot exploata.

Acest lucru poate fi evitat cu un patch virtual care protejează site-urile web vechi de a fi exploatate de firewall-uri, sisteme de detectare a intruziunilor (IDS) și WAF-uri. Exploatarea vulnerabilităților poate fi prevenită și prin păstrarea doar a componentelor care sunt efectiv necesare și prin eliminarea celor care sunt neutilizate sau neîntreținute. Cea mai bună metodă este să instalați componente din surse de încredere și să le păstrați la zi.

1. **Jurnalele insuficient detaliate și monitorizarea slabă**

Un atac de hacker de succes sau o scurgere de date nu este întotdeauna ușor de detectat. Adesea, răufăcătorii nu numai că primesc acces neautorizat la sistemele informaționale, dar le găzduiesc luni sau ani, rămânând invizibili. Pentru a preveni acest lucru, este necesar să se înregistreze și să urmărească comportamentul unei aplicații web pentru a recunoaște activitatea suspectă în timp util și fie pentru a preveni un atac, fie pentru a minimiza consecințele acestuia.



Surse:

* <https://owasp.org/>
* <https://tryhackme.com>
* <https://app.hackthebox.com>