ASPECTE ALE SECURITĂȚII APLICAȚIILOR WEB

Cristina COLESNICENCO

Universitatea Tehnică a Moldovei

Abstract: *Siguranța aplicațiilor web este o componentă centrală a oricărei afaceri web. Vulnerabilitățile aplicațiilor web sunt unele dintre defectele cele mai frecvente, care duc la încălcări ale datelor care implică o defecțiune sau o slăbiciune a sistemului într-o aplicație web. Defectele de proiectare ale aplicațiilor pot fi exploatate pentru a compromite securitatea aplicației. Lucrarea prezintă principalele aspecte legate de securitatea aplicațiilor Web reale și oferă detalii despre posibilele riscuri de securitate (de exemplu, SQL injection sau Scripting Cross-Site) pentru site-urilor Web. De asemenea, materialul descrie soluții diferite pentru a preveni sau rezolva posibilele atacuri periculoase.*

Cuvinte cheie *: Aplicaţii WEB, vulnerabilitate, securitate, atac.*

1. Cross-site scripting (XSS)

Atacurile de tip XSS, reprezintă o vulnerabilitate de securitate care permite infractorilor să injecteze scripturi malware în paginile web vizualizate de utilizatori, să modifice codul pe care o aplicație o livrează unui utilizator care este executat în browser-ul web al utilizatorului.

Varietatea atacurilor bazate pe XSS este aproape nelimitată, dar include în mod obișnuit transmiterea datelor private, cum ar fi cookie-urile sau informații sensibile păstrate de browser, către atacator, redirecționarea victimei la conținutul web controlat de atacator sau efectuarea altor operații rău intenționate pe dispozitivul utilizatorului sub masca site-ului vulnerabil. Scripturile malware pot chiar să rescrie conținutul paginii HTML.

Un exemplu concludent în acest caz, este atacul efectuat în anul 2017, pe eBay, în care Hackerii au exploatat vulnerabilitățile XSS stocate. Atacatorii au exploatat venerabilitățile de scripting ale site-ului eBay pentru a fura acreditările contului. Vulnerabilitatea ce a permis acest lucru, este că eBay a permis atacatorilor să includă JavaScript în anunțurile licitaționale. Atacatorii au folosit scripturi malware pe o varietate mare de articole de valoare mică, incluzând listări legitime care au fost deja postate de pe conturile eBay de renume, ei au compromis aceste conturi și au adăugat informații suplimentare. Pentru a remedia această problemă, eBay a blocat complet utilizarea conținutului activ (cum ar fi JavaScript). Acest lucru este implementat ca un control tehnic prin utilizarea cadrelor iframe cu Politica de securitate a conținutului și cu restricțiile sandbox.

O modalitate de a stoca date pe un site web este de a utiliza o bază de date. Există mai multe tipuri diferite de baze de date, cum ar fi o bază de date Structured Query Language (SQL) sau o bază de date Extensible Markup Language (XML). Ambele atacuri XML și SQL injecție exploatează deficiențe în program, cum ar fi nevalidarea corectă a interogărilor bazei de date.

2. Injecția XML

Când se utilizează o bază de date XML, o injecție XML este un atac care poate corupe datele. După ce utilizatorul oferă intrare, sistemul accesează datele solicitate printr-o interogare. Problema apare atunci când sistemul nu examinează corect solicitarea de intrare furnizată de utilizator. Criminalii pot manipula interogarea programând-o pentru a se potrivi nevoilor lor și pot accesa informațiile din baza de date. Toate datele sensibile stocate în baza de date sunt accesibile infractorilor și pot face orice număr de modificări pe site. Atacatorul poate sigila întreaga bază de date și poate chiar să se înregistreze ca administrator al site-ului.

Prevenirea injectării XML poate fi făcută prin gestionarea și dezinfectarea corectă a oricărei intrări de utilizator înainte de a se ajunge la codul principal al programului. Cea mai bună metodă este de a considera că toate intrările de utilizator sunt nesigure și de a monitoriza corespunzător această intrare. Majoritatea tipurilor de atacuri injectabile XML pot fi prevenite prin eliminarea pur și simplu a tuturor citatelor simple și duble de la intrarea utilizatorului.

3. Injecție SQL.

Cyber criminalul exploatează o vulnerabilitate prin introducerea unei instrucțiuni SQL rău intenționate într-un câmp de intrare. Sistemul nu filtrează corect intrarea utilizatorului pentru caracterele dintr-o instrucțiune SQL. Criminalii folosesc injectarea SQL pe site-uri sau orice bază de date SQL. Criminalii pot

sparge o identitate, pot modifica datele existente, pot distruge date sau pot deveni administratori ai serverului de baze de date.

Un firewall pentru aplicații web (WAF) poate detecta și bloca atacurile SQL de bază. IDS-urile bazate pe rețea pot monitoriza toate conexiunile la serverul de baze de date și pot semnala activitate suspectă. Singura modalitate sigură de a preveni atacurile SQL Injection este validarea intrărilor și interogările parametrizate, inclusiv declarațiile pregătite. Codul aplicației nu trebuie să utilizeze direct intrarea. Dezvoltatorul trebuie să dezinstaleze toate intrările, nu numai intrările din formularul web, cum ar fi formularele de conectare.

4. Buffer overflow

Buffer overflow este unul dintre cele mai grave bug-uri care pot fi exploatate de un atacator, pentru că este foarte greu de identificat și corectat, mai ales dacă software-ul este format din milioane de linii de cod. Este aproape imposibil de eliminat în întregime acest tip de eroare. Această eroare apare atunci când există mai multe date într-un buffer decât se poate ocupa, determinând datele să se deplaseze în spațiul de stocare adiacent. Informațiile suplimentare, care trebuie să meargă undeva, pot să se reverse în spațiul de memorie alăturat, coruperea sau suprascrierea datelor deținute în acel spațiu. Acest excedent are de obicei un accident de sistem, dar creează, de asemenea, posibilitatea ca un atacator să execute un cod arbitrar sau să manipuleze erorile de codificare pentru a solicita acțiuni rău intenționate.

Pentru a preveni depășirea tamponului, dezvoltatorii de aplicații C/C++ ar trebui să evite funcțiile standard ale bibliotecilor care nu sunt bifate, cum ar fi gets, scanf și strcpy. Cea mai fiabilă modalitate de a evita sau de a preveni depășirile buffer-ului este de a folosi protecția automată la nivel de limbaj. O altă soluție este verificarea limitelor la momentul executării, ceea ce împiedică depășirea buffer-ului prin verificarea automată a faptului că datele scrise într-un buffer sunt în limitele acceptabile.

5. Managementul sesiunilor și autentificări corupte

Funcțiile de autentificare și de gestionare a sesiunilor nu sunt întotdeauna implementate corect pentru aplicații. Aceste tipuri de slăbiciuni pot permite unui atacator să capteze sau să ocolească metodele de autentificare utilizate de o aplicație web. Când se întâmplă acest lucru, un atacator ar putea compromite parolele, cheile sau jetoanele de sesiune. Scopul unui așa atac este de a prelua unul sau mai multe conturi, iar atacatorul să obțină aceleași privilegii ca și utilizatorul atacat.

Autentificarea multi-factor este una dintre cele mai bune moduri de a se apăra împotriva autentificării corupte, deoarece poate împiedica astfel de lucruri ca atacurile de refolosire a credențialelor furate. De asemenea, dezvoltatorii și administratorii nu ar trebui să livreze sau să implementeze acreditări de administrator în aplicații. În cele din urmă, ar trebui să creeze funcții care verifică parolele slabe și limitează numărul de încercări greșite de autentificare.

Se poate deci observa cât de variate pot fi atacurile la aplicațiile Web, ceea ce înseamnă că atât dezvoltatorii, dar și administratorii trebuie în permanență să implementeze diferite soluții de securitate și să monitorizeze funcționarea corectă a aplicațiilor.

Bibliografie

1. Seth Fogie (Author), Jeremiah Grossman (Author), Robert Hansen (Author), Anton Rager (Author),
2. Petko D. Petkov, *XSS Attacks: Cross Site Scripting Exploits and Defense*, 2007.
3. Prakhar Prasad, Mastering Modern Web Penetration Testing, 2016.
4. *https://www.veracode.com/security/buffer-overflow*
5. *https://hdivsecurity.com/owasp-broken-authentication-and-session-management*