Autori: Marian Gușatu, Ruxandra F. Olimid
Departamentul de Informatică, Universitatea din București

- Laboratorul 11 -

Practici de securizare a codului

Disclaimer: Pe parcursul acestui curs/laborator vi se vor prezenta diverse noțiuni de securitate informatică, cu scopul de a învăța cum să securizați sistemele. Toate noțiunile și exercițiile sunt prezentate în scop didactic, chiar dacă uneori se presupune să gândiți ca un adversar. Nu folosiți aceste tehnici în scopuri malițioase! Acestea pot avea consecințe legale în cazul comiterii unor infracțiuni, pentru care **deveniți pe deplin răspunzători**!

1. Securizarea codului

Marcați cu *Adevărat* (A) sau *Fals* (F) afirmațiile de mai jos.

a) Ca să analizați/testați securitatea aplicației, ajută să gândiți ca un atacator.
b) Pentru că sunt foarte multe, din punct de vedere al logicii/design-ului aplicației, nu încercați să acoperiți toate cazurile posibile pentru a preveni un comportament neașteptat.
c) Întotdeauna validați câmpurile de input, atât ca format (tip de date, protejare împotriva SQL injection, etc.) dar și ca valori (dimensiuni, valori minime/maxime, verificări între diferite câmpuri de input; ex. data de început a unei activități anterioară datei de final, prețurile să aibă valori pozitive, etc.)
d) Aveți în vedere vulnerabilități de tip buffer overflow.
e) În general nu e o practică bună să stocați log-uri, pentru că ocupă spațiu și cresc timpul de așteptare al utilizatorului.
f) Oferiți cât mai multe detalii posibile utilizatorilor când eșuează autentificarea prin username și parolă sau când implementați mecanisme de recuperare a parolei, pentru a facilita accesul acestora (spre exemplu menționați „Adresa de e-mail nu corespunde unui cont activ” la încercarea de a recupera parola prin e-mail).
g) Dacă folosiți baze de date, în mod normal la ștergerea unei înregistrări folosiți DELETE, pentru a nu mai păstra nicio urmă a acesteia (afirmația nu face referire la ștergerea permanentă a unor date personale, conform GDPR [1]).
h) Nu rețineți parole în clar.
i) Hardcodați parole în cod.

Amintiți-vă aceste aspecte atunci când dezvoltați o aplicație, spre exemplu lucrarea de licență sau alte proiecte!

 Puteți citi mai multe despre securitatea aplicațiilor și diferite vulnerabilități pe pagina OWASP [2,3]!

2. Înregistrarea utilizatorilor

 Realizați o aplicație web cu o pagină care să conțină un formular pentru înregistrarea unui nou utilizator pe baza adresei de e-mail. Menționați cel puțin 3 aspecte de securitate care trebuie avute în vedere și validați input-ul în mod corespunzător pentru a rezolva cazurile menționate.

3. Logarea utilizatorilor

 Realizați o aplicație web cu o pagină care să conțină un formular pentru login. Menționați cel puțin 3 aspecte de securitate care trebuie avute în vedere și validați input-ul în mod corespunzător pentru a rezolva cazurile menționate.

4. Securizarea aplicațiilor

 Considerăm scenariul în care trebuie să dezvoltați o aplicație web (frontend si backend) care să conțină utilizatori cu diverse roluri/permisiuni. Faceți o listă cu cel puțin 5 cazuri în logica/design-ul/implementarea aplicației care ar putea fi exploatate de un atacator din cauza unei dezvoltări non-conforme/nesecurizate a aplicației și care ar putea avea un impact major. Precizati întreg procesul: ce acțiuni întreprinde atacatorul, cum realizează atacul, care este impactul, cum se poate preveni/rezolva o astfel de problemă.

Exemplul 1. Un atacator încearcă diverse manipulari ale link-ului aplicației și reușește să acceseze o pagină la care ar trebui să aibă acces doar un utilizator cu rol de administrator. Această problemă apare deoarece dezvoltatorul nu a realizat o securizare a paginilor pe roluri. Impactul este major deoarece adversarul ar putea dobândi acces la informații confidențiale, modifica drepturile altor utilizatori, etc. O modalitate de prevenție ar fi verificarea permisiunilor de acces la nivel de pagină. Acest exemplu este **acceptat**.

Exemplul 2. Un atacator încearcă să își editeze profilul și își poate schimba data de naștere dar nu ii se actualizează vârsta. Aceasta problemă apare deoarece dezvoltatorul nu a apelat funcția de calculare a vârstei la editarea acesteia. Impactul este major pentru că utilizatorul realizează astfel de anomalii și este posibil să îi fie negat accesul la conținut/acțiuni (ex. dacă este minor

vs. major). Acest exemplu **nu este acceptat** pentru că este un bug care nu are un impact care ar putea aduce beneficii clare atacatorului. Dacă însă impactul este bine motivat și se schimbă contextul, atunci putem lua în considerare un exemplu similar.

5. Testarea aplicațiilor

Utilizați OWASP ZAP [4] pentru a verifica securitatea aplicațiilor web. Puteți folosi pentru testare aplicația de la exercițiile precedente, o altă aplicație web personală (ex. lucrarea de licență) sau aplicația de test *OWASP Juice Shop* [5]. Ce opțiune ați folosit (*Automated scan*, *Manual exploring*)? Ce vulnerabilități ați descoperit? Faceți o scurtă prezentare a modului de utilizare și a rezultatelor obținute.

Referințe bibliografice

1. General Data Protection Regulation (GDPR). Accesibil la: https://gdpr-info.eu/ Ultima accesare: decembrie 2021.
2. Open Web Application Security Project (OWASP). Accesibil la: https://owasp.org/ Ultima accesare: decembrie 2021.
3. OWASP. *Vulnerabilities*. Accesibil la: https://owasp.org/www-community/vulnerabilities/ Ultima accesare: decembrie 2021.
4. OWASP Zed Attack Proxy (ZAP). Accesibil la: https://www.zaproxy.org/ Ultima accesare: decembrie 2021.
5. OWASP. *OWASP Juice Shop.* Accesibil la: https://owasp.org/www-project-juice-shop/ Ultima accesare: decembrie 2021.