**SOSA Lab 4 - Ivan Futivić 0036522493**

**Opis instalacije i pokretanja alata**

* Korištene verzije:
  + OWASP Mutillidae 2.11.4
  + OWASP ZAP 2.12.0
* Instalacija:
  + OWASP Mutillidae
    - OWASP Mutillidae pokrenut je pomoću docker compose datoteke dostupne u webpwnized repozitoriju: https://github.com/webpwnized/mutillidae-docker
    - Nakon pokretanja, web sučelju je moguće pristupiti pomoću adrese localhost:82, a LDAP admin konzoli pomoću localhost:81
    - Tijekom inicijalnog pristupa potrebno je bilo resetirati bazu podataka i importati .ldif datoteku koja se nalazi u istom repozitoriju
  + OWASP ZAP
    - OWASP ZAP instaliran je na macOS uređaj pomoću službene instalacijske datoteke
    - Nakon pokretanja programa, proxy je u web pregledniku postavljen pomoću "manual explore" opcije koja pokrene odvojenu instancu preglednika s unaprijed postavljenim proxyjem
    - A screenshot of a manual

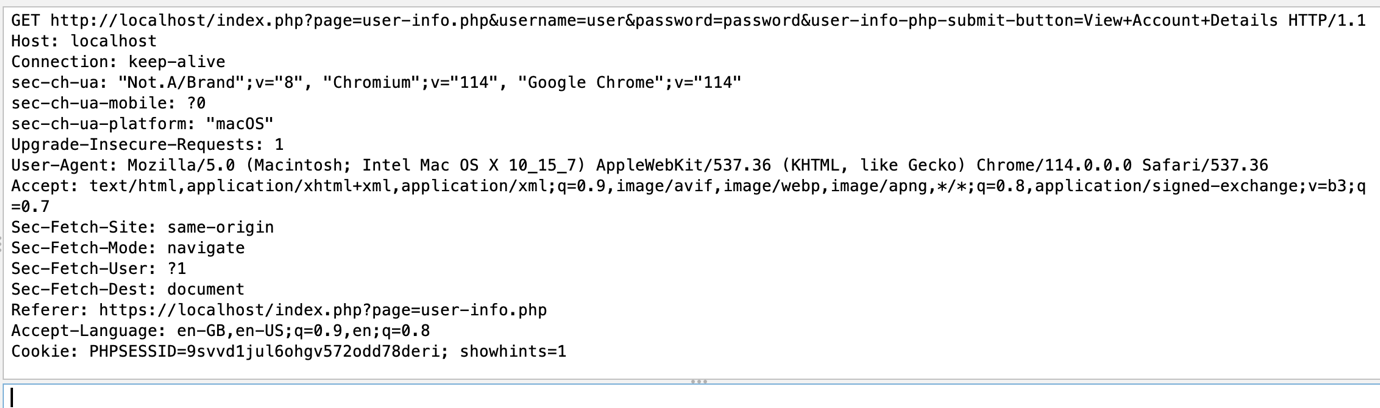
      Description automatically generated with low confidence

**OWASP 2017 - A1 - Injection (SQL) - SQLi Extract Data - User Info (SQL)**

Na stranici unesemo nasumične podatke i kliknemo "View Account Details" kako bismo generirali zahtjev.  
A screenshot of a login form

Description automatically generated with low confidence

U ZAP-u možemo vidjeti taj request te da se username i password šalju   
kao parametri u URL-u.



Sada možemo konfigurirati fuzzer da na mjestu username parametra postavlja vrijednosti koje bi mogle uzrokovati SQL injection.A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Potrebno je za navedeni payload zamijeniti sve "--" oznake komentara s "#" zbog problema s ovom verzijom MySQL-a.

Kako bismo našli sve uspješne SQL injectione pretražujemo rezultate fuzzinga koji ne sadrže izraz "0 records found".

A screenshot of a computer

Description automatically generated with medium confidence

Od navedenih requestova većina ih sadrži stack trace uzrokovan greškom u SQL upitu, ali dva requesta sadrže izraz "23 records found" i popis svih korisnika u bazi.

A picture containing text, screenshot, font, line

Description automatically generated

Ovime smo uspjeli izvući podatke o korisnicima iz baze.

Ovo ukazuje na nedostatak validacije unosa što nam je omogućilo da umjesto korisničkog imena unesemo izraze koji se mogu interpretirati kao SQL naredbe.

Kako bismo ovo spriječili potrebno je koristiti parametrizirane upite (prepared statements) i raditi validaciju unesenih podataka.

**OWASP 2017 - A1 - Injection (Other) - Command injection - Echo Message**

Na stranici unesemo nasumične podatke i kliknemo "Echo message" kako bismo generirali zahtjev.

A screenshot of a computer screen

Description automatically generated with low confidence

Vidimo da se radi o POST requestu koji podatke prenosi u tijelu zahtjeva.

A screenshot of a computer

Description automatically generated with medium confidence

S obzirom da ZAP ne dolazi s listom primjera za command injection koristio sam listu dostupnu u ovom repozitoriju:   
https://github.com/payloadbox/command-injection-payload-list

A screenshot of a computer

Description automatically generated

Nakon pokretanja možemo vidjeti da su mnogi zahtjevi uspješno izveli command injection i vratili nam neke informacije o sustavu.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Ovime smo uspjeli izvesti command injection i dobiti pristup osjetljivim informacijama o sustavu.

Ovo ukazuje na nedostatak validacije unosa što nam je omogućilo da unosimo izraze koji će se izvršiti kao naredbe u ljusci.

Kako bismo ovo spriječili potrebno je maknuti mogućnost izravnog izvođenja komandi u ljusci. Umjesto toga potrebno je koristiti funkcije iz programskog jezika koje vrše istu funkciju.

Ako se moraju podaci izravno unositi u ljusku potrebno je raditi bolju validaciju unosa.

**OWASP 2017 - A2 - Broken Authentication and Session Management - Authentication Bypass- via Brute Force**

Unesemo nasumične podatke za korisnika admin kako bismo generirali zahtjev.

A screenshot of a login screen

Description automatically generated with medium confidence

Pokušamo isto napraviti sa nekim (vjerojatno) nepostojećim korisnikom i vidimo da dobimo različitu poruku.

A picture containing text, screenshot, line, font

Description automatically generated

Ovo nam ukazuje na to da postoji korisnik admin, pa možemo pokušati pomoću fuzzinga pronaći njegovu lozinku.

U ovom POST zahtjevu se također username i password šalju u tijelu.

Kao input za fuzzing koristio sam popis milijun najčešćih lozinka na internetu koji je dostupan ovdje:

https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-100000.txt

Kako bih skratio potrebno vrijeme za brute force napad koristio sam samo lozinke iz popisa koje su sadržavale riječ "admin".   
(iz prijašnjeg SQL injectiona sam dobio informaciju da admin ima lozinku adminpass)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Kako bih saznao je li pronađena ispravna lozinka pretražio sam sve odgovore sa HTTP kodom 302 (Found).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Ovime smo saznali da je lozinka za korisnika "admin" zapravo "adminpass".

Sada na početnoj stranici možemo vidjeti da smo se uspješno prijavili kao admin.

A black text on a white background

Description automatically generated with medium confidence

Ovo ukazuje na nedostatak potrebnih mjera za zaštitu od brute force napada.

Kako bismo ovo spriječili potrebno je implementirati neke od tih mjera kao što su maksimalan broj dopuštenih pokušaja prijave i timeout nakon dovoljno neuspjelih unosa.

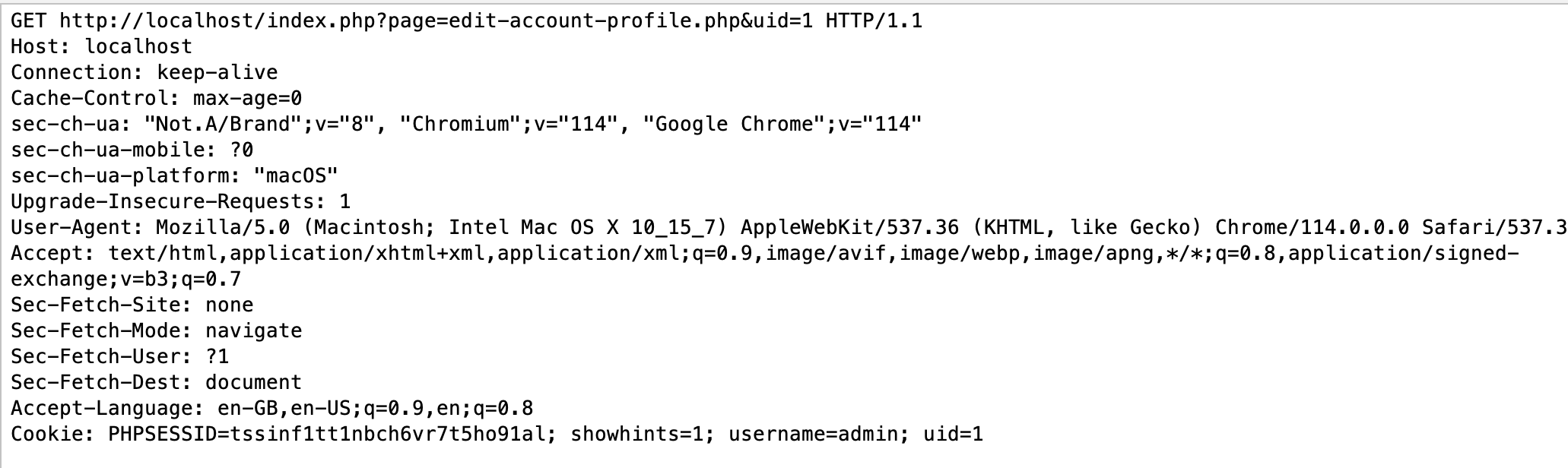
**OWASP 2017 - A2 - Broken Authentication and Session Management - Username enumeration - Edit User Profile**

Nakon uspješnog logina možemo za trenutnog korisnika mijenjati username, password i signature.

A screenshot of a login screen

Description automatically generated with medium confidence

Možemo vidjeti da GET zahtjev za pristup toj stranici sadrži "uid" parametar u URL-u.



S obzirom da je taj parametar jednak broju 1 za admin korisnika možemo s fuzzerom probati druge numeričke vrijednosti.

Koristimo Numberzz fuzzer sa brojevima od 1 do 1000.

A screenshot of a computer

Description automatically generated

U odgovorima možemo pretražiti korisnička imena nekih korisnika (npr. john) i na taj način vidjeti njihove podatke.

A screenshot of a computer

Description automatically generated with medium confidence

Također možemo pretražiti sve odgovore koje ne sadrže prazni signature kako bismo dobili popis svih korisnika.

A screenshot of a computer

Description automatically generated with medium confidence

Ovime smo dobili pristup podacima svih korisnika.

Ovo nam ukazuje na korištenje enumeracije kod ID-jeva korisnika i na nedostatak kontrole pristupa web stranicama.

Kako bismo ovo spriječili potrebno je koristiti nasumično generirane ID-jeve za korisnike (npr. UUID) i potrebno je imati kontrolu pristupa koja omogućava pristup ovoj stranici samo ako je taj isti korisnik trenutno prijavljen.