Penetraciono testiranje

Penetraciono testiranje spada u etičko hakovanje i podrazumeva testiranje računarskih sistema, mreže ili veb aplikacija kako bi se pronašle ranjivosti koje haker može da iskoristi i zloupotrebi. Penetraciono testiranje se može automatizovati pomoću softverskih alata ili se može vršiti ručno. U oba slučaja, proces obuhvata prikupljanje informacija o ciljanoj aplikaciji pre testiranja, identifikaciju mogućih ulaznih tačaka i pokušaj ulaska u sistem. Na kraju je potrebno dokumentovati pronađene ranjivosti u obliku izveštaja.

Izveštaji koje pentesting izgeneriše treba da daju povratnu informaciju organizacijama čije aplikacije se testiraju kako bi trebalo postaviti prioritete što se tiče bezbednosti. Takođe, mogu da pomognu developerima kako bi napravili što bezbednije aplikacije. Namera je motivisati developere da prošire postojeće znanje u domenu bezbednosti kako bi se detektovane ranjivosti sanirale u budućim projektima. Za penetraciono testiranje rent-a-car sistema će se koristiti sqlmap i owaspZAP alati.

SqlMap

Ovaj open-source alat za pentestiranje omogućava automatizaciju detektovanja i zloupotrebe SQL injection nedostataka i preuzimanje servera baze podataka. Neke od mogućnosti ovog alata su:

- Puna podrška za mySQL, postgresSQL, microsoftSQL, MariaDB, H2 (i mnoge druge) sisteme za upravljanje bazom podataka
- Direktno povezivanje sa bazom podataka pomoću DBMS kredencijala, IP adrese i porta i naziva baze podataka
- Može da vrši enumeraciju korisnika, heševa lozinki, permisija, uloga, tabela i kolona
- Automatski prepoznaje format hash lozinke i podržava dictionary-based napade

OwaspZAP

Besplatan open-source alat za početnike u pentestiranju koji omogućava otkrivanje ranjivosti u web aplikacijama. ZAP kreira proxy web server kroz koji prolazi sav „saobraćaj“ veb aplikacije. Za presretanje zahteva i otkrivanje ranjivosti se koristi automatski skener, koji ima mogućnost generisanja izveštaja. ZAP se ponaša kao middle-man između browsera i veb aplikacije kako bi presretao poruke i menjao njihov sadržaj po potrebi. U nastavku su navedeni detektovani problemi pomoću ovog alata i priložena su rešenja.

- **Path Traversal (nivo rizika: VISOK)**

Ova vrsta napada omogućava hakerima da pristupe fajlovima, direktorijumima koji se potencijalno nalaze van korenskog direktorijuma veb aplikacije. Najčešće se napadi vrše na nizom karaktera kao što su “../” i njegove alternative. Pored ovog, česta je upotreba i nevalidnih UniCode-encoding unosa koji umesto kose crte sadrže % (%2e%2e%2f, ..%255c…).

- Detektovani problemi: parametri koji nisu zaštićeni od ovakvog unosa - *kidsSeats, max_price, mileage, mileageLimit, min_price, rating, startDate, city*
- Rešenje: Iz data transfer objekata koji se šalju i prihvataju kroz http zahteve su uklonjeni parametri koji se ne koriste (max_price, min_price). U svakoj metodi gde se ovi parametri koriste, prethodno je pozvana validacija nad poljima pomoću regex izraza. Svaki izraz je formiran tako da proverava dužinu, tip unosa, opseg i karakteristike pojedinačnog parametra. Sprovedena je provera validacije svakog polja u svakom objektu aplikacije, i svakog parametra koji se šalje kroz HTTP zahtev.

➢ SQL injection (nivo rizika: VISOK)

Vrsta napada u kojem se ubrizgava SQL kod koji može da manipuliše podacima u korištenoj bazi podataka, naruši njihov integritet ili ih učini nedostupnim.

- Detektovan problem: uspešno izvršen napad nad city parametrom sa [AND 1=1 --] i [AND 1=2 –]
- Rešenje: pored validacije na bekendu, urađen i escaping svih parametara dobijenih sa fronta

➢ Buffer Overflow (nivo rizika: SREDNJI)

Ovaj tip greške karakteriše prekomerno upisivanje u memorijske lokacije pozadinskih procesa kojima nije dozvoljeno pristupati. Najčešće ovaj tip napada dovodi do zaustavljanja aplikacije ili njenog nepredviđenog ponašanja.

- Detektovan problem: 500 internal server error za zahtev čiji je content-length: 2398
- Rešenje: za svaki response je izvršena provera dužine odgovora, postaviti maksimalnu dužinu odgovora koji dolazi sa backend-a

➢ Cross-Domain Misconfiguration (nivo rizika: SREDNJI)

Pogrešna CORS konfiguracija na veb serveru dozvoljava čitanje zahteva sa proizvoljnih domena trećih strana (third party domains) koristeći neovlašćene API-je na ovom domenu. Veb pretraživači koji se danas koriste podržavaju Same-Origin politiku i onemogućavaju trećim stranama da čitaju odgovor ovakvih zahteva, što donekle smanjuje rizik od napada i svrstava ga u srednji nivo.

- Detektovan problem: Access-Control-Allow-Origin: *
- Rešenje: postavljen Access-Control-Allow-Origin header svakog zahteva na https://localhost:8099 kako bi se dozvolilo slanje i čitanje zahteva samo sa tog domena.

➢ X-Frame-Options Header Not Set (nivo rizika: SREDNJI)

X-Frame-Options header je deo HTTP odgovora i može da se koristi kao indikator koji će pretraživaču dati dozvolu da renderuje stranice u `<frame>`, `<iframe>`, `<embed>` ili `<object>` tagovima. Postavljanje ovog hedera onemogućava napadačima da ugrade svoj sadržaj na tuđe stranice.

- Detektovan problem: ne postoji X-Frame-Options header u HTTP response-u
- Rešenje: X-Frame-Options: DENY
  S obzirom da naša aplikacija ne zahteva renderovanje u okviru nekog <iframe> taga, treba postaviti X-Frame-Options na DENY vrednost što će sprečiti renderovanje u <iframe> kako sa drugih sajtova, tako i iz naše aplikacije i onemogućiti izvršavanje Click-jacking napada.

➢ Web Browser XSS Protection Not Enabled (nivo rizika: NIZAK)

Veb pretraživači imaju svoj mehanizam zaštite od XSS napada koji treba da bude omogućen. Ovaj problem se detektuje kod zahteva u čijem odgovoru se može naći xss sadržaj. Da bi se omogućio ovakav mehanizam zaštite, potrebno je postaviti X-XSS-Protection: 1.

➢ Application Error Disclosure (nivo rizika: NIZAK)

Stranica na kojoj je detektovan ovaj problem sadrži ispis greške ili upozorenja koji napadaču mogu otkriti lokaciju fajlova i otvara mogućnost napadaču da dublje istraži ispis i izazove ozbiljnije napade.

- Detektovan problem: ispis greške na https://localhost:8099/car
- Rešenje: ukoliko zahtev vrati grešku, redirektovati korisnika na implementirane stranice za grešku (Error404 i Error500)

➢ Cross Site Scripting Weakness (Reflected in JSON Response) (nivo rizika: NIZAK)

XSS napad reflektovan in JSON odgovoru koji može dovesti do ranjivosti korisnika koji će konzumirati ovaj odgovor. U ovom slučaju je svrstan u nizak nivo opasnosti iz razloga što Content-Type nije HTML.

- Detektovan problem: izvršen napad <script>alert(1);</script> nad https://localhost:8099/advertisement/addNew u *mileage* parametru
- Rešenje: pozvati OwaspEncoder i uraditi escaping parametra nad kojim je izvršen napad

➢ Information Disclosure - Debug Error Messages (nivo rizika: NIZAK)

Odgovor sadrži uobičajene debug poruke karakteristične za Apache veb server na kom je aplikacija podignuta.

- Detektovan problem: https://localhost:8099/car ispisuje debug poruku ako dođe do greške
- Onemogućiti ispis debug poruka prilikom postavljanja aplikacije u produkciju

➢ X-Content-Type-Options Header Missing (nivo rizika: NIZAK)

Omogućava starijim verzijama Internet Explorera i Chrome-a MIME-sniffing nad sadržajem odgovora sa backend-a, što potencijalno može da dovede po pogrešne interpretacije odgovora, u drugom formatu u odnosu na očekivani.

- Detektovan problem: zahtevi ne sadrže X-Content-Type-Options header
- Rešenje: postaviti X-Content-Type-Options: nosniff za sve stranice aplikacije, uključujući i one koje služe za error handling jer sui one podložne skreniranju sadržaja i napadima.