Izveštaj o penetracionom testiranju

1. Uvod

Penetraciono testiranje je proaktivni načini testiranja veb aplikacija izvršavanjem napada koji su slični stvarnim napadima, a mogu da se dese bilo kada. Ovo testiranje se vrši na kontrolisani način, u cilju pronalaženja što više nedostataka u bezbednosti i obezbeđivanja povratnih informacija kako da se izbegnu rizici koje izazivaju ovi nedostaci.

To je efikasan način za identifikaciju ranjivosti u sistemima i otkrivanju da li se ranjivost može eksploatisati ili ne. Penetraciono testiranje se reguliše ugovorom između izvršioca i vlasnika sistema koji treba da se testira. Potrebno je definisati oblast testa, radi identifikovanja sistema koji će biti testirani, i pravila angažovanja koja određuju na koji će način biti izvršeno testiranje.

Uz penetraciono testiranje se uglavnom koristi i procena ranjivosti koja je mnogo opširnija od penetracionih testova. Krajnji rezultat procene ranjivosti je izveštaj u kojem se opisuju pronađene ranjivosti - najteže su izlistane na vrhu liste, a one koje predstavljaju manji rizik nalaze se niže u izveštaju.

2. Alat za testiranje – OWASP ZAP

Za potrebe ovog projekta korišćen je alat OWASP ZAP. OWASP ZAP je jedan od najpopularnijih svetskih alata za bezbednost, deo je OWASP zajednice. Dizajniran je tako da je pogodan za upotrebu od strane ljudi različitog iskustva u bezbednosti, i kao takav je idealan za developere i testere koji su novi u pentracionom testiranju. Pogodan je za različite platforme. Kreira proxy između klijenta i našeg web sajta. Dok prolazimo kroz sve funkcionalnosti našeg sajta on beleži sve akcije i zatim napada naš sajt poznatim tehnikama.

Alat nudi dva načina skeniranja, pasivno i aktivno.

Pasivno skeniranje je bezazleno testiranje koje gleda samo odgovore i traži poznate ranjivosti. Takođe, ovaj način skeniranja ne menja podatke na web sajtu. Sa pasivnim skenom se ne može detktovati SQL Injection.

Aktivno skeniranje napada web sajt koristeći pozante tehnike da bi pronašao ranjivosti. Aktivno skeniranje modifikuje podatke i može da ubaci maliciozne skripte na web sajt. U našem slučaju korišćeno je aktivno skeniranje za pronalaženje ranjivosti.

Kako bi se dobila kompletna struktrura sajta potrebno da je se uradi *crawl* celog veb sajta i na taj način dođe do svih funkcionalnosti i svih mogućih akcija. Ovaj proces se naziva *spidering*

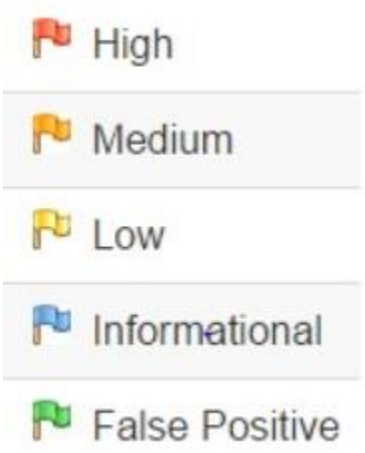
i koristi se u slučaju kada je teško ili komplikovano manuelno doći do svih aspekata sajta. U našem slučaju korišćen je ajax spider.

Ovaj alat ima mogućnost generisanja izveštaja sa detektovanim ranjivostima. Za ovaj projekat napravljeni su izveštaji za tri frontend aplikacije:

- Literarno udruženje
- Koncentrator plaćanja
- Banka

Izveštaji su priloženi u folderu reports**.**

Ranjivosti koje mogu biti detektovani alatom mogu biti različitog nivoa rizika. Tipovi ranjivosti izlistani od najvišeg ka najnižem nivou rizika prikazane su na slici: (Slika 1)



Slika 1

3. Rezultati testiranja

Nakon aktivnog skeniranja rezultati testiranja su pokazali da naše aplikacije nemaju ranjivosti visokog nivoa. Međutim, kako bismo dodatno testirali naš sistem pokušali smo manuelno da izvršimo poznate napade kao što su SQL Injection i XSS napad.

SQL Injection je bezbednosna ranjivost koja dozvoljava napadaču da utiče na upite koje aplikacija pravi ka bazi podataka, i time može omogućiti napadaču da pristupi podacima koji mu inače ne bi bili dostupni ili da mu omogući da izvrši akcije nad bazom koje nisu predviđene. Uspešan SQL Injection napad može rezultovati neautorizovanom pristupu osetljivim podacima kao što su lozinke, podaci sa kreditne kartice ili ličnim informacijama korisnika.

U našem slučaju pokušali smo izvršiti SQL Injection napade na input polja u postojećim formama sa određenim podacima koji se koriste za postizanje malicioznih upita, kao što je: " or ""=". Međutim, naše aplikacije nisu dozvolile te napade zato što koristimo Spring Security i kredencijali u bazi su heširani, takođe upiti koje koristimo ka bazi su parametrizovani i eskejpuju specijalne karaktere.

XSS (Cross Site Scripting) napadi su tip injection napada u kojima se maliciozne skripte ubacuju na sajtove. XSS napadi se javljaju kada napadač koristi veb aplikaciju da šalje maliciozan kod različitim krajnjim korisnicima, uglavnom u formi skripti koje se izvršavaju u browseru. Browser krajnjih korisnika nema način da zna da tim skriptama ne bi trebao da veruje i izvršiće ih.

U našem slučaju to nije moglo biti izvšeno jer pored pravila validacije koja postoje na front-u i back-u, a koja pomažu sprečavanju ovog napada, react biblioteka će svaki tekst koji je unet posmatrati kao tekst i neće ga nikad izvršavati kao skriptu. Time, iako bi maliciozna skripta bila uneta u bazu, tamo gde se ona treba prikazati, biće posmatrana isključivo kao tekst za prikaz i neće biti izvršena.

Prilikom penetracionog testiranja može doći do DoS (Denial of Service) umesto da izvršilac testa dobije pristup sistemu. Zbog toga mnogi izvršioci ne pokreću takve testove da bi izbegli nenamerno izazivanje pada sistema. Sistemi koji nisu testirani na DoS napade su podložniji napadima provalnika.

Prilikom penetracionog testiranja našeg sistema nije izvšen DoS napad.

U nastavku su prikazane pronađene ranjivosti dobijene penetracionim testiranjem koje nisu visokorizične i ne narušavaju bezbednost sistema (Tabela 1)

Naziv ranjivosti	**Nivo rizika**
Cross-Domain Misconfiguration	Medium
X-Frame-Options Header Not Set	Medium
CSP: Wildcard Directive	Medium
Absence of Anti-CSRF Tokens	Low
Incomplete or No Cache-control and Pragma HTTP Header Set	Low
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low
X-Content-Type-Options Header Missing	Low

Tabela 1

Rešenje za svaku od ovih ranjivosti dato je u izveštajima koji su generisani na osnovu rezultata penetracionog testiranja.