24. SIGURNOSNI TESTOVI

Sigurnosno testiranje može se opisati kao vrsta testiranja softvera čiji je **cilj identifikovanje ranjivosti koje potencijalno mogu dopustiti zlonamerni napad na softver**. U sigurnosnom testiranju testeri koriste pristupe zasnovane na riziku, odnosno pristupe koji se zasnivaju na na načinu razmišljanja napadača, da bi na odgovarajući način procenili sigurnost softvera. Utvrđivanjem rizika u sistemu i stvaranjem testova vođenih tim rizicima, tester se može usredsrediti na područja koda u kojima će napad verovatno i uspeti.

Prema QA TestLab-u, postoji 7 vrsta sigurnosnih testiranja:

- Skeniranje ranjivost (Engl. "Vulnerability scanning") se sprovodi proverom sistema skeniranjem poznatih ranjivosti.
- Sigurnosno skeniranje (Engl. "Security scanning") podrazumeva proveru mrežnih i sistemskih slabosti.
- Penetracijski testovi (Engl. "Penetration testing") simuliraju napad zlonamernog koda te proveravaju potencijalne ranjivosti na pokušaje spoljnih neovlaštenih upada u sistem.
- Procena rizika (Engl. "Risk assessment") analizira sigurnosne rizike uočene u organizaciji, kao rezultat daju se preporuke kontrole i mere za smanjenje rizika.
- Revizija ranjivosti (Engl. "Security auditing") je interna revizija sistema s ciljem pronalaska sigurnosnih nedostataka.
- Procena sigurnosnog držanja (Engl. "Posture assessment") odnosi se na sigurnosni status i osviještenost o važnosti sigurnosti unutar sustava ili organizacije.
- Etičko hakovanje (Engl. "Ethical hacking") je metoda legalnog provaljivanja u računare i uređaje unutar organizacije.

Tehnike testiranja sigurnosti aplikacija:

1) **Pristup aplikacij**i – Tester treba da kreira nekoliko korisničkih naloga s različitim pravima i ulogama. Zatim treba da koristi aplikaciju uz pomoć tih naloga i trebao bi da potvrdi da svaka uloga ima pristup samo svojim modulima, ekranima, obrascima i menijima. Testovi treba da provere kvalitet lozinke, prijavu korisnika, oporavak lozinke, funkciju odjave korisnika, promenu lozinke, test za sigurnosno pitanje I njegov odgovor itd.
2) **Zaštita podataka** - Tester bi trebao da pretraži lozinke korisničkih naloga u bazi podataka, poslovno kritične i osetljive podatke, trebao bi da proveri jesu li svi takvi podaci memorisani u šifrovanom obliku. Tester treba da potvrdi da se podaci prenose između različitih formi ili prozora samo nakon odgovarajuće enkripcije. Treba osigurati da su šifrovani podaci ispravno dešifrovani. Posebnu pažnju treba obratiti na različite radnje potvrde, tj. 'submit' funkcije. Mora se proveriti da se informacije, kada se prenose između klijenta i servera, ne prikazuju u adresnoj traci web pretraživača u razumljivom formatu. Drugi način testiranja zaštite podataka je provera korištenja odgovarajućim alogritmima.
3) **Brutalni napadi** – "Brutal Force Attack" se uglavnom izvodi pomoću softverskih alata. Koncept je takav da se korištenjem važećeg korisničkog naloga pokušava pogoditi lozinka, pokušavajući se ponovo i ponovo prijavljivati. Jednostavan primer zaštite od takvog napada je suspenzija računa na kratko vreme, kao što to čine sve

aplikacije za slanje e-pošte poput Yahooa, Gmaila i Hotmaila. Ako se posle određenog broja uzastopnih pokušaja (uglavnom 3) korisnik ne uspe prijaviti, tada se taj račun blokira na neko vreme (30 minuta do 24 sata). Kako testirati Brute-Force Attack: Tester mora proveriti da li je neki mehanizam blokade naloga dostupan i da li radi dobro. Mora pokušati da se prijavi s nevažećim korisničkim nalogom i lozinkom, alternativno, kako bi bio siguran da softverska aplikacija blokira račun ako se kontinuirano pokušava prijaviti. Ako aplikacija to radi, tada je sigurna od napada grubom silom.

4) **SQL Injection (ubrizgavanje)** - U ovom pristupu zlonameran SQL skript koriste hakeri kako bi manipulisali web sajtom. Postoji nekoliko načina zaštite od takvih pokušaja. Za sva polja za unos na web stranici, dužine polja trebaju biti definisana dovoljno mala da ograniče unos bilo kog skripta. Tester mora da osigura da su maksimalne dužine svih polja za unos definisane i implementirane, da definisana dužina polja za unos ne prihvata nikakav unos skripte, kao ni unos oznake(a).
5) **Pristupne tačke usluge** - Danas kompanije zavise jedna o drugoj i međusobno sarađuju, a isto važi i za aplikacije, posebno za web stranice. U tom slučaju, oba saradnika trebaju definisati i objaviti neke pristupne tačke jedan za drugoga. Ako postoji veliki ciljani broj korisnika aplikacije, tada pristupne tačke trebaju biti dovoljno otvorene da olakšaju pristup svim korisnicima, dovoljno prilagodljive da ispune sve zahteve korisnika i dovoljno sigurne da se nose sa svim sigurnosnim situacijama. U nekim slučajevima pristupne tačke mogu biti zapečaćene za neželjene aplikacije ili osobe. To zavisi od poslovnog domena aplikacije i njenih korisnika.
6) **Upravljanje sesijom** - Web sesija je niz HTTP zahteva i odgovora povezanih s istim korisnikom. Proverava se kako se upravlja sesijom u web aplikaciji, upotreba promenljivih, njihovih vrednosti, kreiranje i uništavanje sesije. Može se testirati istek sesije nakon određenog vremena mirovanja, prekid sesije nakon maksimalnog trajanja, prekid sesije nakon odjave, proveriti opseg i trajanje kolačića sesije, testirati može li jedan korisnik imati više istovremenih sesija itd.
7) **Rukovanje greškama** - Testiranje za rukovanje greškama uključuje: proveru kodova grešaka: npr. 408 isteklo vreme zahteva, 400 loši zahtevi, 404 nije pronađeno, itd. Da bi se ovo testiralo, trebate napraviti određene zahteve na stranici tako da se ovi kodovi grešaka vrate. Kod greške biće vraćen s detaljnom porukom. Ova poruka ne bi trebala sadržati nikakve kritične informacije koje se mogu koristiti u svrhe napada na softver.
8) **Specifične rizične funkcionalnosti** - Uglavnom, dve rizične funkcije su plaćanja i učitavanje datoteka. Za učitavanje datoteka mora se prvenstveno testirati da li je bilo neželjenog ili zlonamernog učitavanja datoteka, koje treba biti ograničeno. Za plaćanja se prvenstveno mora testirati ranjivosti ubrizgavanja, nesigurno kriptografsko memorisanje, prekoračenje memorijskih bafera, pogađanje lozinke itd.

Bezbednost aplikacija podrazumeva da je obezbeđen autorizovan pristup funkcijama softvera i podacima koji moraju biti zaštićeni, tj. neautorizovani pristup nesme biti dozvoljen. Bezbednost ima dva apsekta: zaštita podataka i zaštita pristupu podacima.

Desktop aplikacija trebala bi biti sigurna ne samo u pogledu pristupa, već i u pogledu organizacije i čuvanja podataka. Slično tome, web aplikacije zahtevaju još veću sigurnost u pogledu pristupa, uz zaštitu podataka. Programer bi trebao uraditi **web aplikaciju** tako da bude imuna na SQL injekciju, napade grubom silom i XSS (cross-site scripting). Slično tome, ako web-aplikacija omogućuje daljinske pristupne tačke, one takođe, moraju biti sigurne.

Mnogo je različitih uslova koji će uticati na izbor određene metode ili alata za sprovođenje sigurnosnih testova. Iako ne postoji jedinstveno rešenje ili metoda za sprovođenje ove vrste testiranja od izuzetne je važnosti zaštita podataka, aplikacije ili cele organizacije. Preduzimanje pravovremenih mera u zaštiti sistema svakako može uticati na reputaciju organizacije.