Napredni razvoj programske potpore za web

Drugi projekt – Detaljne upute za izvođenje projekta

SQL umetanje (*SQL Injection*)

U aplikaciji pokazati barem dva od četiri poznata načina napada SQL umetanjem:

1. Tautologija
2. Ilegalni upiti
3. Upit UNION
4. Ulančani upiti

Programski kod i primjeri nalaze se u predavanju te u aplikaciji DVWA.

U sprečavanju SQL umetanja onemogućiti neposredno interpretiranje naredbi (umetati unos iz sučelja neposredno u SQL upit). Validirati i filtrirati unos od korisnika. Dodatno je moguće dopustiti ograničen unos naredbi (*whitelisting*). Ako je u aplikaciji definirano više korisničkih uloga, potrebno je minimizirati ovlasti nad bazom podataka kako bi se spriječio pristup neželjenim podacima. Ne prikazivati specifične poruke grešaka ili, još bolje, bilo kakve poruke. Moguće je koristiti „pripremljene izjave" (*prepared statements*) i pohranjene procedure (*stored procedures*).

Loša autentifikacija (*Broken Authentication*)

Odabrati jednu od dvije ranjivosti: 1) pogoditi ime i lozinku korisnika ili 2) ukrasti identifikator sjednice i lažno se predstaviti.

Ako je odabrana ranjivost 1) onda generirati slabu lozinku i demonstrirati *brute force* napad, vertikalni ili horizontalni napad kako je opisano u predavanju.

Ukoliko je odabrana ranjivost 2) onda je u aplikaciji potrebno demonstrirati otmica sjednice *(session hijacking)*, npr. ukrasti identifikator sjednice košarice kupca u online trgovini (*session fixation*) koristeći bilo koji postupak s kojim se pokušava odrediti *Session ID* korisnika koristeći ugrađene ranjivosti sustava.

Obje ranjivosti su opisane u predavanjima i u dodatnoj literaturi na OWASP-ovim stranicama (npr. https://owasp.org/www-community/attacks/Session_fixation).

Koristiti općenite, a ne specifične poruke grešaka. Za upravljanje pristupom koristiti neke od tehnika višefaktorske autentifikacije (npr. CAPTCHA ili dodatna pitanja), tehnike hashiranja ili *password salting* za čuvanje lozinki. Za sprečavanje otmice sjednice generirati dovoljno dug identifikator sjednice i vlastite tokene. Ako je u aplikaciji definirano više korisničkih uloga, onda uloge trebaju imati minimalno dovoljni skup ovlasti. Implementacija funkcionalnosti autentifikacije treba biti jednostavna, centralizirana i standardizirana.

Nesigurna pohrana osjetljivih podataka (*Sensitive Data Exposure*)

Ova ranjivost odnosi se na mogućnost zlonamjernog pristupa osjetljivim podacima i mjestima na kojima se takvi podaci nalaze (npr. relacijska baza podataka, web stranice s javnim pristupom,

poslužiteljski logovi, backup). Podaci na koje se odnose odredbe GDPR također spadaju po osjetljive podatke.

U otklanjanju ove ranjivosti potrebno je Identificirati sve takve podatke i mjesta na kojima se nalaze te onemogućiti im pristup.

Vanjski XML entiteti (*XML External Entity*, XXE)

Pokazati mogućnost slanja proizvoljne XML datoteke na poslužitelj, parsiranja XML datoteke na poslužitelju, dohvaćanja sadržaja čvorova XML datoteke i dvije sljedeće funkcionalnosti:

- Pokretanje JavaScript koda (npr. <script>javascript:alert(“”);</script>) zapisanog u čvor parsirane XML datoteke
- Dohvaćanje datoteke na poslužitelju i ispis sadržaja datoteke putanjom zapisanom u čvor parsirane XML datoteke

U otklanjanju ranjivosti pokazati da se sadržaj čvorova parsirane XML datoteke očisti (sanitizira) kako bi se onemogućilo pokretanje JavaScript koda upisanog u XML datoteku i dohvaćanje datoteka na poslužitelju.

Cross-site scripting (XSS)

Demonstrirati jednu od tri vrste XSS sigurnosna nedostatka:

- Jednokratni XSS sigurnosni nedostatak (reflektirani)
- Trajni XSS sigurnosni nedostatak (pohranjeni)
- Lokalni ili DOM XSS sigurnosni nedostatak

Pri tome koristiti barem jedan karakterističan način umetanja zlonamjernog JavaScript koda:

- SCRIPT oznaka
- IMG src atribut
- Sadržaj kolačića
- Manipulacija CSS-om
- JavaScript funkcije za obradu događaja u web stranicama

I na taj način ili prikazati sadržaj kolačića korisnika (document.cookie) ili preusmjeriti korisnika na poslužitelj ili određenu web stranicu napadača.

U otklanjanju ranjivosti pokazati da implementiran XSS sigurnosni nedostatak više ne funkcionira te nije moguće prikazati sadržaj kolačića ili preusmjeriti korisnikov web preglednik.

Loša kontrola pristupa (*Broken Access Control*)

U ovom sigurnosnom nedostatku potrebno je pokazati neovlašten pristup nekom URL-u, kao što je prikazano na predavanjima, koristeći oznaku uloge u URL-u.

U ranjivosti onemogućiti napadaču pokretanje funkcionalnosti i usluga na koje nema pravo, ili pristup podacima i korisničkim računima drugih korisnika, ili obavljanje privilegiranih akcija. Koristiti ispravnu

autorizaciju i sigurne reference na objekte ili web stranice. Eliminirati javno dostupne izravne reference i zamijeniti ih sa javnim neizravnim referencama na interne izravne reference. Ukoliko se koriste reference na objekte onda obaviti provjeru formata parametra, provjeru prava pristupa za korisnika i provjera pristupa objektu (čitanje, pisanje, promjena vrijednosti objekta). Za svaki javno dostupni URL web aplikacije treba dopustiti pristup samo autentificiranim korisnicima, provjeriti ovlasti za pristup i postupiti u skladu s njima, i zabraniti pristup svemu na što logirani korisnik nema pravo, posebno konfiguracijama, logovima, izvornim kodovima.

Nesigurna deserijalizacija (*Insecure Deserialization*)

Kroz korisničko sučelje web aplikacije omogućiti proizvoljni unos serijaliziranog objekta. Objekt se deserijalizira (parsira) na poslužitelju i sadržaj se prikazuje u korisničkom sučelju.

U otklanjanju ranjivosti provjeravati očekivane tipove i dobivene tipove podataka u serijaliziranom objektu. Prikazati greške.

Lažiranje zahtjeva na drugom sjedištu (*Cross Site Request Forgery*, CSRF)

U ovom sigurnosnom nedostatku pokazati kako se žrtva samostalno prijavljuje na ranjive web stranice gdje dobiva Session ID od poslužitelja. Nakon toga žrtva otvara link u kojem je ubačen napadačev HTTP GET zahtjev u IMG src atribut (npr. žrtva otvara drugu web stranicu ili neku poruku u kojoj se nalazi slika). Na napadačevoj web stranici ispisuje se Session ID žrtve.

U otklanjanju ove ranjivosti implementirati barem jednu funkcionalnost:

- pohrana tokena u sesiji i dodavanje u HTML forme (obrasce) i linkove
- koristiti HTTP POST umjesto HTTP GET