SECURITY DESIGN PATTERNS LAB

1. Priprema

Predlog za Čitanje

1. OWASP Application Security Verification Standard (ASVS): https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

Pripremni zadatak

1. Svaki tim bi trebalo da kreira listu bezbednosnih dizajn obrazaca (*pattern*, šablon) koje je iskoristio na svojim prethodnim projektima, označavajući koje ASVS zahteve je ispunio primenjujući te obrasce.

2. Uvod

Prilikom integracije bezbednosnih mehanizama u softverski sistem, dva aspekta bi trebalo razmotriti. Prvo bi trebalo identifikovati potrebu za vrstom bezbednosne kontrole, kako bi se prepoznalo da je neki bezbednosni obrazac neophodan. To je cilj modelovanja pretnji (threat modeling) na nivou dizajna, ili analize bezbednosnog dizajna.

Nakon odabira odgovarajućeg obrasca, podjednako je važno dobro ga implementirati. Dobra implementacija se svodi na konfiguraciju bezbednosne kontrole, gde se mora odabrati pouzdan provajder, ispitati da li verzija koja se koristi u implementaciji ima ranjivosti, i istražiti koje su najbolje prakse za konkretnu kontrolu. Loša konfiguracija može dovesti do toga da sistem ima isto toliko ranjivosti koliko bi ih bilo da te bezbednosne kontrole i nema (ako ne i više[1]).

3. Zadaci

Sledeći zadaci su fokusirani na aktivnost istraživanja, gde se od studenata očekuje da istraže javno dostupne resurse i dokumente kako bi otkrili, analizirali, i povezali informacije i tako dobili zahteve (*requirements*) za bezbednu implementaciju nekog od bezbednosnih dizajn obrasca.

a. Hešovanje lozinki

Retko postoji potreba da se lozinke za autentikaciju čitaju. Nisu česti slučajevi korišćenja (*use case*) gde je zaista potreban upit “SELECT passwords FROM Users”. Zbog toga, nije potrebno skladištiti lozinke u običnom tekstu (*plaintext*), niti je potrebno skladištiti ih u šifrovanom (*encrypted*) obliku. Umesto toga, skladištenje lozinki u savremenim sistemima podrazumeva upotrebu heš funkcija. Prilikom registracije korisnika, lozinka se hešuje i dobijeni heš se skladišti. Prilikom prijave na sistem, prosleđena lozinka se hešuje, i rezultat se poredi sa prethodno sladištenom heš vrednosti.

Zadatak

Dizajnirati mehanizam hešovanja sa ciljem da se zaštiti poverljivost (*confidentiality*) korisničkih lozinki.

[1] OpenSSL Heartbleed vulnerability, http://heartbleed.com/

- Istražiti različite algoritme i odabrati najbezbedniji;
- Ispitati konfiguracione parametre odabranog algoritma, i otkriti koja bi to bila preporučena praksa za konfiguraciju;
- Odabrati pouzdanog provajdera;
- Istražiti da li poslednja verzija za implementaciju ima ozbiljnijih ranjivosti;
- Specificirati zahteve za bezbednu implementaciju heš mehanizma koristeći sve do sada nabrojano.

b. Mehanizam revizije (auditing)

Log datoteke imaju važnu ulogu u održavanju softvera, jer pružaju informacije potrebne za otkrivanje problema prilikom rada softvera. Međutim, log datoteke doprinose i bezbednosti sistema, jer pružaju uvid u događaje i aktere (*non-repudiation*). Logove mogu koristiti alati za monitoring kako bi se detektovali maliciozni akteri i sumnjivo ponašanje.

Zadatak

Dizajnirati mehanizam za logovanje događaja koji odgovara na sledeće zahteve:

- Log datoteke moraju pružiti informacije potrebne za razrešavanje problema;
- Svi događaji za koje su akteri bitni moraju biti zapisani, sa dovoljno informacija kako akteri ne bi mogli da poriču odgovornost (non-repudiation). Potrebno je obezbediti lako izdvajanje tih događaja;
- Stavke log datoteke ne smeju sadržati osetljive podatke;
- Mehanizam za logovanje mora biti pouzdan, mora obezbediti dostupnost i integritet log datoteka;
- Stavke log datoteke moraju precizno iskazati vreme nastanka;
- Mehanizam za logovanje mora stremiti ka tome da su logovi uredni, da je “pretrpanost” minimalizovana.

Zadatak je istražiti kako se svaki od ovih zahteva može ispuniti, i specificirati konkretne korake implementacije za dizajnirani mehanizam. Korišćenje konkretnih rešenja je dozvoljeno, ako dato rešenje ispunjava sve zahteve, ili se može proširiti tako da ispunjava.

c. Dodatne bezbednosne kontrole

Prateći pristup koji je iskorišten da se reše prethodna dva zadatka, moguće je analizirati bilo koji bezbednosni dizajn obrazac i specificirati zahteve za njegovu bezbednu implementaciju.

Zadatak

Analizirati bezbednosne kontrole implementirane na prethodnim projektima, i zaključiti do koje mere se implementirane konfiguracije kontrola razlikuju od preporučenih bezbednosnih konfiguracija.