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 bezbednosti 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.

Svi zadaci u tekstu ispod su deo celine. Ideja je osmisliti delove sistema za čuvanje i deljenje tajni poput lozinki, ključeva i tome sličnog sadržaja. Primer takve aplikacije: https://github.com/Infisical/infisical.

a. Enkripcija tajni (pa i lozinki)

Retko postoji potreba da se lozinke za autentikaciju čitaju, te se skladištenje lozinki u većini savremenih sistema svodi na upotrebu heš funkcija. Međutim, softver koji lozinke posmatra kao tajne koje je potrebno s vremena na vreme otkriti, mora se osloniti na mehanizam enkripcije.

Zadatak

Dizajnirati mehanizam enkripcije sa ciljem da se zaštiti poverljivost (*confidentiality*) korisničkih lozinki, ali tako da je lozinku moguće po potrebi pročitati.

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

- Istražiti različite algoritme za generisanje ključa za enkripciju/dekripciju na osnovu glavne (master) lozinke i odabrati najbezbedniji. Primer: PBKDF;
- Istražiti različite simetrične algoritme za enkripciju/dekripciju i odabrati najbezbedniji;
- Ispitati konfiguracione parametre odabranih algoritama, i otkriti koje bi to bile preporučene praksa za konfiguraciju;
- Odabrati pouzdane provajdere;
- Istražiti da li poslednje verzije za implementaciju imaju ozbiljnijih ranjivosti;
- Specificirati zahteve za bezbednu implementaciju mehanizama za kreiranje ključa i enkripciju 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.
    - Istražite kako se koristi **logrotate** alat za rotaciju logova u tradicionalnim sistemima, sa osvrtom na njegove poznate ranjivosti (npr. log injection i race condition problemi)
    - Alternativno istražite:
        - Docker log rotaciju (log-driver, max-size, max-file) ili
        - Cloud-native rešenja kao što su AWS CloudWatch Logs ili GCP Logging

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. Istražite kako funkcionišu komponente ELK stack-a (Elasticsearch, Logstash, Kibana) i kako se međusobno integrišu za prijem, obradu i vizualizaciju logova, sa naglaskom na bezbednosne događaje. Posebno obratiti pažnju na sledeće:

- Kako se podešava pipeline za prijem logova (npr. od aplikacija, sistema, WAF-a...)
- Kako se logovi indeksiraju i čuvaju u Elasticsearch-u (high-level objašnjenje + šta je neophodno od konfiguracije)
- Kako se vizualizuju i filtriraju u Kibani (high-level objašnjenje)

c. Višefaktorska autentikacija

Kako mnogo toga u zamišljenom sistemu zavisi od glavne (master) lozinke, potrebno je razmisliti i o tome kako ograničiti posledice njenog otkrivanja. Radi toga, potrebno je razmisliti o višefaktorskoj autentikaciji.

ZADATAK

Istražiti načine implementacije višefaktorske autentikacije u web aplikacijama.

- Istražiti i prokomentarisati tipove višefaktorske autentikacije;
- Fokusirati se na (najmanje) dva faktora (npr. lozinka + TOTP);
- Odabrati najmanje dva faktora (npr. lozinka + TOTP (primer: Google Authenticator));
- Objasniti kako se implementiraju odabrani faktori. Na primer, za TOTP bi bilo potrebno objasniti kako se implementira generisanje, verifikacija i obnavljanje TOTP koda;
- Navesti najčešće greške i bezbednosne propuste u implementaciji MFA sistema. Na primer, kod TOTP bi tu trebalo spomenuti problem sinhronizacije vremena, pogađanje backup koda, i slično;
- Na primeru ELK okruženja istražiti o objasniti kako se MFA integriše. Na primer, admin login u Kibanu.