Zadanie 1 - Modelovanie hrozieb

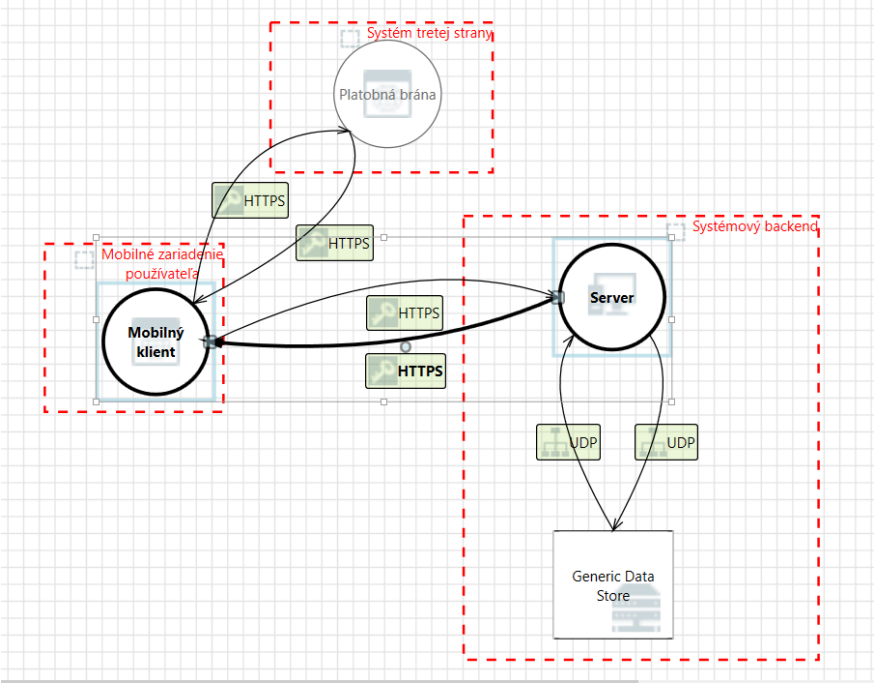
I-UPB: Úvod do počítačovej bezpečnosti

October 4, 2021

1 Úvod

Zadanie pojednáva o modelovaní hrozieb a ich ošetrení podľa metodiky STRIDE pre bikesharing službu. Služba komunikuje so službou tretej strany - platobnou bránou. Pri tejto komunikácií sú prenášané citlivé dáta používateľa. Zároveň počas jazdy na vypožičanom bicykli systém periodicky reportuje polohu používateľa.

2 Data flow diagram



Na diagrame vidíme ako tečú dáva v aplikácií. Na základe týchto tokov vieme zistiť zraniteľnosti pomocou metodiky STRIDE.

3 Trust boundaries

Znázornené trust boundaries boli zvolené na základe architektúry systému. Server a generic data store bežia na jednom fyzickom serveri - trust boundary "Systémový backend". Klientská aplikácia mobilného klienta je izolovaná ako proces na telefóne používateľa, to definuje trust boundary. Platobná brána je služba tretej strany a preto má definovaný svoju vlastnú trust boundary. Komunikácia v každej znázornenej trust boundary je privátna.

4 Zraniteľnosti systému

4.1 Spoofing

- Komunikácia medzi serverom a Generic data store môže byť odchytávaná a prichádzajúce/odchádzajúce dáta môžu byť útočníkom menené. Ochrana pre týmto útokom je implementácia autentifikácie medzi serverom a generic data store.

4.2 Tampering

- Všetky znázornené HTTPS volania sú REST volanie ktoré obsahujú JSON payload. Toto vytvára zraniteľnosť voči JSON injection útoku. Ochrana proti tomuto útoku je escapovanie/sanitácia JSON payloadu.
- Pokiaľ HTTPS komunikácia dovolí prístup ku zdielanej pamäti volanej služby je možné vykonať kód ktorý môže spušťať služby a programy na volanej službe. Ochranou proti takémuto útoku je spúšťať volané webové služby s pod účtom s obmedzenými privilégiami tak aby nemohol spušťať/ukončovať služby a programy ku ktorým byu nemal mať prístup.
- Pokiaľ v HTTPS komunikácií nie je implementovaná kontrola integrity payload-u tak útočník môže obchytávať packety a znova ich prepoužiť. Tým to spôsobom môže odchytiť napríklad zahashované heslo a znova ho prepoužiť. Ochrana je pridanie kontroly integrity payloadu v komunikácií napríklad podla timestamp-u a checksum.

4.3 Repundation

Pri komunikácií cez definované trust boundaries môže napadnutý klient tvrdiť že nedostal odpoveď od servera. Ochrana proti tomuto útoku je vytváranie acess a audit logov aplikácie.

4.4 Information disclosure

- Využívanie vlastnej autentifikačnej metódy je nebezpečne kvôli veľkému riziku ľudskej chyby programátora pri programovaní tohto mechanicku. Bezpečné je používať overené systému pre autentifikáciu a autorizáciu ktoré by v systéme boli ako autentifikačný server a servery v ktorých sa spracovávajú dáta aplikácie by boli iba ako resource server a autentifikovali sa voči autentifikačnému serveru.

4.5 Denial of service

- Pokiaľ ktorákoľvek časť systému spadne alebo nebude dlho odpovedať celý systém buď spadne alebo bude mať dlhé odozvy. Ideálnym riešením pre serverové časti je vytvoriť viac inštancií ideálne na rôznych fyzických serveroch a dať pred tieto inštancie loadbalancer ktorý bude rozdeľovať volania podľa vyťazenie jednotlivých serverov. čo sa týka databáze tak jedných so spôsobov ako zabezpečiť že dáta nezmiznú je replikácia pomedzi viacero inštancia databázy. No tento prístup prichádza s nutnosťou synchronizácie. Ak sa bude manipulovať s dátami je nutné zosynchroznycovať zmenu do všetkých inštancií.

4.6 Elevation of Privilege

- Pokiaľ serverové aplikácie (Server, Platobná brána) neimplementujú ochranu proti útokom typu file inclusion, XSS alebo SQL injection tak útočník môže tieto útoky zneužiť na vykonávanie príkazov na serveroch ako administrátor alebo meniť oprávnenia používateľov na OS servera.