Zadanie 1 – STRIDE

Vypracovali: Samuel Michalčík, Marek Štrba

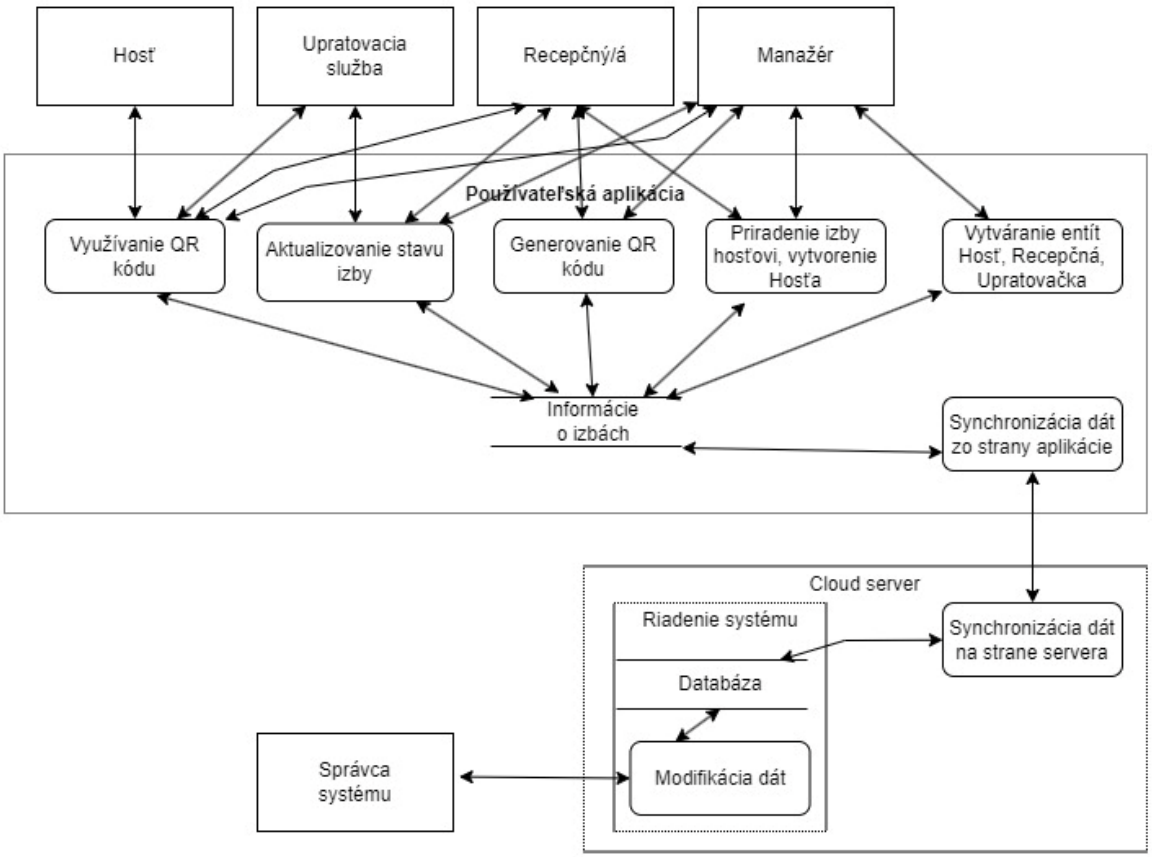
1. Architektúra systému

Naším navrhovaným systémom je aplikácia na správu hotela (hotelových služieb, izieb, evidenciu a pod.). Aplikácia obsahuje autentifikáciu a autorizáciu používateľov, ktorí disponujú rôznymi privilégiami a metódami/funkciami na obsluhu systému. Typy entít sme zoradili v poradí od najsilnejšej entity so všetkými privilégiami, po koncových používateľov – ubytovaných hostí s najviac obmedzenými privilégiami. Predpokladáme, že každá entita má prístup k dátam a metódam, ktoré ovláda aj entita hierarchicky pod nimi, teda napríklad manažér má okrem iného všetky privilégia ako recepčná, upratovačky či ubytovaný hosť, ale nemá privilégiá ako správca systému. Aplikácia funguje ako fullstack webová aplikácia s databázou bežiacou na vzdialenom serveri.

Typy entít:

- Správca systému
    - Má kontrolu nad celým systémom (backend), vie modifikovať všetky údaje
    - Vie vytvárať entity všetkých typov
- Manažér
    - Má prístup k čítaniu všetkých dát
    - Dokáže vytvárať nové entity typu recepčná, upratovačka, ubytovaný hosť, entity typu manažér modifikovať nevie
    - Spravuje všetky financie hotela (napr. navrhovanie cien za izby)
- Recepčná
    - Vie vytvoriť novú entitu ubytovaného hosťa, iné entity vytvoriť nevie
    - K voľnej izbe dokáže priradiť ubytovaného hosťa a vygenerovať mu pin/QR na odomknutie izby cez čítačku
- Upratovačka
    - Dokáže zmeniť stav izby na uprataný/neuprataný, prípadne nahlásiť poruchu
    - Súčasne má upratovačka prístup do všetkých izieb
- Ubytovaný hosť
    - V aplikácii má priradený pin/QR s prístupom výlučne do svojej izby

2. Data-flow diagram



3. Hrozby podľa STRIDE a ich mitigácia

Spoofing

Najväčšia zraniteľnosť typu „spoofing“ spočíva v prevzatí kontroly respektíve v predstieraní, že útočník je zamestnanec hotela. Útočník môže odoslať phisingový email ubytovanému hosťovi a predstierať identitu napríklad recepčnej a vyžiadať jeho QR kód, prípadne prevziať fyzickú kontrolu nad zariadením recepčnej, či upratovačky.

Zmiernenie hrozby: Hrozbu po softvérovej stránke vieme zmierniť správne implementovanou autentizáciou, prípadne zvýšiť jej nároky z hesla na odtlačok prstu či vykonať viacfaktorové overenie. Z hľadiska používania v praxi vieme hrozbu znížiť napríklad poučením personálu, aby nenechával svoje zariadenia odomknuté a rovnako aby ubytovaný hosť za žiadnych okolností neposkytoval svoj QR kód od izby.

Tampering

Metódu „tampering“ by mohol využiť správca systému a manipulovať dátami vo svoj prospech, napríklad vytvoriť nového ubytovaného hosťa s priradeným QR kódom a teoreticky sa zdarmo ubytovať, alebo poškodiť hotel znižovaním kapacity, modifikáciou stavu izieb a podobne.

Zmiernenie hrozby: Dáta prenášané medzi mobilnou aplikáciou a servermi by mali byť šifrované pomocou protokolov, ako je HTTPS, aby sa zabránilo zachyteniu a úpravám. Zároveň by sme sa mali snažiť o obfuskáciu kódu, aby sme zabránili reverznému inžinierstvu.

Repudiation

V tejto metóde zraniteľnosti môže útočník kombinovať predošlú, kde manipuloval dátami. Keďže je potencionálne technicky zdatný a chce uškodiť svôjmu zákazníkovi, logy, alebo históriu manipulácie dát obfuskuje napríklad zmenou súboru, alebo úplné zmazanie dát.

Zmiernenie hrozby: Zmierniť hrozbu tohto útoku by mohlo byť dosiahnuté napríklad dôkladnou a transparentnou logovacou aplikáciou, kde nie je možná zmena napríklad času, alebo dátumu zmeny. Riziko by sa potencionálne dalo úplne odstrániť, ak by logovacia aplikácia bola na štýl read only.

Information Disclosure

Zraniteľnosť systému z pohľadu „information disclosure“ spočíva v nedostatočnom alebo nesprávnom šifrovaní dát, napríklad emailov so vstupným QR kódom a citlivými osobnými údajmi. Problémom by mohla byť aj nedostatočná autentifikácia v podobe slabého hesla a nesprávne ukladaných prihlasovacích údajov v databáze.

Zmiernenie hrozby: Na zmiernenie hrozieb je potrebné zabezpečiť šifrovanie dát, napríklad emailovú komunikáciu, heslá v databáze ale nastaviť aj validáciu hesla (minimálne požiadavky na heslo) pri vytváraní účtu pre jednotlivé entity.

Denial of service

Útočník pri útoku DoS vie napríklad preťažiť komunikáciu aplikácie so serverom, pri procese aktualizovania dát aplikácie a servera. Vedel by takto potencionálne znefunkčniť chod aplikácie, keďže by sa dáta neukladali. Vedel by takto tiež zabrániť

novým hosťom pri vstupe do izieb, ktorí čakajú napríklad na priradenie QR kódu a recepčná nie je v tom čase prítomná.

Zmiernenie hrozby: Zmierniť hrozbu DoS by sa dalo riadnou konfiguráciou Firewall a Routerov, kde by Firewall blokoval podozrivé pohyby, napríklad podozrivé IP adresy, alebo veľa requestov rovnakej adresy.

Elevation of privilege

Zraniteľnosť môže vzniknúť pri vytváraní novej entity a nesprávnom nastavení privilégií. Rovnako by útočník mohol získať oprávnenia kvôli chýbajúcim alebo chybným kontrolám autorizácie.

Zmiernenie hrozby: Za najvhodnejší spôsob ako minimalizovať „elevation of privilege“ je využiť whitelisting a minimalizácia privilégií, teda zakázať všetko a jednotlivým entitám priradiť role a len ich výhradne povolené privilégiá. Každý prístup do aplikácie musí byť riadne autorizovaný, nekontrolovať len autentifikáciu ale aj oprávnenia pred každým jednotlivým úkonom.

4. Attack tree (popisuje spoofing, predstieranie identity za cieľom modifikácie dát)