Kućni sigurnosni sistem

Bezbednost u sistemima elektronskog poslovanja

2022/2023

I Uvod

Kućni sigurnosni sistem je sistem koji se koristi za obezbeđivanje imovine, bilo da se radi o stanu, vikendici ili kući (u daljem tekstu objekat). Uz pomoć ovog sistema, može da se prati da li se nešto nepredviđeno desilo, kao što je nestanak struje, nasilno otvaranje vrata i prozora, pomeranje elemenata u objektu itd. Sistem treba da bude lak za upotrebu i rukovanje, a sva upozorenja i obaveštenja treba da budu razumljiva i intuitivna za korisnika.

Kućni sigurnosni sistem se sastoji od velikog broja softverskih podsistema, različitih internih alata i informacionih sistema. Najznačajnije aplikacije su *Moja kuća, Admin aplikacija* i različiti *uređaji*.

Aplikacija *Moja kuća* služi za bežičnu kontrolu, monitoring i upravljanje sistemom jednog objekta. Sve promene u okviru jednog objekta generišu različiti *uređaji* kao što su: kamere, alarmi, uređaji za praćenje rasvete, uređaji za otvaranje kapija, brava itd.

Admin aplikacija se koristi za konfiguraciju sistema i očuvanje bezbednosti.

Sistem raspolaže sa velikom količinom osetljivih podataka, te predstavlja metu za različite vrste napada. Potrebno je obezbediti ceo sistem, tako da osetljivi podaci ne dođu u posed napadača.

II Admin aplikacija

Admin aplikacija je bitna za očuvanje bezbednosti čitavog sistema. Funkcionalnosti ove aplikacija se mogu podeliti u 3 dela. Jedan deo funkcionalnosti predstavlja podršku infrastrukture javnih ključeva (PKI), drugi deo služi za konfigurisanje korisnika u sistemu, a treći deo za konfiguraciju aplikacije *Moja kuća* i uređaja koji se prate.

Ovu aplikaciju može da koristi samo super admin i uz pomoć nje da izvršava sledeće funkcionalnosti:

I. **Infrastruktura javnih ključeva**
   A. Centralizovano kreiranje sertifikata

      Prilikom kreiranja sertifikata, aplikacija treba super adminu da što više olakša popunjavanje podataka, koji su potrebni za sertifikat.

      Omogućiti templejte za sertifikate, gde se templejtom definišu ekstenzije koje će ući u sertifikat, a pre svega namena sertifikata.

   B. Povlačenje sertifikata
   C. Uvid u sertifikate

      Za svaki sertifikat treba da bude prikazano da li je validan ili ne, što znači da aplikacija treba da sadrži servis za proveru da li je sertifikat validan.

   D. Distribuiranje sertifikata

      Super admin može da distribuira sertifikate, te je potrebno osmisliti korake koje će super admin izvršavati. Aplikacija treba da ga podrži prilikom tog procesa, tako da sertifikat bude bezbedno i efikasno instaliran.

U sklopu admin aplikacije potrebno je kreirati i jednu stranicu kojoj mogu svi da pristupe. Na toj stranici novi korisnici mogu da podnesu zahtev za instalaciju aplikacija i ujedno da pošalju zahtev za potpisivanje sertifikata (CSR). Nakon što korisnik podnese zahtev, super admin treba da ga odobri, generiše odgovarajuće sertifikate i distribuira ih.

II. **Korisnici sistema**
   A. Dodavanje korisnika
   B. Menjanje uloge korisnika u sistemu
   C. Brisanje korisnika iz sistema
   D. Pregled svih korisnika (pretraga, filtriranje)

III. Konfiguracija *Moje kuće* i uređaja

A. Definisanje i povezivanje uređaja sa aplikacijom *Moja kuća*

Moja kuća ima konfiguracioni fajl, koji sadrži spisak uređaja koji se prate. Super admin treba da, za svaki uređaj, koji se prati, definiše: putanju, period čitanja poruka i filter u vidu regexa.

B. Prikaz svih logova koje su generisale aplikacije

C. Pretraga logova po različitim poljima, sa mogućnošću upotrebe regularnih izraza

D. Pregled alarma

E. Kreiranje pravila za okidanje alarma

Super admin ima zadatak da prati da li je neko pokušao da izvrši napad na neku od aplikacija u sistemu.

Sve aplikacije inicijalno treba da poseduju više pravila, kao što su npr:

- Neuspešni pokušaji prijave na sistem sa istim korisničkim imenom
- Pojava loga čiji tip je ERROR
- Pojava loga u kom se nalazi IP adresa sa spiska malicioznih IP adresa
- Detekcija suviše učestalih zahteva itd..

Osim inicijalnih pravila, super admin može sam da kreira i doda novo pravilo. Pravilo za okidanje alarma se kreira kombinacijom različitih parametara. Pravilo može da se primenjuje na sve aplikacije ili samo na određenu aplikaciju, jer aplikacije neće pratiti iste tipove uređaja npr. jedan objekat može da ima kamere, a drugi da ih nema.

III Moja kuća

Moja kuća predstavlja softver koji prikuplja, normalizuje i filtrira događaje, kako bi detektovao i alarmirao korisnike o promenama. Ova aplikacija vrši svoj posao centralizovanim skupljanjem i analizom poruka, koje prima od *uređaja*. Upotrebom sistema zasnovanim na pravilima, ovaj alat korelira događaje koji se dešavaju u nekom vremenskom periodu i na osnovu njih odlučuje da li će okinuti nekakav alarm i upozoriti korisnika na nepredviđeno dešavanje.

Funkcionalnosti koje su dostupne korisnicima ove aplikacije su:

A. Pregled svih uređaja
B. Prikaz svih poruka
C. Filtriranje poruka
D. Generisanje izveštaja bitnih aktivnosti u određenom vremenskom periodu

Potrebno je podržati situacije kada korisnik poseduje više objekata koji se obezbeđuju i kada postoji više korisnika koji mogu da prate samo podskup tih objekata.

IV Uređaji

Uređaj predstavlja aplikaciju koja prati različite signale iz okruženja. Svaki uređaj, u skladu sa svojom namenom, prati određenu vrstu promena (više neuspešnih pokušaja otvaranja vrata, paljenje/gašenje svetla, promena temperature, pojava nepoznatnog objekta, isključivanje uređaja, prekid komunikacije itd). Uređaji treba da generišu poruke u standardizovanom formatu, da ih dopune bitnim informacijama i proslede aplikaciji *Moja kuća*.

Pored generisanja različitih stanja uređaja, treba kreirati i skriptu, koja će funkcionisati kao state mašina, tj. simulirati normalno stanje i stanje napada. U stanjima normalnog rada, treba da se generišu događaji koji su relevantni za kreirane alarme, ali ih ne okidaju. U stanjima napada treba da se generišu događaji koji će okinuti neki od kreiranih alarma. Potrebno je definisati više stanja za normalan rad i za napade.

V Nefunkcionalni zahtevi

1. Tehnologije

Tehnologije koje se koriste za implementaciju bilo koje celine ovog sistema su proizvoljne.

Obavezno je korišćenje Sistema za kontrolu verzija git. Kao udaljeni repozitorijum treba koristiti *Github*. Neophodno je da projekat bude u privatnom repozitorijumu, na koji će profil *bezbednost-ftn* biti dodat kao collaborator/reporter.

Komponente za alarmiranje treba da budu bazirana na ECA (Event Condition Action) pravilima, tj. da se za implementaciju koristi Rule-based sistem.

Logovi treba da se čuvaju u noSQL bazi.

Prilikom implementacije svih aplikacija potrebno je konfigurisati bezbednosne funkcije u skladu sa preporučenim, najboljim praksama.

2. Bezbednost resursa

Osetljivi podaci sa kojim aplikacija radi treba da budu obezbeđeni u skladištu, u transportu i tokom upotrebe. Identifikovati osetljive podatke i definisati i implementirati prikladne bezbednosne kontrole. Podaci, čije skladištenje se ne može izbeći, treba da budu šifrovani ili heširani, ukoliko je to prikladno.

Komunikacija između veb-čitača i servera treba da bude zaštićena sigurnom konfiguracijom HTTPS protokola. Sertifikate generisati putem admin aplikacije.

Poruke koje se razmenjuju treba da budu digitalno potpisane od strane uređaja koji ih šalju.

3. Upravljanje korisnicima

Korisnički interfejs alata treba da podrži prikladne mehanizme za autentikaciju i autorizaciju. Autorizacija podrazumeva kontrolu pristupa po RBAC modelu.

Sistem sa svim svojim *endpoint*-ima treba da ima regulisane sve rizike sa aktuelne OWASP Top 10 liste.

VI Zadaci za ocenu 10

Za ocenu 10 potrebno je uraditi jedan od dva dodatna zadatka *Penetration testing* ili *Secure deployment and disposal.*

Penetration testing

Sprovesti penetraciono testiranje veb-aplikacija i servera upotrebom bar dva alata iz grupe: *Nmap, Nikto, dirbuster, sqlmap, OWASP ZAP, Burp Suite*. Penetraciono testiranje je tehnika za testiranje sigurnosti sistema simuliranjem napada, gde se na osnovu rezultata može izvršiti procena sigurnosti sistema. Potrebno je formirati izveštaje penetracionog testa i regulisati ranjivosti. Kroz ovaj zadatak, studenti će naučiti osnove „hakovanja“, odnosno specijalizovanog testiranja uz pomoć alata, čija svrha je identifikacija ranjivosti u softverskom sistemu.

Secure deployment and disposal

Potrebno je izučiti system hardening i secure disposal procedure i najbolje prakse, i definisati protokol kako će se sistem postaviti u produkciju i kako će se bezbedno ukloniti kada dođe end-of-life sistema. Kroz ovaj zadatak, studenti će naučiti šta podrazumeva postavka sistema u produkciju, kako se obezbeđuje infrastruktura (hardware, OS, serveri) na koje softver leže, i o čemu sve treba voditi računa kada se softver vadi iz produkcije.

Kontrolna tačka 1

Za prvu kontrolnu tačku potrebno je implementirati PKI (infrastruktura javnih ključeva) tj. prvi deo funkcionalnosti admin aplikacije:

- slanje i obrada CSR,
- centralizovano kreiranje sertifikata,
- povlačenje sertifikata,
- uvid u sertifikate (pregled sertifikata),
- provera validnosti sertifikata i
- distribuiranje sertifikata.

Raspodela (distribucija) sertifikata ne mora da bude u potpunosti implementirana. Za KT1 je dovoljno da tim osmisli mehanizam kako bi ovaj zahtev realizovao.

Prva kontrolna tačka nosi 8 bodova.
Rok za poslednji commit je 3.04.2023. u 23:59h.

Kontrolna tačka 2

Za drugu kontrolnu tačku potrebno je implementirati autentifikaciju i autorizaciju, drugi deo funkcionalnosti admin aplikacije (rad sa korisnicima sistema) i omogućiti zaštitu od SQL Injection i XSS napada.

Autentifikacija

- MFA (korisničko ime/email, lozinka i PIN)
- Zaključavanje naloga nakon određenog broja neuspešnih prijava
- Polisa za lozinku (dužina lozinke - minimum 12 karaktera, veliko, malo slovo, broj, znak)
- Provera da li se postavljena lozinka nalazi na listi najčešće korišćenih lozinki
- Heširanje lozinke
- JWT token
    - Trajanje tokena
    - Verifikacija tokena - provera da je očekivani algoritam postavljen
    - Zaštita od krađe tokena (jwt + cookie)
    - Mehanizam da proglasimo token nevalidnim

Autorizacija

Potpun RBAC sa ulogama i permisijama (staviti iznad endpoint-a naziv permisije, a ne role) ili neki drugi model poput ABAC-a.

Validacija podataka

Zaštita od XSS, SQL Injection, validacija podataka i na klijentskoj i na serverskoj strani.

Rad sa korisnicima

Dodavanje korisnika i postavljanje objekata (nekretnina) koje mogu da vide, promena uloge korisnika, brisanje i pregled.

Druga kontrolna tačka nosi 14 bodova.
Rok za poslednji commit je 8.05.2023. u 23:59h.

Pitanja i odgovori

1. **Pitanje:** Da li je potrebno da Certificate Authority i RegistrationAuthority budu dve zasebne aplikacije, ili da ih objedinimo u okviru admin aplikacije.

   Odgovor: Možete sve da objedinite u okviru admin aplikacije.

2. **Pitanje:** Da li u sistemu može biti više admina, ako ima, jel svako od koristi isti rootCA za izdavanje sertifikata ili ima svoj intermediate certificate?

   Odgovor: Dovoljno je da imate jednog admina koji se prijavljuje na sistem. Napravite da u sistemu postoji 1 root i 2 intermediate sertifikata i da admin može da izabere koji od tih sertifikata će potpisati novi leaf sertifikat.

3. **Pitanje:** Da li korisniku dati opciju da izabere da mu sertifikat bude CA s obzirom da mu to nikad neće biti dozvoljeno?

   Odgovor: Ne morate korisniku da dajete opciju da odabere da mu sertifikat bude CA.

4. **Pitanje:** Da li korisnik može prilikom kreiranja zahtjeva za sertifikat da bira algoritam ključa koji će biti korišćen zajedno sa dužinom ključa, režimom rada... ili da to bude zakucano u kodu na najbolju trenutnu praksu?

   Odgovor: Možete korisniku da ponudite da bira algoritam i dužinu ključa. Ubacite par predefinisanih opcija i inicijalno označite najbolju opciju.

5. **Pitanje:** Imam pitanje vezano za templejte i ekstenzije sertifikata. Šta se podrazumjeva pod tim, da li moramo da pokrijemo sve ekstenzije kao u keystore exploreru, pošto ih ima dosta? I da li možemo napraviti par templejta i da admin može da bira između tih ili da dozvolimo adminu pravljenje templejta i čuvanje istih?

   Odgovor: Ne morate da implementirate kreiranje templejta i čuvanje istih. Dovoljno je da ponudite templejte koje npr. Keystore Explorer nudi. Omogućite adminu da sam može da dodaje/briše ekstenzije, koje su označene nakon što je izabran templejt.

6. **Pitanje:** Zanima nas gde se kreiraju javni i privatni ključ prilikom kreiranja zahteva za sertifikat (CSR), tj. da li treba da ih kreira aplikant, pa da u zahtevu prosledi samo svoj javni a privatni sačuva, ili treba da ih generiše adminska aplikacija i da na neki način privatni ključ vrati korisniku (mailom ili nekim requestom)?

   Odgovor: Najbolja praksa je da se privatni ključ generiše sa CSR na serveru gde će taj sertifikat kasnije biti i instaliran. Nekad se oslanjamo na eksterne alate koji generišu CSR, pa samim tim i par ključeva. Zato i imamo

keystore-ove (specijalni fajlovi) koji mogu bezbedno da čuvaju par ključeva. Što se naše aplikacije tiče, možete sve da generišete u okviru adminske aplikacije. Takođe, možete da ponudite i opciju da korisnik unese već kreirani CSR.

U redu je da podržite obe opcije ili samo jednu od te dve (korisnik unosi kreirani CSR i/ili CSR se generiše u okviru adminske aplikacije).

7. **Pitanje:** Ko sve ima mogućnost slanja zahtjeva, da li samo registrovani korisnici ili i neregistrovani (u ovom slučaju prilikom odobrenja CSR se i kreira korisnik)?

   Odgovor: Sami možete da odlučite da li će korisnici na početku biti registrovani ili ne.

8. **Pitanje:** Šta je najbolja praksa za čuvanje zahtjeva, u fajlu, bazi ili keystoru (mada za ovo nisam siguran da može).

   Odgovor: CSR ne sadrži nikakvu posebnu tajnu, tako da ne morate previše da brinete oko njegovog skladištenja. Kada se kreira sertifikat, CSR nema neki poseban značaj. Naravno, ne morate CSR svima da prikažete. Svako će moći da vidi šta radite, na osnovu podataka koji su uneti u sam CSR, a nema potrebe za tim.

9. **Pitanje:** Kada kreiramo sertifikat, da li se onda fizički brišu CSR fajlovi?

   Odgovor:

10. **Pitanje:** Kada se kreira keystore, da li da omogućimo adminu da ga kreira, ili da mi samo napravimo keystore po nekom defaultnom nazivu i šifri kada se pokrene aplikacija?

    Odgovor: Ne morate keystore programski da kreirate. Slobodno generišite keystore pre pokretanja aplikacije.

11. **Pitanje:** Šta treba da čuvamo u bazi admin aplikacije? Da li sertifikati treba da se čuvaju u bazi ili u java keystore-u?

    Odgovor: Studentima je prepušteno da sami osmisle šta će čuvati u bazi admin aplikacije. Same sertifikate čuvate u keystore-u, a ako vam je potrebno, neke podatke možete da čuvate i u bazi admin aplikacije.

12. **Pitanje:** Biblioteke za python koje mogu biti od koristi?

    Odgovor: certifi, pyjks, PyJWT, OpenSSL

13. **Pitanje:** Kako aplikacija treba da simulira rad uredjaja (python skripte, GUI prikazi itd???)

    Odgovor: Aplikacija simulira rad uređaja tako što generiše poruke u standardizovanom formatu.

14. **Pitanje:** Kako bi otprilike trebala da izgleda arhitektura projekta? Da li za svaki deo aplikacije (Moja kuca, admin app i uredjaji) treba da postoji poseban backend servis ili to moze da se realizuje u sklopu jedne viseslojne backend aplikacije?

    Odgovor: Studentima je prepušteno da osmisle arhitekturu sistema. Možete da idete u smeru mikroservisne arhitekture (admin i moja kuća su odvojene aplikacije) ili da admin i moja kuća budu u sklopu iste aplikacije. Uređaji treba da budu odvojena celina.

15. **Pitanje:** Koliko bi trebalo kreirati frontend aplikacija?

    Odgovor: 1-2, zavisi od toga da li odvajate admin i moja kuća aplikaciju.

16. **Pitanje:** Koje sve baze podataka treba da imamo?

    Odgovor: Za logove je obavezan noSQL, koji ujedno može i da bude jedina baza.

17. **Pitanje:** Da li je potrebna neka aplikacija koja bi pristupala zajednickom bazom podataka sa svim logovima?

    Odgovor: To treba da bude admin aplikacija.

18. **Pitanje:** Da li admin aplikacija treba da bude spoj auth aplikacije (rad sa entitetima korisnika - login, reg itd) i konfiguracionog dela?

    Odgovor: Da, ali je ok i ako auth deo odvojite u poseban servis.

19. **Pitanje:** Pravila za okidanje alarma se definisu pomocu drools?

    Odgovor: Da, ali ne morate sva da definišete pomoću drools-a.

20. **Pitanje:** Koje sve uloge korisnika postoje?

    Odgovor: Minimum je potrebno da imate admina, vlasnika i stanara, a sve ostale uloge u sistemu možete proizvoljno da izmodelujete.

21. **Pitanje:** Kako se radi sa konfiguracionim fajlovima i sta predstavljaju putanja i filter?

    Odgovor: Konfiguracioni fajl koristimo da definišemo odakle ćemo čitati poruke koje uređaji generišu, da definišemo filter i period čitanja tih poruka. Uređaji šalju poruke, npr:

    poruka1: Lampa je ugašena

    poruka2: Temperatura je ispod 20 stepeni

    Inicijalno je predviđeno da uređaji sve poruke čuvaju u nekom fajlu i da na određeni vremenski period, i u skladu sa filterom, Moja kuća preuzima i obrađuje te poruke.

Putanja u konfiguracionom fajlu u tom slučaju je putanja do tog fajla sa porukama.

Period čitanja definiše kada će Moja kuća da preuzme nove poruke koje su sačuvane u fajlu sa porukama.

Filter u vidu regexa koristimo da bismo preuzeli samo poruke koje zadovoljavaju određeni kriterijum. Uređaji sve vreme generišu različite poruke, a Moja kuća preuzima samo poruke koje prođu taj filter. Sve ostale poruke Moja kuća ne obrađuje, tj. ne prikazuje korisniku, ne ulaze u izveštaj, niti okidaju alarme.

Konfiguracioni fajl može da se ručno podesi na početku, pre pokretanja sistema.

Takođe, ovaj deo može i na drugačiji način da se implementira. Npr. poruke čuvamo u bazi i imamo mogućnost da kroz UI dodamo nove uređaje, da ih povezujemo sa objektima, da definišemo period čitanja itd. U ovom slučaju putanja neće biti zaista putanja, već identifikator uređaja, a period čitanja i filter u vidu regexa treba da zadrže svoju funkciju.

Napomena: Uređaji generišu poruke i njih prati Moja kuća, a sve aplikacije u sistemu generišu logove i njih prati admin aplikacija.

22. **Pitanje:** Da li kada se povlači sertifikat treba da se unese i razlog povlačenja?

    Odgovor: Da, omogućite da se unese i razlog povlačenja sertifikata. Možete da ponudite par najčešćih razloga i da omogućite ako treba da se unese neki drugi razlog (koji nije ponudjen).

23. **Pitanje:** Da li korisnik može da traži povlačenje sertifikata ili samo to admin radi?

    Odgovor: Nije obavezno da se dodaje na frontend za korisnika, dovoljno je da admin može da povlači sertifikate.

24. **Pitanje:** Da li će svaki korisnik za sebe zahtevati generisanje sertifikata?

    Odgovor: Da, svaki korisnik može da zahteva generisanje sertifikata.

    Napomena: Kada budete podešavali HTTPS za finalnu odbranu tada ćete najverovatnije generisati nove sertifikate i neće biti obavezno da svaki korisnik koristi svoj sertifikat prilikom prijave na sistem.

25. **Pitanje:** Da li može CSR da se umesto fajla priloži u vidu teksta u odgovarajuću u formu?

    Odgovor: Možete i tako da implementirate.

26. **Pitanje:** Koliko ekstenzija i templejta treba da podržimo u projektu?

    Odgovor: Implementirajte minimum 2 templejta i 2 ekstenzije po templejtu.

27. **Pitanje:** Da li treba da se čuva lanac sertifikata?

    Odgovor: Da, potrebno je čuvati lanac sertifikata.

28. **Pitanje:** Validnost sertifikata

    Odgovor: Potrebno je implementirati metodu koja će raditi proveru validnosti sertifikata tj. datumi, potpis, lanac sertifikata, da li je sertifikat povučen.

29. **Pitanje:** Da li ćemo morati da podešavamo korisničke sertifikate na nivou aplikacije?

    Odgovor: Sertifikate koje generišete za korisnike (koristimo njihov email) nećete morati kasnije da konfigurišete kada budete podešavali HTTPS.

30. **Pitanje:** Kako da potvrdimo identitet korisnika?

    Odgovor: Pošaljite korisniku link na e-mail. Ovaj korak vam ujedno može biti i deo procesa registracije.

    Registracija i prijava na sistem idu za KT2, ali potvrda identiteta korisnika treba da postoji da bi CA (admin) mogao da kreira sertifikat.

31. **Pitanje:** Da li moramo da implementiramo OSCP ili CRL protokol za povlačenje sertifikata?

    Odgovor: Sami možete da implementirate povlačenje sertifikata kako želite?

32. **Pitanje:** Koja je minimalna dužina lozinke?

    Odgovor: Minimalna dužina lozinke treba da bude 12 karaktera.

33. **Pitanje:** Da li možete da nam date primere lozinki koje imaju 12 karaktera, a koje korisnik ne sme da postavi?

    Odgovor:
    123456789012

    qwertyuiopasdf

    password1234

    123456789101

    1q2w3e4r5t6y

    letmeinplease

    monkey123456

    admin123456789

    sunshine12345

    welcome123456

    qwertyuiop12

    abcdefghijkl

iloveyou1234
football1234
letmein123456
monkey123456
welcome12345