



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVI SAD



Grupa 14

Teodora Kadić, PR20-2015

Igor Kuzmanović, PR31-2015

Bogdan Žugić, PR59-2015

Nevena Miletić, PR62-2015

Zadatak 28

Sigurnost i bezbednost u elektroenergetskim sistemima
- Primenjeno softversko inženjerstvo -

Novi Sad, 16.11.2018.

Sadržaj

1. OPIS REŠAVANOG PROBLEMA	3
2. TEORIJSKE OSNOVE	4
3. DIZAJN IMPLEMENTIRANOG SISTEMA	8
4. TESTIRANJE SISTEMA	11

1. OPIS REŠAVANOG PROBLEMA

Cilj implementacije je razvoj servisa za upravljanje bazom podataka koja sadrži poverljive podatke kriptovane tajnim ključem servisa primenom AES simetričnog algoritma. U ovoj implementaciji korišćeno je blokovsko šifrovanje AES algoritma sa ključem dužine 256 bita. Baza podataka (.txt fajl) sadrži događaje generisane od strane validnih korisnika. Spisak validnih događaja je unapred definisan i čuva se u okviru resursnog (.resx) fajla. U bazi podataka se uz odgovarajuću poruku skladište informacije o jedinstvenom identifikatoru entiteta, jedinstvenom identifikatoru klijenta koji generiše događaj i Timestamp kad je događaj generisan.

Klijent i servis se obostrano autentifikuju posredstvom sertifikata (ChainTrust) nakon čega servis razmenjuje tajni ključ, enkriptovan RSA algoritmom (koristeći javni ključ klijentovog sertifikata), sa klijentom tako da samo taj klijent može da ga dekriptuje. Koristeći ovaj tajni ključ svaki klijent može interno da pristupi bazi podataka i dekriptuje je.

Dodatno, servis omogućava klijentima sledeće akcije nad bazom podataka:

- Update (ažuriranje postojećeg sadržaja)
- Add (dodavanje novog entiteta)
- Delete (uklanjanje postojećeg entiteta),

u zavisnosti od toga da li klijent ima permisiju za izvršavanjem svake od pomenutih akcija.

Sve pokušaje uspešnog i neuspešnog pristupa bazi servis loguje u custom Windows Event Log-u. Radi detektovanja Denial-of-Service napada omogućeno je konfigurisati broj neuspešnih pokušaja izmene ili brisanja nad istim entitetom u određenom periodu. Alarm se šalje Intrusion Detection System-u (IDS) nakon prelaska ove granice. IDS generiše i ispisuje alarm na konzoli. Komunikacija između servisa i IDS komponente se ostvaruje putem Windows autentifikacionog protokola.

2. TEORIJSKE OSNOVE

AUTENTIFIKACIJA

Autentifikacija predstavlja bezbednosni mehanizam kojim se obezbeđuje validacija identiteta u okviru sistema.

- **Sertifikat X.509** (.cer) predstavlja digitalni identitet korisnika izdat od strane sertifikacionih tela (certification authority, CA). Sertifikat sadrži podatke o vlasniku sertifikata, period važenja sertifikata i informacije o izdavaocu sertifikata. U sertifikat se ugrađuje javni ključ korisnika (uz identifikator algoritma primenjenog za generisanje ključa, npr. RSA).. Svaki sertifikat je digitalno potpisan od strane sertifikacionog tela koje ga izdaje čime se potvrđuje da sertifikat zaista pripada podnosiocu zahteva. Na ovaj način je takođe moguće detektovati izmene u okviru samog sertifikata jer digitalni potpis obezbeđuje integritet podataka.

- **PKCS#12** (.pfx) predstavlja zaštićenu arhivu koja se najčešće sastoji od X.509 sertifikata (.cer) i odgovarajućeg privatnog ključa (.pvk).

- **Chain of Trust** predstavlja listu uređenih sertifikata koji počinje sa korisničkim sertifikatom. Svaki naredni sertifikat u listi predstavlja izdavača prethodnog sertifikata. Poslednji sertifikat u listi je sertifikat glavnog sertifikacionog tela, odnosno Root CA. Jedan od načina provere validnosti korisničkog sertifikata je prolazak kroz ovu listu sertifikata i utvrđivanje da li su svi sertifikati validno izdati dok se ne dođe do glavnog sertifikacionog tela.

- **Windows Authentication** predstavlja autentifikaciju korisnika korišćenjem ugrađenih Windows User-a. Podržava protokole NTLM i Kerberos.

NTLM (NT Lan Manager) je autentifikacioni protokol zasnovan na challenge-response autentifikacionoj šemi, čime je omogućena autentifikacija bez slanja poverljivih podataka (šifre). Problem kod ovakvih protokola je činjenica da servis mora da zna originalnu šifru svakog klijenta kako bi mogao da validira pristigli odgovor. Dodatno, izostaje verifikacija servisnog identiteta od strane klijenta, odnosno protokol ne omogućuje obostranu autentifikaciju.

Kerberos je dvosmerni autentifikacioni protokol koji se zasniva na trećoj strani od poverenja i razmeni ticketa u cilju uspostavljanja bezbedne obostrane autentifikacije učesnika u komunikaciji bez razmene šifri. Kerberos je namenjen za domenska okruženja gde uslugu treće strane od poverenja ima posebno konfigurisani server, tzv. domen kontroler (DC). DC predstavlja autoritet na nivou celokupnog domena kome pripada skup računara i korisničkih naloga.

AUTORIZACIJA

Autorizacija je bezbednosni mehanizam kojim se proverava pravo korisnika za izvršenje određene funkcionalnosti sistema.

- **Role-Based-Access-Control (RBAC)** je mehanizam koji omogućava kontrolu pristupa sistemu autorizovanim korisnicima. RBAC se interpretira i kao role-based security. Svakom korisniku u sistemu se dodeljuju određena prava/uloge (roles) koje sadrže određene permisije. Permisije omogućavaju pristup različitim servisima/delovima sistema.

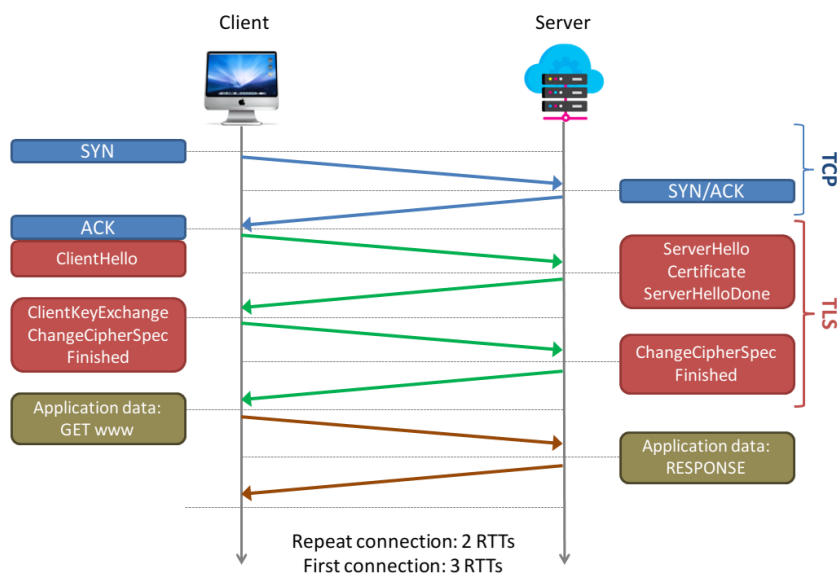
POVERLJIVOST PODATAKA

- **Bezbedan TCP kanal (TLS over TCP)** se postiže primenom TLS (Transport Layer Security) Handshake protokola (slika 1), koji je odgovoran za autentifikaciju i razmenu ključeva neophodnih za sigurnu komunikacionu sesiju.

Handshake protokol upravlja koracima:

- Postavljanje seta kriptografskih algoritama u cilju da obe strane koriste ista podešavanja
- Autentifikacija servera i po potrebi klijenta
- Razmena ključeva kad je obezbeđena sigurna sesija

Server i klijent se autentifikuju uz pomoć javnih i privatnih ključeva, najčešće tako što pošiljalac poruku enkriptuje svojim privatnim ključem a primalac uspeva da je dekriptuje javnim ključem pošiljaoca i time se uveri da je poruka legitimna.

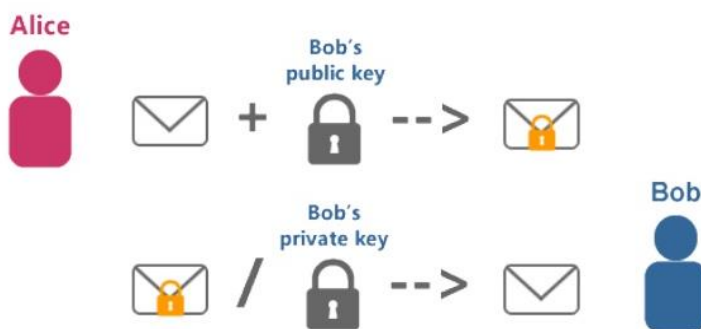


Slika 1. TCP i TLC Handshake protokol

- **Rivest-Shamir-Adleman (RSA)** je kriptografski algoritam (slika 2) koji asimetrično vrši enkripciju. To znači da ovaj algoritam zahteva dva ključa, jedan za enkriptovanje, i jedan za dekriptovanje. Dva ključa su najčešće poznati kao "javni" i "tajni".

Podaci enkriptovani javnim ključem mogu biti dekriptovani samo njegovim korespondentnim tajnim ključem.

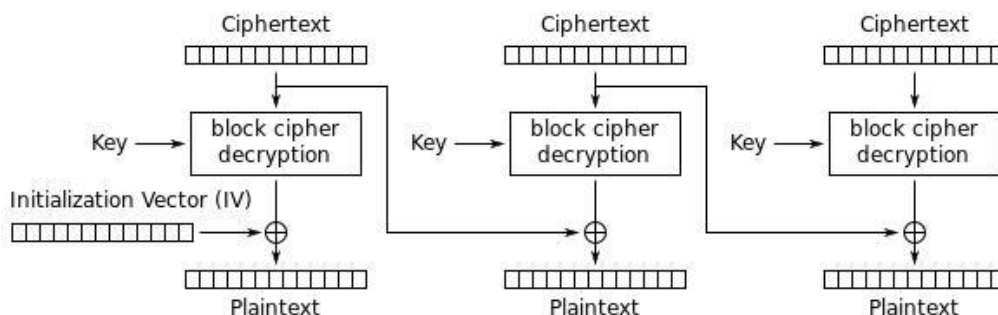
Algoritam uzima javni ključ od strane kojoj se šalje i njime enkriptuje poruku čime obezbeđuje da će samo ta strana moći da poruku i dešifruje.



Slika 2. Primena RSA algoritma

- **Advanced Encryption Standard (AES) - Cipher Block Chaining (CBC)** je algoritam koji koristi simetričan ključ, gde se enkripcija i dekripcija na klijentskoj i servisnoj strani vrši uz pomoć istog skrivenog ključa. Režim rada AES-a je CBC, koji dodatno usložnjava podatke koji se enkriptuju.

CBC (slika 3) radi tako što podatke podeli na predefinisane blokove memorije koje enkriptuje korak po korak. Prvi blok se enkriptuje uz pomoć prosleđenog ključa i nasumice generisanog vektora inicijalizacije (IV) dok se svaki sledeći blok enkriptuje uz pomoć ključa i podataka dobijenih na osnovu prethodnog bloka.



Slika 3. AES CBC algoritam

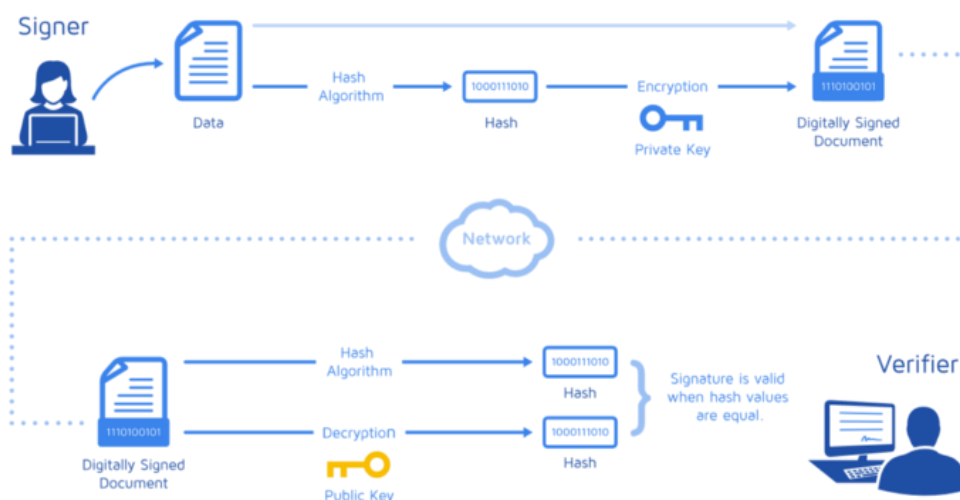
- **SecureString klasa .NET-a** reprezentuje tekst koji omogućava dodatnu meru bezbednosti, time što izbegava skladištenje osetljivih podataka (npr. šifra) u radnoj memoriji. Za običan string, koji se ne može ručno ubaciti u red za garbage collection (tj. kada se fizički briše iz memorije), ne može se predvideti kada će osetljivi podaci biti izbrisani, što predstavlja rizik pristupa osetljivim informacijama.

SecureString, međutim, omogućava ručno brisanje pomoću sopstvene *Dispose* metode, pored standardnog brisanja tokom garbage collection-a. Mana SecureString-a je činjenica da se tekstu koji je sadržan ne može pristupiti direktno, već se mora preneti u radnu memoriju radi čitanja ili konvertovanja u običan string pri čemu se gubi prethodno dobijena poverljivost informacija.

INTEGRITET PODATAKA

- **Secure Hashing Algorithm (SHA)** je hashing algoritam koji je često korišćen kod bezbednosti vezane sa komunikacione protokole kao što su TLS, SSL, SSH, itd... Njegova najveća upotreba je provera integriteta poruke u komunikaciji između klijenta i servisa (slika 4).

Klijent potpisuje svoje poruke svojim privatnim ključem, a servis nakon primanja iste poruke koristi klijentov javni ključ kako bi uporedio integritet poruke s obzirom da će i najmanja izmena originalne poruke dovesti do potpuno drugačijeg rezultata hash funkcije.



Slika 4. Primena digitalnog potpisa

3. DIZAJN IMPLEMENTIRANOG SISTEMA

Arhitektura sistema se zasniva na 3 komponente koje međusobno komuniciraju putem WCF tehnologije na transportnom nivou koristeći TCP/IP model uz korišćenje bezbednog komunikacionog kanala TLS (Transport Layer Security) i različitih tipova autentifikacije.



Slika 5. Arhitektura sistema

Komunikacija između WCFClient-a i WCFService-a podrazumeva dva umrežena računara koji razmenjuju binarne poruke kroz bezbedan kanal. Obe komponente koriste sertifikate kako bi se autentifikovale jedna drugoj i utvrdile mere korišćene za bezbednu komunikaciju. Ispravnost sertifikata se određuje metodom *Chain of Trust* koja je prethodno objašnjena. Kako bi se ova komunikacija omogućila potrebno je prethodno konfigurisati računare kao i same komponente:

1. Potrebno je instalirati već izgenerisan RootCA.cer sertifikat u *Trusted Root Certification Authorities* odeljku na klijentskoj i servisnoj mašini. Ovaj sertifikat će predstavljati glavno sertifikaciono telo za sve validne sertifikate.
2. Sa servisne strane potrebno je instalirati sertifikat WCFService.pfx u *Personal* odeljku kako bi WCFService mogao da pristupi njegovom privatnom ključu i iskoristi sertifikat da se autentifikuje klijentima. U konfiguracionom fajlu WCFService-a, u okviru *serviceCertificate* sekcije naznačiti koji će se sertifikat koristiti.
3. Sa klijentske strane potrebno je instalirati jedan od već kreiranih sertifikata, u *Personal* odeljku. Svaki sertifikat predstavlja različitog korisnika i sadrži različite uloge koje u sistemu sadrže različite nivoe pristupa. Nakon instaliranja sertifikata potrebno je u konfiguracionom fajlu WCFClient-a, u okviru *clientCertificate* sekcije naznačiti koji će se sertifikat koristiti.

Komunikacija između WCFService-a i IDSService-a se takođe zasniva na bezbednom kanalu ali koristi Windows autentifikaciju. Obe komponente treba da budu pokrenute od strane validnih Windows usera, a takođe je moguće ručno podesiti dodatnu autorizaciju koja će biti objašnjena u nastavku. Za potrebe ove komunikacije je potrebna konfiguracija računara i samih komponenti:

1. Potrebno je na računaru napraviti dva proizvoljna korisnika koji će predstavljati korisnike WCFService-a i IDSService-a, a zatim u okviru konfiguracionih fajlova oba servisa podesiti sledeće parametre: *WCFServiceUser* je korisnik koji pokreće WCFService, *IDSServiceUser* je korisnik koji pokreće IDSService i *IDSServiceClient* je očekivani identitet korisnika koji poziva *Alarm* metodu IDSService-a čime se omogućuje autorizacija.

Ostala podešavanja komponenti se mogu naći u okviru njihovih konfiguracionih fajlova. Potrebno je naznačiti da je za kreiranje custom Windows Event Log-a potrebno pokrenuti WCFService kao Administrator računara na kojem se komponenta nalazi ili ga ručno kreirati upotrebom PowerShell-a u Administratorskom režimu.

WCFSERVICE je komponenta koja nam omogućava autentifikaciju, autorizaciju i auditing korisnika koji se služe bazom podataka kao i detekciju Denial-of-Service napada i alarmiranje. Operacije WCFSERVICE klase implementiraju IWCFService interfejs.

public interface IWCFService:

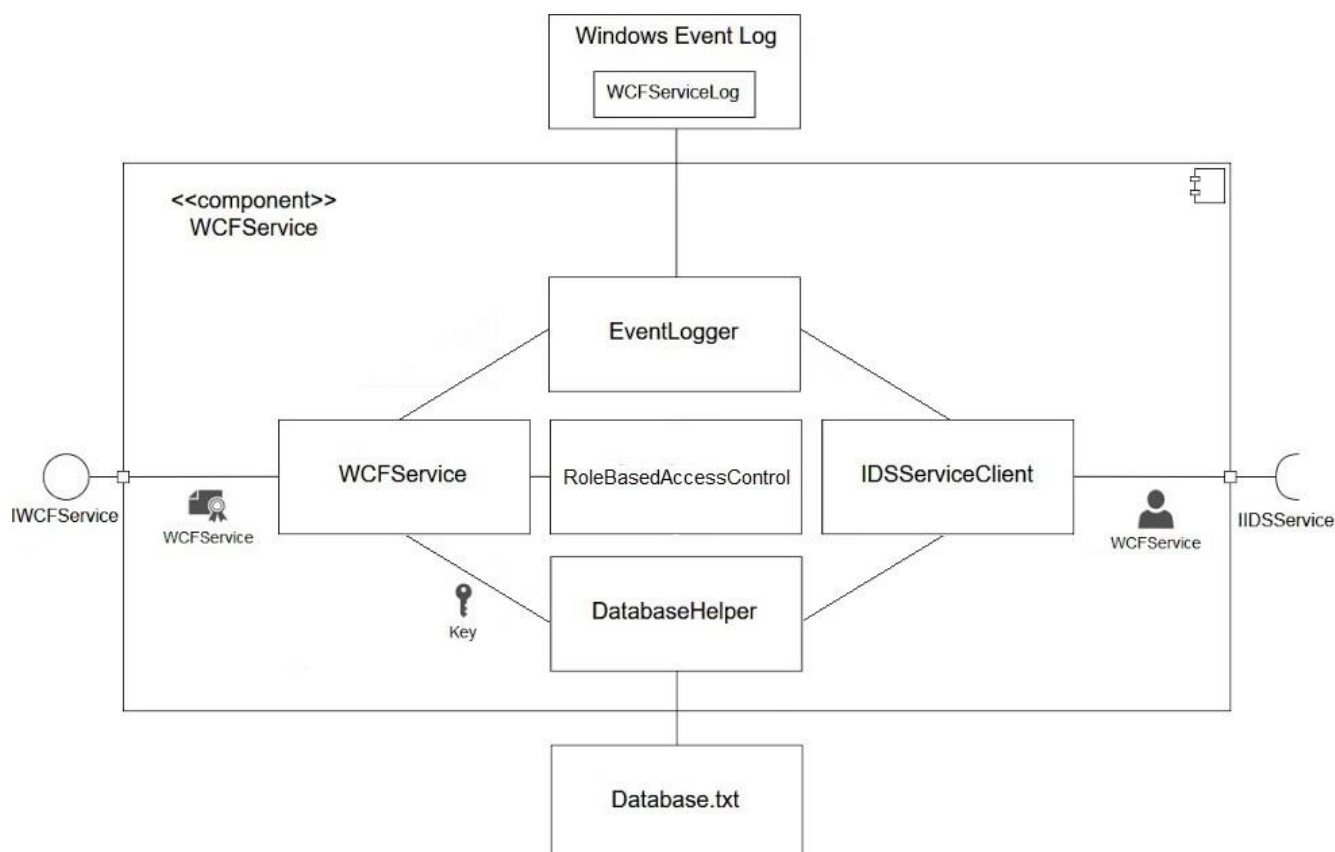
byte[] CheckIn()

bool Add(**string** content)

bool Update(**int** entryID, **string** content)

bool Delete(**int** entryID)

byte[] ReadFile()



Slika 6. Dizajn WCFSERVICE komponente

Pri pokretanju programa WCFSERVICE potrebno je uneti ključ proizvoljne dužine koji će služiti za enkripciju i dekripciju baze podataka korišćenjem AES algoritma u CBC modu. Pri svakom pozivu *Encrypt* i *Decrypt* funkcije ključ se hash-uje SHA-256 algoritmom kako bi se dobila odgovarajuća veličina ključa.

Prilikom konektovanja klijenta na WCFSERVICE metodom *CheckIn* servis autentifikuje klijenta. Nakon uspešne provere klijentskih kredencijala, servis klijentu šalje svoj tajni ključ kojim je enkriptovana baza podataka s tim da ga prethodno enkriptuje 2048-bitnim javnim ključem klijenta upotrebom RSA algoritma. Klijent, nakon dekriptovanja ključa za pristup bazi podataka svojim privatnim 2048-bitnim ključem, u zavisnosti od svoje uloge i permisija te uloge, ima pristup određenim metodama WCFSERVICE-a. Svi autentifikovani klijenti su po prijemu ključa autorizovani da pristupe bazi podataka, tj. na zahtev će dobiti enkriptovanu bazu od servisa koju će prethodno dobijenim ključem moći da dekriptuju i deserijalizuju.

DatabaseHelper služi za manipulisanje bazom podataka koja je u .txt formatu. Klasa podržava sve metode WCFService-a uz dodatak internih metoda koje se koriste za pomoć pri generisanju slobodnog ID-a za nove upise, upis i čitanje same baze podataka.

EventLogger omogućava vođenje istorijata aktivnosti u custom Windows Event Log-u. Evidentirajući svaku uspešnu autentifikaciju, uspešnu ili neuspešnu autorizaciju (pokušaj modifikovanja baze podataka) i svaki zahtev za iščitavanje baze podataka. U sebi ima sistem za detektovanje malicioznih pristupa nad svakim entitetom i *Alarm* metodom obaveštava IDSService preko IDSServiceClient-a nad kojim je entitetom to detektovano. Detektovanje malicioznih pristupa se svodi na čuvanje poslednjih N (konfigurabilan broj pokušaja) neuspešnih pristupa za svaki entitet i računanje da li razlika u vremenu između najnovijeg i najstarijeg neuspešnog pristupa prelazi granicu M (konfigurabilan vremenski period u sekundama).

IDSServiceClient služi kao proxy za konekciju sa IDSService-om i sadrži sva potrebna podešavanja i kredencijale.

RoleBasedAccessSecurity je zadužen za autorizaciju klijenata. On sadrži veze između Role-a i Permission-a, dok su same Role i Permission-i sadržani u okviru odgovarajućih enumeracija. Provera se vrši time što se u okviru klijentovog sertifikata ispituje atribut *OrganizationalUnits* (OU). Sve pronađene organizacione jedinice se pretvaraju u Role. Pronađene Role se redom ispituju sve dok se ne pronađe ona koja sadrži traženi Permission.

WCFClient predstavlja komponentu sistema koja se može distribuirati na lokalne računare i predstavlja korisnika WCFService-a. Njen cilj je da kroz konzolu omogući korisniku da koristi WCFService obavljanjem akcija kao što su dodavanje, brisanje, ažuriranje event-ova i čitanje baze. Event-ovi se nalaze u okviru resursnog fajla i imaju format sličan statusnim porukama HTTP odgovora. Pri pokretanju programa, automatski se poziva metoda *CheckIn* gde se od WCFService-a dobavlja privatan ključ za dešifrovanje baze podataka.



WCFServiceClient je deo komponente koji služi kao proxy za komunikaciju između korisnika i WCFService-a. Ona sadrži sva potrebna podešavanja i kredencijale klijenta potrebnih za korišćenje servisa i implementira interfejs preko kojeg korisnik može da vrši neku od prethodno pomenutih akcija.


IDSService služi za obaveštavanje vlasnika WCFService-a o mogućim detekcijama Denial-of-Service napada na sistem kako bi vlasnik u što kraćem roku mogao da adekvatno odreaguje i odbrani se od napada. Metod *Alarm* je zaštićen od neautorizovane upotrebe i očekivani identitet korisnika se može podesiti u okviru konfiguracionog fajla.

Helpers je deljena biblioteka koja sadrži pomoćne klase za olakšano rukovanje stringovima, RSA i AES enkripciju, rad sa resursnim i konfiguracionim fajlovima i rad sa sertifikatima i kredencijalima.



4. TESTIRANJE SISTEMA


Uspešan Add: Korišćen WCFClient ili WCFSystemAdministrator sertifikat koji kao deo organizacione jedinice sadrže rolu *Client*, a samim tim i permisiju *Add*.

 WCFClient.exe - Shortcut Event type: OK User: user1	 WCFClient.exe - Shortcut Add successful Press any key to continue...
---	--



 WCFClient.exe - Shortcut
[11/14/2018 8:12:51 PM][1][9B32979E935B2EBA4B1986066CE1CCD9][user1 - 200 - OK]
Press any key to continue...
_


Uspešan Update: Korišćen WCFModerator ili WCFSystemAdministrator sertifikat koji kao deo organizacione jedinice sadrže rolu *Moderator*, a samim tim i permisiju *Update*.

 WCFClient.exe - Shortcut Entry ID: 1 Event type: Unauthorized User: userMod_	 WCFClient.exe - Shortcut Update successful Press any key to continue... _
---	--

 WCFClient.exe - Shortcut
[11/14/2018 8:31:56 PM][1][9C123D1E08A4D1BA4A22AF968BFC109F][userMod - 401 - Unauthorized]
Press any key to continue...
_



Uspešan Delete: Korišćen WCFAdministrator ili WCFSystemAdministrator sertifikat koji kao deo organizacione jedinice sadrže rolu *Administrator*, a samim tim i permisiju *Delete*.

 WCFClient.exe - Shortcut Entry ID: 1_	 WCFClient.exe - Shortcut Delete successful Press any key to continue... _
--	--



 WCFClient.exe - Shortcut
Entry list is empty

Press any key to continue...



Neuspešan Add: Korišćen WCFModerator ili WCFAdministrator, čije role ne obuhvataju permisiju *Add*.

 WCFClient.exe - Shortcut Event type: Created User: someuser	 WCFClient.exe - Shortcut [Add] ERROR = Unauthorized Press any key to continue... _
---	---

Neuspešan Update: Korišćen WCFClient ili WCFAdministrator, čije role ne obuhvataju permisiju *Update*.

 WCFClient.exe - Shortcut Entry ID: 1 Event type: Found User: user2	 WCFClient.exe - Shortcut [Update] ERROR = Unauthorized Press any key to continue... _
---	--

Neuspešan Delete: Korišćen WCFClient ili WCFModerator, čije role ne obuhvataju permisiju *Delete*.

 WCFClient.exe - Shortcut Entry ID: 1_	 WCFClient.exe - Shortcut [Delete] ERROR = Unauthorized Press any key to continue... _
--	--

Nespešna autentifikacija klijenta: Korišćenjem sertifikata WCFBadClient sa strane klijenta, koji je potpisan od strane samog sebe, WCFService ustanovljuje da je sertifikat nevalidan i odbija konekciju.

```
WCFClient.exe - Shortcut
[CheckIn] ERROR = The socket connection was aborted. This could be caused
by an error processing your message or a receive timeout being exceeded
by the remote host, or an underlying network resource issue. Local socket
timeout was '00:00:59.9870016'.
Unable to retrieve the private key from the service
Press any key to exit...
```

Neuspešna autentifikacija servisa: Korišćenjem sertifikata WCFBadService sa strane servisa, koji je potpisan od strane samog sebe, WCFClient, pri pokušaju poziva metode *CheckIn* ustanovljuje da servis nema validan sertifikat i ne nastavlja sa radom.

```
WCFClient.exe - Shortcut
[CheckIn] ERROR = The X.509 certificate CN=WCFBadService chain building failed. The cert
ificate that was used has a trust chain that cannot be verified. Replace the certificate
or change the certificateValidationMode. A certificate chain processed, but terminated
in a root certificate which is not trusted by the trust provider.
Unable to retrieve the private key from the service
Press any key to exit...
```

Izazivanje alarma: Cilj ovog testa je simulacija dva kompromitovana klijenta koji pokušavaju da izvedu Distributed Denial-of-Service (D-DOS) napad. Iskorišćena su dva klijenta koja koriste sertifikat WCFClient koji ima samo permisiju *Add*. Istovremeno, oba klijenta 10 puta na svakih 950ms šalju zahteve. Jedan šalje za *Update*, a drugi za *Delete* nad istim entitetom. WCFService je podešen tako da alarmira IDS ukoliko se nad istim entitetom u roku od 20 sekundi detektuje 10 neuspelih pokušaja modifikacije.

```
WCFClient.exe - Shor...  WCFClient.exe - Shortcut
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
[Update] ERROR = Unauthorized [Delete] ERROR = Unauthorized
Press any key to continue... Press any key to continue...

IDSService.exe - Shortcut
Service is ready
[11/14/2018 10:24:23 PM][WCFService] Entry 1 has too many failed modification attempts
```