



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVI SAD



Grupa 18

Danilo Milošević E3112/2014

Obren Milošević PR41/2015

Bosiljka Bratić PR70/2015

Nebojša Knežević PR104/2015

Zadatak 13

Sigurnost i bezbednost u elektroenergetskim sistemima

- Primenjeno softversko inženjerstvo -

Novi Sad, 13.11.2018.

Sadržaj

1. OPIS REŠAVANOG PROBLEMA.....	3
2. TEORIJSKE OSNOVE.....	4
3. DIZAJN IMPLEMENTIRANOG SISTEMA.....	5
4. TESTIRANJE SISTEMA	6

1. OPIS REŠAVANOG PROBLEMA

Zadatak projekta je pravljenje *Certificate Manager* komponente koja služi za upravljanje sertifikatima. Od nje klijenti zahtjevaju generisanje sertifikata koje oni koriste za međusobnu komunikaciju. Druga uloga *CMS-a* je da u slučaju kompromitovanja nekog izdatog sertifikata, isti povuče, a zatim generiše novi. Takođe, postoji i *Backup Server* na koji se repliciraju svi podaci o izdatim i povučenim sertifikatima.

Komunikacija između *Certificate Manager-a* i klijenata, kao i *Certificate Manager-a* i *Backup Server-a* se ostvaruje putem *Windows Authentication* protokola.

Sve prethodno navedene akcije se evidentiraju u okviru *CMS Windows Event Log-a*.

Zadatak drugog dijela projekta je razvijanje *WCF* klijent-servis modela takvog da se učesnici u komunikaciji međusobno autentifikuju koristeći sertifikate genereisane od strane *CMS* komponente. Obostrana autentifikacija ove dvije komponente se vrši *Chain Trust* validacijom.

Nakon uspješne autentifikacije, klijenti se javljaju servisu na random određeni period u interval od 1-10 sekundi, a servis upisuje u tekstualni fajl poruku u sledećem format:

<ID>;<TimeStamp>;<CommonName>

gdje je ID redni broj upisa u fajl, TimeStamp je vrijeme poziva metode od strane klijenta, a CommonName je atribut sertifikata koji predstavlja ime korisnika.

Server autentifikuje klijenta tako što provjerava da li on pripada jednoj od četiri grupe: *RegionEast*, *RegionWest*, *RegionSouth* i *RegionNorth*.

Dodatno, servis treba da vodi evidenciju o svim klijentskim procesima u okviru *Application Windows Event Log-a*, i to poruku da je uspostavljena nova konekcija ili da je konekcija prekinuta.

2. TEORIJSKE OSNOVE

Bezbednosni mehanizmi koji su korišćeni u projektu su autentifikacija, autorizacija, vođenje istorijata aktivnosti (*auditing*).

Autentifikacija je proces u okviru koga korisnik ili izvor informacija dokazuju da su to za šta se predstavljaju - drugim riječima, proces utvrđivanja identiteta korisnika koji pokušava da pristupi sistemu.

Autorizacija predstavlja bezbednosni mehanizam kojim se proverava pravo korisnika za izvršenje određene funkcionalnosti servisa.

Auditing (vođenje istorijata aktivnosti) je oblik bezbednosnog mehanizma gdje se na osnovu bilježenog toka izvršavanja programa može ustanoviti u kom momentu je nastao problem u sistemu.

Za autentifikaciju se koriste dva autentifikaciona protokola: *Windows Authentication Protocol* koji se zasniva na NTLM-u i *Autentifikacija putem sertifikata sa Chain Trust validacijom*.

NTLM (NT Lan Manager) je autentifikacioni protokol zasnovan na challenge-response autentifikacionoj šemi, čime je omogućena autentifikacija bez slanja povjerljivih podataka (šifre). Iako challenge-response spada u jake autentifikacione šeme jer nema razmjene povjerljivih podataka, Problem ovakvih protokola je činjenica da servis mora da zna originalnu šifru svakog klijenta kako bi mogao da validira pristigli response. Dodatno, u ovako definisanom autentifikacionom protokolu izostaje verifikacija servisnog identiteta od strane klijenta, odnosno ovakav protokol ne omogućuje obostranu autentifikaciju.

Autentifikacija putem sertifikata podrazumijeva provjeru klijentskog sertifikata pri zahtjevu za uspostavu konekcije na osnovu *Chain Trust* validacije.

Chain Trust je način validacije sertifikata gdje se sertifikati kreiraju u hijerarhiji pri čemu je svaki sertifikat povezan sa izdavaocem sertifikata. Izdavaoc sertifikata je dalje povezan sa izdavaocem svog sertifikata. Ovaj proces se ponavlja sve dok se ne dođe do korijenskog sertifikata koji je potpisan od strane samog sebe.

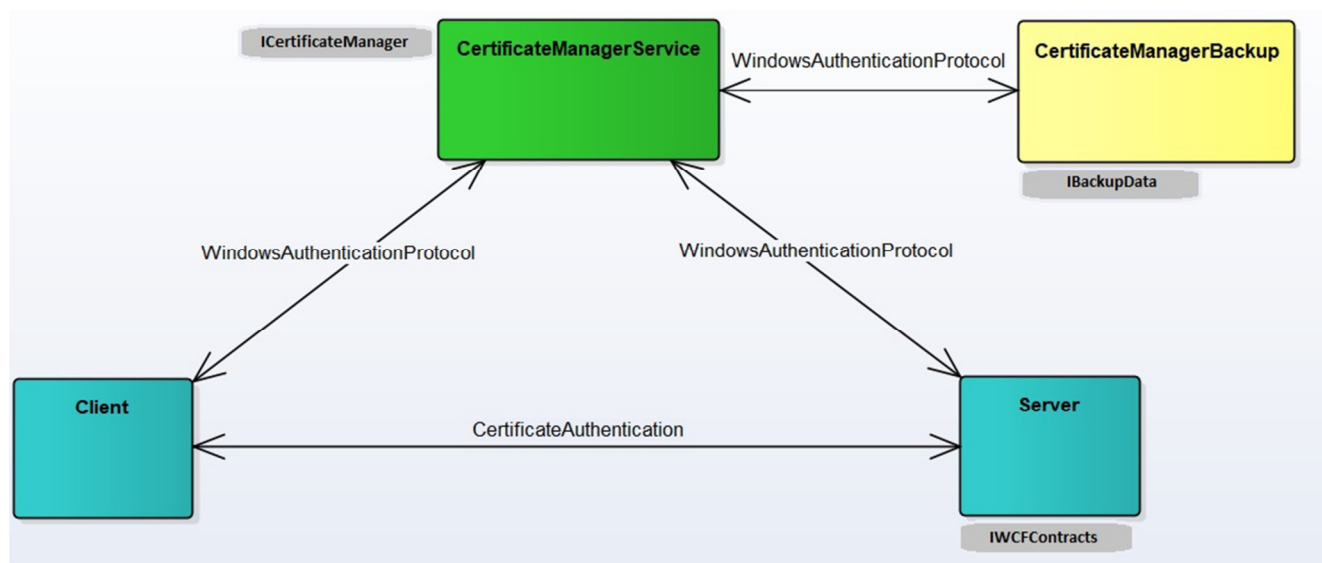
Prvi korak u autentifikaciji sertifikatima je podešavanje *binding-a* tako da podrži autentifikaciju uz pomoć sertifikata. Ovim je definisan tip kredencijala koji se očekuje od drugog učesnika u komunikaciji. Zatim je potrebno da svaki učesnik u komunikaciju podesi svoj sertifikat kojim se predstavlja drugim učesnicima u komunikaciji.

Kao dio obostrane autentifikacije, neophodno je podesiti identitet servisa koji klijent očekuje prilikom uspostavljanja komunikacije. Za to je potrebno proslediti *EndpointIdentity* prilikom uspostavljanja komunikacije.

3. DIZAJN IMPLEMENTIRANOG SISTEMA

Implementacija rješenja se sastoji od sledećih komponenti:

- CertificateManagerService
- CertificateManagerBackup
- Client
- Server
- Contract



Slika 1. – Komponente sistema

Na slici su prikazane komponente sistema zajedno sa interfejsima koje one implementiraju i čije servise izlažu. Takođe, na slici su prikazani i načini komunikacije između svih komponenti.

CertificateManagerService otvara *host* preko koga klijenti zahtjevaju generisanje ili povlačenje sertifikata, putem *ICertificateManager* interfejsa. Za svrhe bekapovanja podataka CMS kreira kanal i šalje podatke koji se repliciraju na *CertificateManagerBackup*. Inicijalno CMS kreira korijenski sertifikat koji je *self-signed* ukoliko već ne postoji, da bi preko njega generisao nove sertifikate klijentima. Informacije o uspjehnosti izvršenih akcija generisanja i povlačenja sertifikata bilježe se u CMS *Windows Event Log*. *CertificateManager* čuva listu svih nevalidnih i kompromitovanih sertifikata koji su povučeni unutar tekstualnog fajla.

CertificateManagerBackup je komponenta koja omogućuje repliciranje podataka u tekstualnu datoteku i ne sadrži funkcionalnosti *CertificateManagerService-a*. Otvara *host* preko interfejsa *IBackupData* i koristi *Windows authentication protocol* za komunikaciju.

Server je komponenta koja se ponaša kao klijent u odnosu na CMS, s kojim komunicira putem *Windows authentication protocol-a*, i putem te komunikacije šalje zahtjev za generisanje, odnosno povlačenje sertifikata. Za potrebe komunikacije sa klijentom, server otvara *host* preko koga ga klijent može pingovati. Klijent i server se autentifikuju preko sertifikata sa *Chain Trust* validacijom. Ova komponenta omogućava evidentiranje svih povezanih klijenata i upisivanje dva tipa događaja unutar *Application Windows Event Log-a*:

- 1) Da je konekcija uspostavljena
- 2) Da je konekcija prekinuta

Pošto klijent periodično javlja serveru to se zapisuje u tekstualnu datoteku u već navedenom formatu. Klijent i server imaju opciju dodavanja prava pristupa svojim nalogima.

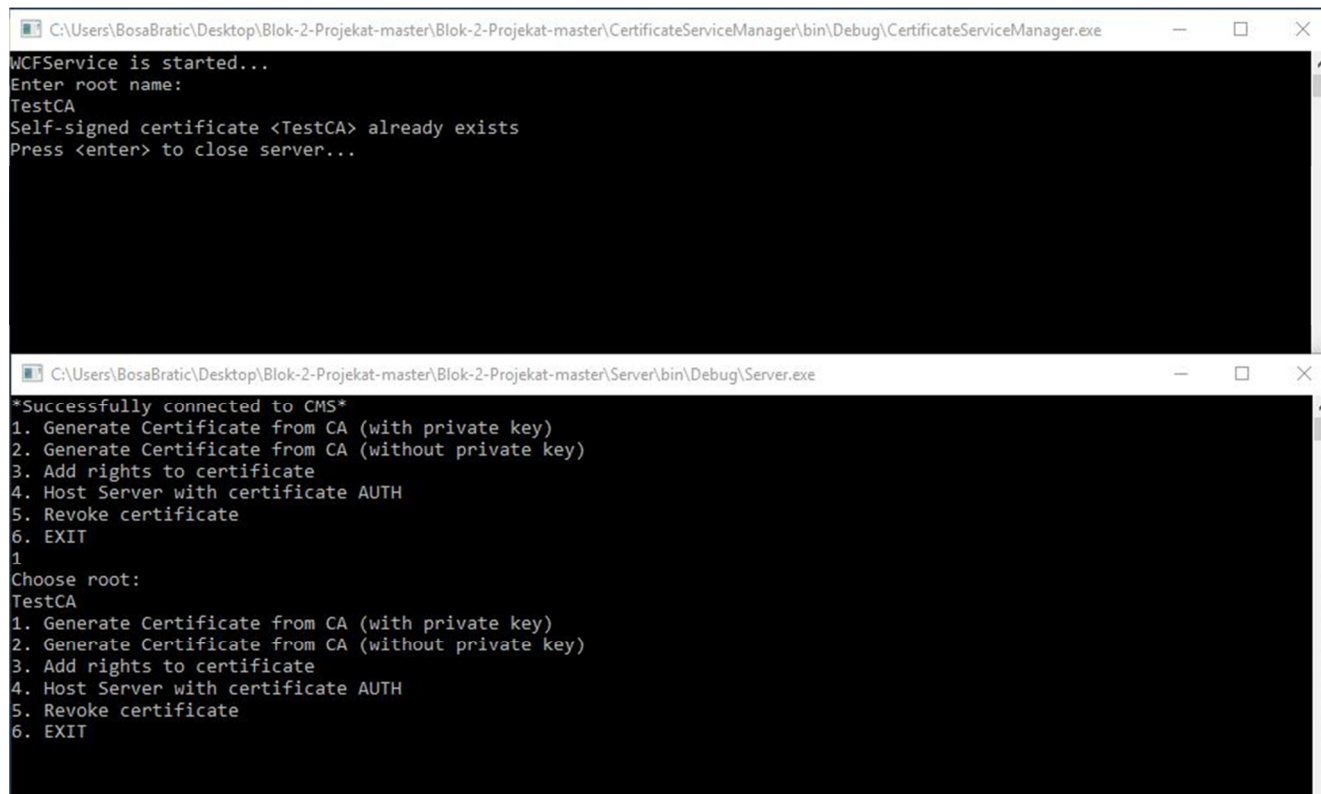
Client je komponenta koja traži od CMS-a sertifikat koji će dalje koristiti za komunikaciju sa serverskom komponentom tako što svoj sertifikat stavi na komunikacioni kanal prema serveru da bi on mogao da bi on mogao da izvrši autorizaciju i autentifikaciju. Klijent periodično pinguje servera da bi on znao da je taj klijent i dalje aktivan.

Contract je biblioteka sa interfejsima i pomoćnim klasama. Svi interfejsi koji se koriste za *WCF Duplex* komunikaciju nalaze se u ovoj biblioteci. Od pomoćnih klasa sadrži *Helper* u kome su metode za “izvlačenje” *CommonName-a* iz sertifikata i metoda za dodavanje prava. Druga pomoćna klasa je *EventLogManager* koja služi za kreiranje loga i upis poruka u log.

4. TESTIRANJE SISTEMA

- Pozitivni test scenariji

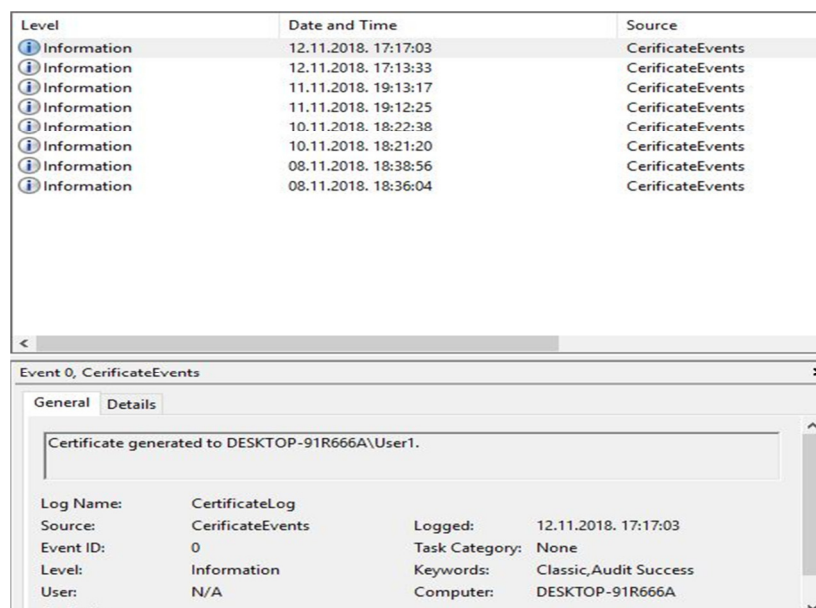
- 1) Uspješno uspostavljena konekcija servera sa CMS-om i generisan sertifikat (Slika 2), a potom upisana poruka u Certificate Event Log (Slika 3)



```
C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\CertificateServiceManager\bin\Debug\CertificateServiceManager.exe
WCFService is started...
Enter root name:
TestCA
Self-signed certificate <TestCA> already exists
Press <enter> to close server...

C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\Server\bin\Debug\Server.exe
*Successfully connected to CMS*
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
1
Choose root:
TestCA
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
```

Slika 2. – Server uspješno dobio sertifikat

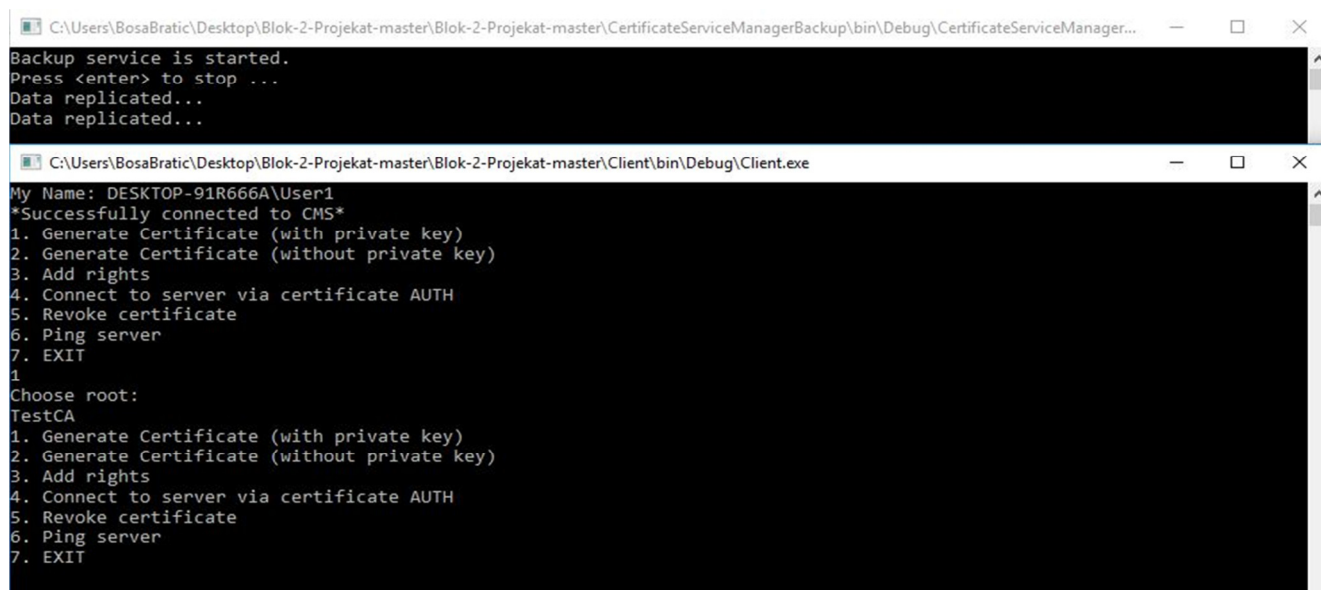


Level	Date and Time	Source
Information	12.11.2018. 17:17:03	CertificateEvents
Information	12.11.2018. 17:13:33	CertificateEvents
Information	11.11.2018. 19:13:17	CertificateEvents
Information	11.11.2018. 19:12:25	CertificateEvents
Information	10.11.2018. 18:22:38	CertificateEvents
Information	10.11.2018. 18:21:20	CertificateEvents
Information	08.11.2018. 18:38:56	CertificateEvents
Information	08.11.2018. 18:36:04	CertificateEvents

Event 0, CertificateEvents	
General	
Certificate generated to DESKTOP-91R666A\User1.	
Log Name:	CertificateLog
Source:	CertificateEvents
Event ID:	0
Level:	Information
User:	N/A
Logged:	12.11.2018. 17:17:03
Task Category:	None
Keywords:	Classic,Audit Success
Computer:	DESKTOP-91R666A

Slika 3 – Zapis u CMS Event logu

- 2) Uspješno uspostavljena konekcija klijenta sa CMS-om i dobijen sertifikat (Slika 4), a potom taj sertifikat repliciran na CertificateManagerBackup (Slika 5)



```
C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\CertificateServiceManagerBackup\bin\Debug\CertificateServiceManager...
Backup service is started.
Press <enter> to stop ...
Data replicated...
Data replicated...

C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\Client\bin\Debug\Client.exe
My Name: DESKTOP-91R666A\User1
*Successfully connected to CMS*
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
1
Choose root:
TestCA
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
```

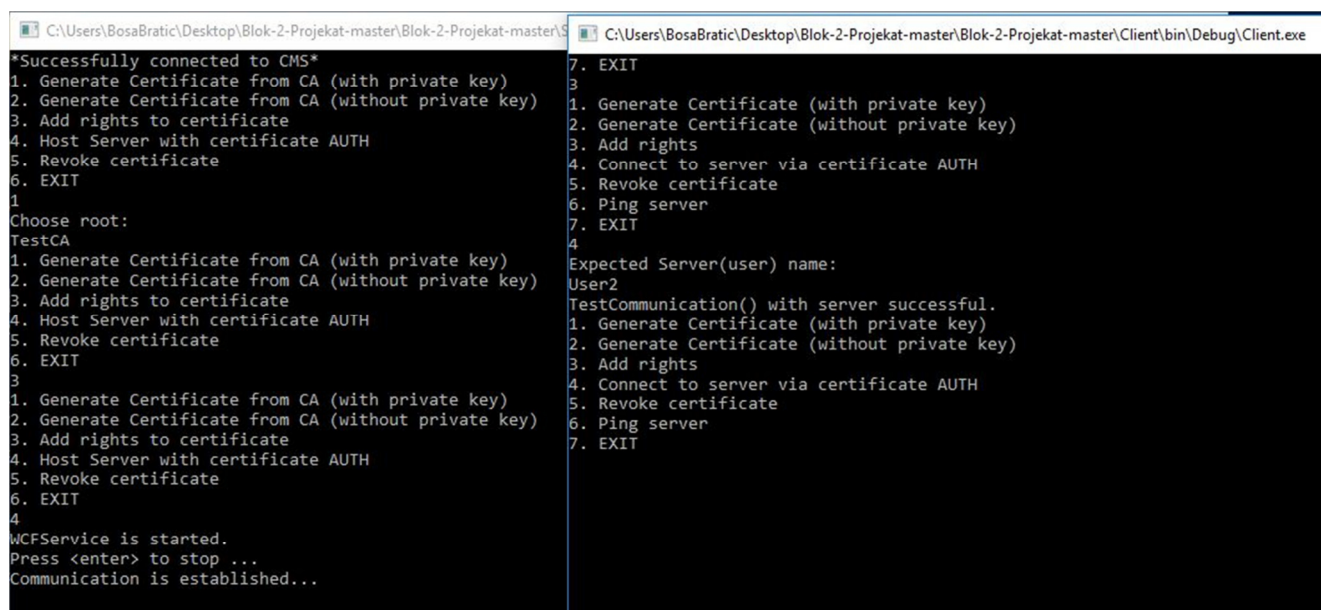
Slika 4. – Klijent uspješno dobio sertifikat koji je odmah repliciran



```
CertListBackup - Notepad
File Edit Format View Help
CN=TestCA, thumbprint: 5B6D343D67FB2EA707245554377B5CCF990E0F0A
CN=User2, OU=RegionWest_RegionEast, thumbprint: 24E4B008D6E96D9B1CC2BFA6D8A4CF1C9F07AB43
CN=User1, OU=RegionWest, thumbprint: F37A48B6633509F87F10250C9E40B670919B14C2
```

Slika 5. – Replicirani podaci na CertificateManagerBackup-u

- 3) Po dobijanju sertifikata uspješno uspostavljena komunikacija klijenta i servera

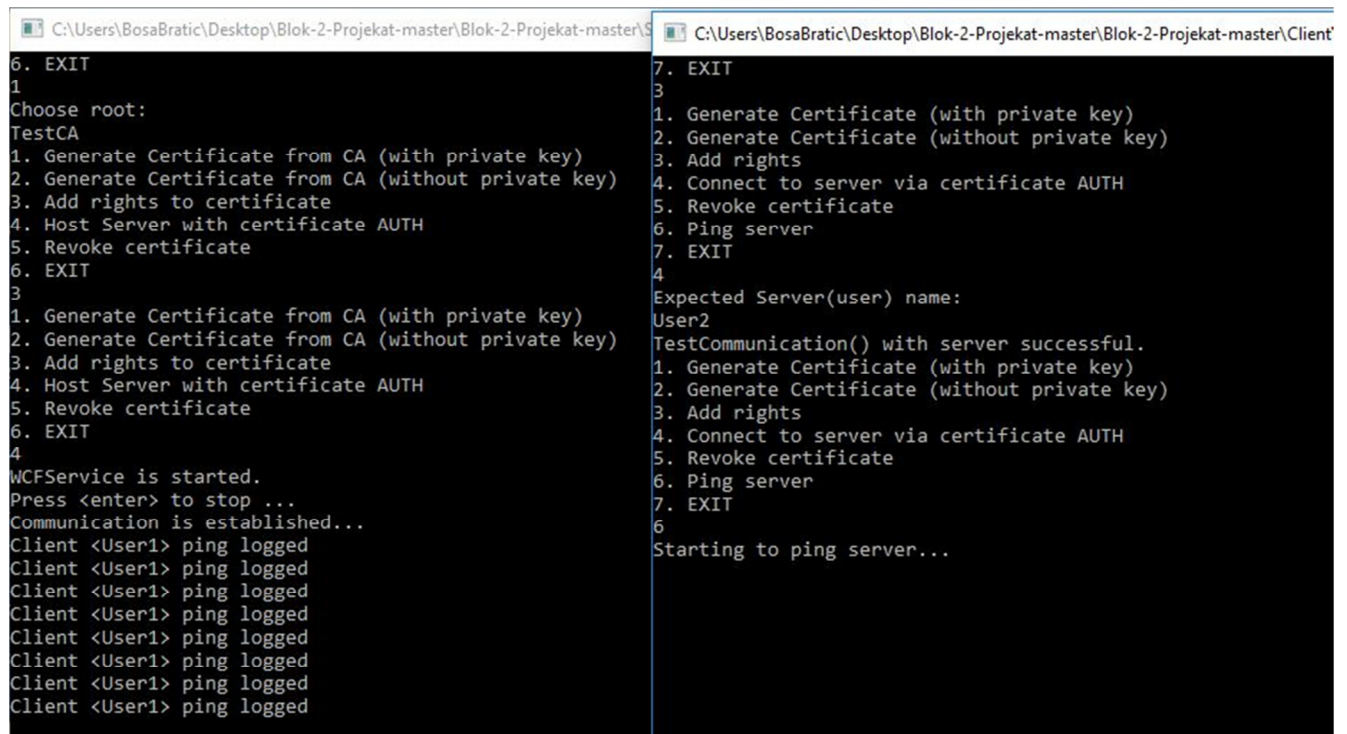


```
C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\
*Successfully connected to CMS*
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
1
Choose root:
TestCA
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
3
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
4
WCFService is started.
Press <enter> to stop ...
Communication is established...

C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\Client\bin\Debug\Client.exe
7. EXIT
3
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
4
Expected Server(user) name:
User2
TestCommunication() with server successful.
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
```

Slika 6. – Klijent i server komuniciraju

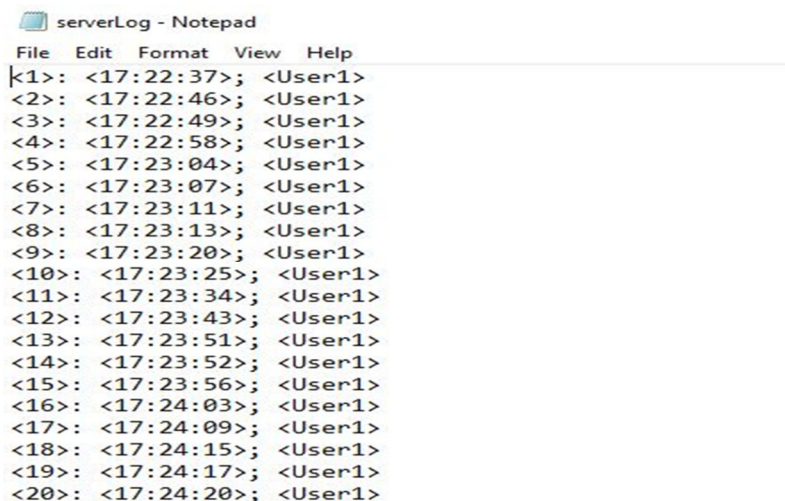
- 4) Nakon uspostave veze klijent periodično pinguje servera (Slika 7) i to server upisuje u fajl (Slika 8), a podatke o povezanom klijentu upisuje u Server Event Log (Slika 9)



```
6. EXIT
1
Choose root:
TestCA
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
3
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
4
WCFService is started.
Press <enter> to stop ...
Communication is established...
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged
Client <User1> ping logged

7. EXIT
3
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
4
Expected Server(user) name:
User2
TestCommunication() with server successful.
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
6
Starting to ping server...
```

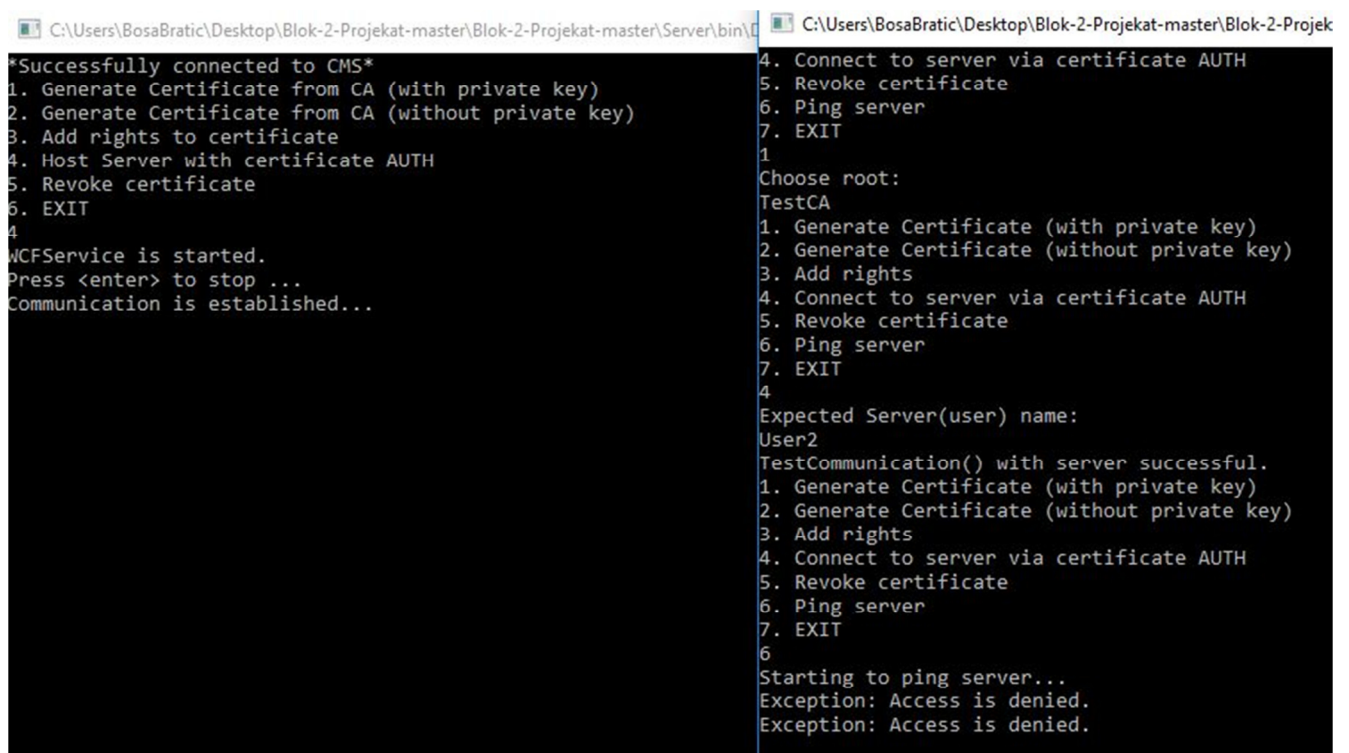
Slika 7. – Pingovanje servera od strane klijenta



```
serverLog - Notepad
File Edit Format View Help
<1>: <17:22:37>; <User1>
<2>: <17:22:46>; <User1>
<3>: <17:22:49>; <User1>
<4>: <17:22:58>; <User1>
<5>: <17:23:04>; <User1>
<6>: <17:23:07>; <User1>
<7>: <17:23:11>; <User1>
<8>: <17:23:13>; <User1>
<9>: <17:23:20>; <User1>
<10>: <17:23:25>; <User1>
<11>: <17:23:34>; <User1>
<12>: <17:23:43>; <User1>
<13>: <17:23:51>; <User1>
<14>: <17:23:52>; <User1>
<15>: <17:23:56>; <User1>
<16>: <17:24:03>; <User1>
<17>: <17:24:09>; <User1>
<18>: <17:24:15>; <User1>
<19>: <17:24:17>; <User1>
<20>: <17:24:20>; <User1>
```

Slika 8. – Log fajl servera

- 2) Klijent nije autorizovan zato što ne pripada jednoj od 4 korisničke grupe (koje su uslov autorizacije)



```
C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\Server\bin\Debug>
*Successfully connected to CMS*
1. Generate Certificate from CA (with private key)
2. Generate Certificate from CA (without private key)
3. Add rights to certificate
4. Host Server with certificate AUTH
5. Revoke certificate
6. EXIT
4
WCFService is started.
Press <enter> to stop ...
Communication is established...

C:\Users\BosaBratic\Desktop\Blok-2-Projekat-master\Blok-2-Projekat-master\Server\bin\Debug>
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
1
Choose root:
TestCA
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
4
Expected Server(user) name:
User2
TestCommunication() with server successful.
1. Generate Certificate (with private key)
2. Generate Certificate (without private key)
3. Add rights
4. Connect to server via certificate AUTH
5. Revoke certificate
6. Ping server
7. EXIT
6
Starting to ping server...
Exception: Access is denied.
Exception: Access is denied.
```

Slika 12. – Izgled klijenta i servera prilikom neautorizovanog pokušaja pinga