

CENG421 – Network Programming

FINAL PROJECT

Developing an FTP Server/Client with SSL(OpenSSL) Application

IBRAHIM ESER - 220206018

TURGUT KALFAOĞLU

Abstract

This project aims to develop a File Transfer Protocol Server/Client Application securely. This security process will be obtained with Secure Sockets Layer (SSL) and Transport Layer Security (TLS). This server/client application sends a file from server to client. In this project, we will transfer a text and image file from server to client. We will use OpenSSL.

Introduction

One of the most used applications for file transfer is FTP. "File Transfer Protocol", or FTP, was developed as a file transfer protocol. It provides file transfer between two computers connected to the Internet. If you want to transfer your files to your website, you can do it easily thanks to the FTP application. It provides fast transfer of high-dimensional data between two computers. You can use the FTP application to download files as well as upload files.

Security is also the other important term for Network Applications. Many security protocols give us encrypted keys, and authentication information then we have the more secure client/server or peer-to-peer communication. OpenSSL is an open-source implementation of the SSL and TLS protocols. The main library built with the C programming language implements the basic encryption graph.

Implementation of the Project

A. Required Tools

Installing GCC

The first step is to get the C compiler, gcc, installed.

```
sudo apt-get install build-essential
```

or

```
sudo apt-get install gcc
```

Installing OpenSSL

OpenSSL can be tricky. You can try your distribution's package manager with the following commands:

```
sudo apt-get install openssl libssl-dev
```

#NOTE#

Since Linux distribution that I use in Virtualbox is constantly crashing, I have completed some of the processes I have done there through Windows Subsystem for Linux (WSL). Ubuntu and Debian WSL.

1. Creating Authentication Certificate for SSL

```
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5$ mkdir certificate_ibrahim
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5$ cd certificate_ibrahim/
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ openssl req -newkey rsa:2048 -keyout root_key.pem -out root_request.pem
Generating a RSA private key
.....+++++
writing new private key to 'root_key.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:Izmir
Locality Name (eg, city) []:Urla
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IZTECH
Organizational Unit Name (eg, section) []:EEE
Common Name (e.g. server FQDN or YOUR name) []:ibrahimeser
Email Address []:eseribrahim07@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ openssl x509 -req -in root_request.pem -signkey root_key.pem -out root_certificate.pem
Signature ok
subject=C = TR, ST = Izmir, L = Urla, O = IZTECH, OU = EEE, CN = ibrahimeser, emailAddress = eseribrahim07@gmail.com
Getting Private key
Enter pass phrase for root_key.pem:
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
root_certificate.pem  root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ cat root_certificate.pem root_key.pem > root.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
root.pem  root_certificate.pem  root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ openssl req -newkey rsa:2048 -keyout CA_key.pem -out CA_request.pem
Generating a RSA private key
.....+++++
writing new private key to 'CA_key.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:Izmir
Locality Name (eg, city) []:Urla
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IZTECH
Organizational Unit Name (eg, section) []:EEE
Common Name (e.g. server FQDN or YOUR name) []:ibrahimeser
Email Address []:eseribrahim07@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
CA_key.pem  CA_request.pem  root.pem  root_certificate.pem  root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ openssl x509 -req -in CA_request.pem -CA root.pem -CAkey root.pem \
-p CA
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ openssl x509 -req -in CA_request.pem -CA root.pem -CAkey root.pem -CAcreateserial -out CAcert.pem
Signature ok
subject=C = TR, ST = Izmir, L = Urla, O = IZTECH, OU = EEE, CN = ibrahimeser, emailAddress = eseribrahim07@gmail.com
Getting CA Private Key
Enter pass phrase for root.pem:
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
CA_key.pem  CA_request.pem  CAcert.pem  root.pem  root_certificate.pem  root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ cat CAcert.pem CA_key.pem root_certificate.pem > CA.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$ ls
CA.pem  CA_key.pem  CA_request.pem  CAcert.pem  root.pem  root_certificate.pem  root_key.pem  root_request.pem
eseribrahim07@DESKTOP-BUBHDS8: /mnt/c/Users/eseri/ceng421_hw5/certificate_ibrahim$
```

2. Developing an FTP server and an FTP client application.

2.1. Compiling server and client

Compiling server side

```
debian07@DESKTOP-BUBHDS8: /mnt/c/ceng421_project/file_server
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_server$ ls
certificate.ibrabim ftpserver ftpserver.c image.png server.pem text.txt
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_server$ gcc -o ftpserver ftpserver.c -lcrypt
o -lssl
ftpserver.c: In function 'main':
```

Compilation requires that the crypto and ssl libraries from the OpenSSL kit be linked in with

`gcc -o ftpserver ftpserver.c -lcrypto -lssl`

Compiling client side

```
debian07@DESKTOP-BUBHDS8: /mnt/c/ceng421_project/file_client
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ls
ftpclient ftpclient.c
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ gcc -o ftpclient ftpclient.c -lcrypt
o -lssl
```

`gcc -o ftpclient ftpclient.c -lcrypto -lssl`

Running the server examples requires a PEM-style certificate. Running the server requires that the certificate be in the same directory as the server executable.

3. Running FTP server/client app with OpenSSL

```
debian07@DESKTOP-BUBHDS8: /mnt/c/ceng421_project/file_server
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_server$ ls
certificate.ibrabim ftpserver ftpserver.c image.png server.pem text.txt
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_server$ ./ftpserver
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
Reading file text.txt
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
Reading file image.png
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
Reading file image.png
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_server$ _

debian07@DESKTOP-BUBHDS8: /mnt/c/ceng421_project/file_client
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ls
ftpclient ftpclient.c
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ./ftpclient 127.0.0.1 text.txt /mnt/
c/ceng421_project/file_client/new_text_1.txt
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ./ftpclient 127.0.0.1 image.png /mnt
/c/ceng421_project/file_client/new_image_1.png
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ls
ftpclient ftpclient.c new_image_1.png new_text_1.txt
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ cat ./new_text_1.txt
merhaba, nasilsin?
000$00t#10*0] [.v0a0200EEdebian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ ./ftpclient 127.0.0.1 image.png /mnt
/c/ceng421_project/file_client/new_image_1.png
Connection made with [version,cipher]: [TLSv1,ECDHE-RSA-AES256-SHA]
Destination file is found and please enter a file that does not exist!
debian07@DESKTOP-BUBHDS8:/mnt/c/ceng421_project/file_client$ _
```

`./ftpserver`

`./ftpclient [IP(Local Network)] [File at Server] [Destination File]`

`./ftpclient 127.0.0.1 text.txt /mnt/c/ceng421_project/file_client/new_text_1.txt`

`./ftpclient 127.0.0.1 image.txt /mnt/c/ceng421_project/file_client/new_image_1.txt`

Resources

- [1] Davis, K., Turner, J. W., & Yocum, N. (2004). *The Definitive Guide to Linux Network Programming*. Apress.
- [2] Winkle, L. V. (2019). *Hands-on network programming with C: Learn socket programming in C and write secure and optimized network code*. Packt Publishing.