SSL

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893. This work is in the public domain, and cannot be copyrighted.

1 Overview

This lab requires that you use SSL certificates to authenticate devices on a simulated industrial control system network shared by Programmable Logic Controlers (PLCs) and Human Machine Inteface (HMI) devices. The concepts covered by this lab are applicable to pairs of clients and servers, e.g., a web broswer and a web server.

1.1 Background

The student is expected to have separately learned about the basic elements of PKI certificates, e.g., public/private key pairs, Certification Authorities, signing requests and certificate chains. If the student is engaded in independent study, several tutorial videos that cover public key cryptography are at:
`https://my.nps.edu/web/c3o/movies`
Tutorials on the use of the `openssl` utility can be found on the web, and details can be viewed using "`man openssl`".

The student is expected to have at least a basic understanding of the Linux command line, the basics of the file system, and the ability to use `scp` to copy files from one computer to another.

2 Lab Environment

This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer ssl
```

A link to this lab manual will be displayed.

All user ids in the lab are `admin` **and all passwords are** `password`.

3 Network Configuration

This lab includes two sumulated PLCs, two HMI devices, and a certification authority as shown in Figure 1. When the lab starts, you will get several virtual terminals, one connected to each component.

The host names of each component are per the diagram. The /etc/hosts files allow use of these host names instead of explicit ip addresses.

Initially, the `plc1` and the `hmi1` components have PKI certificates and keys provided by the `ca`. The `hmi1` component includes a `client_ssl` program that sends instructions to a PLC using client-authenticated TLS. The `plc1` component includes a `service_ssl` service that receives instructions from

hmi components. The SSL connection utilized by the client and server side of this communication are both authenticated using keys and certificates generated using the CA component.

The `plc2` and the `hmi2` components initially lack keys and certificates. They include `client` and `server` programs that are functionally identical to those on the `plc1` and `hmi1` components, except that they do not use SSL.

The ca component is configured for signing certificates within the "example.com" domain, and was used to generated and sign the initial set of certificates.

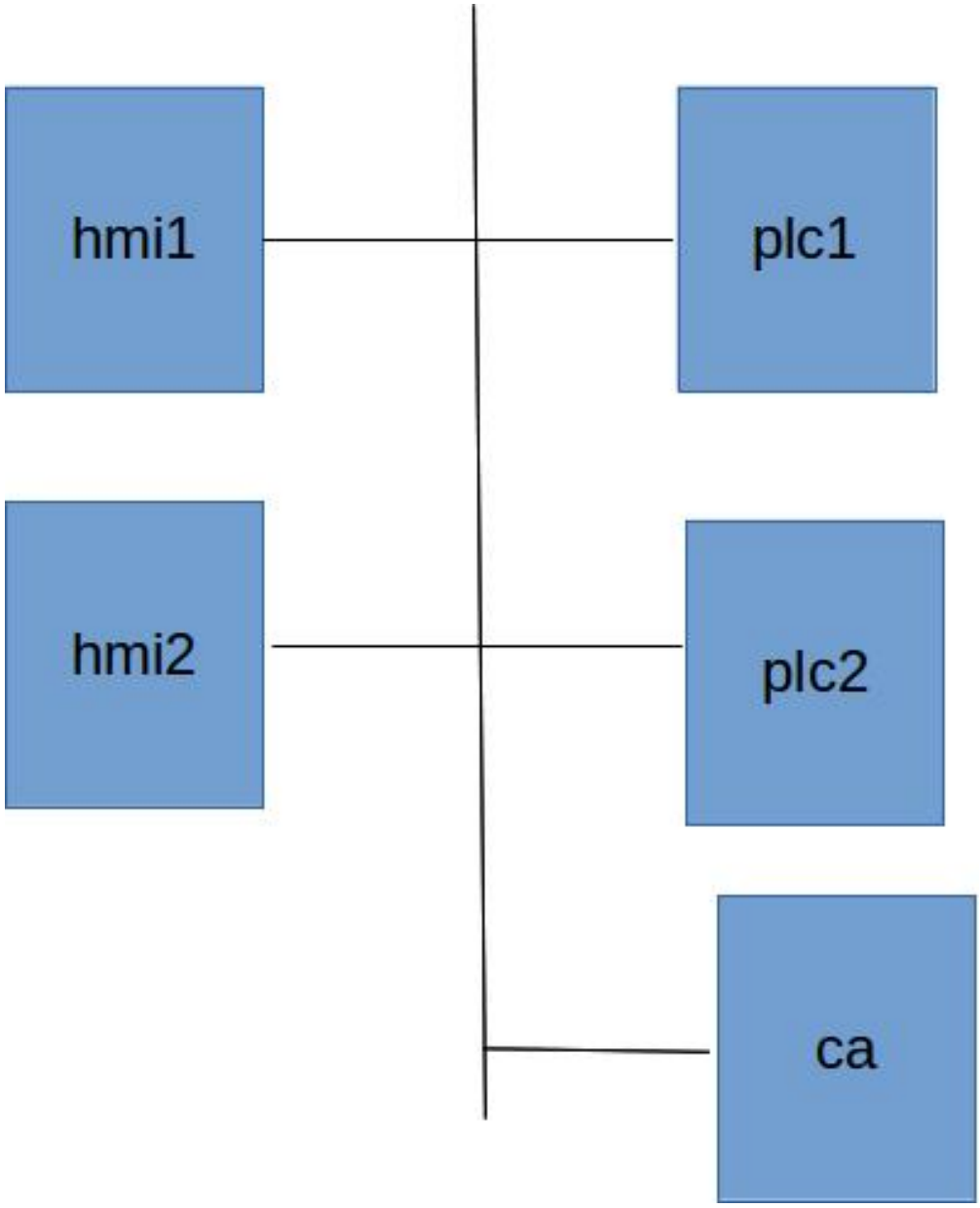

Figure 1: Network topology for the ssl lab

4 Lab Tasks

4.1 Explore

Start the server_ssl service on the PLC1 component:

```
./server_ssl
```

and the server service on PLC2:

```
./server
```

Then start wireshark on each of the two HMI components to allow you to view the network traffic.

```
wireshark &
```

Select the `eth0` device.

Then use the client programs on the HMI components to communicate with the PLCs and observe the network traffic. On the hm1 component:

```
./client_ssl plc1 This is an instruction
```

and on the hmi2 component:

```
./client plc2 This is an instruction
```

What differences do you see in wireshark when you communicate between plc1 and hmi1 as opposed to communication between plc2 and hmi2?

Try to send instructions from hmi2 to plc1. What happens, and why? Try using the `client_ssl` program on hmi2 to communicate with each PLC.

Then try to send instructions from hmi1 to plc2. Again, what happens and why?

4.2 Generate certficates and keys

Use the `openssl` utility on the CA component to generate keys and certificates for the `hmi2 and plc2` components. As an example, the key generation, signing requests and certificate signing operations that were used for `plc1` are provided below:

```
# plc1 key gen
openssl genrsa -out intermediate/private/plc1.example.com.key.pem 2048
chmod 400 intermediate/private/plc1.example.com.key.pem

# plc1 cert signing request
openssl req -config intermediate/openssl.cnf \
      -key intermediate/private/plc1.example.com.key.pem \
      -subj '/CN=plc1.example.com/O=Example./C=US/ST=CA' \
      -new -sha256 -out intermediate/csr/plc1.example.com.csr.pem

# sign plc1 cert
openssl ca -batch -config intermediate/openssl.cnf \
      -extensions server_cert -days 375 -notext -md sha256 \
      -in intermediate/csr/plc1.example.com.csr.pem \
      -out intermediate/certs/plc1.example.com.cert.pem
```

Before running `openssl` commands, change your directory to the ca directory:

```
cd ~/ca
```

Generate certificates and keys for plc2, and then for hmi2. Note when signing the hmi2 certificate, you should not include the "`-extensions server_cert`" option because you are signing a client certificate.

Use scp [1] to transfer files from the CA to the appropriate component. You will also want to transfer the certificate chain (i.e., the root and intermediate certificates) from `intermediate/certs/ca-chain.cert.pem` to the `/certs` directory of the two components. (Note you have plc1 and hmi1 as working examples).

4.3 Demonstrate communication between all 4 components

After installing the certificates and keys, start the `server_ssl` service on each of the PLC components. If you have properly installed certificates and keys on hmi2 and plc2, then you should be able to use the `client_ssl` program to send instructions to either of the PLCs from either of the HMI components.

5 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.

[1] The user ID is admin, and the password is "passsword".