Applied Cryptography Lab-06 Manual

20 October 2022 22:23

Prerequisites

Labsetup files - https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_PKI/

Task 1: Becoming a certificate authority (CA)

Firstly, copy the /usr/lib/ssl/openssl.cnf file to your working directory
Then create the following files and directories in the working directory:

```
pki_lab
  - demoCA
      - certs (dir)
      - crl (dir)
      - newcerts (dir)
      - index.txt (blank text file)
      - Serial (contains a 4 digit number, no line ending)
```

Creating certificate authority

Command

```
$ openssl req -x509 -newkey rsa:4096 -sha256 -days 3650 \
 -keyout ca.key -out ca.crt \
 -subj "/CN=www.modelCA.com/O=Model CA LTD./C=US" \
 -passout pass:dees
```

Remember the passphrase, you'll have to use it in later tasks!

Viewing the contents of files generated

Commands

```
$ openssl x509 -in ca.crt -text -noout
$ openssl rsa -in ca.key -text -noout
```

Take a screenshot and note your observations

Task 2: Generating a Certificate Request for the web server

Step 1 - Generate a public/private key pair

Command

```
$ openssl req -newkey rsa:2048 -sha256 \
 -keyout server.key -out server.csr \
 -subj "/CN=www.bank32.com/O=Bank32 Inc./C=US" \
 -passout pass:dees \
 -addext "subjectAltName = DNS:www.bank32.com, \
  DNS:www.bank32A.com, \
  DNS:www.bank32B.com"
```

The keys will be stored in server.key
Again, keep track of the passphrase used.
View the created file using the command:

```
$ openssl req -in server.csr-text -noout
$ openssl rsa -in server.key -text -noout
```

Take a screenshot and note your observations

Task 3: Generating a Certificate for your server

Command

```
openssl ca -config openssl.cnf -policy policy_anything \
 -md sha256 -days 3650 \
 -in server.csr -out server.crt -batch \
 -cert ca.crt -keyfile ca.key
```

Viewing the contents of files generated
Command

```
$ openssl x509 -in server.crt -text -noout
```

Take a screenshot and note your observations

Task 4: Deploying Certificate in an Apache-Based HTTPS Website

Step 1 - Setting up the required files

Copy the files server.crt, server.key and ca.crt to Labsetup/image_www/certs and rename them to bank32.crt, bank32.key and modelCA.crt respectively.

Step 2 - Building docker

Navigate to Labsetup and run the following commands

Commands

```
$ docker-compose build
$ docker-compose up
# in a different terminal
$ dockps
# Note the id of the container
$ docksh <id of container>
# Inside the docker shell
% service apache2 start
```

Step 3 - Setting up DNS

Open /etc/hosts in a text editor as root (in the seed vm)
Add the following entry at the end

```
10.9.0.80 www.bank32.com
```

Step 4

Open firefox and navigate to https://www.bank32.com

Take a screenshot and note your observations

Step 5

1. Go to about:preferences#privacy
2. At the bottom, under certificates, click on "View Certificates", then "import"
3. Select the ca.crt that you generated and import it
4. Ensure to check the "trust this CA to identify websites"
5. Open https://www.bank32.com again

Take a screenshot and note your observations

Question

Since bank32.com points to 10.9.0.80, if we use https://10.9.0.80 instead, we will be connecting to the same web server. Please do so, describe and explain your observations

Task 5: Launching a Man-In-The-Middle Attack

Step 1: Setting up the malicious website.

In Task 4, we have already set up an HTTPS website. We will use the same Apache server to impersonate www.example.com. To achieve

that, we will follow the instruction in Task 4 to add a VirtualHost entry to Apache's SSL configuration file: the ServerName should be www.example.com, but the rest of the configuration can be the same as that used in Task 4.

Step 2: Becoming the man in the middle

Add the following entry to the victim's /etc/hosts file:

```
10.9.0.80 www.example.com
```

Step 3 - Browse the target website

Open https://www.example.com in firefox and note your observations.