Networks and Systems Security I - Monsoon 2023

CSE 352/552

November 6, 2023

Exercise 2 (total points: 75)

Due date: Nov 20. Time: 23:59 Hrs.

There are three parts to this exercise. Each of these would involve the use of the VMs created in the previous exercises. You need to attempt all three parts.

1 Directory Encryption using `eCryptfs` (total points: 25)

This assignment aims to familiarize you with using `eCryptfs` directory encryption scheme. You are expected to perform the following tasks:

1. Install `eCryptfs` on your VMs that you used in the previous exercise.
2. Create a subdirectory which would be encrypted directory.
3. Mount it as an encrypted directory using `eCryptfs`. Use either your name as the passphrase.
4. Once mounted every file subsequently added to this directory would be encrypted and stored.
5. Create a temporary file – `temp1` and add some plaintext to it, while the subdirectory is still mounted using `eCryptfs`.
6. Unmount the directory and validate that the contents of the file `temp1` are encrypted. Print the contents on the screen and take a snapshot of the same (to be submitted).
7. Mount the subdirectory again and decrypt the file using the passphrase and print the contents on the screen.

What you need to submit:

You need to submit a report showing the following:

1. Screenshot for each of the steps required for the setup, showing all the commands executed and their outcomes (10 points).
2. Screenshot showing you display the encrypted file and the unencrypted one (10 points).
3. Description of the commands used, in terms of the arguments supplied and why it was run (*i.e.* the objective of running the command) (5 points).

2 Using OpenSSL Toolkit to Create and Validate Public Key Certificates (total points: 20)

The objective of this assignment is to familiarize you with using `OpenSSL` toolkit to generate public and private keys and certificates. You need two VMs for this, one would act as the generic TLS server and the other as the client.

You need to the following:

1. Create your own CA. The CA should have its own root certificate, and public private keys. Similarly create client and server keys and certificates which need to be signed using the CA certificates. These keys and certificates should be in the client and server VMs respectively.
2. Using `Openssl s_server` launch a server which listens on a chosen port (Say 12345) and presents the CA signed server certificates to any connection. Use the `Openssl s_client` program to connect to the server and validate the certificate. The s client program would also need to know the CA's certificate.
3. Upon connection establishment the server's certificate must be verified. Capture the `openssl s_client` and `s_server` debug messages that prove that the client successfully authenticates the server.
4. Run the server in client verification mode so that both client and server could authenticate one another. The client should present the client certificate whenever connecting to the server. This helps achieve mutual authentication. Here again you need to capture the console output showing messages that prove that the client and server authenticate one another.
5. Run the `s_server` program with `www` option so as to emulate a simple HTTPS server. Your browser needs to be configured to connect to the HTTPS server and present the client certificate. When mutual authentication succeeds, the browser screen should display the client certificate validation information.

What you need to submit:

You need to submit a report showing the following:

1. Screenshot for each of the steps required for the setup, showing all the commands executed and their outcomes (10 points).
2. Description of the commands used, in terms of the arguments supplied and why it was run (*i.e.* the objective of running the command) (10 points).

No partial points for any of the cases.

3 Encrypt NFSv4 Connections using OpenSSL (total points: 30) (Bonus Exercise

The second part of the exercise involves encrypting NFSv4 traffic (Network File System) using OpenSSL. Normally NFSv4 does not use encrypted traffic. For this exercise, you require two VMs. One would act as the NFS server while the other would be the NFS client. You would require to do the following:

1. You would first require to set-up a NFS server on the server VM.
2. The NFS server should export the directory /home/yourusername directory to the NFS client.
3. The NFS client VM should be able to read and write to files in the exported directory.

4. Use `stunnel4` program to create a TLS tunnel between the VM client and server.
5. Make sure that the server side VM uses a self-signed certificate. The root certificate should be exported to the client VM (choose any method to do so).
6. Thereafter make sure that the NFS server exports the directory not to the NFS client but to the `stunnel4` server side IP address.
7. The client should connect to its own end of the `stunnel4` tunnel IP address. This way the NFS client–server traffic should flow via the tunnel.
8. Capture the traffic between the VMs before and after creating the TLS tunnel, thereby showing the encyrpted and unencrypted traffic.

What you need to submit:

You need to submit a report showing the following:

1. Screenshot for each of the steps required for the setup, showing all the commands executed and their outcomes (15 points).
2. Description of the commands used, in terms of the arguments supplied and why it was run (*i.e.* the objective of running the command) (15 points).

No partial points for any of the cases.