SSH-ng Training Session -Task

Ques 1: What is Encryption & Decryption ? Types of Encryption ?

Ans 1: Encryption:

Encryption is a way of encoding the data such that only authorized parties can understand the information. It is the process of converting the human-readable information into the unrecognizable text i.e. Cipher Text. Encryption basically uses cryptographic key .
For example :

“Hello” — encryption —-> “SNifgNi+ukfj0=”
Plain text ciphertext

Encrypted data somehow seems very random , but the encryption proceeds in a very logical manner that allows the receiver to possess the right ke that will help to decrypt the data to turn the cipher text into the original information. The encryption process uses the Encryption algorithm to transform this data into ciphertext.

Types of Encryption

There are two types of encryption :

1. Symmetric Encryption

   In symmetric encryption , there is only one key. The same key is used for encryption as well as decryption. Therefore , a secure method is considered to transfer the key between sender and receiver.

Plaintext —--->Encrypt ((((key))))—-> Ciphertext —---> Decrypt((((Key)))) —------>PlainText

2. Asymmetric Encryption

   In asymmetric encryption, there is a key pair. One key is used for encryption purpose and a different key is used for decryption purpose. There is one public key and one private key. The data encrypted with the public key can only be decrypted with the corresponding private key. This therefore reduces the risk of unauthorized or unlawful access of data.

Decryption:

Decryption is a way of decoding the data. It is the process of converting the unrecognizable data i.e ciphertext back to its unencrypted form i.e human readable form.

Ques2 : What is a Certificate authority , Intermediate CA , Server & Client Cert ?

Ans:

1. Certificate Authority :

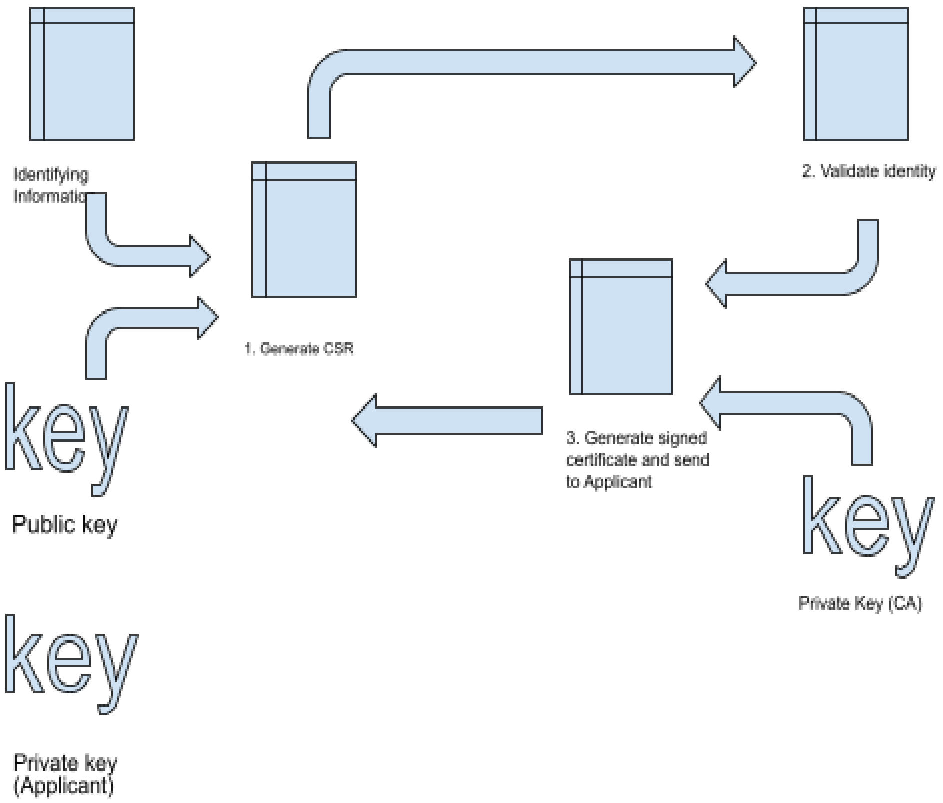
A certificate authority is an organization that validates the identities of entities and bind them with the cryptographic keys by issuing the document known as digital certificates. This digital certificate provides:

1. Authentication
2. Encryption
3. Integrity

Certificate Authority Process:

Applicant

Certificate Authority (CA)



2. Intermediate Certificate Authority

The intermediate certificate authority (CA) exists in the middle of the chain i.e they just exist in between the Trust anchor , root and the subordinate CA's . Intermediate certificates serve an administrative function which particularly means that each intermediate can be used for purposes such as issuing SSL/TLS or code signing certificates.

It provides a buffer between the end-entity certificate and the root CA , protecting the private root key.

3. Server Certificate :

Server Certificates are used to authenticate the identity of a server. When it is installed on a website , an SSL certificate turns the protocol on the website from http to https and indicates that the website is authenticated now. Users can now know after seeing the certificate that the website belongs to the said entity . SSL certificates also facilitate Encryption. Any information which a user sends to the server is protected from the 3rd party.
Example: SSL Certificate

4. Client Certificate:

A client certificate can be defined as a certificate which is used to authenticate the identity of the person who requests that can be an email user , website user or a remote server. A client certificate ensures that the server is communicating with the authenticated user. This is used to validate the identity of the client. No encryption of Data takes place in case of the client certificates. They are based on the PKI.
For example: Email Client Certificates.

Ques 3: Generate a self-signed certificate using OpenSSL with root CA certs , Intermediate CA certs , Server and Client certificates , and attach the steps that you used to generate it. Attach the final certificates that you have generated?

Ans: Build the Chain of Trust in order to get a self -signed certificate with rootCA certs , Intermediate CA , server and client certs.
The chain of trust goes on like root certs -> intermediate certs-> end-entity user certificate

STEP 1: Creating the root CA certificate :

- Generate the private key :
  - □ openssl genrsa -des3 -out rootCA.key 4096

- Create and Self-Sign the root certificate
  - □ openssl req -new -x509 -days 365 -key rootCA.key -out rootCA.cr

STEP 2: Begin With Creating Intermediate CA certificate

- Create the Certificate Key:
  - □ openssl genrsa -out IntermediateCA.key 4096

- Create the CSR :
  - □ openssl req -new -key IntermediateCA.key -out IntermediateCA.csr

- Provide the details that are asked

- Verify the CSR:
  - □ Openssl req -in IntermediateCA.csr -noout -text

- Generate the certificate using this csr and key along with the CA root key:
  - □ openssl x509 -req -days 1000 -in IntermediateCA.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out IntermediateCA.crt

STEP 3: Creating the Server Certificate:

- Creating the certificate key:
  - □ openssl genrsa -out server.key 2048

- Creating the CSR:
  - □ openssl req -new -key server.key -out server.csr

- Generate the Certificate using this CSR and Key along with the IntermediateCA root key:
  - □ Openssl x509 -req -in server.csr -CA IntermediateCA.crt -CAkey IntermediateCA.key -set_serial 01 -out server.crt -days 500 -sha1

STEP 4: Creating the Client Certificate:

- Creating the certificate key:
  - □ openssl genrsa -out client.key 2048

- Creating the CSR:
  - □ openssl req -new -key client.key -out client.csr

- Generate the Certificate using this CSR and Key along with the IntermediateCA root key:
  - □ Openssl x509 -req -in client.csr -CA IntermediateCA.crt -CAkey IntermediateCA.key -set_serial 01 -out client.crt -days 500 -sha1

- Verify The Certificate using :
  - □ Openssl x509 -in <name>.crt -text -noout

Ques 4: How does SSH Works ?

Ans: SSH which stands for ‘Secure shell’. It provides a secure way for two computers to connect remotely. It use encryption so that the hackers cannot interpret the traffic between two connected devices.

SSH is basically a client-server based protocol. The protocol allows the device which is requesting for information or services which is basically the client to connect to another device which is the server. Being a client-server based protocol , an SSH client have to initiate an SSH session with an SSH server. The connection setup is done by the SSH client. After this , the server responds by sending the client a public cryptography key. So , how basically this is done is that public key cryptography is used to verify the identity of SSH server, and then symmetric key encryption and algorithms are used to maintain the data in ciphertext. In this way , the integrity and privacy of data transmission is maintained between the client and server in both the directions.

Steps that are basically involved in creating an SSH session are:

— 1. Client initiates the connection by contacting server —>

← 2. Sends server public key —

SSH Client SSH Server

← 3.Negotiate parameters and open source channel —

— 4. User login to server host operating system — >

Ques 5: What is key forwarding in SSH? How do you forward keys? Mention a use case?

Ans: Key Forwarding:

SSH key forwarding is the concept in which we can authenticate ourself without actually storing the private keys on the lost server.
It is usually implemented using the SSH Agent Forwarding. To forward the key , we have to let the SSH agent connect to the server and act as us , this does not send our keys . it just lets the remote server access our ssh agent and verify our identity.

Use Case: It can be used to copy files from one server to another without copying the files on our localhost, which could take more time and resources , if we copy to our localhost then to the other server.

Ques 6: What is agent Forwarding in SSH? How do you perform agent forwarding with SSH? Mention a use case?

Ans: Agent Forwarding in SSH:

SSH agent forwarding can be thought of as a mechanism where the SSH client allows the SSH server to use the local agent on the server as if the user is locally present on the system.
SSH agent forwarding can be used to make the deploying process to a server very simple. It allows you to use the local SSH keys instead of leaving keys sitting on your server.
SSH agent Forwarding allows you to use private , local SSH key remotely without worrying about leaving confidential information on the server .

Setting up SSH agent :

1. Check is the SSH key is set up and working fine.
   If not then add SSH keys using
   Ssh-add ~/.ssh/id_rsa
2. Open up the file /.ssh/config. If doesn't exist create it.
3. In that file enter the following test:
   Host <server's domain name>
   ForwardAgent yes

Use case:

Consider that you’re connected to a remote server , and you want to git pull some code that you’re storing on github. Now you wish to use SSH authentication but you don’t have the private keys on the remote server though you have it on your machine.

To solve this problem, what you need to do is that you can open your local SSH agent to the remote server . This will not send the keys over the internet , it will just let a remote server access your local SSH agent and verify the identity.

Created By: Janvi