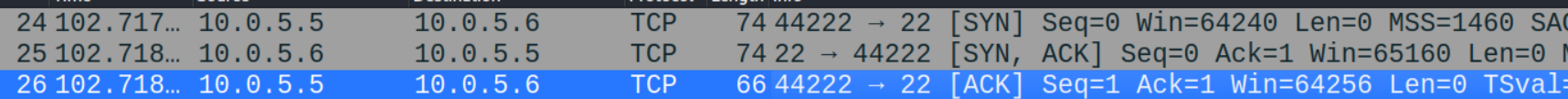
Understanding SSH Authentication

Ssh uses either password based authentication or public key based authentication

Let’s examine password based auth first

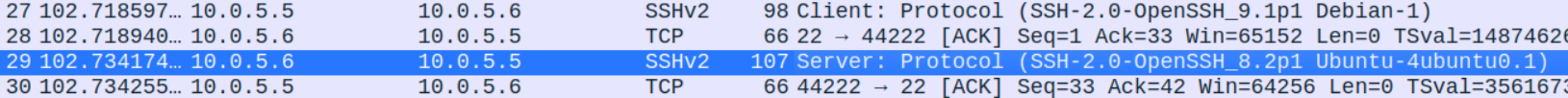
As for TCP based connection 3-three handshake takes place first



A ssh session gets established in two phases

Phase I: A secure communication channel and encryption is established to protect future communications.

Packet 1&2: Client and Server exchanges the protocol version of ssh being used and acknowledges the messages



Packet 3&4: Key Exchange Init during which they exchange the encryption and hashing algorithms supported on each end

Table

Description automatically generated

Packet 5&6: Diffie-Hellman key Exchange Init by client and Server responds with Diffie-Hellman key Exchange Reply and new keys

Now client responds with new keys

These new keys are nothing but shared secret session key calculated using Diffie Hellman, which is identical on both ends.

Using these new keys a secure encrypted channel is created and from here all communication goes encrypted

Phase II: Authenticate the user and find if he is authorized to access the server.

The Server prompts the client for username and password and all this communication is encrypted using shared secret session key.

The password based authentication is not recommended because they can be cracked.

**Using Public Key Authentication**

Before we establish public key authentication we need to generate a key pair, as this is asymmetric method one key(public) will be used for encrypting the data and other key(private) would be used for decrypting the data.

Command for generating a key-pair

**ssh-keygen -t rsa -b 4096 -f ~/.ssh/ssh-server.key -C "My ssh-server key"**

This would create two keys public is stored in ~/.ssh/ssh-server.key.pub

Private is stored in ~/.ssh/ssh-server.key

Now this public has to be moved to the server to authorize ourselves when we connect.

Command to move the public key to server

**ssh-copy-id -i $HOME/.ssh/ssh-server.key.pub** [**seed@10.0.5.6**](mailto:seed@10.0.5.6)

Go to servers authorized keys under .ssh folder you would find this key there

Text

Description automatically generated

Before we proceed to use public key based authentication we need to disable password based authentication

For that go to /etc/ssh/sshd\_config file and set PermitRootLogin to no

Also restart the sshd service using sudo systemctl reload ssh

If there is a passphrase set for the private key it would ask you to enter it.

In-order to get rid of it use ssh-agent and ssh-add

Ssh-agent bash

Ssh-add ~/.ssh/my-ssh-server.key

**Now I have a question how client verifies the server, like if it is the same server client wishes to connect**.

For this purpose we have something called ssh key fingerprint.

When you connect to the server for the first time you may get a message like authenticity of the server cannot be established.

Text

Description automatically generated

Login to the server and run below command

ssh-keygen -lf /etc/ssh/ssh\_host\_ed25519\_key.pub

If both values match we can confirm we are connecting to the same server.

Ok now lets get back to the flow in phase2

* The client begins by sending an ID for the key pair it would like to authenticate with to the server.
* The server checks the authorized\_keys file of the account that the client is attempting to log into for the key ID.
* If a public key with a matching ID is found in the file, the server generates a random number and uses the public key to encrypt the number.
* The server sends the client this encrypted message.
* If the client actually has the associated private key, it will be able to decrypt the message using that key, revealing the original number.
* The client combines the decrypted number with the shared session key that is being used to encrypt the communication, and calculates the *MD5 hash* of this value. MD5 is a message-digest algorithm that uses the hash function to generate a 128-bit hash value.
* The client then sends this MD5 hash back to the server as an answer to the encrypted number message.
* The server uses the same shared session key and the original number that it sent to the client to calculate the MD5 value on its own. It compares its own calculation to the one that the client sent back. If these two values match, it proves that the client was in possession of the private key and the client is authenticated.

If you want to debug the whole process on server you can use the following command

/usr/sbin/sshd -d