Man in the Middle - ssh

Security questions

1. Explain the steps above. Explain the purpose/meaning of each step

1. create ssh key on your HL LiveCD ssh-keygen
   This generates a asymetric key to identify my with the private key.

2. copy the ssh public key /root/.ssh/id_rsa.pub to the hackinglab/alpine-openssh-server docker into ~./ssh/authorized_keys
   This allows the user to login with rsa-ssh without a password.

3. try to login from your LiveCD Linux via SSH MITM into the SSH Server: ssh -l hacker -p 10022 localhost
   The MITM proxy don't have the private key and therefore can't prove his identity to the upstream as the requester.

4. on your LiveCD: disable password authentication in your HL LiveCD /etc/ssh/ssh_config (PasswordAuthentication no)
   Switching from password authentication to ssh authentication.

5. try again to login from your LiveCD Linux via SSH MITM into the SSH SERVER by using public/key auth ssh -l hacker -p 10022 localhost
   This doesn't allow the MITM server to authenticate itself as the requester on the upstream.

6. if you did the ssh pub key installation into hackinglab/alpine-openssh-server correct, you can test pub key authentication using this command ssh -l hacker -p 22 localhost
   It authenticates me as the given user by using generated key.

2. Explain why public/key auth is really preventing MitM

1. An SSH key pair, which includes a public and private cryptographic key, is generated by a computer.
2. The public key is stored on the server that you log into, while the private key is stored on your computer.
3. When you attempt to log in, the server will check for the public key and then generate a random string and encrypt it using this public key. This encrypted message can only be decrypted with the associated private key.
4. The server will send this encrypted message to your computer. Upon receipt of the message, your computer will decrypt it using the private key and send this message back to the server. If everything matches up, you’re good to go.

As shown from the source, the client can make sure the server has the public key and the server can make sure the client is has the private key for authentication. In addition, the client has the possibility to sign some relevant session attributes for the session. Also, rsa-keys are longer than passwords andasymmetric.

3. Explain the purpose of editing the ssh client configuration

The client only support authentication with keys.

4. Explain why 2FA would not fix the problem of ssh MitM

With 2FA the server don't prove itself to be real, only the clientside is authenticated withit. A MitM would no help, because a ssh tunnel already was established between the client and MitM proxy.

2FA would only protect the user if the password is in some list, but not againt a MitM.

5. What is the lesson learned?

Only use ssh-key authentication and protect these keys on offline-device or HSM. (Or on my Yubikey)