**Unit Name:** Network Security and Resilience

**Unit Code:** TNE30009

**Title:** NSR/AS Lab 4 – Public and Private key Encryption

**Name:** S M RAGIB REZWAN

**ID:** 103172423

Contents

[Steps take to set up the login 2](#_Toc133311110)

[How the login works 4](#_Toc133311111)

[How does diffie hellman work? 6](#_Toc133311112)

# Steps take to set up the login

In order to set up the log in, the following sequences of steps were needed to be performed:

Setp, what did you do? Why did you do it? Proof or way to verify.. screenshot

1. Setting up Virtual Machine (VM):

In here, the command “ifconfig” has been used on both VMs in order to find out their ip addressed. Once this has been done, the command “ping <ipaddress>” has been done on both VMs in order to ensure that:

* 1. Their TCP/IP protocol was working (via pinging their own ip)
  2. They can communicate with the other VM (via pinging the ip address of the other VM)

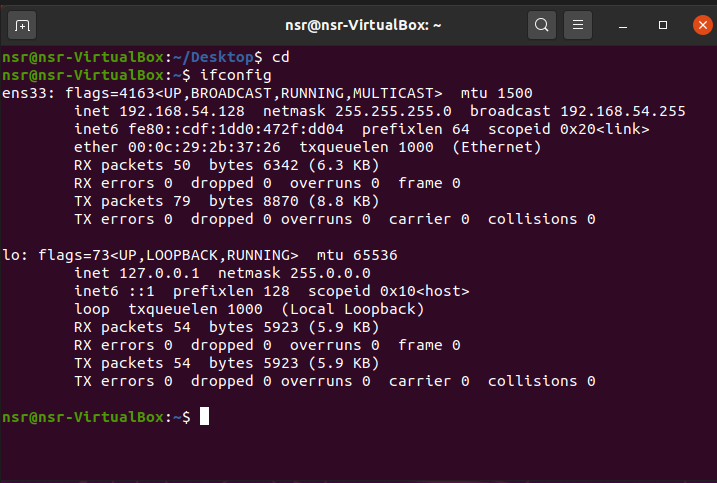


Figure 1: IP of VM1

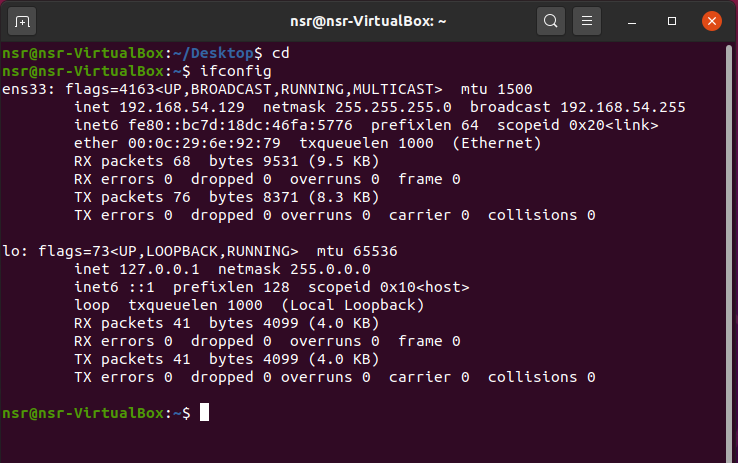


Figure 2: IP of VM2

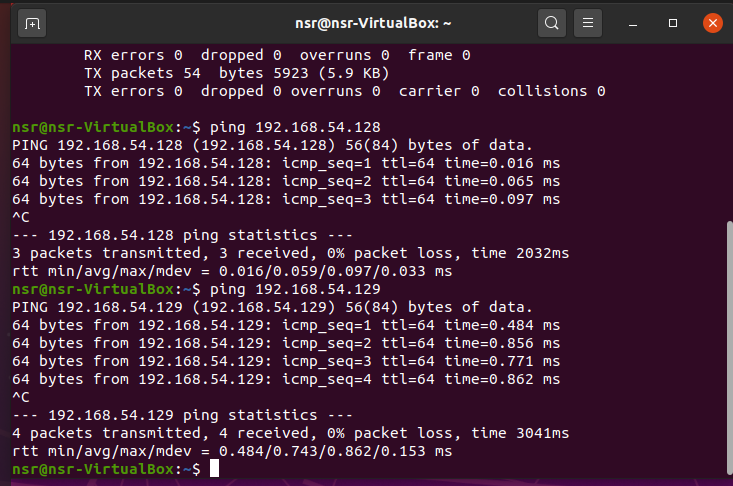


Figure 3: self pinging and pinging VM2 from VM1

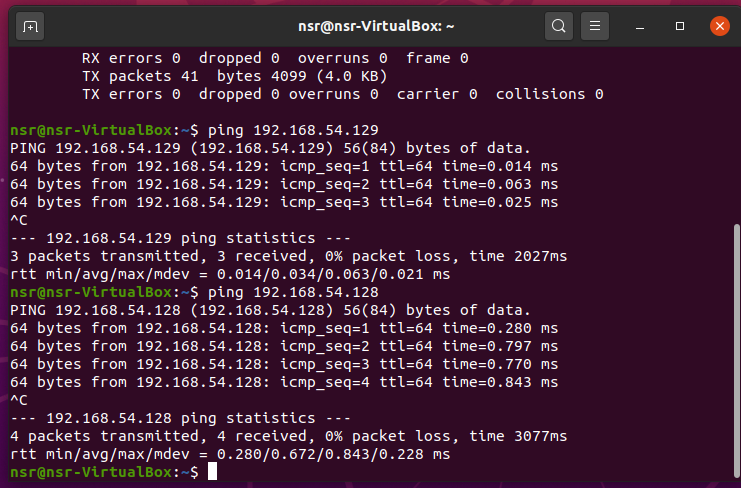


Figure 4: self pinging and pinging VM1 from VM2

1. Installing OpenSSH on VM1

This had been done by using the commands “sudo apt-get update” and “sudo apt-get install openssh-server”. These commands update the VM and installs the OpenSSH server on it. This has been done to set up the VM1 as an ssh server and ensure that we can pass the key in the later steps using the ssh

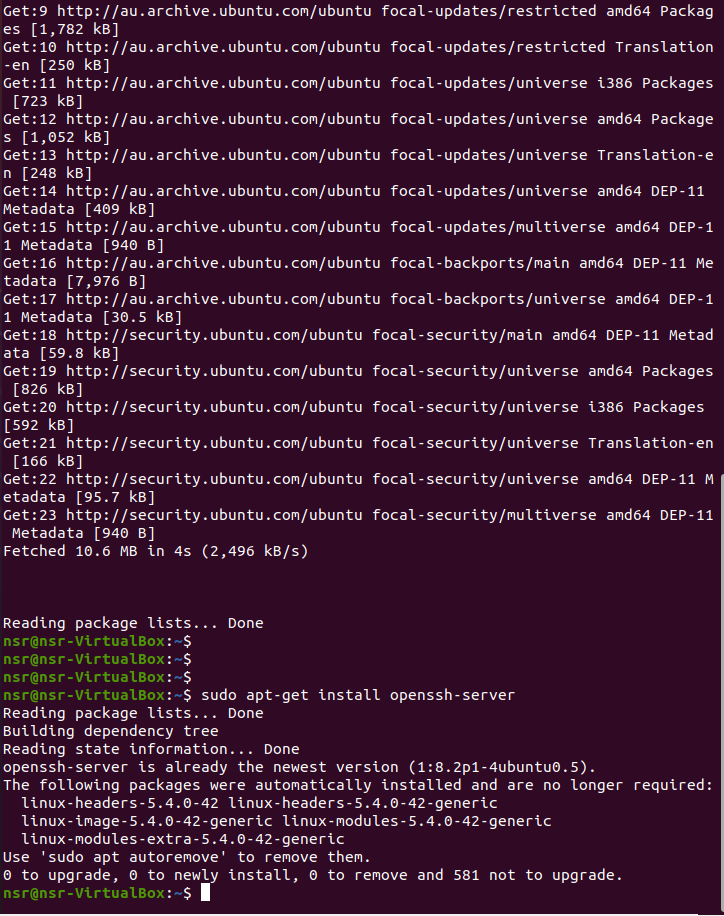


Figure 5: OpenSSH setup on VM1

1. Generating private and public keys on VM2:

This has been done using the commands ”ssh-keygen –t rsa”. This basically asks it to generate the private and public key pair for VM2 using ssh’s key generation fucntion. After this, we accept the defaults which ensure that the key pair is stored in /home/nsr/.ssh directory (as that is its default) and is stored without any passphrase (ie the user doesn’t need to say any password or anything in order to prove its validity).

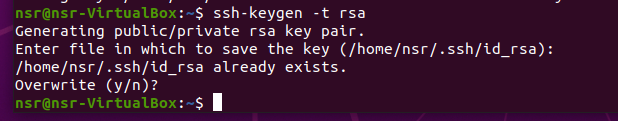


Figure 6:ssh’s Key generation on VM2

This can be verified by by going into the /home/nsr/.ssh directory can cat-ing the files id\_rsa (for private key) and id\_rsa.pub (for public key) usin he commands: “cat /home/nsr/.ssh/id\_rsa” and “cat /home/nsr/.ssh/id\_rsa.pub”

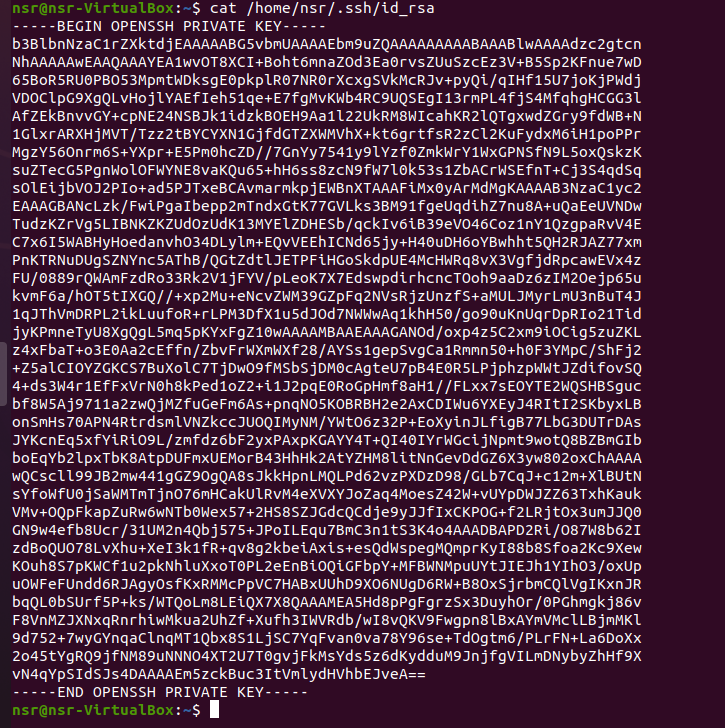


Figure 7:private key generate on VM2

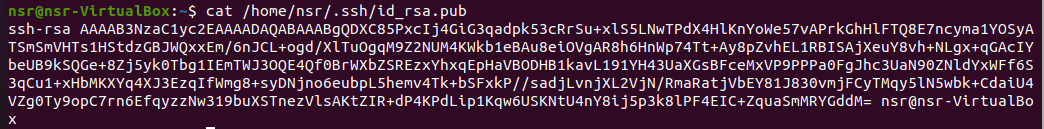


Figure 8:public key generate on VM2

1. Transferring the public key from VM2 to VM1:

This has been done using the command “ssh-copy-id –f [nsr@192.168.54.128](mailto:nsr@192.168.54.128)”. It basically forcefully transfers the public key from vm2 to vm1 and stored it as a authorized key in the /home/nsr/.ssh/authorized\_key file. This will be used later on to allow the users to log into vm1 from vm2 using ssh without any password. This in turn can be verified using the command “cat /home/nsr/.ssh/authorized\_key”

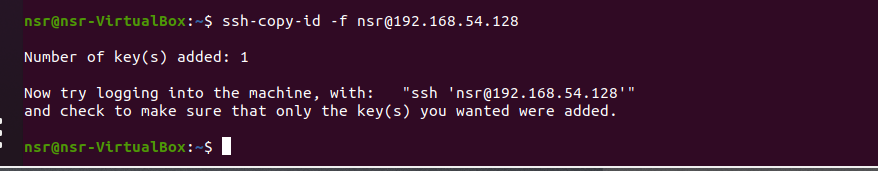


Figure 9:public key being passed from VM2 to VM1

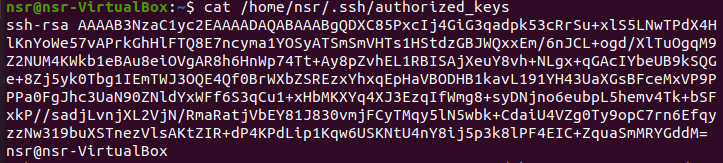


Figure 10:verifying that the public key is indeed stored in that file as an authorized key in VM1

1. Logging in to VM1 from VM2 using ssh without password:

This has been done using the command “ssh –l nsr 192.168.54.128”. It basically tries to log into vm1 for vm2 using the ssh tunnel (as vm1 has been designated as ssh server). Here to can be noticed that no password has been needed to be entered to allow us to gain access. This can be further ensured by typing ifconfig and checking that the ipconfig of vm1 can indeed be seen

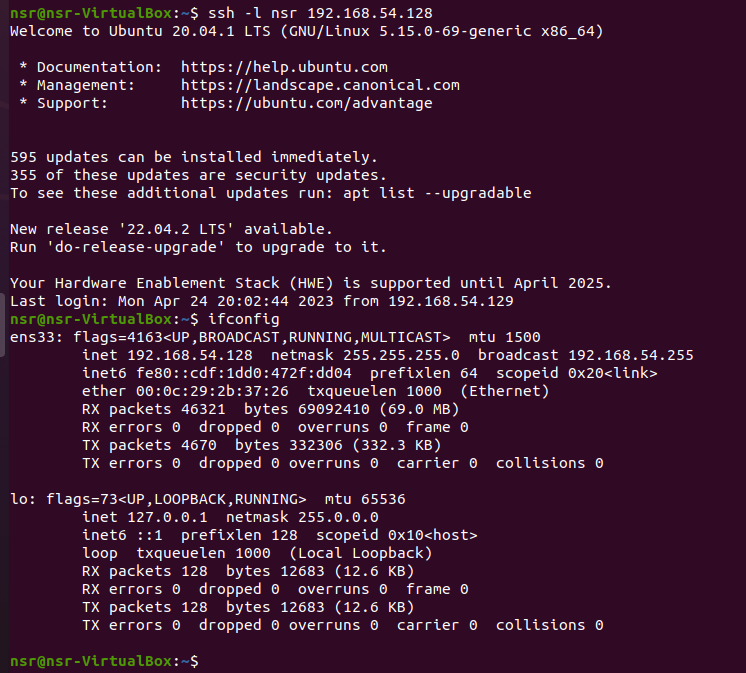


Figure 11:ssh into vm1 from vm2 without using any password, alongside ifconfig for proof

# How the login works

In order to understand why VM1 allows VM2 to ssh into it without password, one must first view the wireshark packets being passed into it through ens33 interface. There we can see the following information in following the tcp stream:

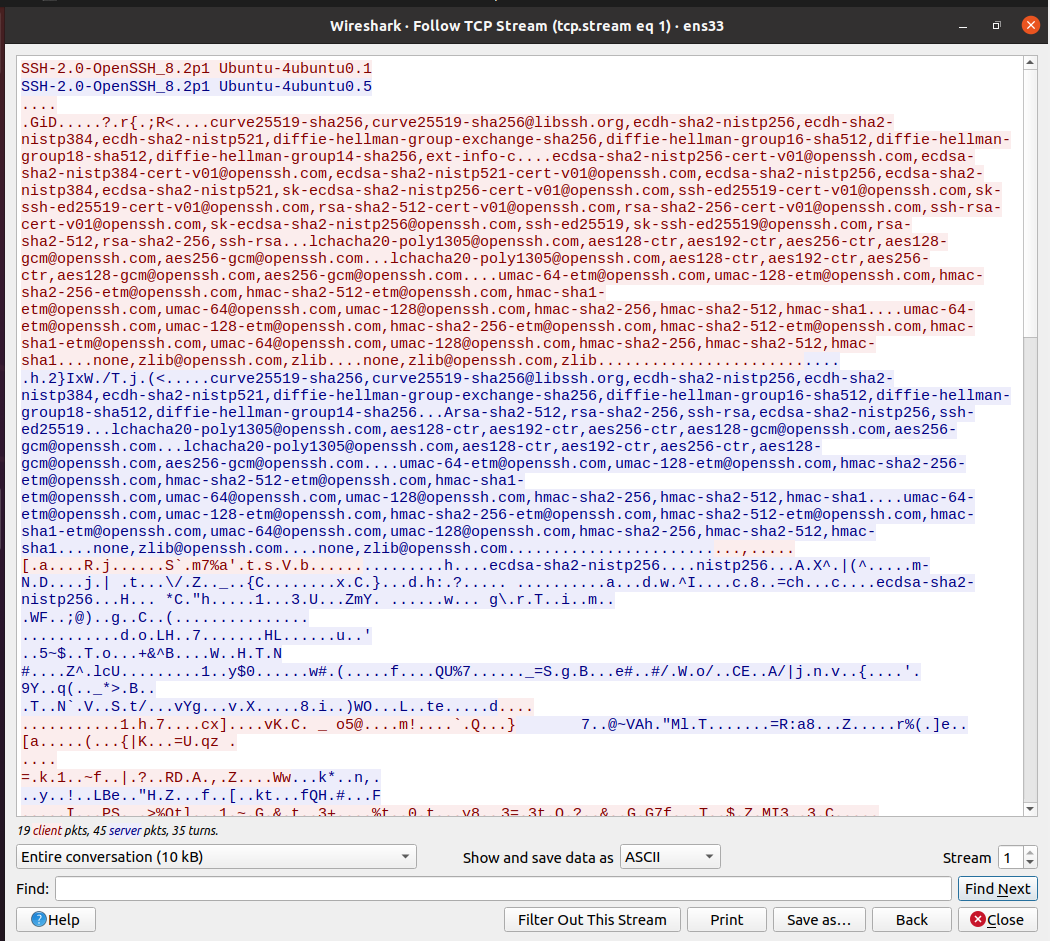


Figure 12:follow the tcp stream on ethernet’s ens33

Here, we can see that information about ssh server versions, type of ssh, gid, encryption algorithms, etc has been communicated between the client (VM2) and server(VM1). But all of these information look extremely complicated (and almost gibberish) when viewed together. Hence, it would be better to observe the individual packets that were passed during the log in instead:

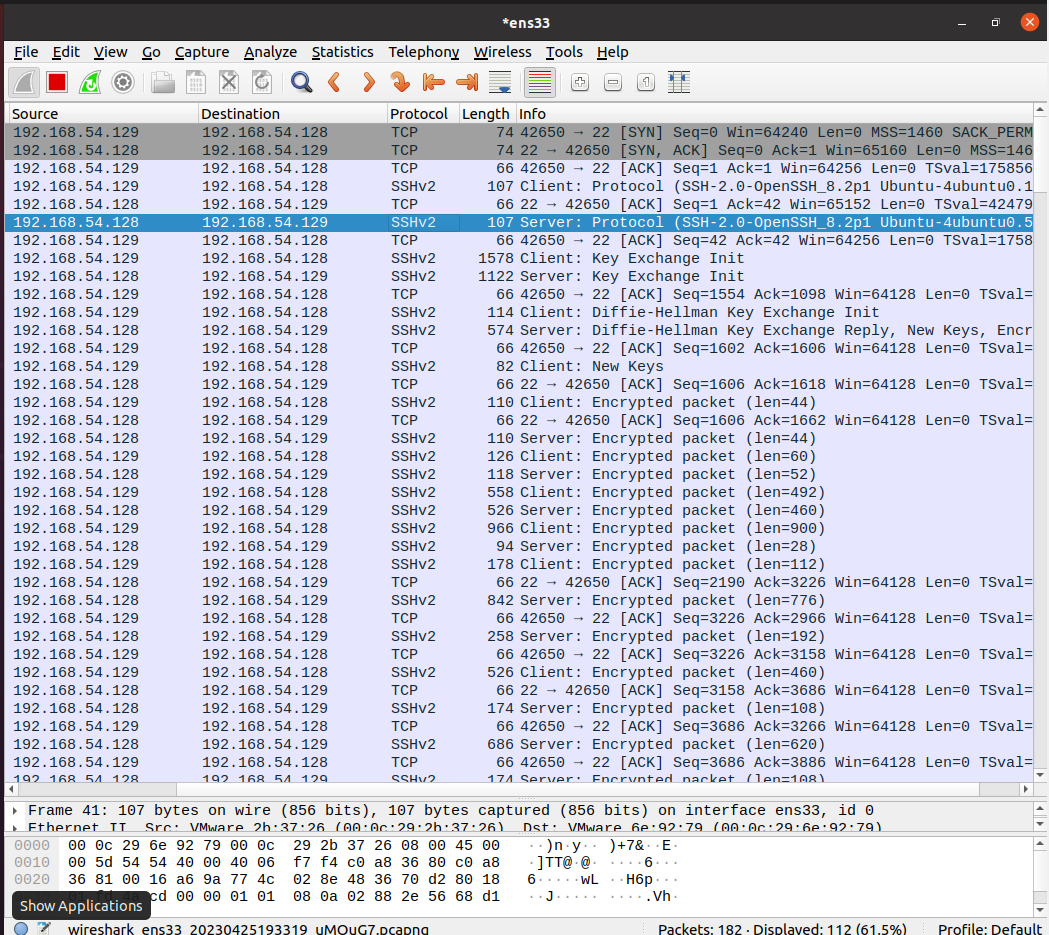


Figure 13: Individual packets that has been passed through ens33 (detected by wireshark)

Here, basically the following sequence of actions had taken place during login:

1. The 3 way hand shake had been performed between the VMs to ensure that they are ready to communicate. This had been done by the VM2 sending SYN packet, VM1 responding with SYN and ACK packet and VM2 responding with the ACK packet.
2. After that, client (VM2) and the server(VM2) had exchange information about which protocols and their versions they were using to communicate (here it was SSH, but using OpenSSH’s version) . Each time the information was passed, it was acknowledges by the VM from the other end before the VM from the other end gave out details regarding its own protocol and version
3. After this, a key exchange had occurred between the two VMs where they determined the cookie, encryption algorithms used and its lengths, mac algorithm used and its length, compression algorithm used and its length. This would be used later on for diffie hellman
4. Once the above step has been completed and acknowledged, the diffie hellman key exchange had taken place in order to perform the key exchange. Here the client had sent the “e” value that will be used for diffie hellman. Seeing this, the server had responded back with “f” value of diffie hellman. These are basically the secret numbers that are being passed between the two VMs.(**details spoken in diffie hellman process)**
5. During the above step, when the server sends back the Diffie hellman response, it also sends an authentication message to the client. So the client responds to the authentication message using shared secret (calculated using diffie hellman process) as the key for the encryption. In that authentication packet, the new symmetric key that will be used to encrypt the messages from here on out is sent to the server. Thus, when the server receives it, it uses the key to encrypt the message from then on.
6. Since all of these have been successful the server knows it can trust the connection with the client and thus doesn’t ask for the password anymore

# How does diffie hellman work?

Basically in the beginning, the two VMs mutually decide on two numbers, the modulus (p) and base (g). After this, they create their own secret numbers (ie VM2 creates a and VM1 creates b). When this had been done, both VMs perform the following calculation:

A= ga mod p [for VM2]

B= gb mod p [for VM1]

Once this has been done, they send these values across to the other VM during the Differ Hellman key exchange process. Once this has been done, the following calculations are performed:

S= Ba mod p [for VM2]

S= Ab mod p [for VM1]

These in turn result in the same secret S value. Thus the VM2 then sends a new symmetric key (which will be used to encrypt the communication form here on out) using the encryption created using the S key to VM1. Then VM1 using the S key (that it calculated from its own end) and when it sees the new keys hidden inside the packet, it ensures that VM1 is valid and thus accepts to encrypt the packet using the new symmetric key from that moment forward.

Hence, throughout the process, the eves droppers are not able to find out the symmetric keys the machines have decided to use to encrypt their messages and thus are unable to break the encryption.