

Laboratory Session 4

Introduction

The purpose of this lab is to investigate the use of public key encryption for authentication. We will be using a public / private key pair to authenticate an SSH session.

In public / private key cryptography different, but related keys are used to encrypt and decrypt. A message encrypted with the public key can only be decrypted with the private key and vice-versa. Public keys are not secret. Private keys are.

These keys can be used for authentication. Possession of a private key that corresponds to a public key can be regarded as proof of identity. In this approach to authentication, a challenge is issued to the person who holds the key pair. They then encrypt the challenge with their private key which becomes the response. The challenger then uses the public key to decrypt the response. If it is the challenge then they have proven they have the private key.

We will use the Virtual Machines VM1 and VM2 from the first three labs. You will set up an SSH server on VM1 and log into it from VM2 using the public key. You should be able to log in without using a password.

Method

1. Set up the Virtual Machines used in the previous labs. Check connectivity between the two VMs. Note the IP addresses of VM1 and VM2 using **ifconfig**.

2. Install OpenSSH on VM1 and VM2. Do the following on both machines.

Type the following into a command line terminal:

```
sudo apt-get update
```

```
sudo apt-get install openssh-server
```

You may be asked for a password. All passwords are **user**.

3. Generate the public/private key pair on VM2

Type the following into a command line terminal: **ssh-keygen**

```
-t rsa
```

Accept all the defaults. Do not include a passphrase.

You should now have a private key pair in the directory **/home/nsr/.ssh**. Display the private key by typing **cat /home/nsr/.ssh/id_rsa** and the public key by **cat /home/nsr/.ssh/id_rsa.pub**

4. Still on VM2, transfer the public key to VM1. In the following instruction **<VM1 IP address>** is the IP address of VM1 as noted in step 1.

```
ssh-copy-id nsr<VM1 IP address>
```

You should now be able to see the file **/home/nsr/.ssh/authorized_keys** on VM1 which should contain the contents of your public key. Again examine it using **cat /home/nsr/.ssh/authorized_keys**

5. From VM2 try to log into VM1 using ssh. You should not need a password. If you are asked for a password you have done something wrong. Check what you have done and try again.

To login from VM2 to VM1 use the following where **<VM1 IP address>** is the IP address of VM1 as noted in step 1. (Note: the following command uses the letter "l" not the number "1").

```
ssh -l nsr <VM1 IP address>
```

Do a screen capture showing the successful login.

- You should now be logged into VM1. Check by typing **ifconfig**. If successful log out and log back in but this time capturing the exchange of packets with Wireshark.

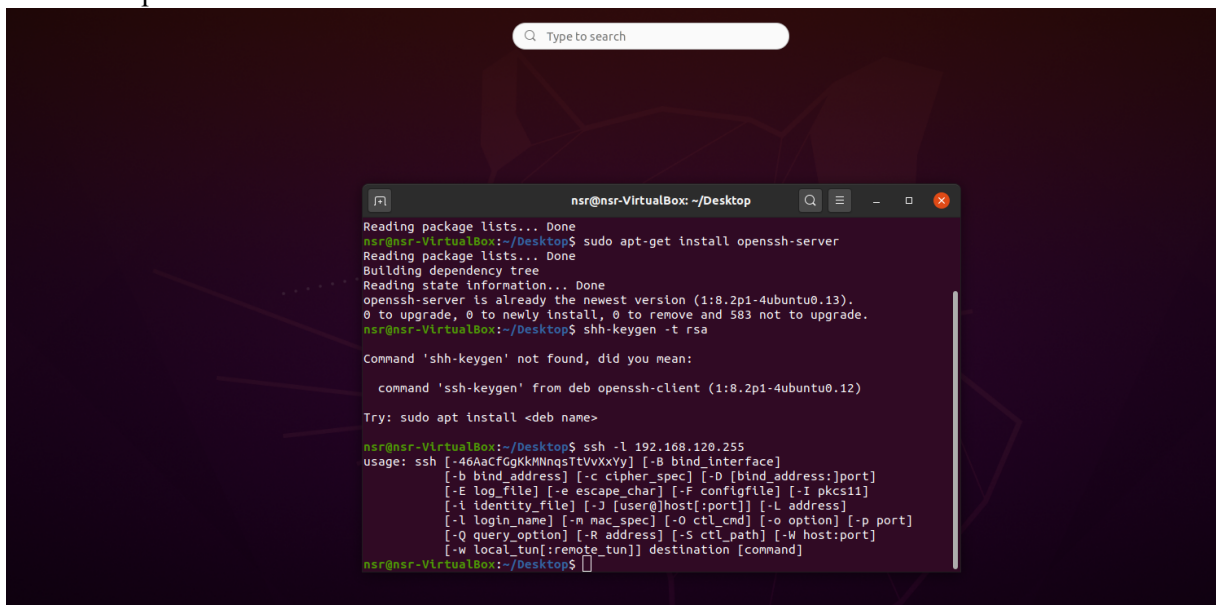
Laboratory Session 4

- You should see the three way handshake, a Diffie-Hellman key exchange and authentication messages. Do a screen capture to show the instructor.

Assessment of this lab

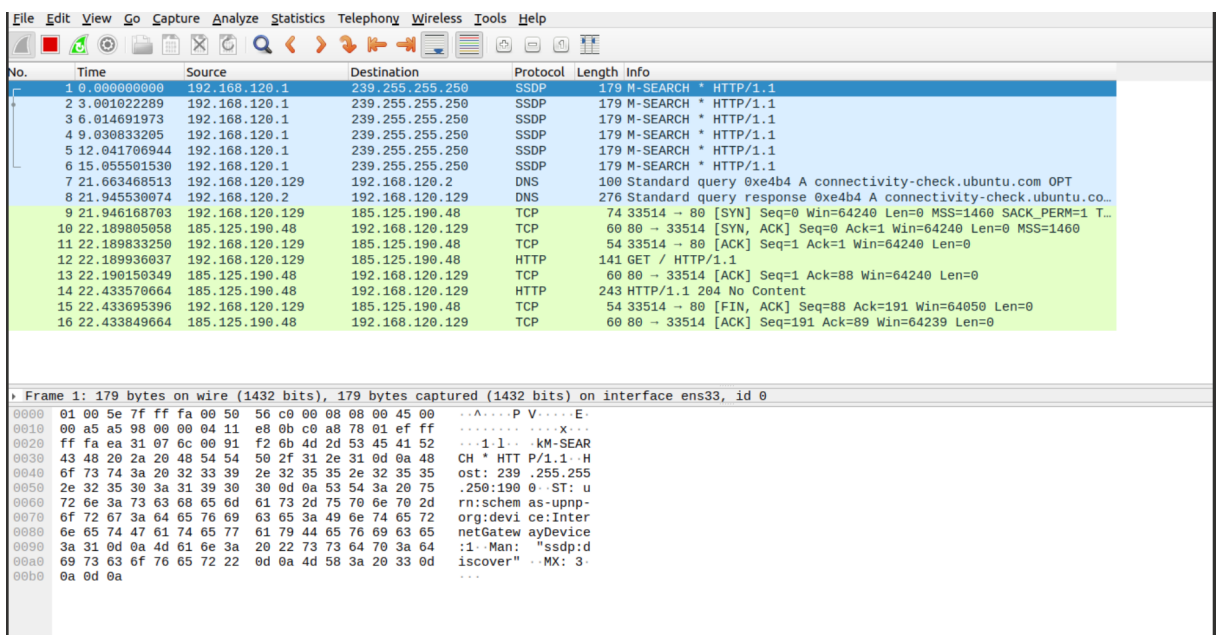
This lab will be assessed by showing the following to the lab supervisor:

- Screen dumps showing a successful log in using the public/private key pair. You should not be asked for a password.



```
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox:~/Desktop$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.13).
0 to upgrade, 0 to newly install, 0 to remove and 583 not to upgrade.
nsr@nsr-VirtualBox:~/Desktop$ ssh-keygen -t rsa
Command 'ssh-keygen' not found, did you mean:
  command 'ssh-keygen' from deb openssh-client (1:8.2p1-4ubuntu0.12)
Try: sudo apt install <deb name>
nsr@nsr-VirtualBox:~/Desktop$ ssh -l 192.168.120.255
usage: ssh [-46AaCfGgKkMNnqsitTVXxyY] [-B bind_interface]
[-b bind_address] [-c cipher_spec] [-D [bind_address]:port]
[-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
[-i identity_file] [-J [user@]host[:port]] [-L address]
[-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
[-Q query_option] [-R address] [-S ctl_path] [-W host:port]
[-w local_tun[:remote_tun]] destination [command]
```

- A wireshark screen dump showing the exchange of SSH messages generated as a result of your login.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
2	3.001022289	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
3	6.014691973	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
4	9.030833295	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
5	12.041706944	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
6	15.055501530	192.168.120.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
7	21.663468513	192.168.120.129	192.168.120.2	DNS	100	Standard query 0xe4b4 A connectivity-check.ubuntu.com OPT
8	21.945538074	192.168.120.2	192.168.120.129	DNS	276	Standard query response 0xe4b4 A connectivity-check.ubuntu.co...
9	21.946168763	192.168.120.129	185.125.190.48	TCP	74	33514 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
10	22.189005056	185.125.190.48	192.168.120.129	TCP	60	80 → 33514 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
11	22.189833250	192.168.120.129	185.125.190.48	TCP	54	33514 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
12	22.189936037	192.168.120.129	185.125.190.48	HTTP	141	GET / HTTP/1.1
13	22.190150349	185.125.190.48	192.168.120.129	TCP	60	80 → 33514 [ACK] Seq=1 Ack=88 Win=64240 Len=0
14	22.433570664	185.125.190.48	192.168.120.129	HTTP	243	HTTP/1.1 204 No Content
15	22.433695396	192.168.120.129	185.125.190.48	TCP	54	33514 → 80 [FIN, ACK] Seq=88 Ack=191 Win=64050 Len=0
16	22.433849664	185.125.190.48	192.168.120.129	TCP	60	80 → 33514 [ACK] Seq=191 Ack=89 Win=64239 Len=0

Frame 1: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface ens33, id 0

```
0000 01 00 5e 7f ff fa 00 50 56 c0 00 08 08 00 45 00  --A...P V...E.
0010 00 a5 a5 98 00 00 04 11 e8 0b c0 a8 78 01 ef ff  ....X...
0020 ff fa ea 31 07 6c 00 91 f2 6b 4d 2d 53 45 41 52  ...1...kM-SEAR
0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48  CH * HTT P/1.1..H
0040 6f 73 74 3a 20 32 33 39 2e 32 35 35 2e 32 35 35  ost: 239 .255.255
0050 2e 32 35 30 3a 31 39 30 30 0d 0a 53 54 3a 20 75  .250:190 0..ST: u
0060 72 6e 3a 73 63 68 65 6d 61 73 2d 75 70 6e 70 2d  rn:schem as-upnp-
0070 6f 72 67 3a 64 65 76 69 63 65 3a 49 6e 74 65 72  org:devi ce:Inter
0080 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 63 65  netGatew ayDevice
0090 3a 31 0d 0a 4d 61 6e 3a 20 22 73 73 64 70 3a 64  :1. Man: "ssdp:d
00a0 69 73 63 6f 76 65 72 22 0d 0a 4d 58 3a 20 33 0d  iscover" --MX: 3.
00b0 0a 0d 0a
```

- A short explanation in very broad terms of how the log in works.

You can log into the server without a password.