Week 1

Session 2

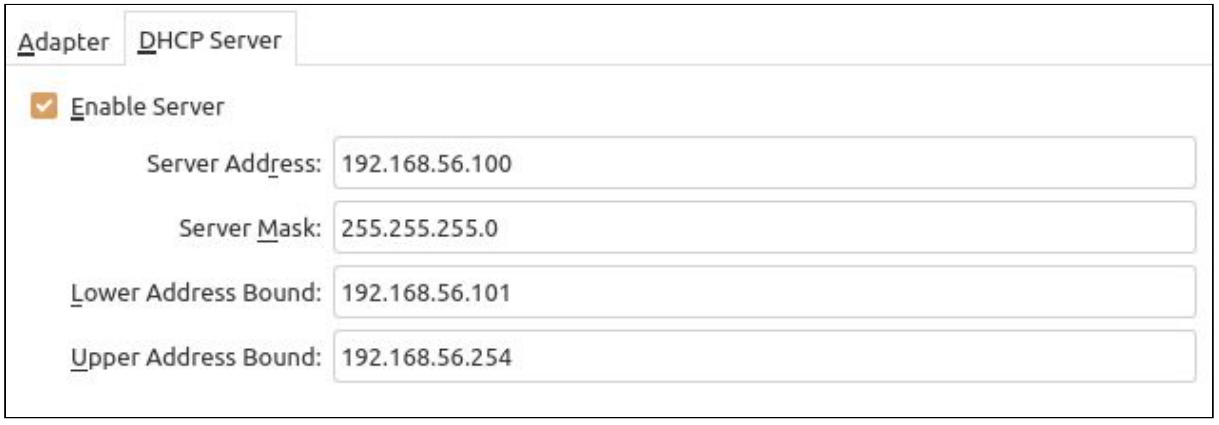
Daemons&Services

Olga Chernukhina

0. Preparation

I used VMs from the previous lab, so I disabled DHCP on server:
`sudo systemctl stop dnsmasq.service`

Also I enabled the default VirtualBox DHCP server:



1. Daemons

Install SSH Daemon on your server (not clients).

1. In the server terminal: `sudo apt-get install openssh-server`
2. Enable the ssh service `sudo systemctl enable ssh`

```
serv@server:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:7.6p1-4ubuntu0.3).
0 upgraded, 0 newly installed, 0 to remove and 24 not upgraded.
serv@server:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/system
Executing: /lib/systemd/systemd-sysv-install enable ssh
```

Demonstrate the usage of control scripts, systemctl/service for managing the daemon.

1. Start the ssh service `sudo systemctl start ssh`
   Check the result by `service --status-all`

```
[ + ]  rsyslog
[ - ]  screen-cleanup
[ + ]  ssh
[ + ]  udev
[ + ]  ufw
```

2. Restart the ssh service `sudo systemctl restart ssh`
   Check the result by `service --status-all`

```
[ + ]  rsyslog
[ - ]  screen-cleanup
[ + ]  ssh
[ + ]  udev
[ + ]  ufw
```

3. Reload the ssh service `sudo systemctl reload ssh`
   Check the result by `service --status-all`

```
[ + ]  rsyslog
[ - ]  screen-cleanup
[ + ]  ssh
[ + ]  udev
[ + ]  ufw
```

4. Stop the ssh service `sudo systemctl stop ssh`
   Check the result by `service --status-all`

```
[ + ]  rsyslog
[ - ]  screen-cleanup
[ - ]  ssh
[ + ]  udev
[ + ]  ufw
```

Check your active ports (netstate -plant / netstate -plante), and describe which network services are active and what they do.

```
serv@server:~$ sudo netstat -plant
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      834/
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3446/
tcp6       0      0 :::22                   :::*                    LISTEN      3446/
```

Port 22 is usually occupied by **ssh** service that allows *secure remote* login. We can check this by `sudo systemctl status ssh`

```
Oct 27 15:27:12 server sshd[3446]: Server listening on 0.0.0.0 port 22.
Oct 27 15:27:12 server sshd[3446]: Server listening on :: port 22.
```

Port 53 is used by **DNS** service which is responsible for resolving domain names to ip and vice versa.

2. SSH

2.1 Standard usage

Connect to your server from one of the clients.

I use `ssh <username>@<server-ip>`

```
ubuntu@ubuntu:/etc/network$ ssh serv@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established
.
ECDSA key fingerprint is SHA256:0YOyk7HSSCEZ1utkLVwvihXZVjYryjUKAhm409iHda4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ECDSA) to the list of known hosts.
serv@192.168.56.103's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-122-generic x86_64)
```

Create a directory inside the server with remote command execution

The ssh connection is open:

```
serv@server:~$ mkdir test_directory
```

Check that the directory has appeared on the server machine:

```
serv@server:~$ ls
test_directory
```

Install wireshark on the server, open it remotely from the client.

Installation is performed via `sudo apt-get install wireshark`

Check that it is installed successfully:

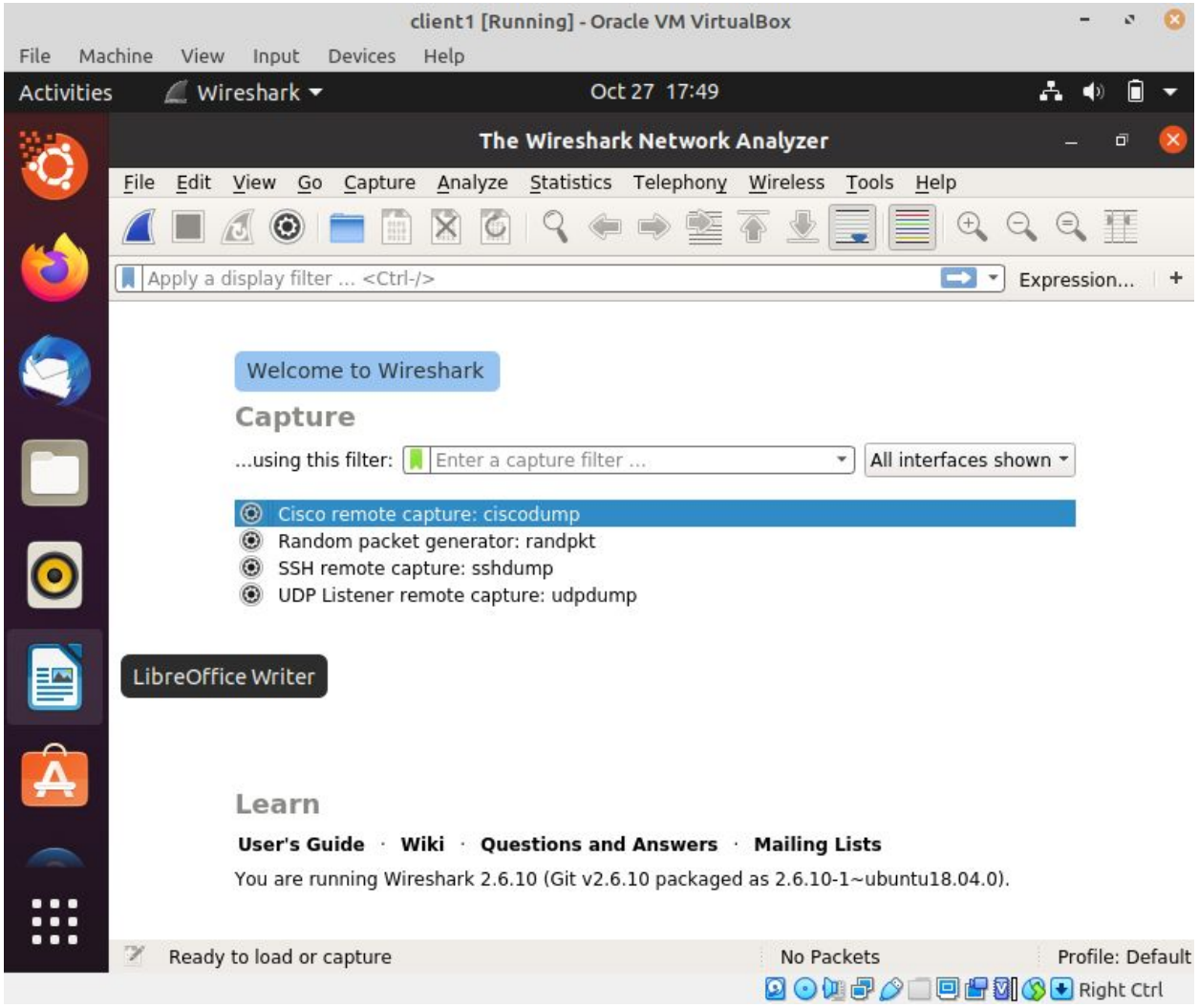
```
serv@server:~$ wireshark --version
Wireshark 2.6.10 (Git v2.6.10 packaged as 2.6.10-1~ubuntu18.04.0)
```

To allow graphical content to be transmitted to the client, it is necessary to make sure X11 forwarding is enabled:

In `/etc/ssh/sshd_config` file:

```
X11Forwarding yes
```

Then launching wireshark with -X flag: `ssh -X serv@192.168.56.103 wireshark`



2.2 Configuration

Change the default port to something else between 10000 and 65500.

In `/etc/ssh/sshd_config` file:

```
Port 33333
```

Make sure root is not allowed to ssh to the server.

In `/etc/ssh/sshd_config` file:

```
PermitRootLogin no
```

Disable X11 forwarding in the server.

In `/etc/ssh/sshd_config` file:

```
X11Forwarding no
```

After this - `sudo systemctl restart ssh`

Check and see if your configuration is working properly from the client.

Port configuration:

With default login:

```
ubuntu@ubuntu:/$ ssh serv@192.168.56.103
ssh: connect to host 192.168.56.103 port 22: Connection refused
```

With port specified:

```
ubuntu@ubuntu:/$ ssh -p 33333 serv@192.168.56.103
serv@192.168.56.103's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-122-generic x86_64)
```

Root login:

```
ubuntu@ubuntu:/$ ssh -p 33333 root@192.168.56.103
root@192.168.56.103's password:
Permission denied, please try again.
```

X11 forwarding:

```
ubuntu@ubuntu:/$ ssh -p 33333 -X  serv@192.168.56.103
serv@192.168.56.103's password:
X11 forwarding request failed on channel 0
```

3. SCP

Copy a file from your Desktop to the directory that you recently created. (Remember port has changed)

I created a file first:

```
ubuntu@ubuntu:~$ sudo touch test_file.txt
```

Then copied it with specifying the port and destination folder:

```
ubuntu@ubuntu:~$ scp -P 33333 test_file.txt serv@192.168.56.103:test_directory
serv@192.168.56.103's password:
test_file.txt                                 100%    0     0.0KB/s   00:00
```

Check that the file is on the server:

```
serv@server:~$ cd test_directory
serv@server:~/test_directory$ ls
test_file.txt
```

Copy a directory from your Desktop to the server

On the client machine I created a directory with a file inside:

```
ubuntu@ubuntu:~$ mkdir dir1
ubuntu@ubuntu:~$ cd dir1
ubuntu@ubuntu:~/dir1$ sudo touch file1.txt
```

Then copied it with specifying the port:

```
ubuntu@ubuntu:~$ scp -P 33333 -r dir1 serv@192.168.56.103:
serv@192.168.56.103's password:
file1.txt                                     100%    0     0.0KB/s   00:00
```

Check that everything is copied to the server:

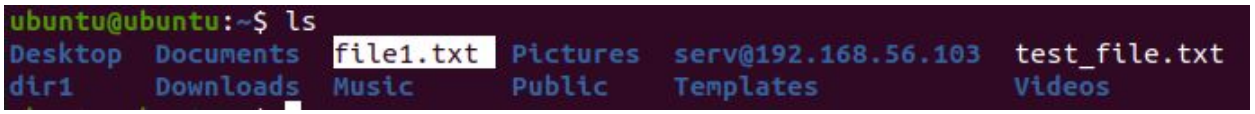
```
serv@server:~$ ls
dir1  test_directory
serv@server:~$ ls dir1
file1.txt
```

Copy a file from your server to your Desktop

Copy recently uploaded file to the current directory with specifying the port:

```
ubuntu@ubuntu:~$ scp -P 33333 serv@192.168.56.103:dir1/file1.txt @
serv@192.168.56.103's password:
file1.txt                                     100%    0     0.0KB/s   00:00
```

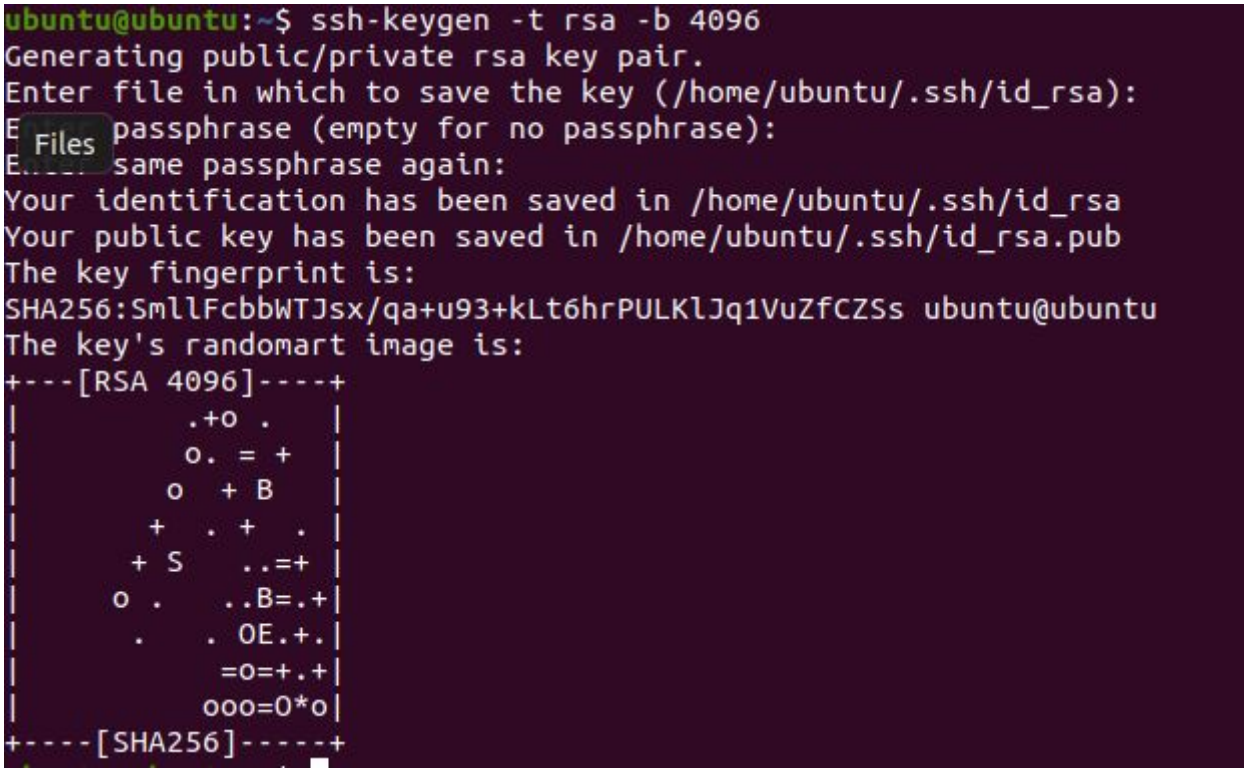
Check it's present on the client machine:



Bonus

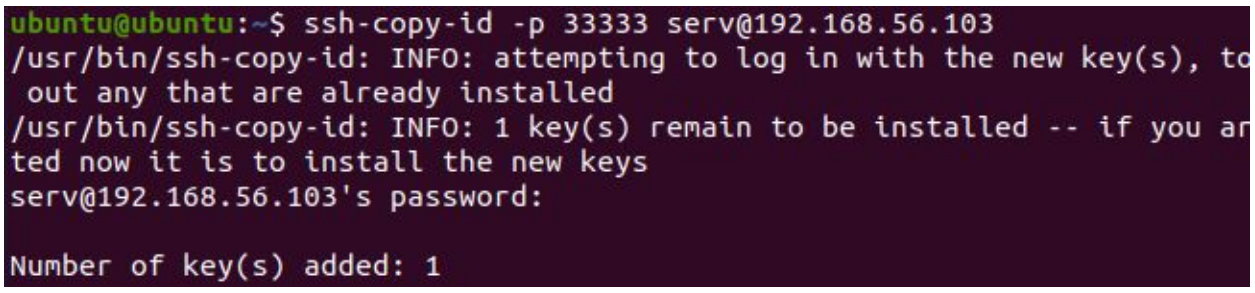
Use public-key cryptography to login to the server (password-less login)

Generate a pair of keys on the client machine:



Here I generated an RSA 4096 bits key.

Then transfer the public key to the server:



Login without the password:

```
ubuntu@ubuntu:~$ ssh -p 33333 serv@192.168.56.103
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-122-generic x86_64)
```

Create an SSH tunnel (any type) and use it.

Remote port forwarding:

In `/etc/ssh/sshd_config` file GatewayPorts should be set to yes.

The server will listen on port 44444 and tunnel all traffic coming to this port to the client machine on port 55555

```
ubuntu@ubuntu:~$ ssh -p 33333 -R 44444:127.0.0.1:55555 serv@192.168.56.103
```

Unfortunately, I was not able to open the browser in the 3rd VM to open server_ip:44444 and check it works, since the CPU was overloaded (I didn't even know it's possible to reach more than 100 percent):

Ideally, it should have given access to an internal service to the outside client, which is useful in cases where someone doesn't have public IP and still wants a secure connection to be possible for a remote node.

Resources:
https://www.cyberciti.biz/faq/ubuntu-linux-install-openssh-server/
https://askubuntu.com/questions/42444/the-list-of-running-daemons
https://superuser.com/questions/237057/how-do-i-make-ubuntu-server-get-ipv4-address/1262469
https://linux-notes.org/nastrojka-x11-forwarding-ispol-zuya-ssh-v-unix-linux/
https://linuxize.com/post/how-to-setup-passwordless-ssh-login/