# Laboratory 6: Firewalls and SSH in Linux using UFW and OpenSSH

In today’s lab, we will explore how to set a personal firewall in a Linux machine using a configuration tool called Uncomplicated Firewall (UFW). Moreover, we will learn how to establish a SSH connection between two VMs.

## 1. UFW

The Linux environment provides a packet filtering system called *netfilter*. This system is traditionally manipulated using the *iptables* suite of commands. Due to the difficulty of becoming proficient in managing these two features, several front-end solutions have been created to manage them. UFW is one of the most popular, since it offers both a framework to manage *netfilter* and simplified *iptables* commands.

By default, UFW blocks all incoming connections and allows all outbound connections once it is initiated. These default policies are defined in the /etc/default/ufw file and also in the files located in the /etc/ufw directory which end in .rules. To change them, we can use the command *sudo ufw <allow/deny> <policy>*. For instance, if you want to use your VM as a server and you want the world to access your website, you need to make sure the default TCP port (80) is open by issuing the command *sudo ufw allow 80/tcp*.

Table 1 shows a list of some of the different commands for UFW:

|  |  |
| --- | --- |
| Command | Specification |
| enable | Enables the firewall. |
| disable | Disables the firewall. |
| default ARG | Set default policy. |
| logging LEVEL | Set logging to LEVEL. |
| allow ARGS | Add allow rule. |
| deny ARGS | Add deny rule. |
| reject ARGS | Add reject rule. |
| limit ARGS | Add limit rule. |
| delete RULE|NUM | Delete a rule. |
| insert NUM RULE | Insert a RULE at a certain NUM. |
| reload | Reload firewall. |
| reset | Reset firewall |
| status | Show firewall status. |
| status numbered | Show firewall as numbered list of rules. |
| status verbose | Show verbose firewall status. |
| show ARG | Show firewall report. |
| version | Display version information. |
| app list | Shows the packages in your system. |
| App info PACKAGE | Info on the ports used by PACKAGE. |

TABLE 6.1. Options for the ufw front-end.

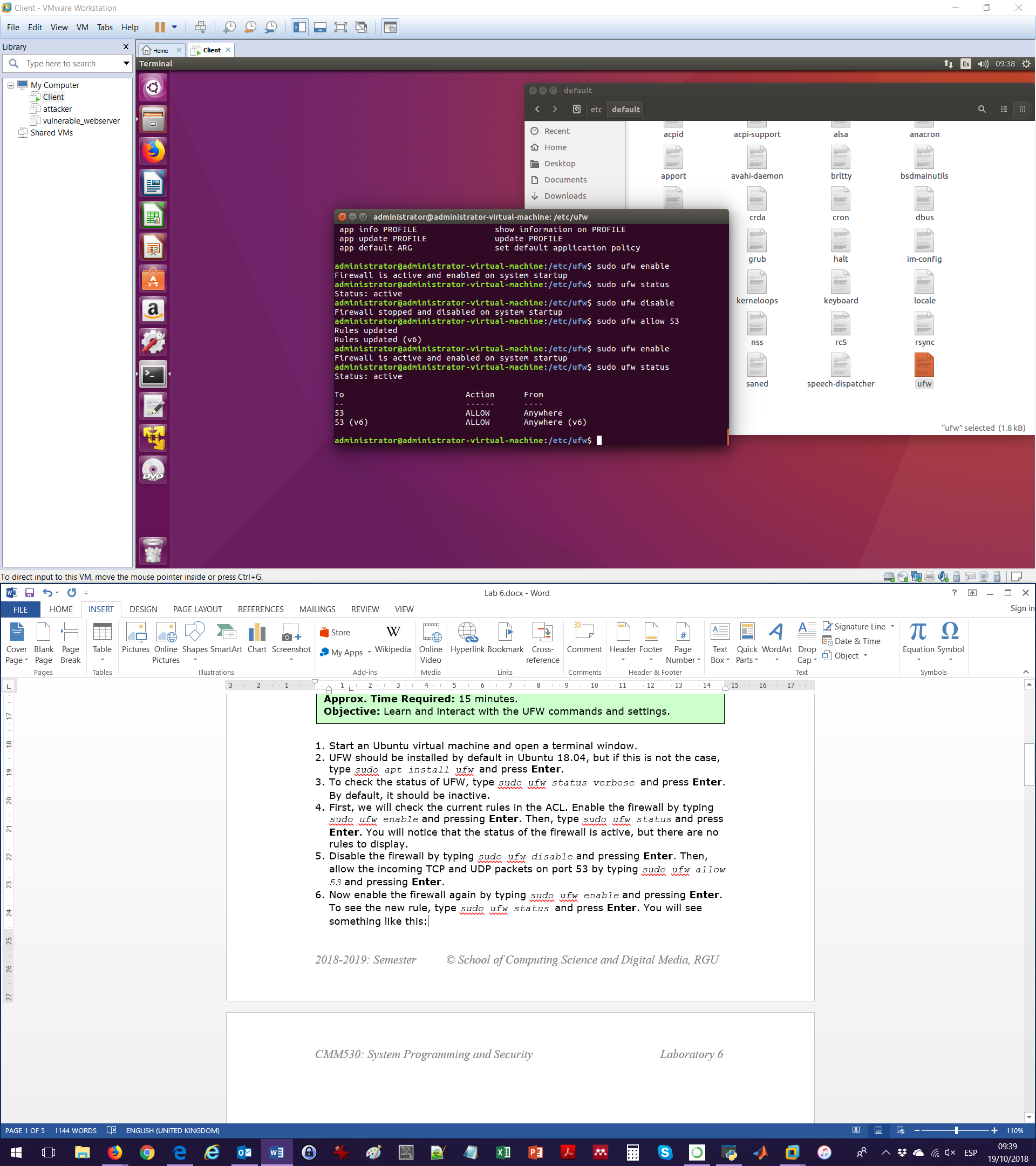
### Activity 6.1

### Installing and Configuring UFW

**Approx. Time Required:** 15 minutes.

**Objective:** Learn and interact with the UFW commands and settings.

1. Start an Ubuntu virtual machine and open a terminal window.
2. UFW should be installed by default in Ubuntu 18.04, but if this is not the case, type *sudo apt install ufw* and press **Enter**.
3. To check the status of UFW, type *sudo ufw status verbose* and press **Enter**. By default, it should be inactive.
4. First, we will check the current rules in the ACL. Enable the firewall by typing *sudo ufw enable* and pressing **Enter**. Then, type *sudo ufw status* and press **Enter**. You will notice that the status of the firewall is active, but there are no rules to display.
5. Disable the firewall by typing *sudo ufw disable* and pressing **Enter**. Then, allow the incoming TCP and UDP packets on port 53 by typing *sudo ufw allow 53* and pressing **Enter**.
6. Now enable the firewall again by typing *sudo ufw enable* and pressing **Enter**. To see the new rule, type *sudo ufw status* and press **Enter**. You will see something like this:



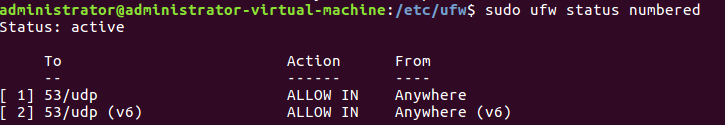
Notice that the rule has been enabled for both for IPv4 and IPv6.

In case that you don’t see the rules, you can use the command *sudo ufw reload*.

1. You can also specify which protocol you will enable for each port. To do so, first we will delete the existing rule. One option is to type *sudo ufw delete allow 53* and press **Enter.** Another option is to erase the rules by number. To see the number of each rule, type *sudo ufw status numbered* and press **Enter.** You will see that the IPv4 rule is number 1, and the IPv6 rule is number 2. In this case you will need to erase both rules individually by doing two times *sudo ufw delete 1* and pressing **Enter**.

You have to do *sudo ufw delete 1* two times instead of *sudo ufw delete 1* and *sudo ufw delete 2* since when you eliminate the first rule, then the second becomes the first one and thus there is no second rule to eliminate.

1. Disable the firewall, then type *sudo ufw allow 53/udp* and press **Enter.** Now check the status of the firewall and observe the difference.



1. Now we will deny the access for a specific IP to the VM. To do so:
   1. Check the IP address of your host machine (this can be done in the terminal using the *ipconfig* command in Windows or the *ifconfig* command in Unix-based systems.
   2. Then, check the IP address of your guest VM using *ifconfig*.
   3. Ping from the host to the guest VM by typing *ping \*\*IP\_GUEST\*\** and pressing **Enter** to verify that there is connectivity between both machines.
   4. In the guest VM, disable the firewall.
   5. Then, type *sudo ufw deny from \*\*IP\_HOST\*\** and press **Enter**.
   6. Enable the firewall.
   7. Go to the host machine, close and open the command prompt/terminal and then ping the VM by typing *ping \*\*IP\_GUEST\*\** and pressing **Enter**. Your request should now time out.
   8. If you ping from the guest to the host by typing *ping \*\*IP\_HOST\*\** and pressing **Enter**, you should get a response.

Another form to denying pings (from everyone) is by editing the /etc/ufw/before.rules file (using *sudo gedit* for instance) and changing ACCEPT to DROP in the five rules after the line that says *# ok icmp codes*.

1. Save a VM snapshot of this progress. Then, reset the firewall to its original settings by disabling the firewall, then typing *sudo ufw reset* and pressing **Enter**. Enable and check the status to verify.

## 2. SSH Connection

### Activity 6.2

### Creating a SSH connection

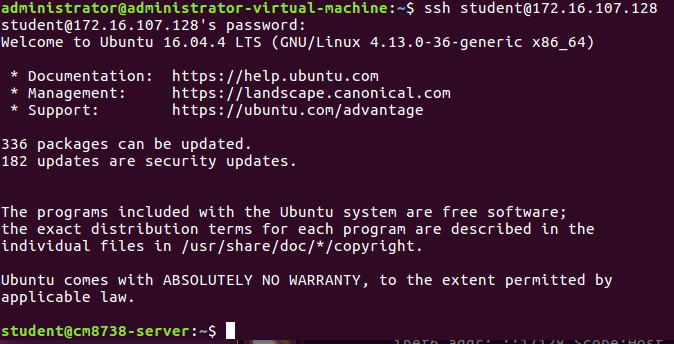
**Approx. Time Required:** 15 minutes.

**Objective:** Create a SSH Connection between two computers.

1. Start a “client” Ubuntu virtual machine and a “server” nested VM within the client.
2. On both VMs, install OpenSSH by typing *sudo apt-get install openssh-server* and pressing **Enter**.
3. In the server, check that UFW is installed.
4. In the server’s firewall, add a rule which allows incoming SSH connections from the client VM. To do so, type *sudo ufw allow from \*\*IP\_CLIENT\*\* to any port ssh* and press **Enter**. This will enable port 22.

If you need to set a SSH daemon (for instance in port 4422) then you need to specify this in the command by using, for instance, *sudo ufw allow from \*\*IP\_CLIENT\*\* to any port 4422*.

1. Check the IP of the server by using *ifconfig*.
2. To connect from the client to the server, in the client VM type *ssh username@ipadress* and press **Enter**. To terminate the connection, use *exit*.



If you are using another port other than 22 (for instance in port 4422) then you need to first open the SSH configuration file on the server (located in /etc/ssh/sshd\_config) by typing *sudo gedit /etc/ssh/sshd\_config* and pressing **Enter**. In line 5, change the port number to 4422. Moreover, you need to use the command *ssh username@ipadress –p4422*.

**References:**

<https://wiki.ubuntu.com/UncomplicatedFirewall>

<https://linuxize.com/post/how-to-setup-a-firewall-with-ufw-on-ubuntu-18-04/>

<https://www.vultr.com/docs/how-to-configure-ufw-firewall-on-ubuntu-14-04>

<https://linuxconfig.org/how-to-deny-icmp-ping-requests-on-ubuntu-18-04-bionic-beaver-linux>

<https://help.ubuntu.com/lts/serverguide/openssh-server.html.en>