

# ASSIGNMENT REPORT 1 - VMWARE VICTIM MACHINE

CENG 3544, COMPUTER AND NETWORK SECURITY

Ahmet Oral  
ahmetoral@posta.mu.edu.tr

Monday 7<sup>th</sup> June, 2021

## Abstract

In this assignment I installed a victim machine to be used for testing attack scenerios. The victim machines operating system is Ubuntu and it is installed as a virtual machine. I used Vmware Workstation and installed Ubuntu on it. This victim machine can be pinged by the host system and there are 3 network services installed. These services are; ssh, ftp and web. I successfully established an ssh connection between the host pc and victim machine. I also made ftp server secured by creating a certificate and configuring it for encrypting the traffic. Lastly I created the Web Server by using Apache Web Server

## 1 Introduction

In this lab assignment, my goal is to install a victim machine, make it operating and install certain network services. For the operating system of this machine, I will be using Ubuntu which is an open source Debian-based Linux distribution. To be able to run this machine virtually on my Windows OS, I will be using Vmware Workstation. Reason I am using Ubuntu on Vmware is because there are pre-build images available that can be used to set up the machine within minutes without extra configurations.

## 2 Assignments

Lab assignments consists of three main tasks:

- Installing a Virtual Victim Machine to Test Attacks
- Being Able to Ping Victim Machine From Host
- Installing Network Services

Detailed explanation of how each of these tasks solved and which steps are taken to solve them are shown below.

## 2.1 Installing a Virtual Victim Machine to Test Attacks

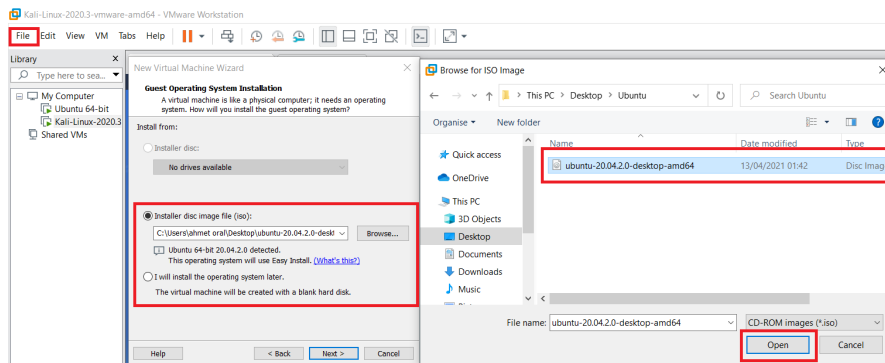


Figure 1: Creating a New Virtual Machine With Downloaded Iso File

I downloaded Ubuntu iso file from it's official website. To install it I clicked file(top left in the image) and clicked "Create new virtual machine" and selecte the iso file I downloaded. After configuring some basics and 10 minute installation, it was ready to use.

## 2.2 Being Able to Ping Victim Machine From Host

```
C:\Users\ahmet oral>ping 192.168.190.133

Pinging 192.168.190.133 with 32 bytes of data:
Reply from 192.168.190.133: bytes=32 time<1ms TTL=64
Reply from 192.168.190.133: bytes=32 time<1ms TTL=64
Reply from 192.168.190.133: bytes=32 time<1ms TTL=64
Reply from 192.168.190.133: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.190.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ahmet oral>
```

Host Pc is running on Windows OS. As seen in the image below, our victim machine can be pinged from the host machine without any problems.

## 2.3 Installing Network Services

### 2.3.1 Installing SSH

SSH (Secure Shell) is a network communication protocol that enables two computers to communicate and share data. Communication between the two computers is encrypted meaning that it is suitable for use on insecure networks.

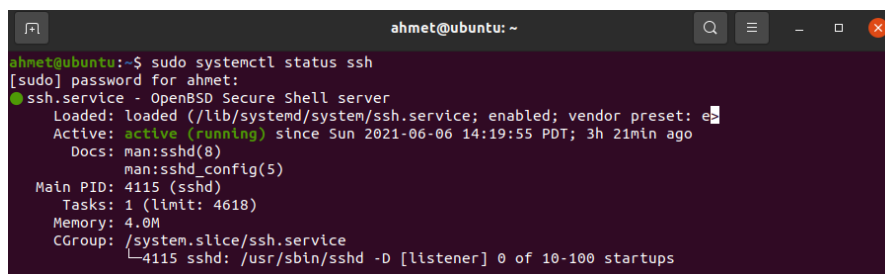
SSH is often used to login and perform operations on remote computers which I will demonstrate it by connecting to our victim machine from the host computer.

The SSH server is not installed by default on Ubuntu desktop systems but it can be easily installed by running the command below:

```
$ sudo apt install openssh-server
```

Once the installation is completed, the SSH service will start automatically. We can verify that the installation was successful and SSH service is running by typing the command below:

```
$ sudo systemctl status ssh
```

A terminal window titled 'ahmet@ubuntu: ~' showing the output of the command 'sudo systemctl status ssh'. The output indicates that the 'ssh.service' is loaded and active (running) since Sun 2021-06-06 14:19:55 PDT, 3h 21min ago. It also shows details about the service's configuration, main PID (4115), tasks, memory usage, and cgroup.

```
ahmet@ubuntu:~$ sudo systemctl status ssh
[sudo] password for ahmet:
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Sun 2021-06-06 14:19:55 PDT; 3h 21min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 4115 (sshd)
    Tasks: 1 (limit: 4618)
   Memory: 4.0M
   CGroup: /system.slice/ssh.service
           └─4115 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Figure 2: SSH Status check

As shown in the image above, SSH service is fully operational.

Ubuntu comes with a firewall configuration tool called UFW. This tool is enabled by default and we should make sure that SSH port open. Command to allow access:

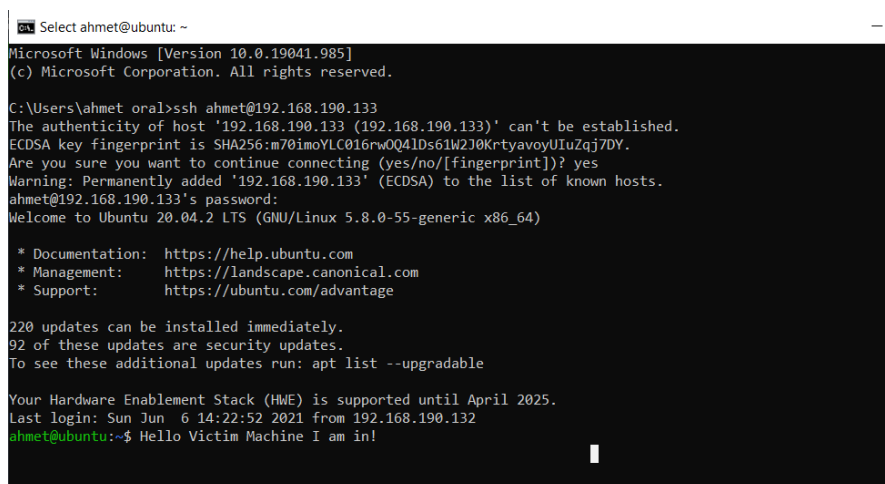
```
$ sudo ufw allow ssh
```

## Connecting to Our Victim Machine From the Host PC Using SSH

Now that SSH is installed and running on our Ubuntu system we can connect to it via SSH from any remote machine. To do this, I opened the cmd in my Windows and entered the commands below:

```
$ ssh ahmet@192.168.190.133
```

"ahmet" is the name of the victim machine and "192.168.190.133" is the Ip of that machine. After executing this line of code I need to enter password of victim machine for establishing connection. Result after I entered the password can be shown in the image below:

A screenshot of a Windows command prompt window. The title bar reads "Select ahmet@ubuntu: ~". The window content shows the execution of the command 'ssh ahmet@192.168.190.133'. It displays the SSH warning about the host's authenticity, the user's confirmation to proceed, and the password prompt. After the password is entered, the user is logged into the Ubuntu system as 'ahmet'. The prompt changes to 'ahmet@ubuntu:~\$'. The user then types 'Hello Victim Machine I am in!'. The background of the terminal window is black with white text. The Windows taskbar is visible at the top, showing the Start button and the taskbar title.

```
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ahmet oral>ssh ahmet@192.168.190.133
The authenticity of host '192.168.190.133 (192.168.190.133)' can't be established.
ECDSA key fingerprint is SHA256:m70imoVLC016rw0Q4lDs61W2J0KrtYavoyUIuZqj7DY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.190.133' (ECDSA) to the list of known hosts.
ahmet@192.168.190.133's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

220 updates can be installed immediately.
92 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Jun  6 14:22:52 2021 from 192.168.190.132
ahmet@ubuntu:~$ Hello Victim Machine I am in!
```

Figure 3: SSH Connection From Host Pc to Victim Machine

### 2.3.1 Installing FTP

The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP works by opening two connections that link the computers trying to communicate with each other. One connection is designated for the commands and replies that get sent between the two clients, and the other channel handles the transfer of data.

FTP was not designed to provide a secure tunnel through which information could travel. Hence, there is no encryption. If a hacker is able to intercept an FTP transmission, they would not have to deal through encryption to make the data usable. That's why I will configure encryption after I installed FTP server on the victim machine.

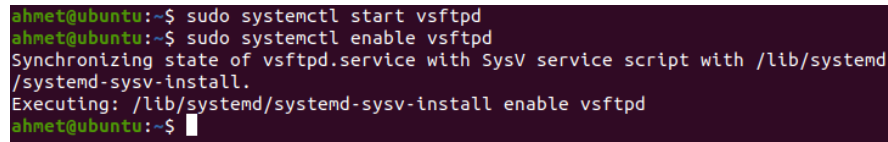
A common open-source FTP utility used in Ubuntu is vsftpd (Very Secure File Transport Protocol Daemon). I will use this tool because it is easy to use and I liked its name.

Command for installing vsftpd is:

```
$ sudo apt install vsftpd
```

After installation, we will launch the service and enable it at startup by run the commands below:

```
$ sudo systemctl start vsftpd
$ sudo systemctl enable vsftpd
```

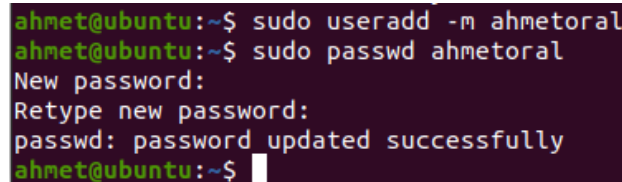


```
ahmet@ubuntu:~$ sudo systemctl start vsftpd
ahmet@ubuntu:~$ sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
ahmet@ubuntu:~$
```

Figure 4: Launching and Enabling the Service

Then we will create a new FTP user with the following commands:

```
$ sudo useradd -m "username"
$ sudo passwd "password"
```

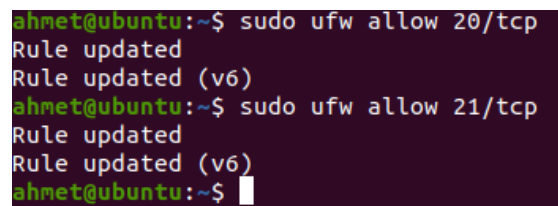


```
ahmet@ubuntu:~$ sudo useradd -m ahmetoral
ahmet@ubuntu:~$ sudo passwd ahmetoral
New password:
Retype new password:
passwd: password updated successfully
ahmet@ubuntu:~$
```

Figure 5: Creating a New FTP User

UFW will block FTP traffic by default. So like we did on the SSH, we should open Ports 20 and 21 for FTP traffic. Command to open these ports:

```
$ sudo ufw allow 20/tcp
$ sudo ufw allow 21/tcp
```



```
ahmet@ubuntu:~$ sudo ufw allow 20/tcp
Rule updated
Rule updated (v6)
ahmet@ubuntu:~$ sudo ufw allow 21/tcp
Rule updated
Rule updated (v6)
ahmet@ubuntu:~$
```

Figure 6: Opening Ports For FTP Traffic

Now that we installed and did the basic configurations, we can start to make our FTP server secured by encrypting the traffic. There are other ways to secure a FTP server such as limiting user access but in this assignment, I will only use encryption.

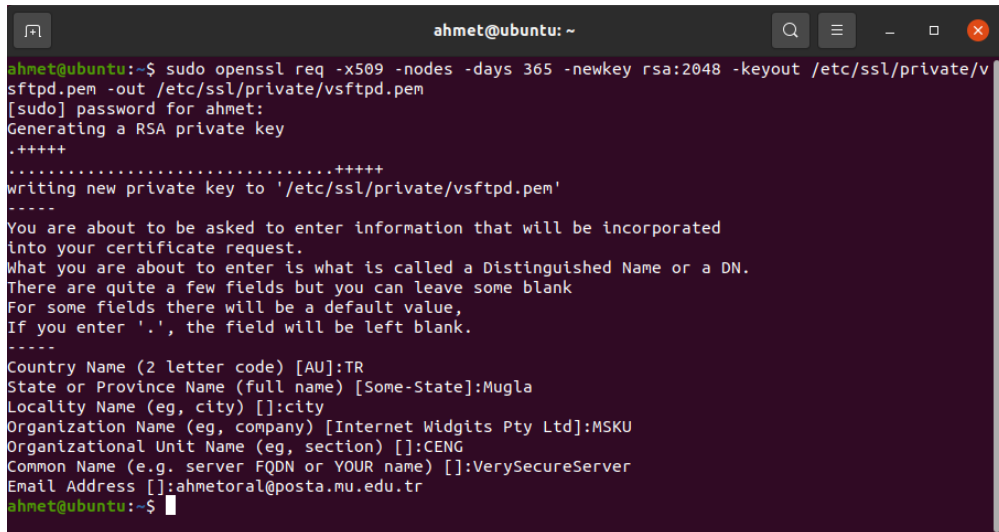
## Encrypting Traffic With FTPS

I will encrypt the traffic by using FTPS – File Transfer Protocol over SSL (Secure Socket Layer).

For this to work, users need to be set up with a shell account on the FTP server. This will add a layer of secure encryption to our FTP traffic.

Creating a new certificate with openssl:

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```



```
ahmet@ubuntu: ~  
ahmet@ubuntu:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem  
[sudo] password for ahmet:  
Generating a RSA private key  
.....  
writing new private key to '/etc/ssl/private/vsftpd.pem'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [AU]:TR  
State or Province Name (full name) [Some-State]:Mugla  
Locality Name (eg, city) []:city  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MSKU  
Organizational Unit Name (eg, section) []:CENG  
Common Name (e.g. server FQDN or YOUR name) []:VerySecureServer  
Email Address []:ahmetoral@posta.mu.edu.tr  
ahmet@ubuntu:~$
```

Figure 7: Creating a New Certificate With openssl

After creating the certificate I need to edit vsftpd.conf as shown below:

Lines below are to define the location of the SSL certificate and key file:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem  
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Next, we have to prevent anonymous users from using SSL, then force all non-anonymous logins to use a secure SSL connection for data transfer and to send the password during login:

```
allow_anon_ssl=NO  
force_local_data_ssl=YES  
force_local_logins_ssl=YES
```

We will set ssl\_enable's value to YES to activate the use of SSL, in addition, since TLS is more secure than SSL, we will restrict VSFTPD to employ TLS instead, using the ssl\_tlsv1\_2 option:

```
ssl_enable=YES  
ssl_tlsv1=YES  
ssl_sslv2=NO  
ssl_sslv3=NO
```

In addition, we can add the options below to boost up FTP server security. When option `require_ssl_reuse` is set to YES, then, all SSL data connections are required to exhibit SSL session reuse; proving that they know the same master secret as the control channel. Therefore, we have to turn it off.

```
require_ssl_reuse=NO
```

Again, we need to select which SSL ciphers VSFTPD will permit for encrypted SSL connections with the `ssl_ciphers` option. This can greatly limit efforts of attackers who try to force a particular cipher which they probably discovered vulnerabilities in:

```
ssl_ciphers=HIGH
```

Now, we set the port range (min and max port) of passive ports.

```
pasv_min_port=40000
```

```
pasv_max_port=50000
```

After these changes the config file is looking like this:

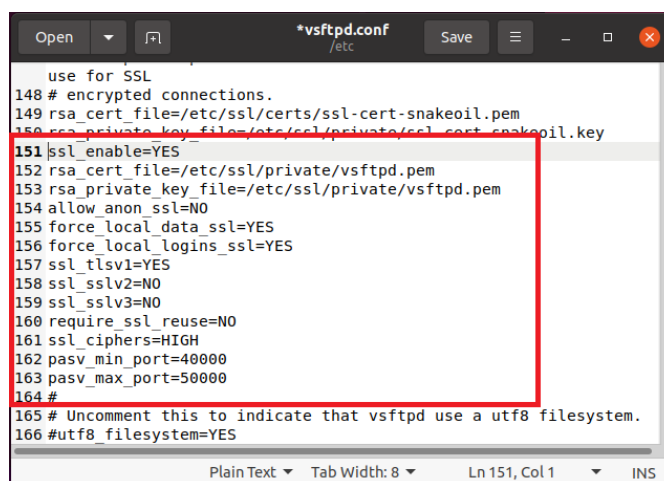
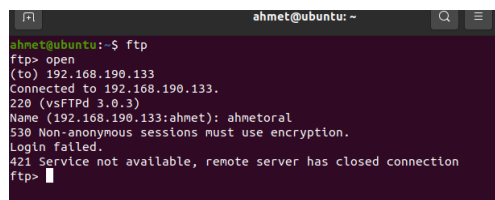


Figure 8: vsftpd.conf file

## Testing FTP server With SSL/TLS Connections

After doing the configurations above, we can test if VSFTPD is using SSL/TLS connections by attempting to use FTP from the command line:



From the image on left, we can see that there is an error informing us that VSFTPD can only allow user to login from clients that support encryption services.

The command line does not offer encryption services thus producing the error. So, to securely connect to the server, we need a FTP client that supports SSL/TLS connections.

## Securely Connecting to a FTP Server

FileZilla is a cross-platform FTP client that supports SSL/TLS connections by default. After installing and allowing the certificate I created, I can successfully connect to our FTP server as can be seen in the image below:

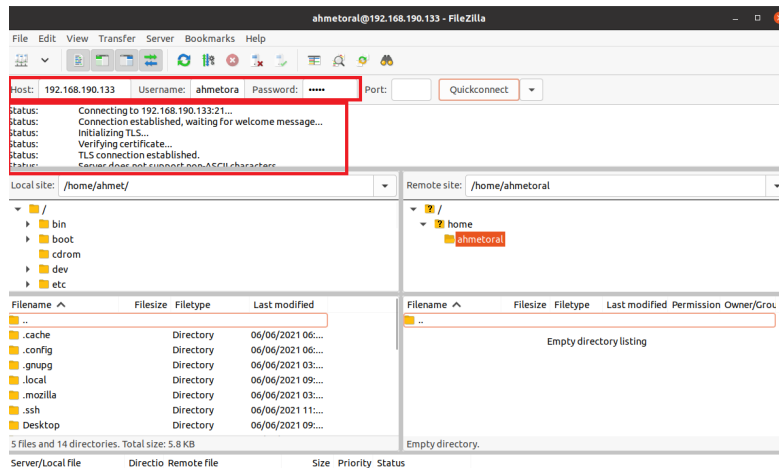


Figure 9: Secure Connection With FileZilla

## Installing WEB Server

Because the report is already too long I will only show the commands I used to install web server and the result.

I used Apache Web Server for installing the Web Server. I installed it by using the command below:

```
$ sudo apt-get install apache2
```

After installation and configuration I can successfully connect to it. Here is the welcome page for Apache:

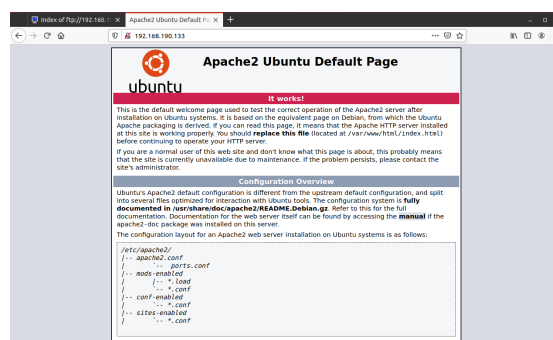


Figure 10: Apache Web Server Welcome Page



### **3 Conclusion**

In this assignment I learned to install and configure a victim virtual machine. I learned to install and configure network services such as; ssh, ftp and web. I was able successfully establish an ssh connection to the victim machine from my host pc. I also configured and secured my ftp server by encrypting it because by default ftp is very unsecure(transferred files are not encrypted) and easily hackable. So by creating a certificate and configuring it I made it a little bit more secured. Lastly I learned to install a Web Server using Apache Web Server. All in all, this assignment has taught me lots of valuable informations and features about network services and I had fun while doing it.