

# Report on Ethical Hacking as a proactive and adversarial approach to secure systems

Student name: **Jones Tobi Ogidiagba**

Student ID: **2056237**

University of Wolverhampton — May, 2022

## 1 Introduction

The goal of this assignment is to carry out and document a penetration testing phase as part of a practical 'offensive security' approach against a known network architecture with distinguishing characteristics and services in order to demonstrate understanding of offensive security. A series of processes and methods are carried out in order to setup the system, identify vulnerabilities in the system, and propose countermeasures to limit the impact of a network security breach.

## 2 Network setup

The scenario involves a Linux machine (Centos) configured as the server, a windows machine, configured as the client and a Linux based machine (Backtrack) configured as the attacker machine. All of the these machines are deployed in Virtual Box and configured in the same LAN network. The server machine (Centos) is equipped with a DHCP server that lease IP addresses to the other machines in the network. Thus, the Centos machine is be able to lease addresses to the windows client machine and the attacker machine on the local segment IP Address segment: 192.168.0.0/24, while the server is allocated a static IP of 192.168.0.1. The network setting on the VirtualBox is changed according to the following:

Address: 192.168.0.1; Subnet mask: 255.255.255.0; Default gateway address: 192.168.0.1.

### 3 DHCP configuration on the server

DHCP package is installed on the Centos machine if not already installed. The following command is run on the shell to find out which packages are already installed on the server:

Command Line

```
# rpm -qa | grep dhcp
```

For my case, DHCP package was not installed. The following command was used to install all packages related to the dhcp. on the server:

Command Line

```
# yum install dhcp*
```

In order to configure the DHCP service, I firstly edit the sample file by copying 'dhcpd.conf.sample' in into '/etc' directory to keep the original file safe in case something went wrong. Then 'dhcpd.conf' file is edited using vi editor. The file as changed is shown in Figure 1 below.

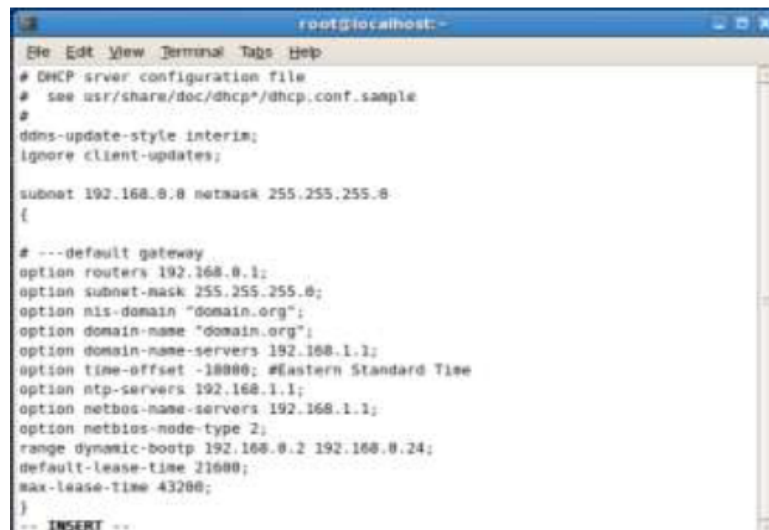


Figure 1: Editing dhcpd.conf file

The commands used to getting dhcp service running are summarised below:

```
Command Line

# cat /usr/share/doc/dhcp -3.0.5/dhcpd.conf.sample > /etc/dhcpd.conf
# vi /etc/dhcpd.conf
# cat /etc/dhcpd.conf
# chkconfig dhcpd on
# service dhcpd start
```

Figure 2 shows the service started and running.

```
[root@localhost ~]# service dhcpd start
Starting dhcpd: [ OK ]
[root@localhost ~]# service dhcpd stop
Shutting down dhcpd: [ OK ]
[root@localhost ~]# service dhcpd restart
Starting dhcpd: [ OK ]
[root@localhost ~]#
```

Figure 2: DHCP service started

## 4 Services Configuration

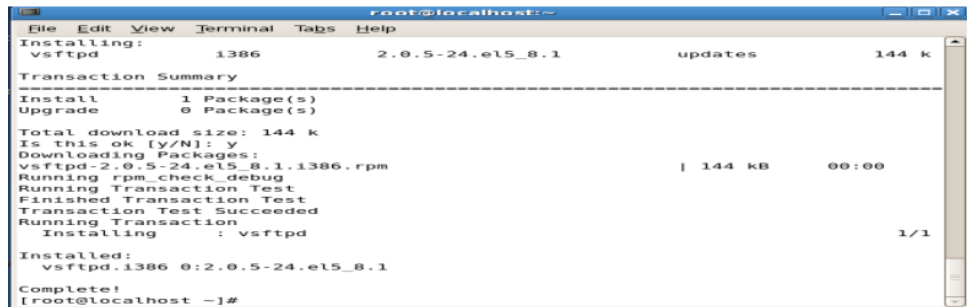
This section describes the services configured on the Centos machine (server) in order to demonstrate attack to the client machine from the attacker machine. In this assessment, I configured FTP and web services.

### 4.1 FTP service (allowing non-anonymous access)

There is always a need to verify that the desired application is already installed. This was accomplished by running the command `rpm -qa | grep vsftpd`. However, just before to this, the centos system was linked to the internet by switching from VirtualBox network to NAT and from static to dynamic IP address. This was followed by the installation of 'vsftpd' with the command `yum install vsftpd`, which was completed successfully as indicated in the following image;

The vsftpd is then configured to allow non-anonymous access. The configuration file was edited using `vi /etc/vsftpd.conf` to make the following changes:

- Anonymous\_enable=NO (for a non anonymous server, meaning user will require authentication to get access)
- Xferlog\_file=/var/log/xferlog (to specify the log file's name, because of the settings 'xferlog\_enabled' and xferlog\_std\_format' to 'YES')



```
root@localhost:~  
Installing:  
vsftpd      1386      2.0.5-24.el5_8.1      updates      144 k  
  
Transaction Summary  
-----  
Install      1 Package(s)  
Upgrade      0 Package(s)  
  
Total download size: 144 k  
Is this ok [y/N]: y  
Downloading Packages:  
vsftpd-2.0.5-24.el5_8.1.i386.rpm      | 144 kB      00:00  
Running rpm_check_debug  
Running Transaction Test  
Finished Transaction Test  
Transaction Test Succeeded  
Running Transaction  
Installing      : vsftpd  
  
Installed:  
vsftpd.i386 0:2.0.5-24.el5_8.1  
  
Complete!  
[root@localhost ~]#
```

Figure 3: Installing vsftpd

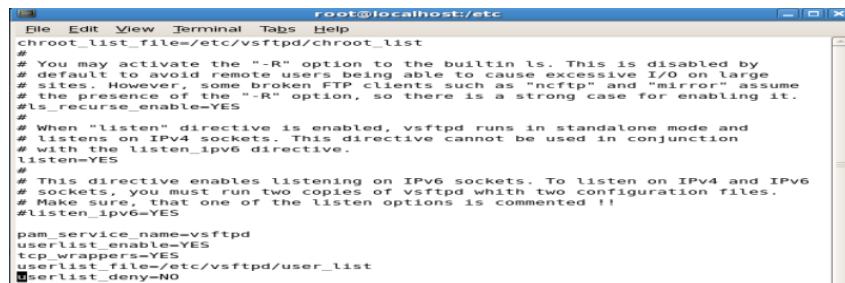
- Chroot\_enabled=yes (to enable the chroot)
- Chroot\_list\_file=/etc/vsftpd/chroot\_list (to provide the directory in which the users' file is stored)

Additionally, to add the users to the 'user\_list', the user list file is specified using the following command:

#### Command Line

```
# userlist_file=/etc/vsftpd/user_list  
# userlist_deny=NO
```

The configuration file is shown in the following diagram.



```
root@localhost:/etc  
File Edit View Terminal Tabs Help  
chroot_list_file=/etc/vsftpd/chroot_list  
#  
# You may activate the "-R" option to the builtin ls. This is disabled by  
# default to avoid remote users being able to cause excessive I/O on large  
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume  
# the presence of the "-R" option, so there is a strong case for enabling it.  
#ls_recurse_enable=YES  
#  
# When "listen" directive is enabled, vsftpd runs in standalone mode and  
# listens on IPv4 sockets. This directive cannot be used in conjunction  
# with the listen_ipv6 directive.  
listen=YES  
#  
# This directive enables listening on IPv6 sockets. To listen on IPv4 and IPv6  
# sockets, you must run two copies of vsftpd with two configuration files.  
# Make sure, that one of the listen options is commented !!  
#listen_ipv6=YES  
pam_service_name=vsftpd  
userlist_enable=YES  
tcp_wrappers=YES  
userlist_file=/etc/vsftpd/user_list  
#userlist_deny=NO
```

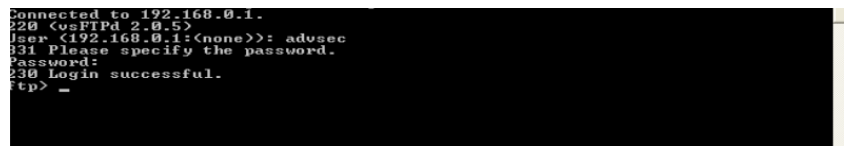
Figure 4: 'vsftpd' configuration file

The next thing was to add users. An example user 'advsec' and then add the user to the 'chroot' and the user list file. The following commands were used to achieve that.

#### Command Line

```
# useradd advsec  
# passwd advsec  
# vi /ect/vsftpd/user_list  
# vi /etc/vsftpd/chroot_list
```

To test the service is correctly set up, I logged in on the client's side (i.e the Windows machine) as shown in Figure 5:

A terminal window showing an FTP session. The text displayed is: 'Connected to 192.168.0.1.', '220 <vsFTPD 2.0.5>', 'User <192.168.0.1:<none>>: advsec', '331 Please specify the password.', 'Password:', '230 Login successful.', and 'ftp> \_'.

```
Connected to 192.168.0.1.
220 <vsFTPD 2.0.5>
User <192.168.0.1:<none>>: advsec
331 Please specify the password.
Password:
230 Login successful.
ftp> _
```

Figure 5: ftp access from client machine

## 4.2 Web services (with CGI support)

### 4.2.1 CGI

is an acronym for common gateway interface, which refers to a standard protocol for connecting external application software to the web. This is a software that has been written in such a way that CGI can accept and return data that complies with the CGI specifications. it might take any form. For instance, this could be C, Perl, Java, or Visual Basic. CGI is the most frequent and ideal method for web servers to interface dynamically with end users. Numerous HTML pages include forms. Applets, Java scripts, and ActiveX controls are all examples of programmes. These are referred to as client-side technologies, whereas the use of CGI is considered server-side.

CGI is commonly used on the web to construct dynamically produced pages. Additionally, it can be written in any language and has a very user-friendly interface. No specific library is required to develop or write the CGI. It must, however, always build a separate process for each HTTP request. Additionally, the database connection must be reopened for subsequent executions of the programme. Additionally, it is somewhat pricey in comparison. Using FASTCGI can help improve performance.

### 4.2.2 HTTP

HTTP stands for Hypertext Transfer Protocol and is a collection of rules for exchanging data such as text, graphics, pictures, sound, video, and multimedia files across the internet. When a web user begins to utilise a web browser, the user is inadvertently utilising HTTP. Which is built on top of TCP/IP (the internet's foundational protocol) When web browsers communicate with web servers, they use the HTTP protocol as well.

In order to get HTTP service running, the Apache HTTP server is configured on the server side (i.e Centos). HTTPD: HTTP is firstly configured with CGI. The HTTP Daemon (HTTPD) was installed and then configured using the following commands:

#### Command Line

```
# yum -y install httpd
# rm -f /etc/httpd/conf.d/welcome.conf
# rm -f /var/www/error/noindex.html
# ln -s /usr/bin/perl /usr/local/bin/perl
```

Note that, the default error page is removed and a link is created for Perl. The options in the httpd configuration file is configured as follows:

```
ServerAdmin root@tobi.org
// the name can be edited according to the domain name //
ServerName eh.tobi.org:80
//edit it according to he domain name server// Go to options and FollowSym-
    Links ExecCGI
// to enable CGI //
AllowOverride All
AllowOverride All DirectoryIndex index.html index.cgi index.php
ServerSignature Off
AddDefaultCharset UTF-8
AddHandler cgi-script .cgi .pl
```

And then the httpd service was started using `/etc/rc.d/init.d/httpd start`. And to test HTTP from the client side, I created a CGI page by using HTTP and `vi /var/www/html/index.html` and made the following changes:

```
<html>
<body>
<div style="width: 100
tobi Page
</div>
</body>
</html>
// width, size, font, weight, text can be assigned in their own desire//
```

On the browser, specifically the Internet Explorer on the Windows machine, type the DNS 'tobi.org' to display the HTTP page.

## 5 Attacks demonstration

The services that I configured previously must be attacked from Backtrack (Linux based attacker machine), which requires that all machines be connected to the same network. The attacks were carried out in the following manner:

### 5.0.1 Launching attacks to the server

Firstly, I launched 'ARMITAGE', a metasploit framework tool for exploitation, after logging into the backtrack. This was accomplished through clicking: Backtrack => Exploitation Tools => Network Exploitation Tools => Metasploit => Armitage. Secondly, after opening Armitage, the tabs were used to scan the network for hosts as follows: Quickscan => Hosts => nmapscan. Using the network address 192.168.0.1 and selecting scan at this point. The findings of the scan are depicted in the figure below.

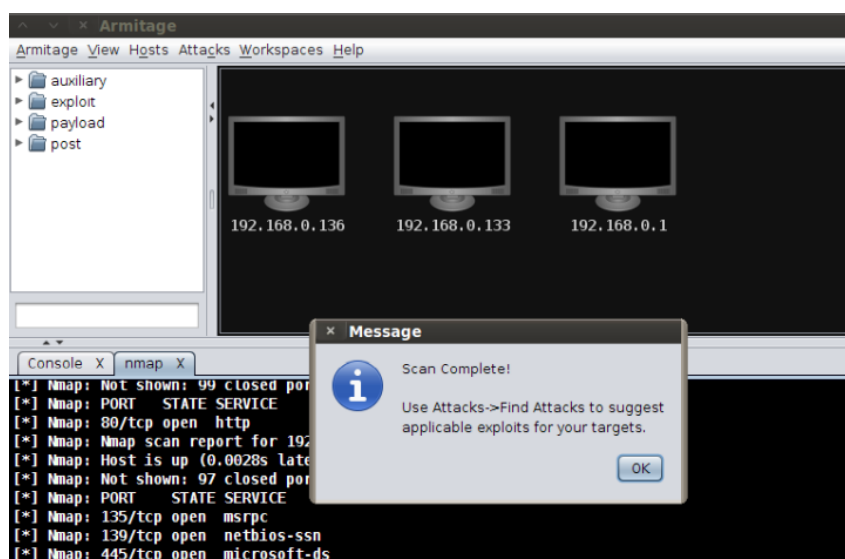


Figure 6:

In order to get detailed information on the operating system of the machines on the virtual network, the following command was used on 'ARMITAGE' console.

#### Command Line

```
$-db_nmap -st -pm -ts -o -open 192.168.0.0/24
```

The following shows the detailed output with information on specific operating system of the machines on the network.

The server is exploited further for the kind services running on it, which can be a pointer to the attacker. The following screenshot shows an example service scan on my Centos machine.

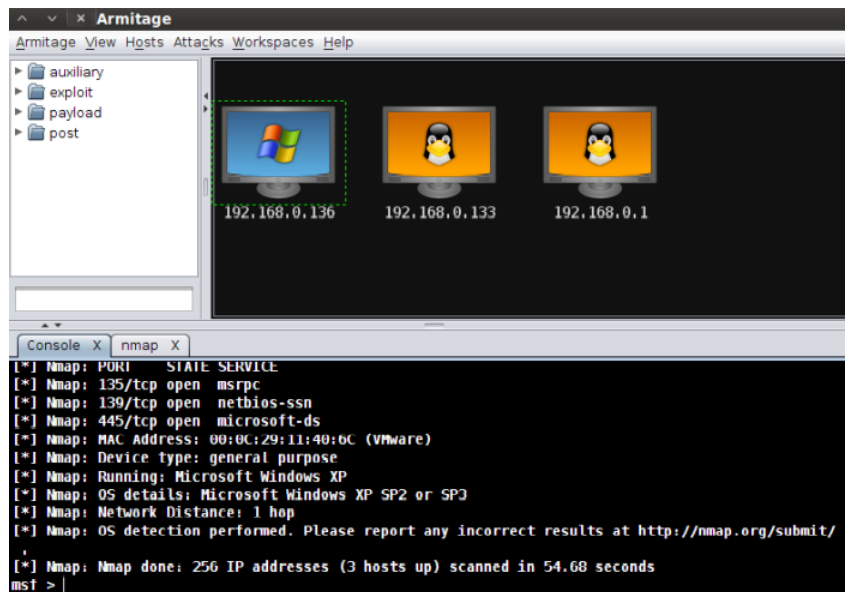


Figure 7:

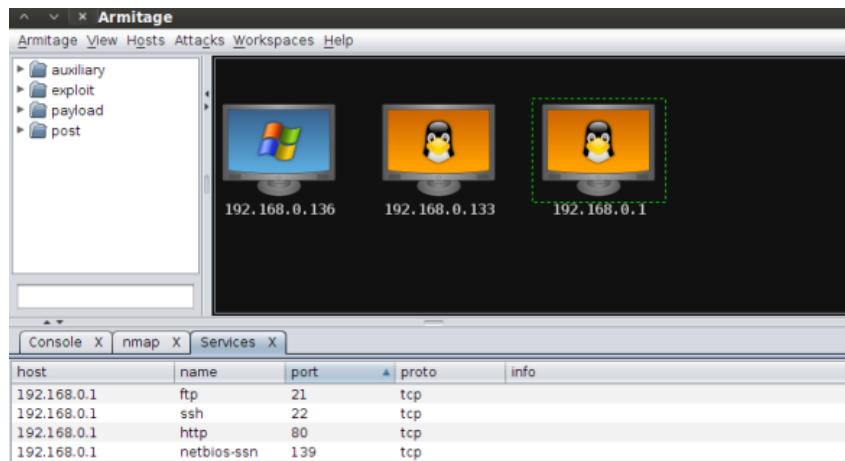


Figure 8:

Consequently, there is now full access to the attacked systems, files can be created or deleted from the attacker machine. I can now look for all the users and their passwords using the services on the victim machine. `-sessions -v` was used to achieve that. The figure below shows the output of the command.

In order to crack the users' passwords, a particular session was selected. To achieve this, `-sessions -i 1` command was used and the output is shown in the figure below.

Additionally, to show the users using the services `-cat /etc/passwd` was used on the console which reveals the following.



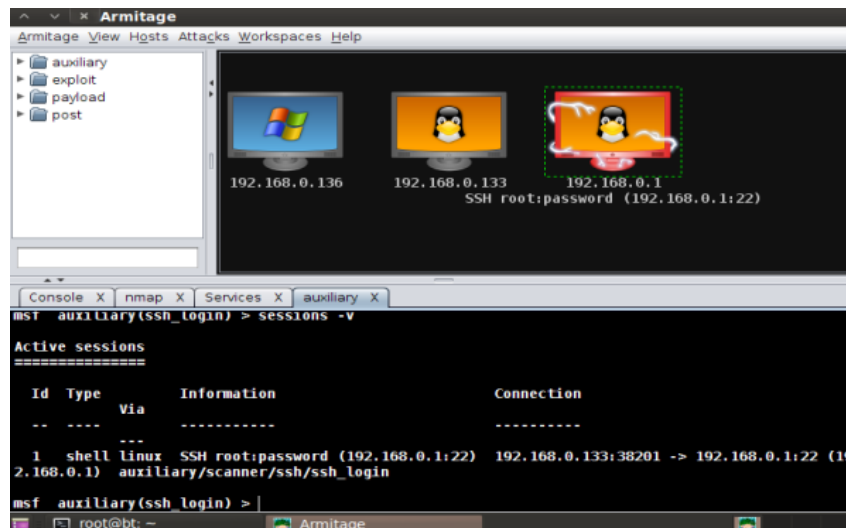


Figure 9:

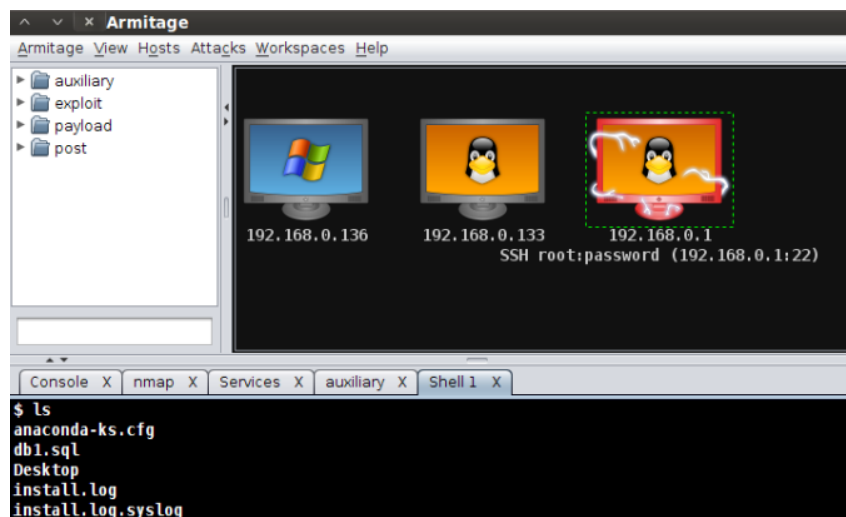


Figure 10:

## 6 Countermeasures (System hardening)

### 6.1 Defence Mechanism in LINUX

The following security measures have been implemented to guard against server attacks: To begin, we entered into the server through SSH and executed the following commands, which rendered the host machine inaccessible to the attacker (whose IP address had already been filtered out). The following are the commands:

#### Command Line

```
# iptables -L -n
# iptables -I INPUT -s 192.168.0.138 -j DROP
```

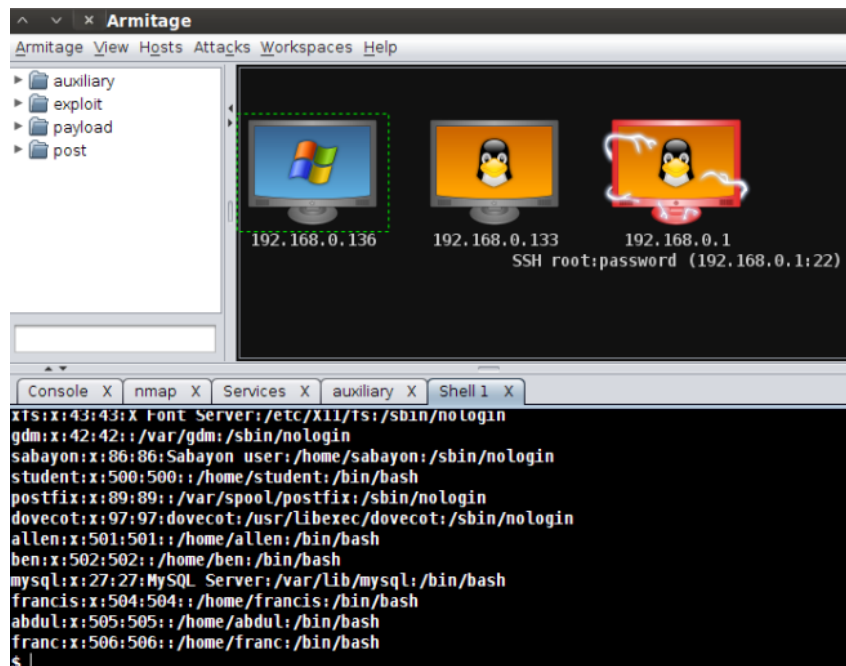


Figure 11:

Where the specified IP address is the attacker's address, as determined by the system administrator. Attempt(s) to breach the system at this time were unsuccessful, as evidenced by the screenshot below.

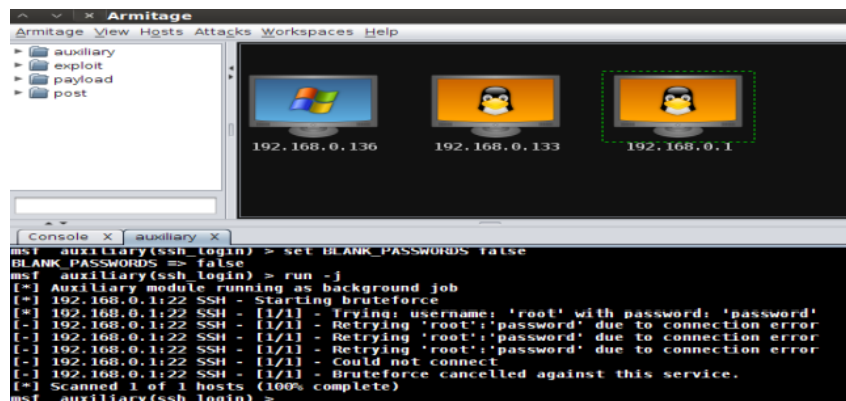


Figure 12:

## 6.2 Defence Mechanism in Windows

To safeguard the Windows computer from malicious activity, I modified the hosts file found at Start>My Computer>Local disk C:>Windows>System32>Drivers>Etc>Hosts. Right clicking the hosts and editing it by inserting the attacking machine's IP address. Additionally, the range of IP addresses that are deemed to be blocked can be included. The figure below illustrates the 'hosts' file.

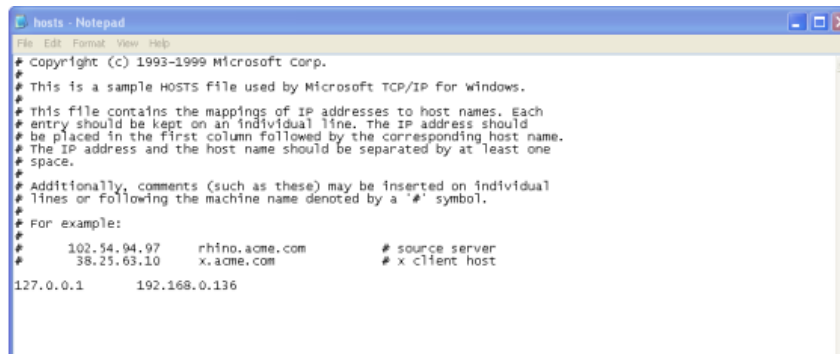


Figure 13:

After modifying the 'hosts' file in Windows, I restarted my computer and returned to Backtrack to re-initiate the attack. However, the Windows file is preventing me from breaking in, since a notice appears stating that the connection timed out, as shown below.

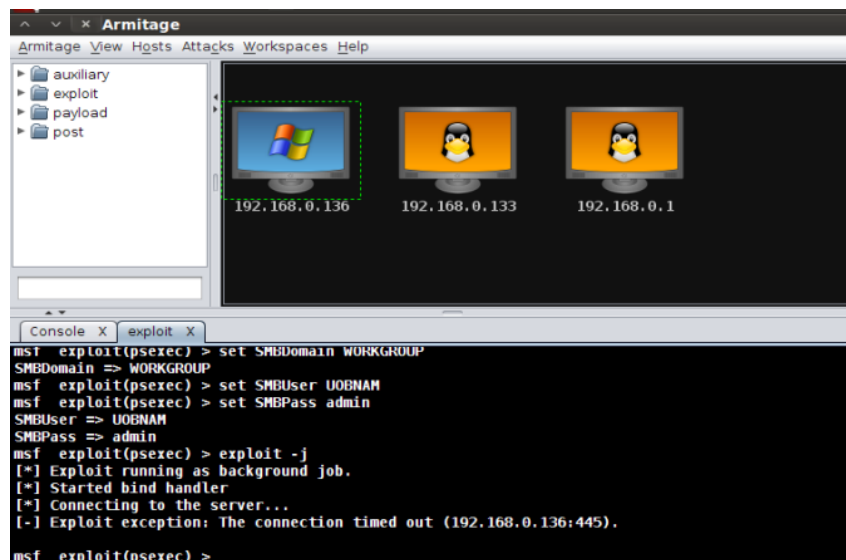


Figure 14:

## References

- James Marshall (2007). HTTP Made Really Easy: A Practical Guide to Writing Clients and Servers. Available at: < <http://www.jmarshall.com/easy/http/> >. [Accessed May 2022].
- David Medinets (n.d.). Perl By Example (pdf). Available at: < <http://affy.blogspot.co.uk/p5be/ch19.htm> >. [May 2022].
- S. Wale, Linux Administration: A beginner's Guide, 5th ed., New York: McGraw-Hill Companies, 2009.