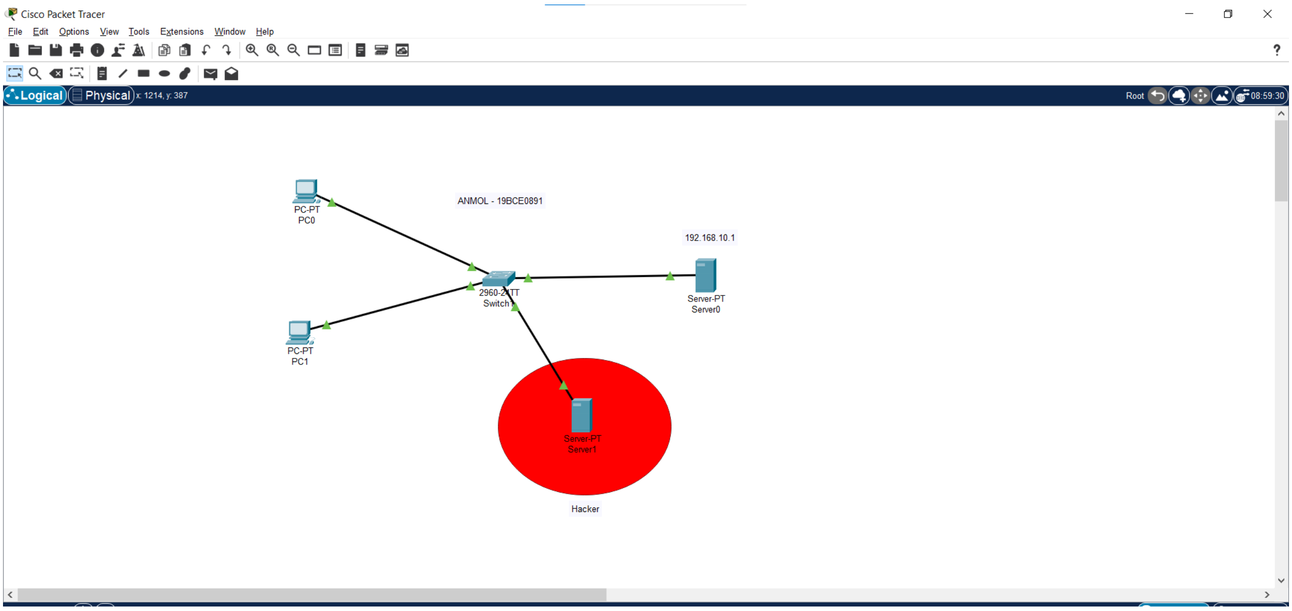
NAME – ANMOL
REG. NO. - 19BCE0891

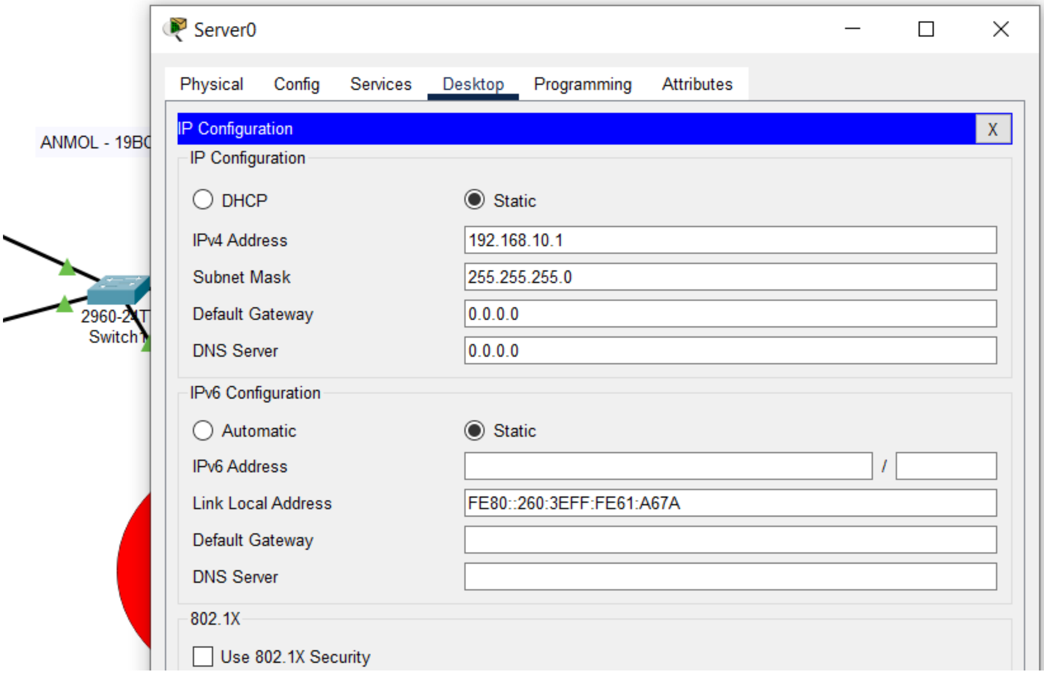
DIGITAL ASSIGNMENT – 6

#6a - DCHP SNOOPING

NETWORK TOPOLOGY



Server0 configuration



IP Configuration

IP Configuration

○ DHCP ● Static

IPv4 Address	192.168.10.1
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

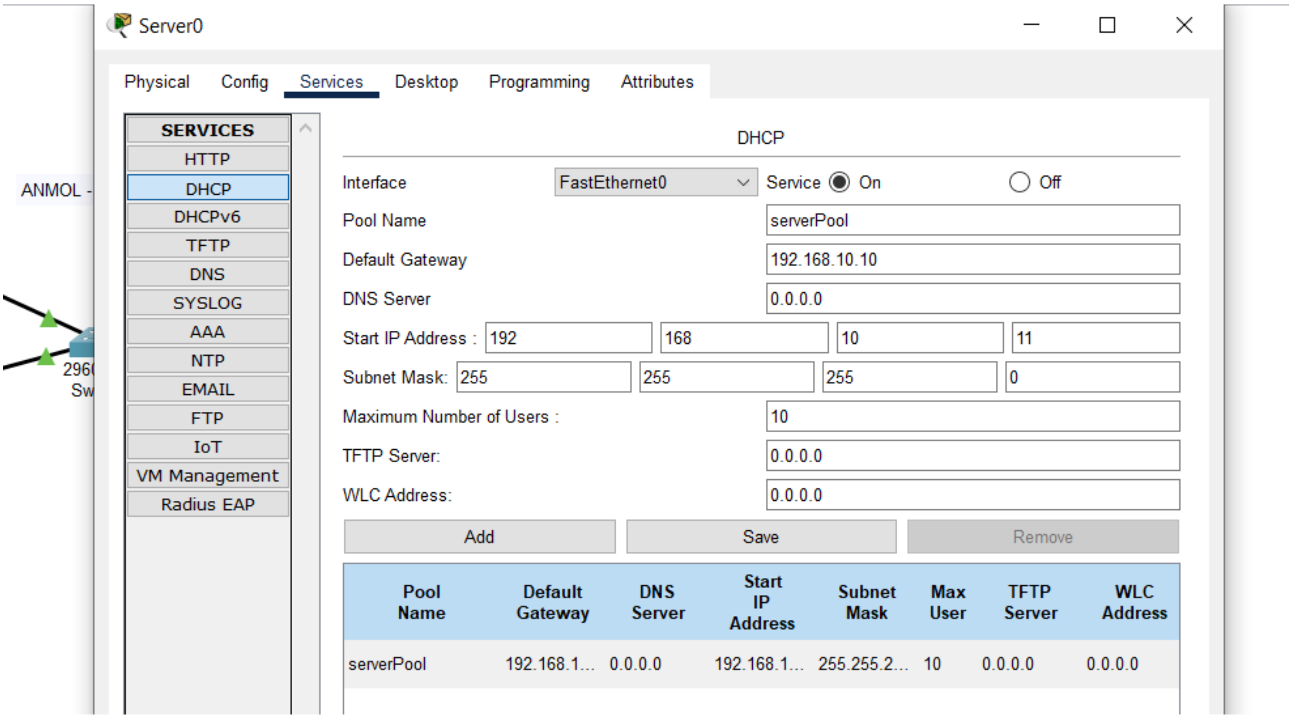
IPv6 Configuration

○ Automatic ● Static

IPv6 Address	/
Link Local Address	FE80::260:3EFF:FE61:A67A
Default Gateway	
DNS Server	

802.1X

□ Use 802.1X Security



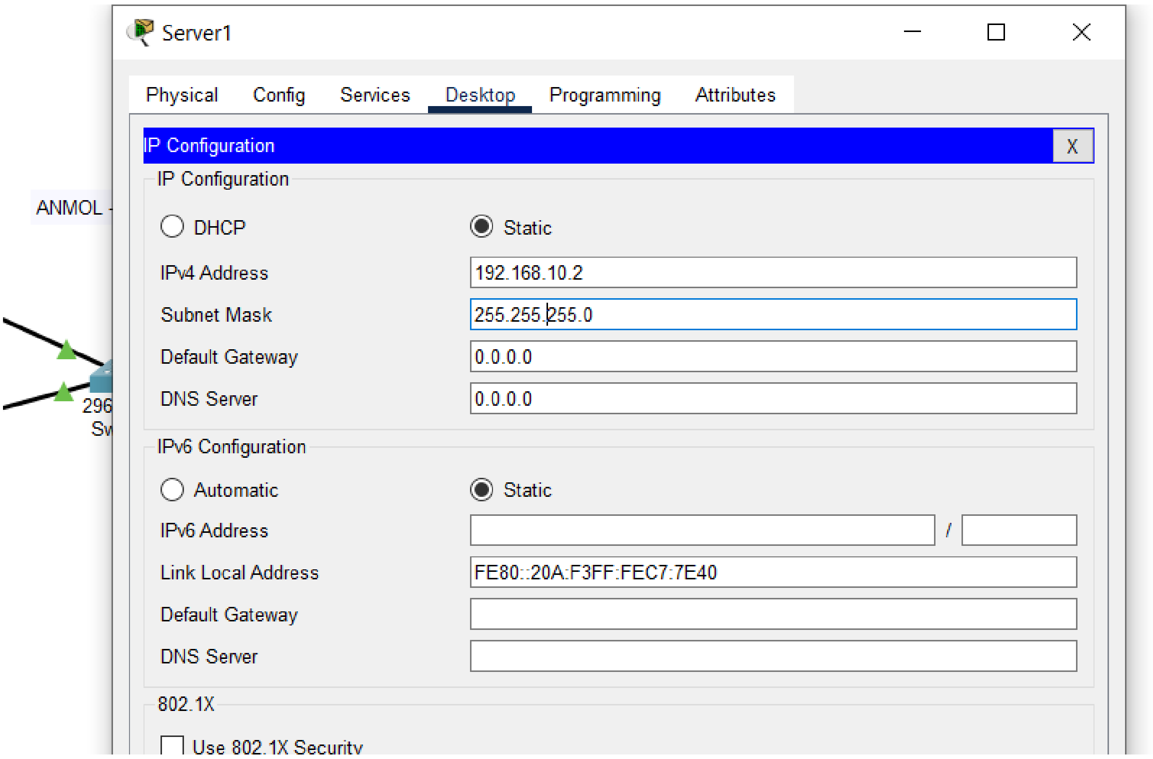
DHCP

Interface: FastEthernet0 Service ● On ○ Off

Pool Name	serverPool
Default Gateway	192.168.10.10
DNS Server	0.0.0.0
Start IP Address :	192 . 168 . 10 . 11
Subnet Mask:	255 . 255 . 255 . 0
Maximum Number of Users :	10
TFTP Server:	0.0.0.0
WLC Address:	0.0.0.0

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.168.1...	0.0.0.0	192.168.1...	255.255.2...	10	0.0.0.0	0.0.0.0

Server1 configuration (hacker)



IP Configuration

○ DHCP ● Static

IPv4 Address	192.168.10.2
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
DNS Server	0.0.0.0

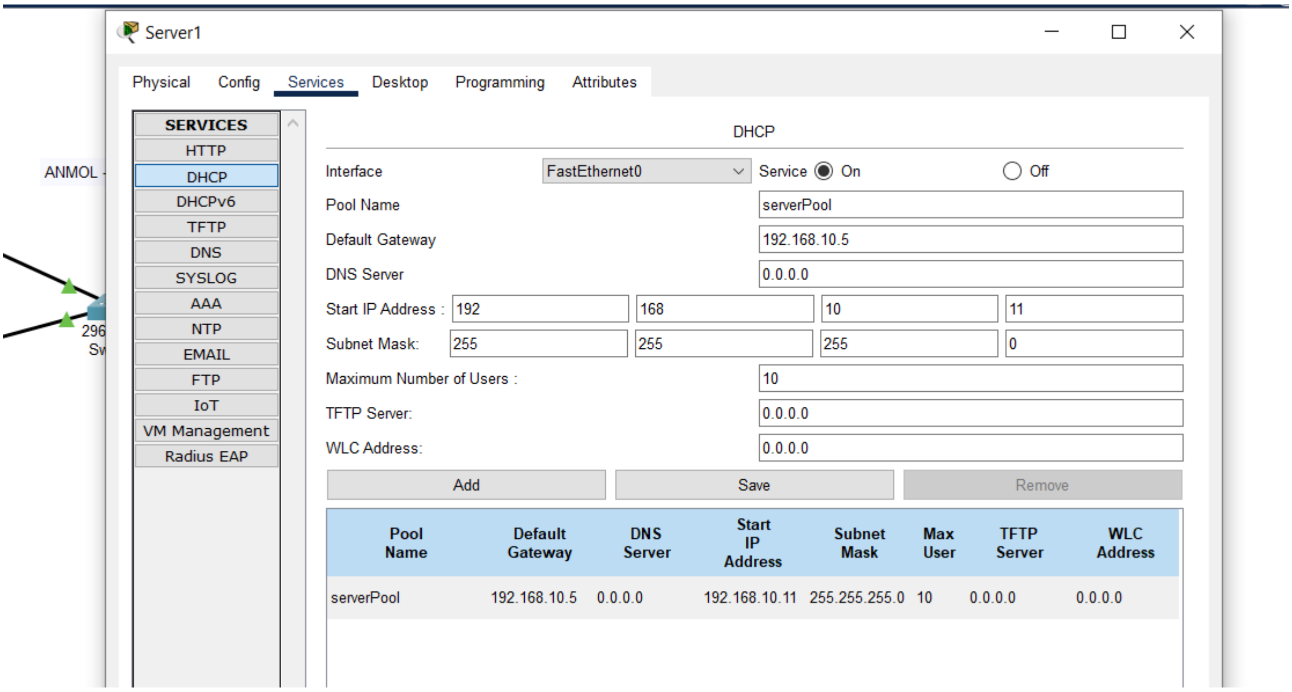
IPv6 Configuration

○ Automatic ● Static

IPv6 Address	/
Link Local Address	FE80::20A:F3FF:FEC7:7E40
Default Gateway	
DNS Server	

802.1X

□ Use 802.1X Security



Server1

Physical | Config | Services | Desktop | Programming | Attributes

DHCP

Interface: FastEthernet0 | Service ● On ○ Off

Pool Name: serverPool

Default Gateway: 192.168.10.5

DNS Server: 0.0.0.0

Start IP Address : 192 | 168 | 10 | 11

Subnet Mask: 255 | 255 | 255 | 0

Maximum Number of Users : 10

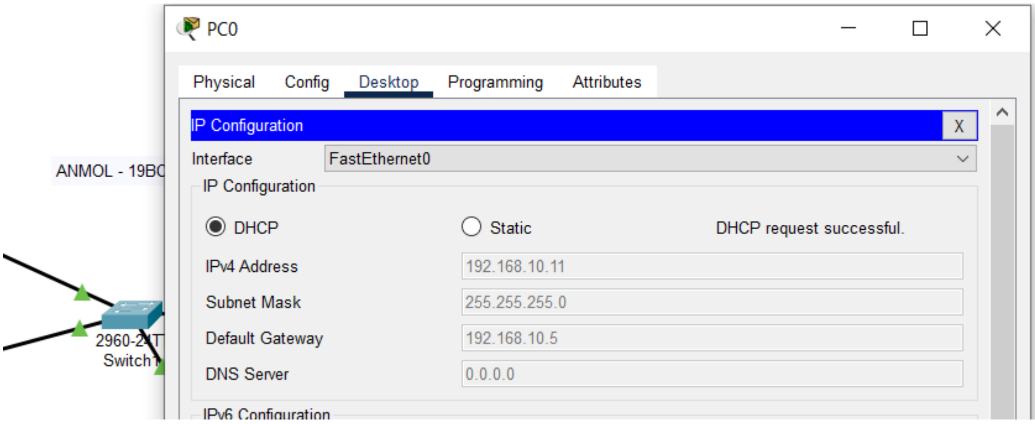
TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Add | Save | Remove

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.168.10.5	0.0.0.0	192.168.10.11	255.255.255.0	10	0.0.0.0	0.0.0.0

Before snooping PC's connect with hacker server



PC0

Physical | Config | Desktop | Programming | Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration

● DHCP ○ Static DHCP request successful.

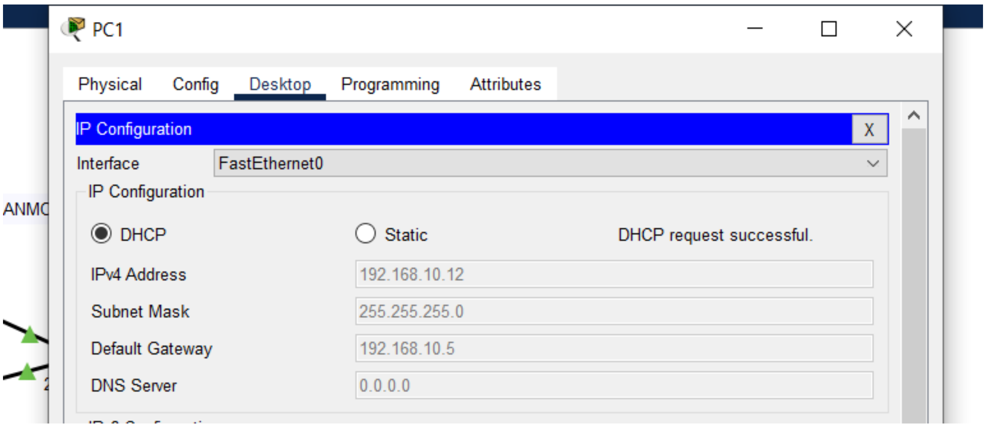
IPv4 Address: 192.168.10.11

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.5

DNS Server: 0.0.0.0

IPv6 Configuration



PC1

Physical | Config | Desktop | Programming | Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration

● DHCP ○ Static DHCP request successful.

IPv4 Address: 192.168.10.12

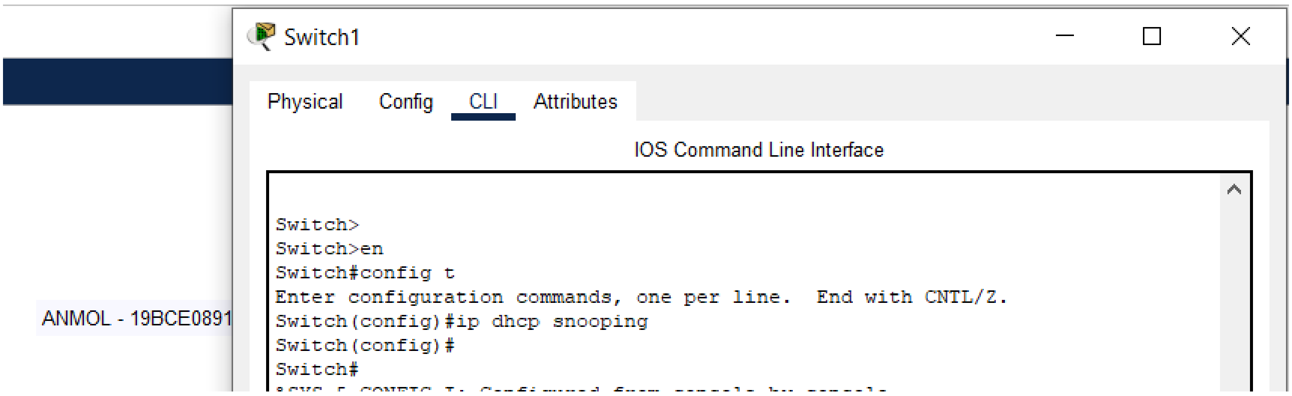
Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.5

DNS Server: 0.0.0.0

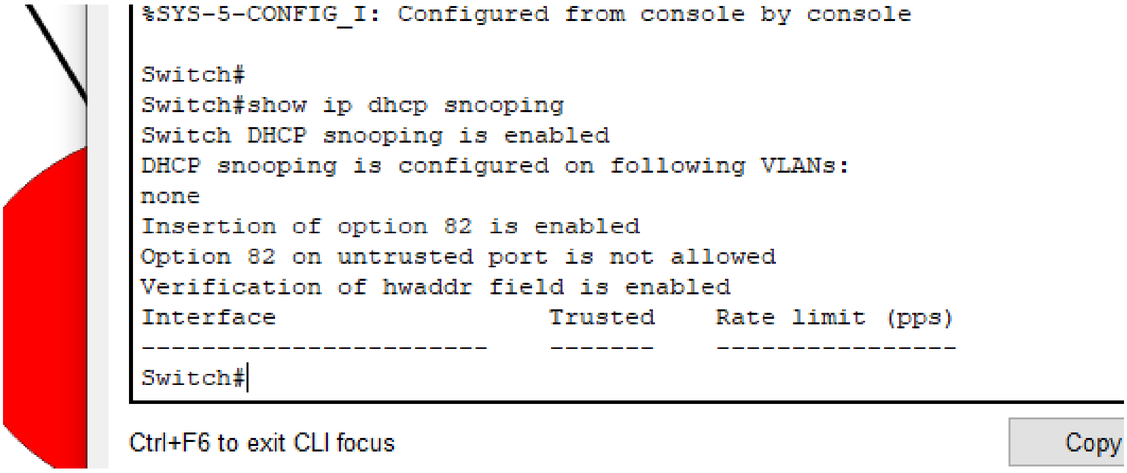
Commands to untrust all

Switch>
Switch>en
Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#ip dhcp snooping
Switch(config)#



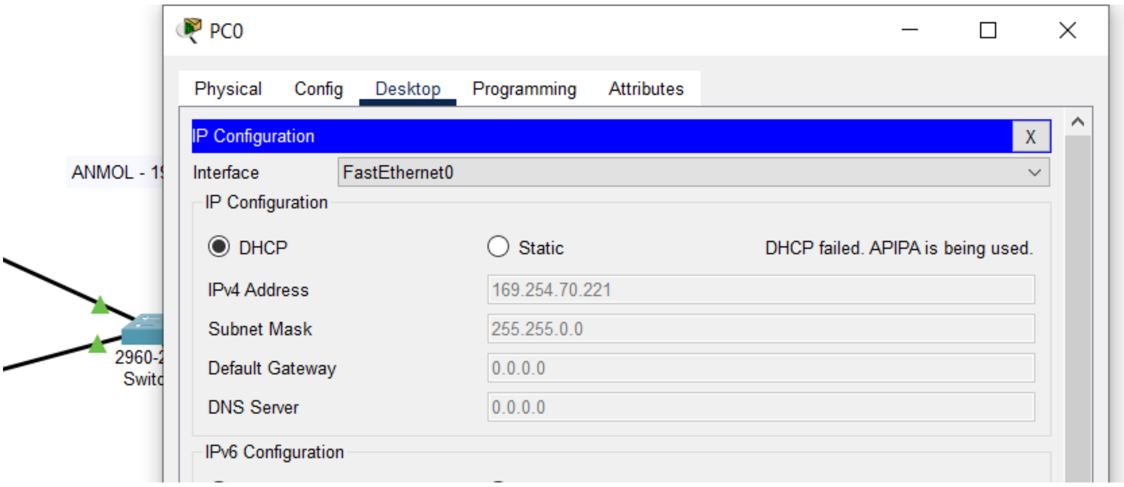
```
Switch>
Switch>en
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#ip dhcp snooping
Switch(config)#
Switch#
```

(all untrusted)



```
%SYS-5-CONFIG_I: Configured from console by console

Switch#
Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
none
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                  Trusted    Rate limit (pps)
-----------------------    -------    ----------------
Switch#
```

(All untrusted causes DHCP failure)



IP Configuration

Interface FastEthernet0

IP Configuration

● DHCP ○ Static DHCP failed. APIPA is being used.

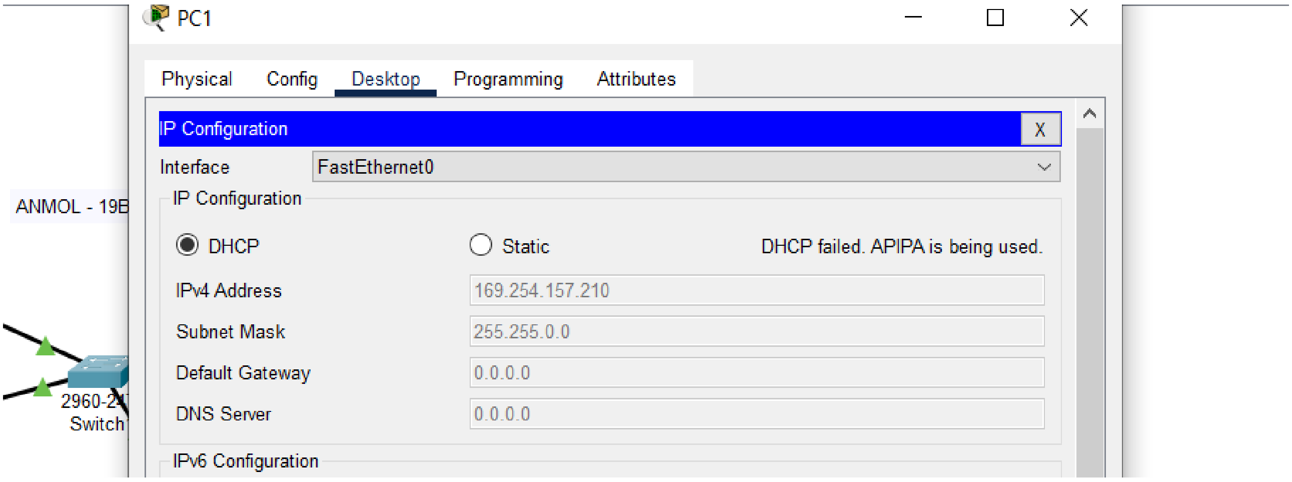
IPv4 Address 169.254.70.221

Subnet Mask 255.255.0.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration



IP Configuration

Interface FastEthernet0

IP Configuration

● DHCP ○ Static DHCP failed. APIPA is being used.

IPv4 Address 169.254.157.210

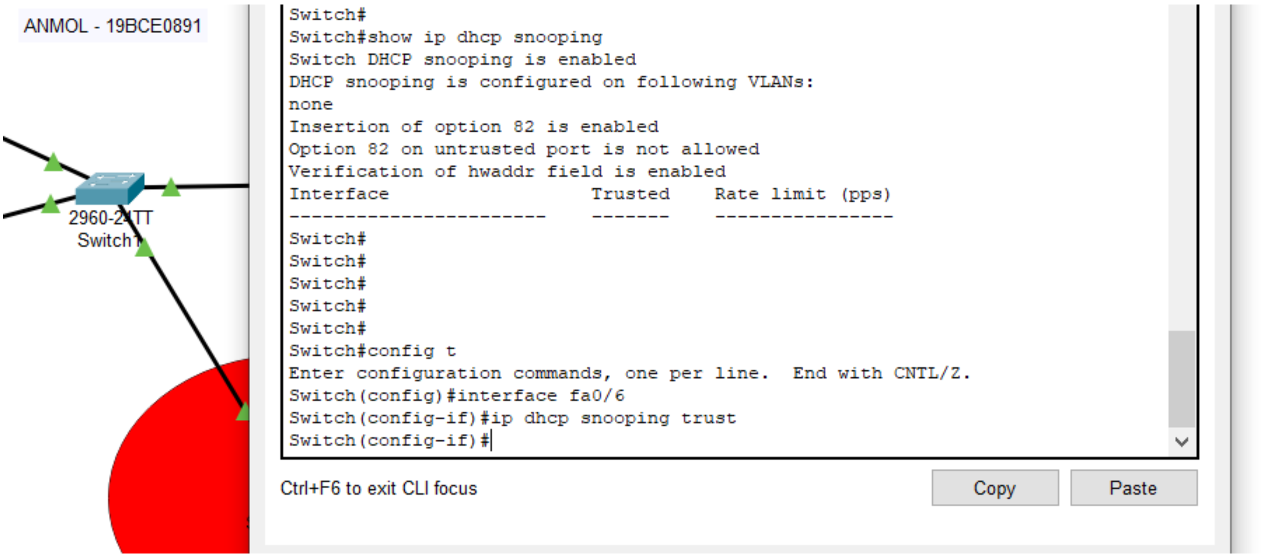
Subnet Mask 255.255.0.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

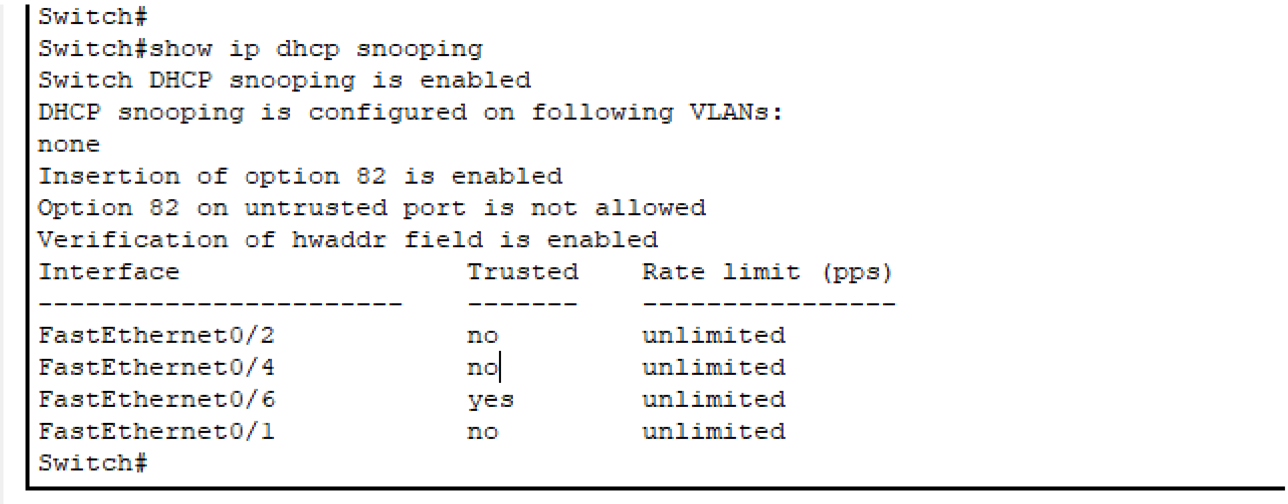
(Trusting server0 for DHCP and remaining all as untrusted)



```
Switch#
Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
none
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                  Trusted     Rate limit (pps)
-----------------------    -------     ----------------
Switch#
Switch#
Switch#
Switch#
Switch#
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#interface fa0/6
Switch(config-if)#ip dhcp snooping trust
Switch(config-if)#
```

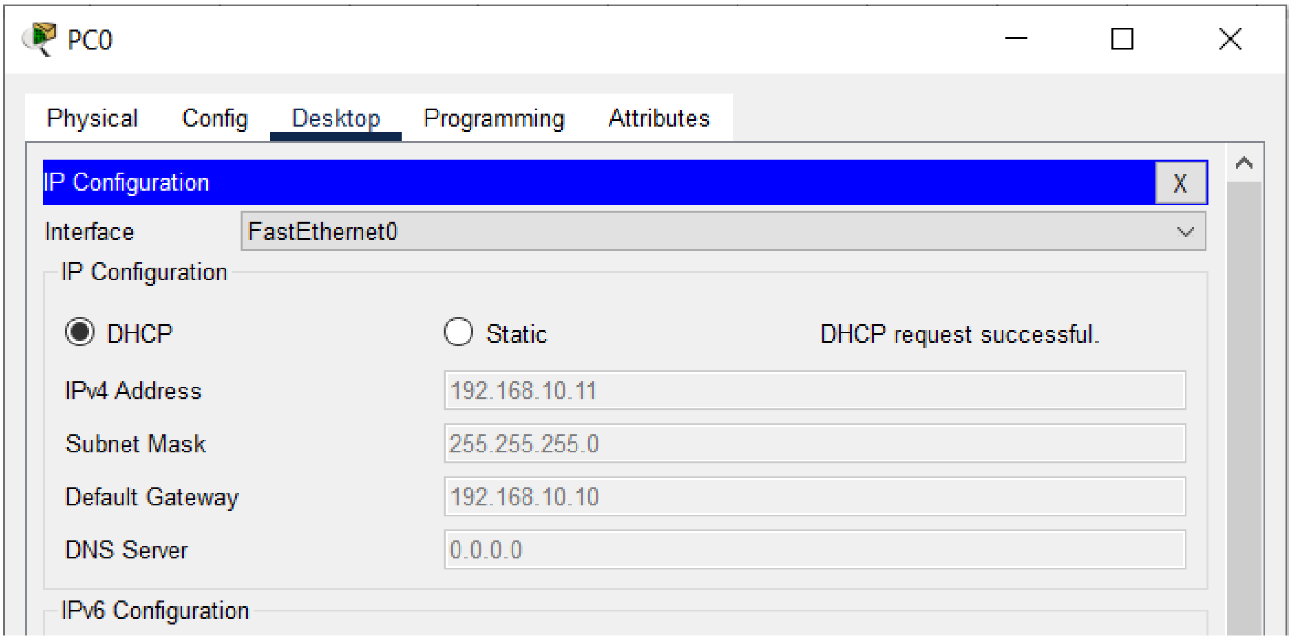
Commands for enabling trusted server

Switch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface fa0/6
Switch(config-if)#ip dhcp snooping trust



```
Switch#
Switch#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
none
Insertion of option 82 is enabled
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface                  Trusted    Rate limit (pps)
-----------------------    -------    ----------------
FastEthernet0/2            no         unlimited
FastEthernet0/4            no         unlimited
FastEthernet0/6            yes        unlimited
FastEthernet0/1            no         unlimited
Switch#
```

(DHCP using trusted server)



Interface: FastEthernet0

IP Configuration

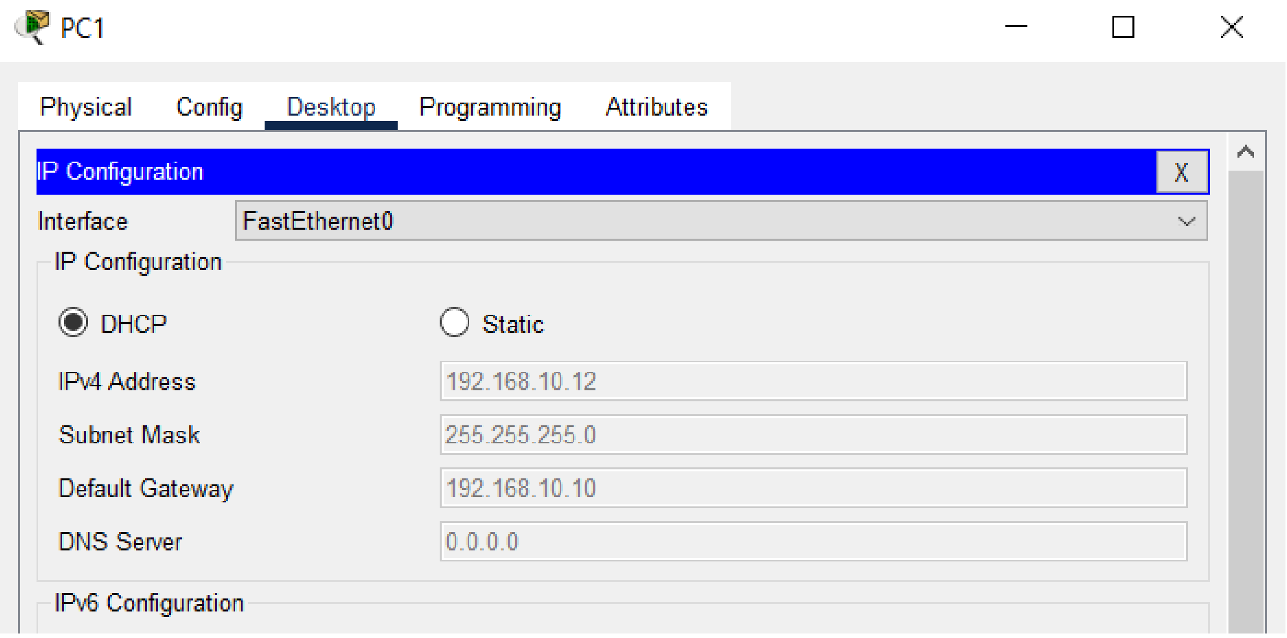
● DHCP ○ Static DHCP request successful.

IPv4 Address: 192.168.10.11

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.10

DNS Server: 0.0.0.0



Interface: FastEthernet0

IP Configuration

● DHCP ○ Static

IPv4 Address: 192.168.10.12

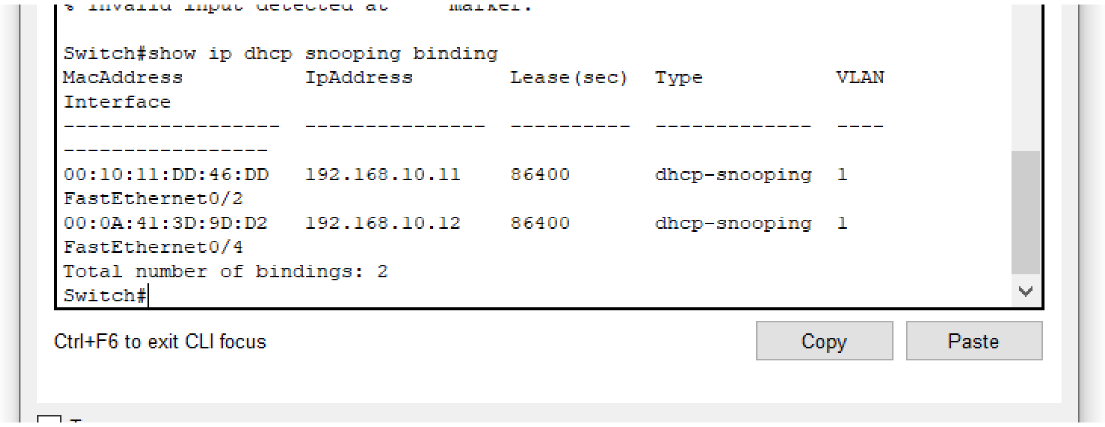
Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.10

DNS Server: 0.0.0.0

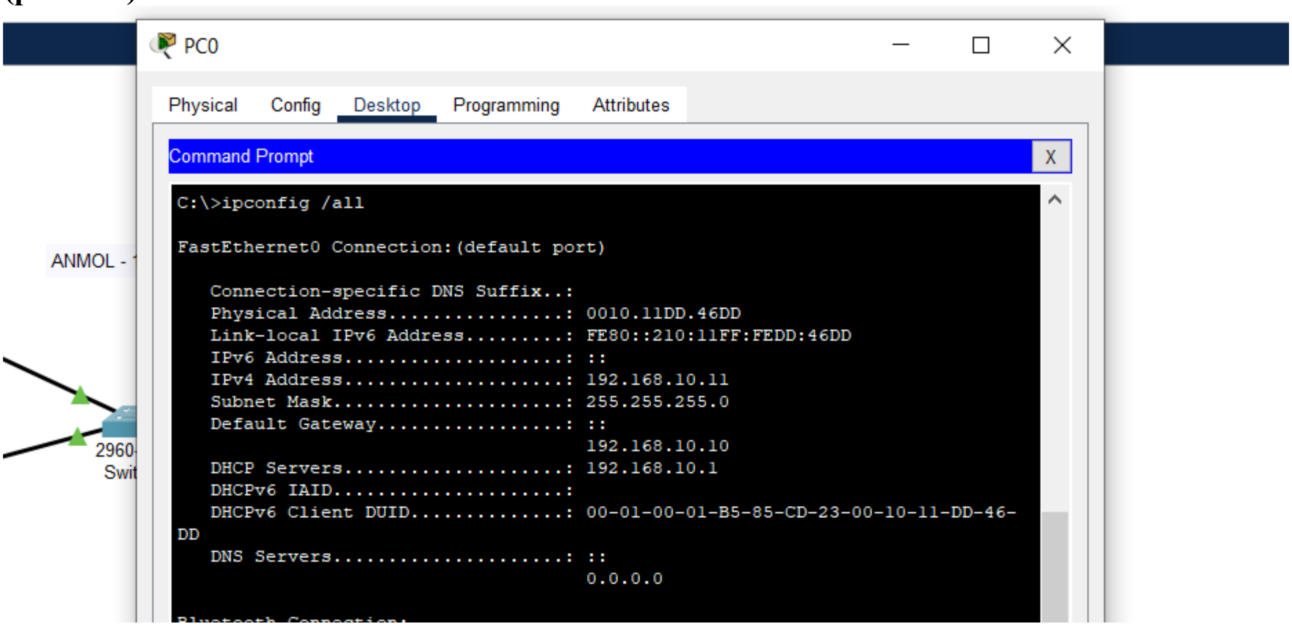
IPv6 Configuration

(Mac address of PC's getting ip address using dhcp by trusted server)



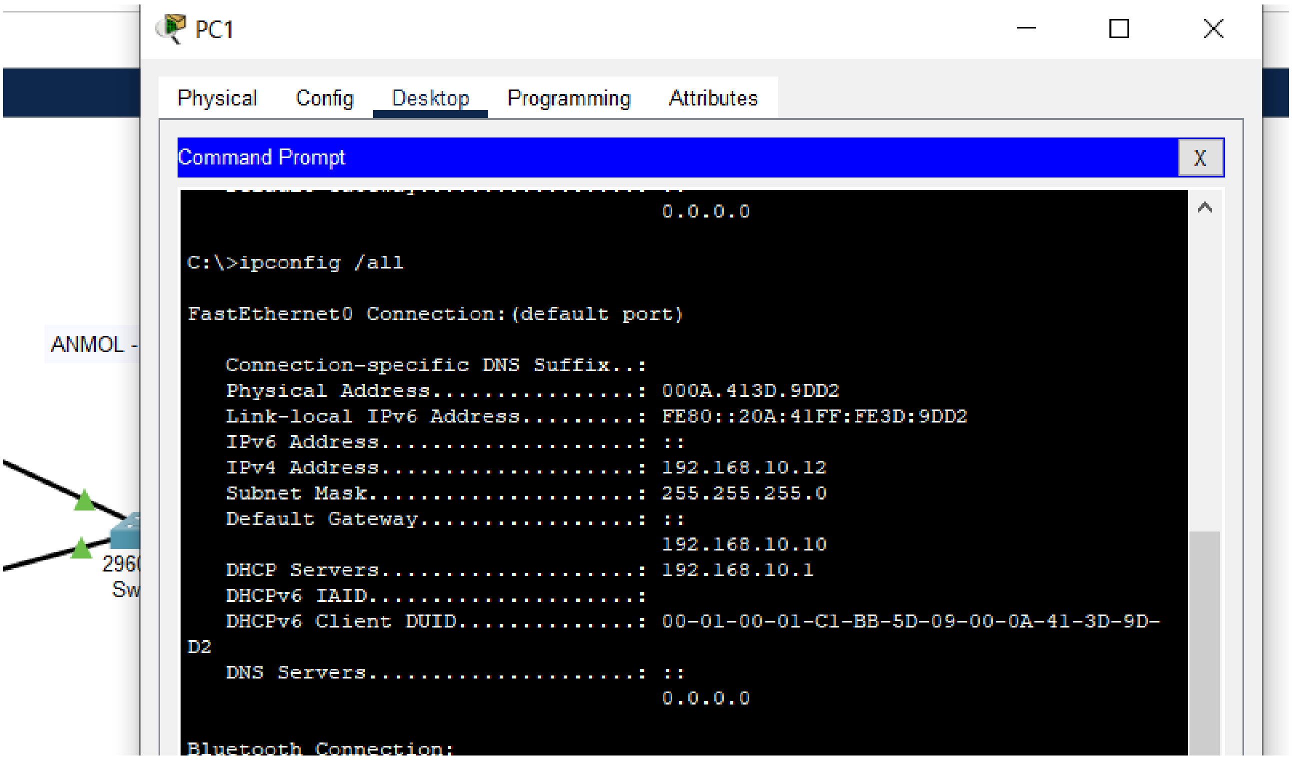
```
% Invalid input detected at    marker.

Switch#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN
Interface
------------------  ---------------  ----------  -------------  ----
-----------------
00:10:11:DD:46:DD   192.168.10.11    86400       dhcp-snooping  1
FastEthernet0/2
00:0A:41:3D:9D:D2   192.168.10.12    86400       dhcp-snooping  1
FastEthernet0/4
Total number of bindings: 2
Switch#
```

(pc0 mac)



```
C:\>ipconfig /all

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Physical Address................: 0010.11DD.46DD
   Link-local IPv6 Address.........: FE80::210:11FF:FEDD:46DD
   IPv6 Address....................: ::
   IPv4 Address....................: 192.168.10.11
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: ::
                                     192.168.10.10
   DHCP Servers....................: 192.168.10.1
   DHCPv6 IAID.....................:
   DHCPv6 Client DUID..............: 00-01-00-01-B5-85-CD-23-00-10-11-DD-46-
DD
   DNS Servers.....................: ::
                                     0.0.0.0

Bluetooth Connection:
```

(pc1 mac)



```
                                     0.0.0.0

C:\>ipconfig /all

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Physical Address................: 000A.413D.9DD2
   Link-local IPv6 Address.........: FE80::20A:41FF:FE3D:9DD2
   IPv6 Address....................: ::
   IPv4 Address....................: 192.168.10.12
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: ::
                                     192.168.10.10
   DHCP Servers....................: 192.168.10.1
   DHCPv6 IAID.....................:
   DHCPv6 Client DUID..............: 00-01-00-01-C1-BB-5D-09-00-0A-41-3D-9D-
D2
   DNS Servers.....................: ::
                                     0.0.0.0

Bluetooth Connection:
```

#6b – PYTHON CODE FOR FILTERING PACKETS (using size, protocol, keywords)

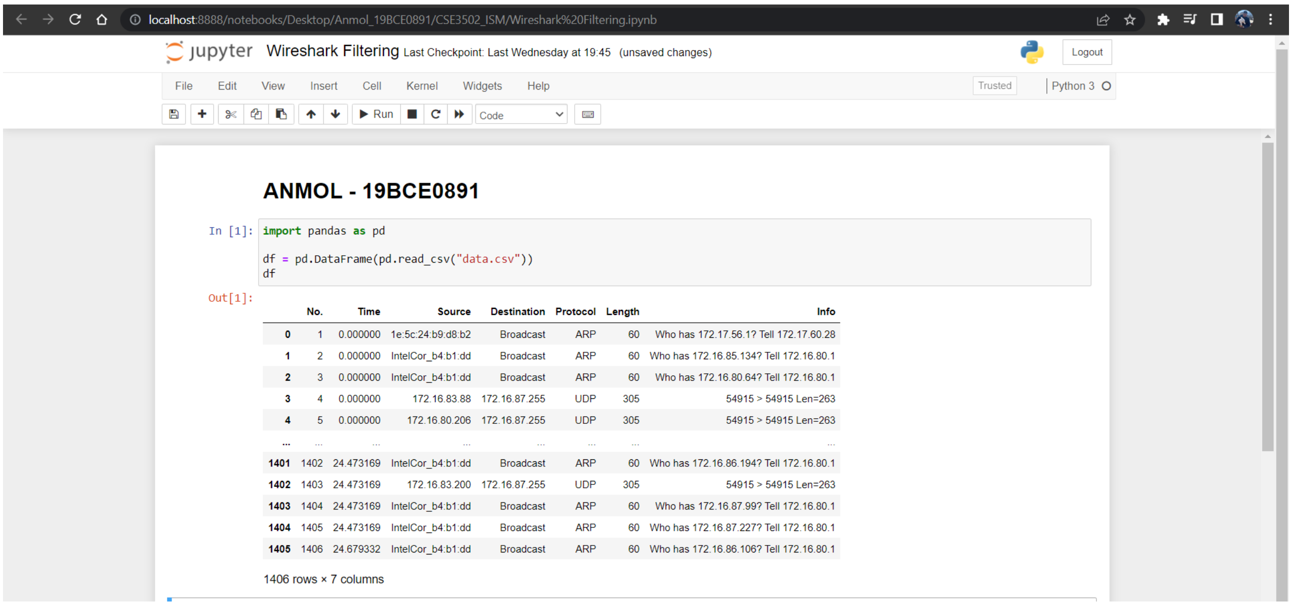
Algorithm :

1. Start Wireshark and browse anything and capture packets and stop it. Export the generated values to CSV.
2. Read CSV and store it in pandas dataframe.
3. Input Size, keywords and Protocol and filter using below code.

CODE :

```
import pandas as pd

df = pd.DataFrame(pd.read_csv("data.csv"))
print(df)

# protocol filtering
inpPrtcl = str(input("Enter Protocol to filter : "))
protocol_filtered = df.loc[df['Protocol'] == inpPrtcl]
print(protocol_filtered)


# size(len) filtering
inpSize = int(input("Enter size (length) to filter : "))
size_filtered = df.loc[df["Length"] == inpSize]
print(size_filtered)


# keywords filter
inpKeyword = str(input("Enter Keyword to filter : "))
keyword_filtered = df.loc[df['Info'].str.contains(inpKeyword)]
print(keyword_filtered)
```

OUTPUT -


localhost:8888/notebooks/Desktop/Anmol_19BCE0891/CSE3502_ISM/Wireshark%20Filtering.ipynb
jupyter Wireshark Filtering Last Checkpoint: Last Wednesday at 19:45 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

ANMOL - 19BCE0891

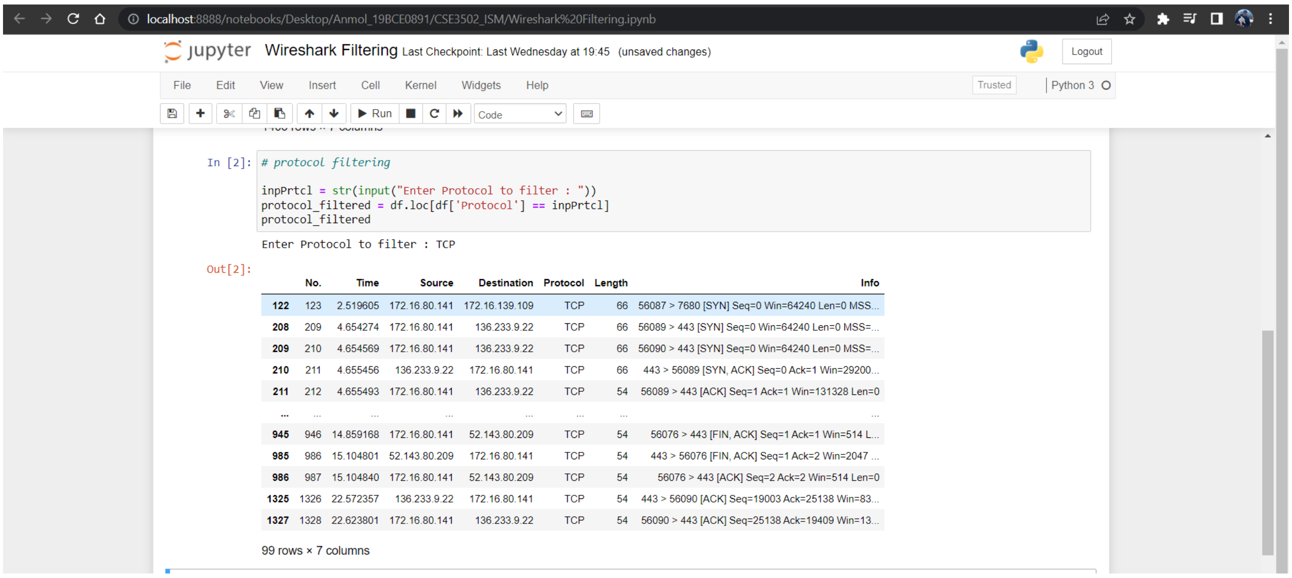
In [1]:

```python
import pandas as pd

df = pd.DataFrame(pd.read_csv("data.csv"))
df
```

Out[1]:

	No.	Time	Source	Destination	Protocol	Length	Info
0	1	0.000000	1e:5c:24:b9:d8:b2	Broadcast	ARP	60	Who has 172.17.56.1? Tell 172.17.60.28
1	2	0.000000	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.85.134? Tell 172.16.80.1
2	3	0.000000	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.80.64? Tell 172.16.80.1
3	4	0.000000	172.16.83.88	172.16.87.255	UDP	305	54915 > 54915 Len=263
4	5	0.000000	172.16.80.206	172.16.87.255	UDP	305	54915 > 54915 Len=263
...	...	...	...	...	...	...	...
1401	1402	24.473169	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.86.194? Tell 172.16.80.1
1402	1403	24.473169	172.16.83.200	172.16.87.255	UDP	305	54915 > 54915 Len=263
1403	1404	24.473169	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.87.99? Tell 172.16.80.1
1404	1405	24.473169	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.87.227? Tell 172.16.80.1
1405	1406	24.679332	IntelCor_b4:b1:dd	Broadcast	ARP	60	Who has 172.16.86.106? Tell 172.16.80.1

1406 rows × 7 columns


localhost:8888/notebooks/Desktop/Anmol_19BCE0891/CSE3502_ISM/Wireshark%20Filtering.ipynb
jupyter Wireshark Filtering Last Checkpoint: Last Wednesday at 19:45 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [2]:

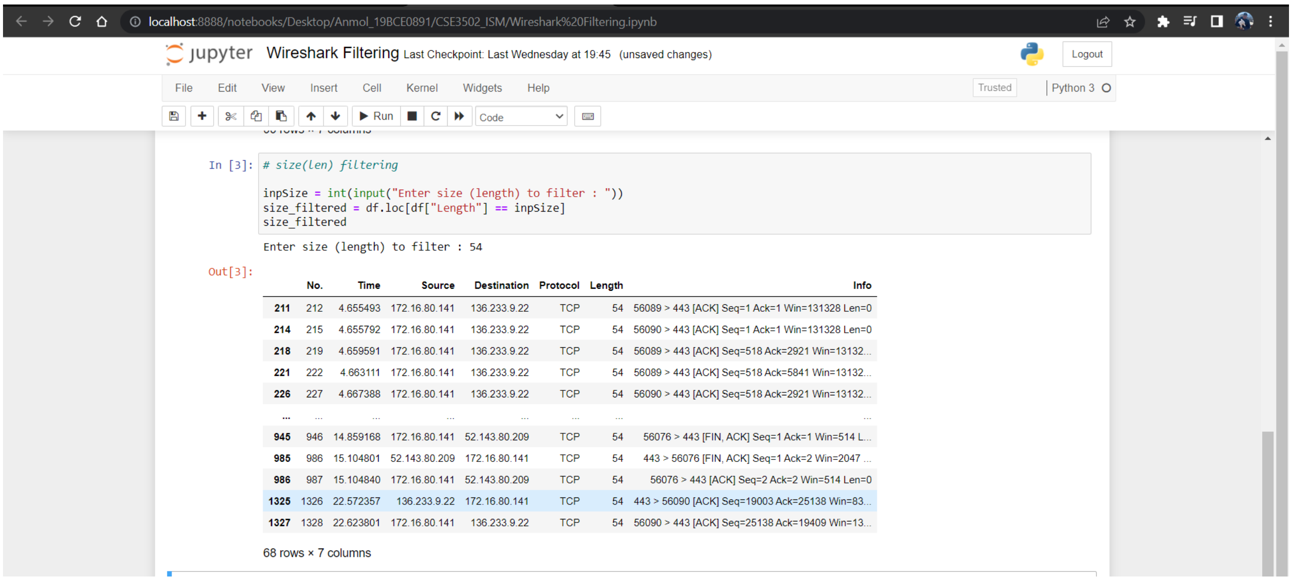
```python
# protocol filtering

inpPrtcl = str(input("Enter Protocol to filter : "))
protocol_filtered = df.loc[df['Protocol'] == inpPrtcl]
protocol_filtered
```

```
Enter Protocol to filter : TCP
```

Out[2]:

	No.	Time	Source	Destination	Protocol	Length	Info
122	123	2.519605	172.16.80.141	172.16.139.109	TCP	66	56087 > 7680 [SYN] Seq=0 Win=64240 Len=0 MSS...
208	209	4.654274	172.16.80.141	136.233.9.22	TCP	66	56089 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
209	210	4.654569	172.16.80.141	136.233.9.22	TCP	66	56090 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
210	211	4.655456	136.233.9.22	172.16.80.141	TCP	66	443 > 56089 [SYN, ACK] Seq=0 Ack=1 Win=29200...
211	212	4.655493	172.16.80.141	136.233.9.22	TCP	54	56089 > 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
...	...	...	...	...	...	...	...
945	946	14.859168	172.16.80.141	52.143.80.209	TCP	54	56076 > 443 [FIN, ACK] Seq=1 Ack=1 Win=514 L...
985	986	15.104801	52.143.80.209	172.16.80.141	TCP	54	443 > 56076 [FIN, ACK] Seq=1 Ack=2 Win=2047 ...
986	987	15.104840	172.16.80.141	52.143.80.209	TCP	54	56076 > 443 [ACK] Seq=2 Ack=2 Win=514 Len=0
1325	1326	22.572357	136.233.9.22	172.16.80.141	TCP	54	443 > 56090 [ACK] Seq=19003 Ack=25138 Win=83...
1327	1328	22.623801	172.16.80.141	136.233.9.22	TCP	54	56090 > 443 [ACK] Seq=25138 Ack=19409 Win=13...

99 rows × 7 columns


localhost:8888/notebooks/Desktop/Anmol_19BCE0891/CSE3502_ISM/Wireshark%20Filtering.ipynb
jupyter
Wireshark Filtering Last Checkpoint: Last Wednesday at 19:45 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [3]:

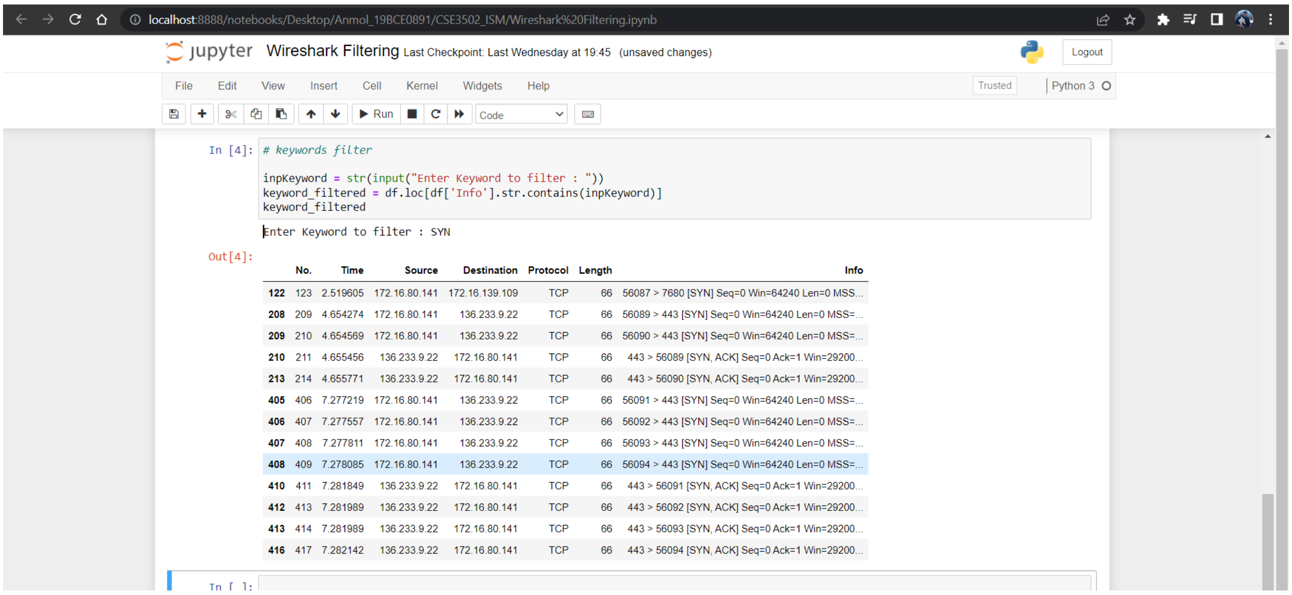
```python
# size(len) filtering

inpSize = int(input("Enter size (length) to filter : "))
size_filtered = df.loc[df["Length"] == inpSize]
size_filtered
```

```
Enter size (length) to filter : 54
```

Out[3]:

	No.	Time	Source	Destination	Protocol	Length	Info
211	212	4.655493	172.16.80.141	136.233.9.22	TCP	54	56089 > 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
214	215	4.655792	172.16.80.141	136.233.9.22	TCP	54	56090 > 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
218	219	4.659591	172.16.80.141	136.233.9.22	TCP	54	56089 > 443 [ACK] Seq=518 Ack=2921 Win=13132...
221	222	4.663111	172.16.80.141	136.233.9.22	TCP	54	56089 > 443 [ACK] Seq=518 Ack=5841 Win=13132...
226	227	4.667388	172.16.80.141	136.233.9.22	TCP	54	56090 > 443 [ACK] Seq=518 Ack=2921 Win=13132...
...	...	...	...	...	...	...	...
945	946	14.859168	172.16.80.141	52.143.80.209	TCP	54	56076 > 443 [FIN, ACK] Seq=1 Ack=1 Win=514 L...
985	986	15.104801	52.143.80.209	172.16.80.141	TCP	54	443 > 56076 [FIN, ACK] Seq=1 Ack=2 Win=2047 ...
986	987	15.104840	172.16.80.141	52.143.80.209	TCP	54	56076 > 443 [ACK] Seq=2 Ack=2 Win=514 Len=0
1325	1326	22.572357	136.233.9.22	172.16.80.141	TCP	54	443 > 56090 [ACK] Seq=19003 Ack=25138 Win=83...
1327	1328	22.623801	172.16.80.141	136.233.9.22	TCP	54	56090 > 443 [ACK] Seq=25138 Ack=19409 Win=13...

68 rows × 7 columns


localhost:8888/notebooks/Desktop/Anmol_19BCE0891/CSE3502_ISM/Wireshark%20Filtering.ipynb
jupyter
Wireshark Filtering Last Checkpoint: Last Wednesday at 19:45 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

In [4]:

```python
# keywords filter

inpKeyword = str(input("Enter Keyword to filter : "))
keyword_filtered = df.loc[df['Info'].str.contains(inpKeyword)]
keyword_filtered
```

```
Enter Keyword to filter : SYN
```

Out[4]:

	No.	Time	Source	Destination	Protocol	Length	Info
122	123	2.519605	172.16.80.141	172.16.139.109	TCP	66	56087 > 7680 [SYN] Seq=0 Win=64240 Len=0 MSS...
208	209	4.654274	172.16.80.141	136.233.9.22	TCP	66	56089 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
209	210	4.654569	172.16.80.141	136.233.9.22	TCP	66	56090 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
210	211	4.655456	136.233.9.22	172.16.80.141	TCP	66	443 > 56089 [SYN, ACK] Seq=0 Ack=1 Win=29200...
213	214	4.655771	136.233.9.22	172.16.80.141	TCP	66	443 > 56090 [SYN, ACK] Seq=0 Ack=1 Win=29200...
405	406	7.277219	172.16.80.141	136.233.9.22	TCP	66	56091 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
406	407	7.277557	172.16.80.141	136.233.9.22	TCP	66	56092 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
407	408	7.277811	172.16.80.141	136.233.9.22	TCP	66	56093 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
408	409	7.278085	172.16.80.141	136.233.9.22	TCP	66	56094 > 443 [SYN] Seq=0 Win=64240 Len=0 MSS=...
410	411	7.281849	136.233.9.22	172.16.80.141	TCP	66	443 > 56091 [SYN, ACK] Seq=0 Ack=1 Win=29200...
412	413	7.281989	136.233.9.22	172.16.80.141	TCP	66	443 > 56092 [SYN, ACK] Seq=0 Ack=1 Win=29200...
413	414	7.281989	136.233.9.22	172.16.80.141	TCP	66	443 > 56093 [SYN, ACK] Seq=0 Ack=1 Win=29200...
416	417	7.282142	136.233.9.22	172.16.80.141	TCP	66	443 > 56094 [SYN, ACK] Seq=0 Ack=1 Win=29200...

In [ ]: