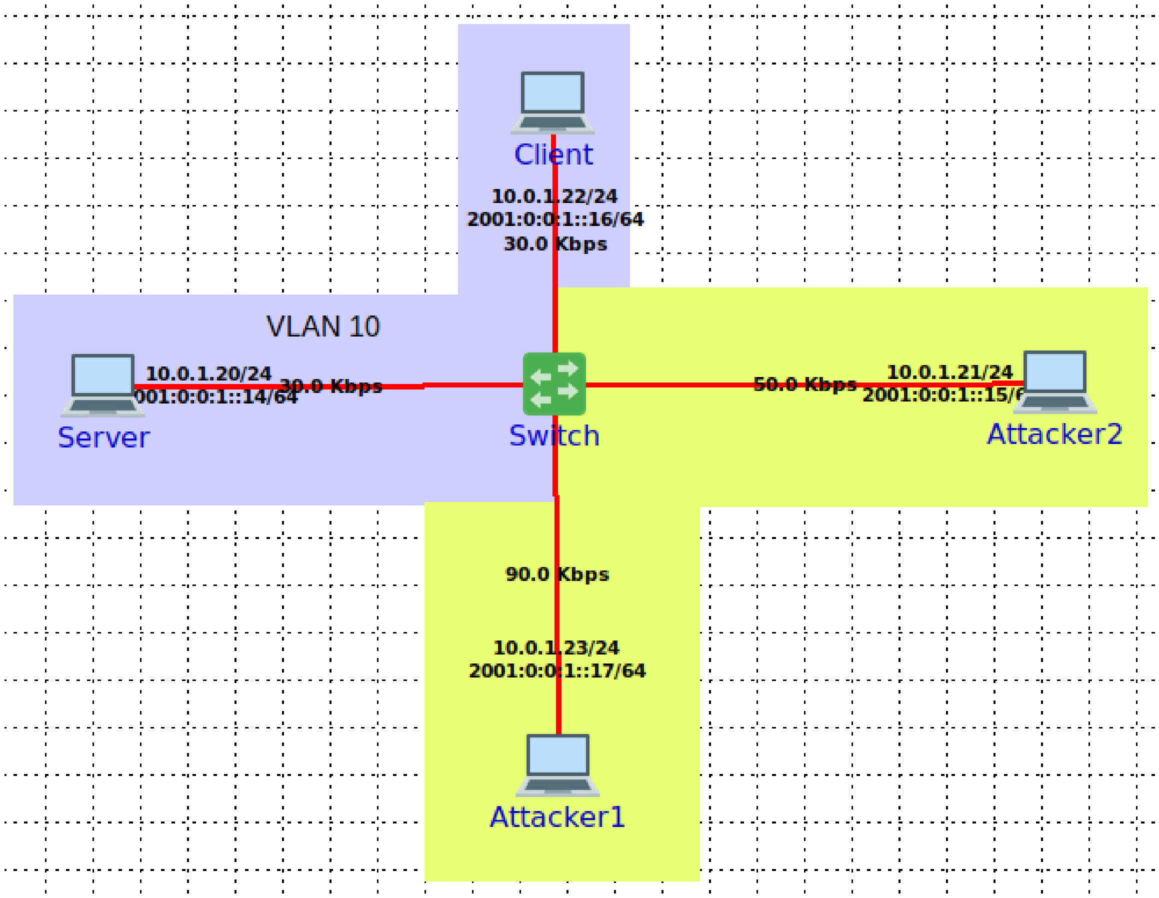
Network Security – CSCI_6541_80

Namana Y Tarikere – G21372717

Homework Assignment – 3

Access Control using VLANs (5pts)

1. **Load and run the scenario the scenario hw3.xml.**



2. **We are going to use Wireshark. In all the questions below, make sure to change Wireshark view to include full <u>date</u> for each packet captured as shown below**

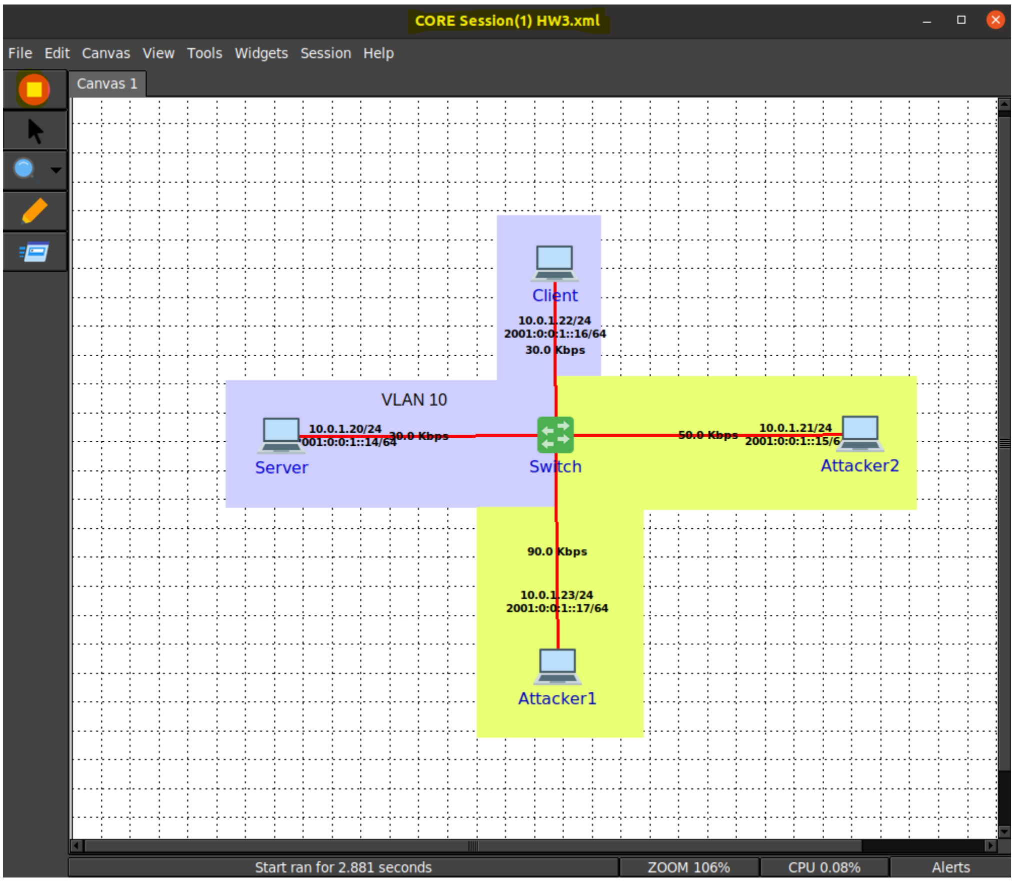
   In Wireshark console, Go to View -> Time Display Format -> Choose Date and Time of Day

3. **Make sure all name resolutions are turned off**

   In Wireshark console -> Go to View -> Name Resolution -> Uncheck all the name resolutions and ensure they are turned off

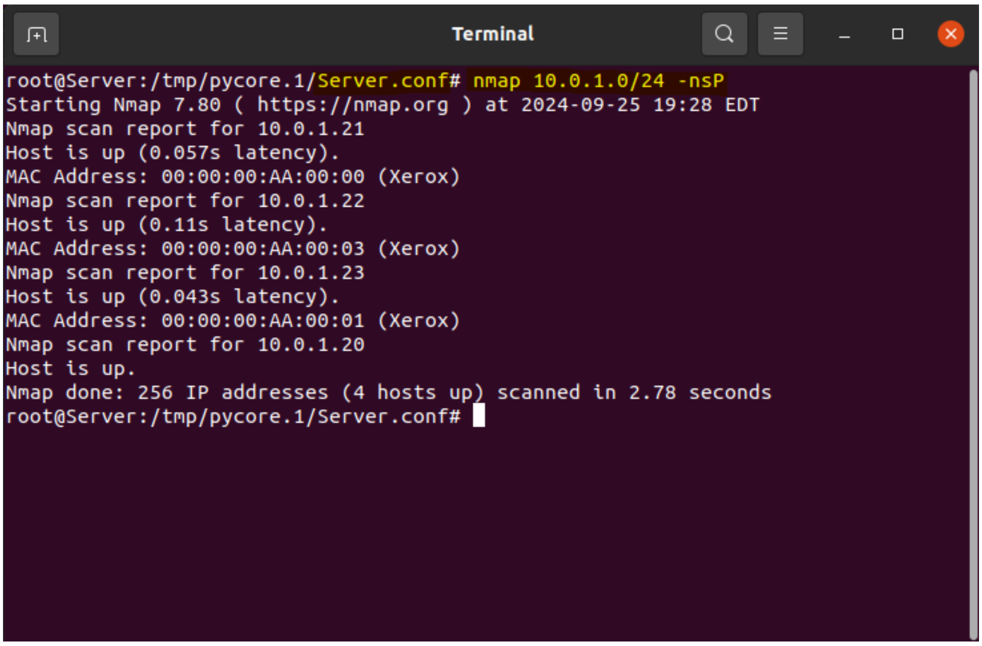
4. **Run the scenario:**

   Command used: **core-gui**



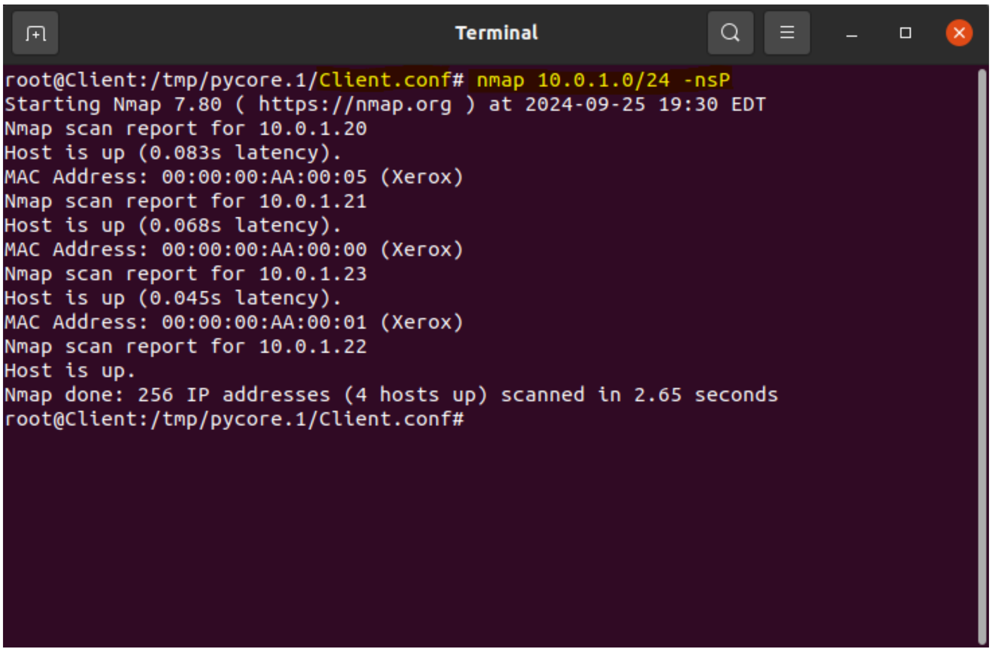
5. **(0.5 point) Run an nmap host scan from Client node to scan for all hosts on the 10.0.1.0/24 subnet. An nmap host scan tries to identify what hosts are up and reachable in the subnet provided.**
   a. **Run: nmap 10.0.1.0/24 –nsP.**
   b. **Show a screenshot of the result.**
   c. **Do the same thing from Client and report what it sees.**

Nmap from Server: The Nmap scan conducted on the Server reported that the server can see 4 active hosts - Server, Client, Attacker1 and Attacker 2 which are up and running as shown below:



```
root@Server:/tmp/pycore.1/Server.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 19:28 EDT
Nmap scan report for 10.0.1.21
Host is up (0.057s latency).
MAC Address: 00:00:00:AA:00:00 (Xerox)
Nmap scan report for 10.0.1.22
Host is up (0.11s latency).
MAC Address: 00:00:00:AA:00:03 (Xerox)
Nmap scan report for 10.0.1.23
Host is up (0.043s latency).
MAC Address: 00:00:00:AA:00:01 (Xerox)
Nmap scan report for 10.0.1.20
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.78 seconds
root@Server:/tmp/pycore.1/Server.conf#
```

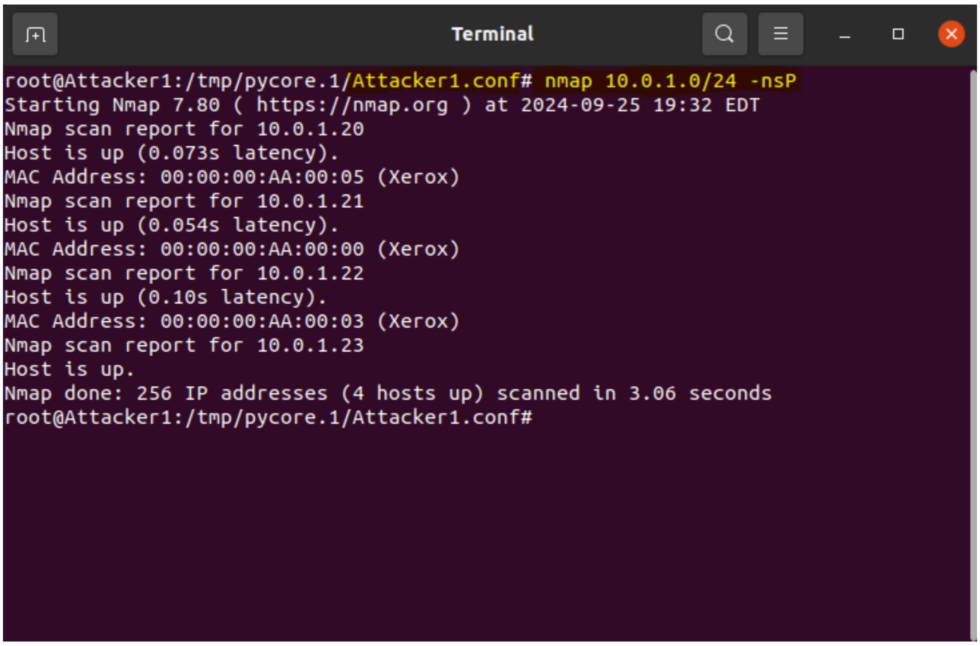
Nmap from Client: The Nmap scan conducted on the Client reported that the Client can see 4 active hosts - Server, Client, Attacker1 and Attacker 2 which are up and running as shown below:



```
root@Client:/tmp/pycore.1/Client.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 19:30 EDT
Nmap scan report for 10.0.1.20
Host is up (0.083s latency).
MAC Address: 00:00:00:AA:00:05 (Xerox)
Nmap scan report for 10.0.1.21
Host is up (0.068s latency).
MAC Address: 00:00:00:AA:00:00 (Xerox)
Nmap scan report for 10.0.1.23
Host is up (0.045s latency).
MAC Address: 00:00:00:AA:00:01 (Xerox)
Nmap scan report for 10.0.1.22
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.65 seconds
root@Client:/tmp/pycore.1/Client.conf#
```

6. **(0.5 point) Repeat 5 from Attacker 1. Does the Attacker 1 see both client and server?**

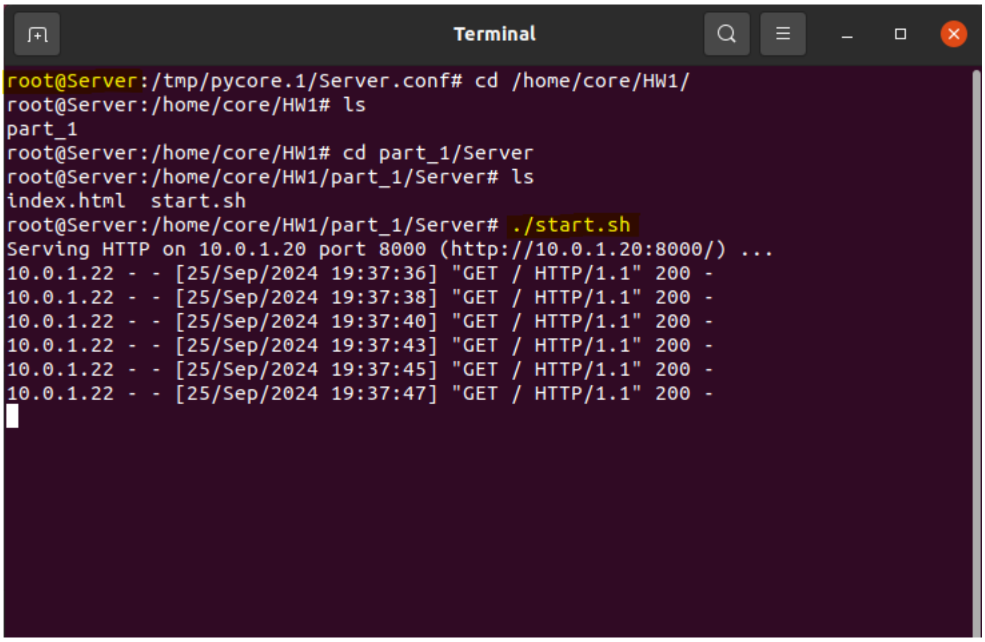
   Nmap from Attacker 1: The Nmap scan conducted on the Attacker 1 reported that the attacker can see 4 active hosts - Server, Client, Attacker1 and Attacker 2 which are up and running. Yes, Attacker 1 can see both the Client (IP: 10.0.1.22) and the Server (IP: 10.0.1.20) as shown below:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 19:32 EDT
Nmap scan report for 10.0.1.20
Host is up (0.073s latency).
MAC Address: 00:00:00:AA:00:05 (Xerox)
Nmap scan report for 10.0.1.21
Host is up (0.054s latency).
MAC Address: 00:00:00:AA:00:00 (Xerox)
Nmap scan report for 10.0.1.22
Host is up (0.10s latency).
MAC Address: 00:00:00:AA:00:03 (Xerox)
Nmap scan report for 10.0.1.23
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.06 seconds
root@Attacker1:/tmp/pycore.1/Attacker1.conf#
```

7. **Run the HTTP Server from HW1 on Sever. Show screenshot of server running**

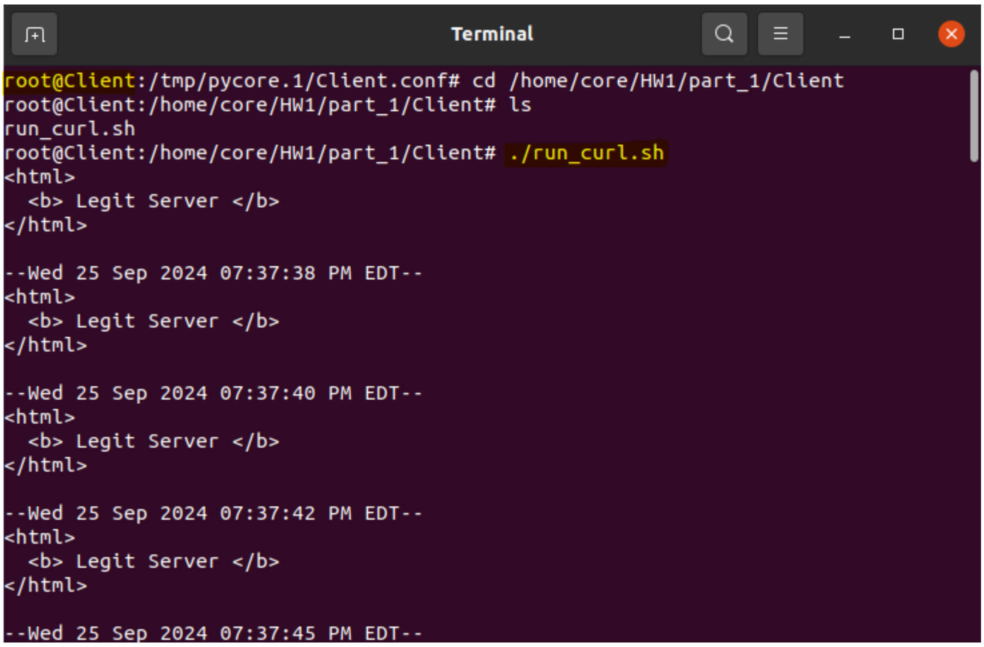
   Navigating to start.sh script from HW 1 in the Server node and hosting the HTTP server using **./start.sh** command as shown below:



```
root@Server:/tmp/pycore.1/Server.conf# cd /home/core/HW1/
root@Server:/home/core/HW1# ls
part_1
root@Server:/home/core/HW1# cd part_1/Server
root@Server:/home/core/HW1/part_1/Server# ls
index.html  start.sh
root@Server:/home/core/HW1/part_1/Server# ./start.sh
Serving HTTP on 10.0.1.20 port 8000 (http://10.0.1.20:8000/) ...
10.0.1.22 - - [25/Sep/2024 19:37:36] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 19:37:38] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 19:37:40] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 19:37:43] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 19:37:45] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 19:37:47] "GET / HTTP/1.1" 200 -
```

8. **Run the run_curl.sh script from HW1 on Client. Show screenshot of client running**

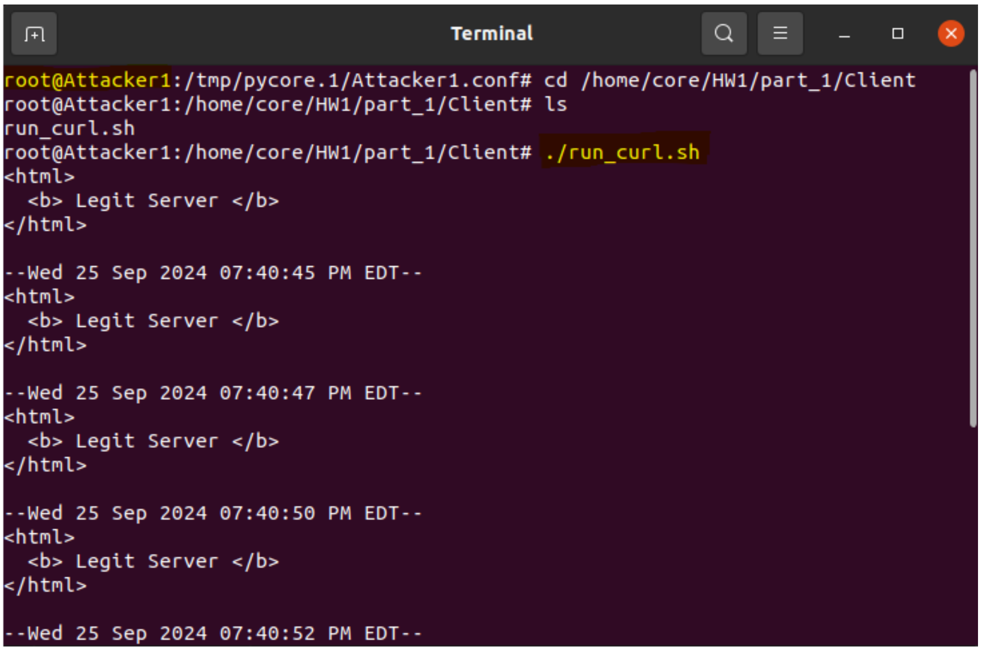
   Navigating to run_curl.sh script from HW 1 in the Client node and running the client script using the **./run_curl.sh** command as shown below:



```
root@Client:/tmp/pycore.1/Client.conf# cd /home/core/HW1/part_1/Client
root@Client:/home/core/HW1/part_1/Client# ls
run_curl.sh
root@Client:/home/core/HW1/part_1/Client# ./run_curl.sh
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:37:38 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:37:40 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:37:42 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:37:45 PM EDT--
```

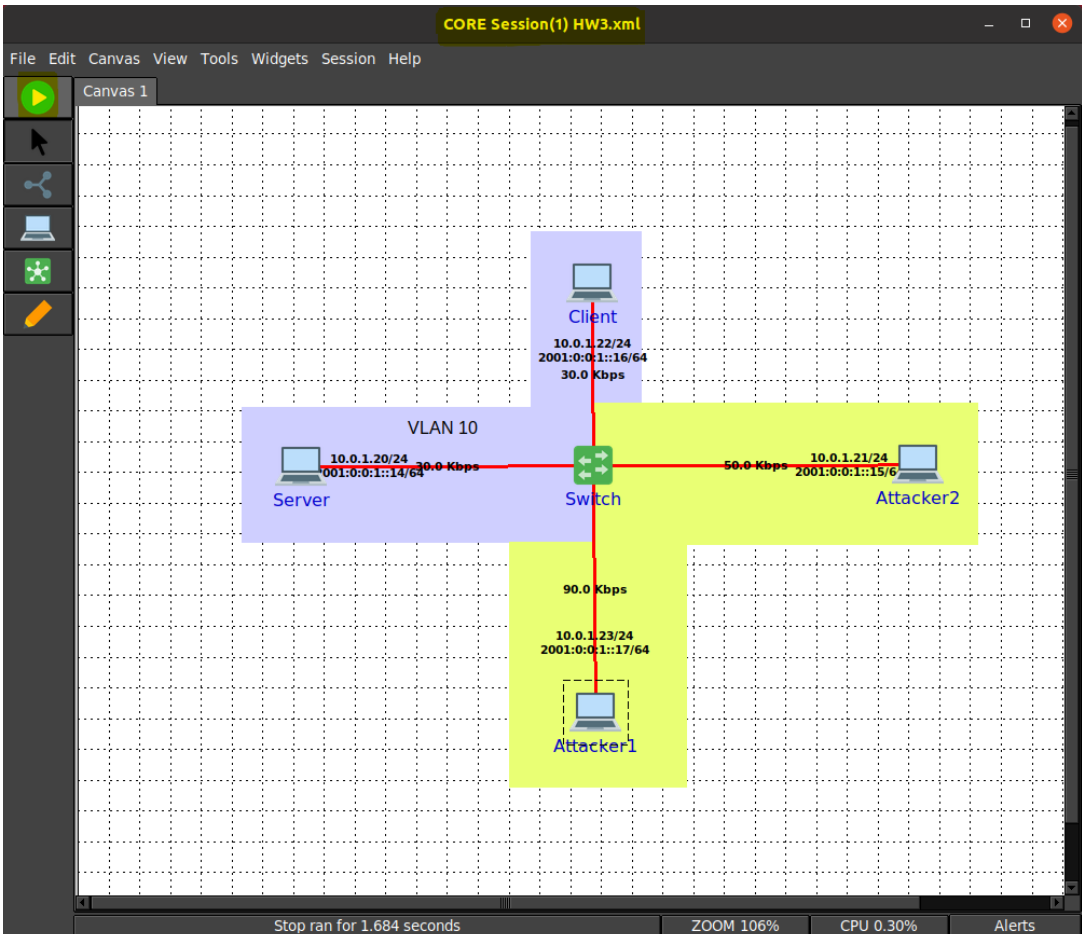
9. **(0.5 point) If you run the same run_curl.sh script from Attacker 1, will it be able to connect to server? Show screenshot to support your findings.**

   Navigate to run_curl.sh script from HW 1 in the Attacker 1 node and run the client script using the **./run_curl.sh** command. Here we can see that the Attacker 1 is receiving the HTTP page content from the Server since it is in the same network as the Server as shown below:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# cd /home/core/HW1/part_1/Client
root@Attacker1:/home/core/HW1/part_1/Client# ls
run_curl.sh
root@Attacker1:/home/core/HW1/part_1/Client# ./run_curl.sh
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:40:45 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:40:47 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:40:50 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 07:40:52 PM EDT--
```

10. Stop the scenario

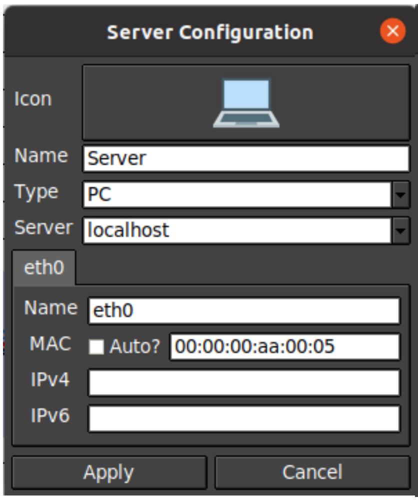


Stop ran for 1.684 seconds

11. Do the following for Server, Client, Attacker 1, and Attacker 2 nodes

a. **Right click on node and select configure.**
b. **Remove all the address assignments by clicking on the trash icon next to the addresses.**

Removing the IPV4 and IPV6 addresses from Server and Client configurations:

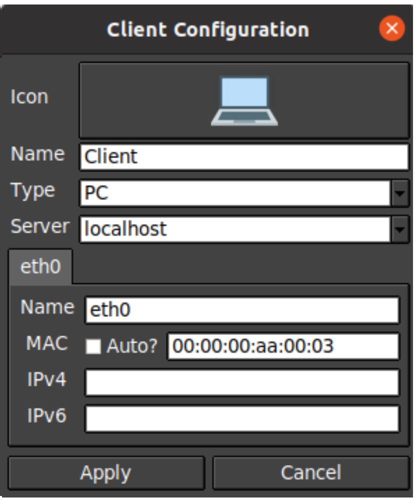


Server Configuration

Icon	
Name	Server
Type	PC
Server	localhost

eth0

Name	eth0
MAC	□ Auto? 00:00:00:aa:00:05
IPv4	
IPv6	



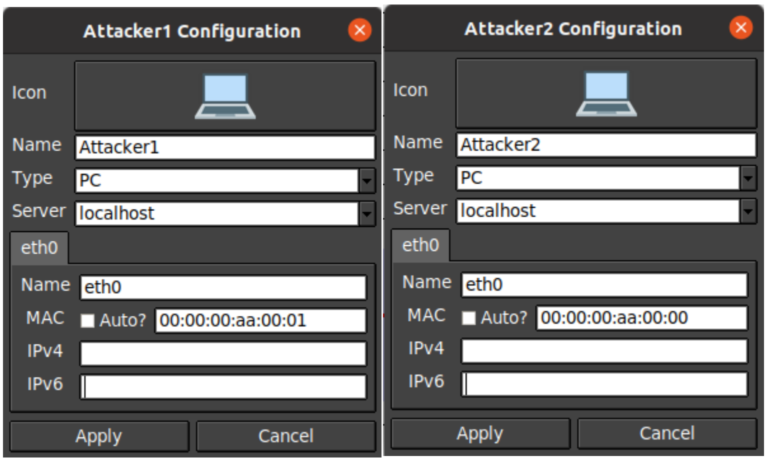
Client Configuration

Icon	
Name	Client
Type	PC
Server	localhost

eth0

Name	eth0
MAC	□ Auto? 00:00:00:aa:00:03
IPv4	
IPv6	

Removing the IPV4 and IPV6 addresses from Attacker 1 and Attacker 2 configurations:



Attacker1 Configuration

Icon

Name: Attacker1

Type: PC

Server: localhost

eth0

Name: eth0

MAC: □ Auto? 00:00:00:aa:00:01

IPv4:

IPv6:

Apply | Cancel

Attacker2 Configuration

Icon

Name: Attacker2

Type: PC

Server: localhost

eth0

Name: eth0

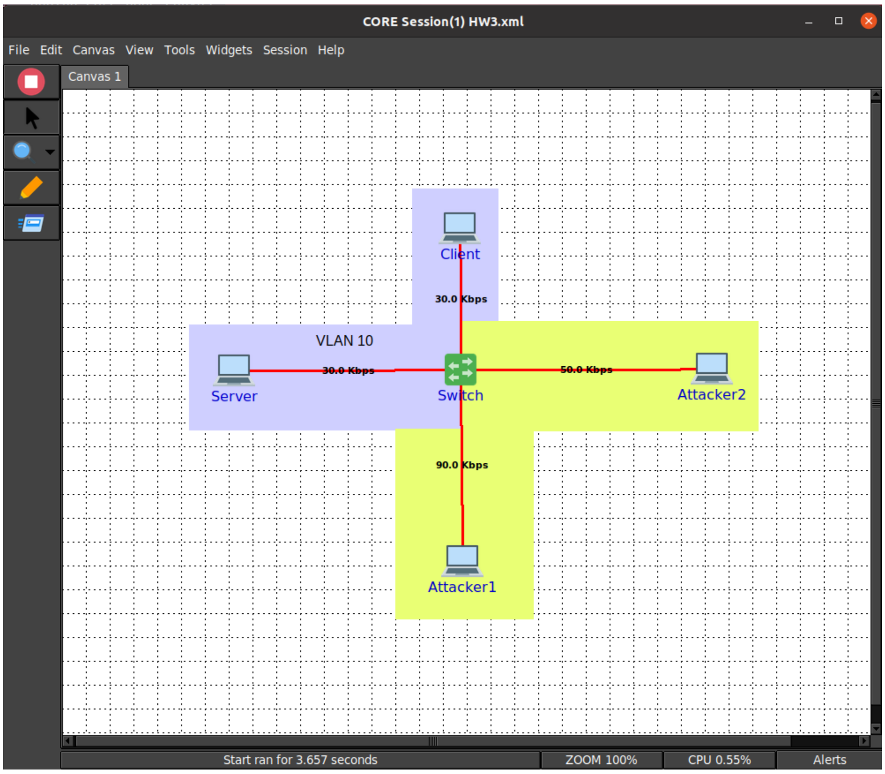
MAC: □ Auto? 00:00:00:aa:00:00

IPv4:

IPv6:

Apply | Cancel

This should be your view after:

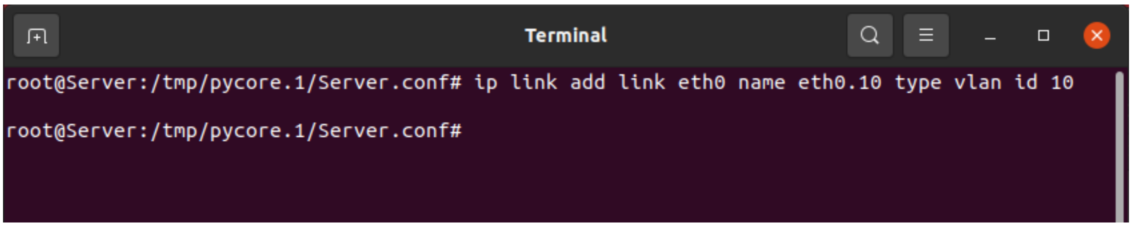


CORE Session(1) HW3.xml

12. Add VLAN interfaces to nodes Client, Server, Attacker 1, and Attacker2 based on the color grouping shown in the above figure. You can use the ip link add command:

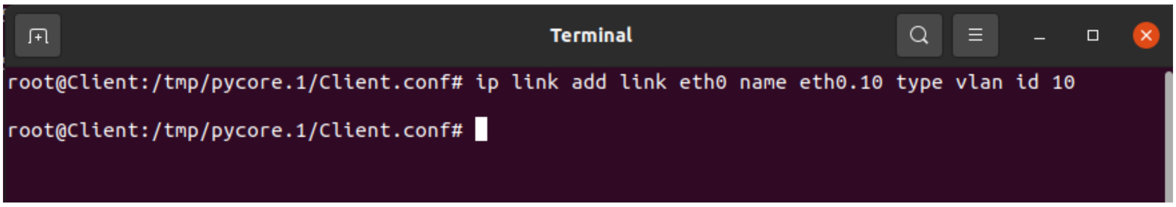
a. For instance, on Client: ip link add link eth0 name eth0.10 type vlan id 10

Server: Adding a new VLAN interface to the Server node using the command **ip link add link eth0 name eth0.10 type vlan id 10** as shown below:



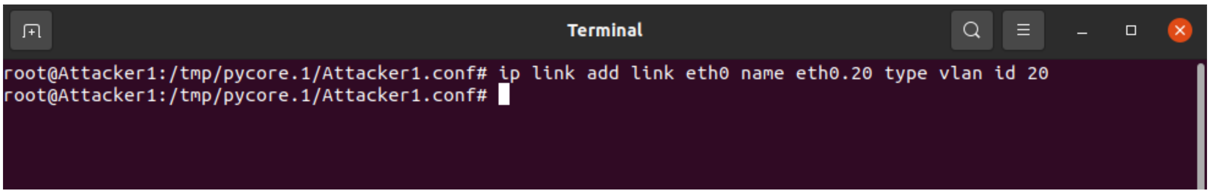
```
root@Server:/tmp/pycore.1/Server.conf# ip link add link eth0 name eth0.10 type vlan id 10

root@Server:/tmp/pycore.1/Server.conf#
```

Client: Adding a new VLAN interface to the Client node using the command **ip link add link eth0 name eth0.10 type vlan id 10** as shown below:



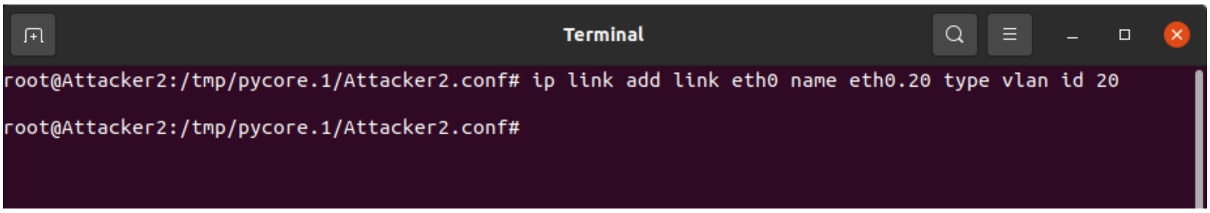
```
root@Client:/tmp/pycore.1/Client.conf# ip link add link eth0 name eth0.10 type vlan id 10

root@Client:/tmp/pycore.1/Client.conf#
```

Attacker 1: Adding a new VLAN interface to the Attacker 1 node using the command **ip link add link eth0 name eth0.20 type vlan id 20** as shown below:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# ip link add link eth0 name eth0.20 type vlan id 20
root@Attacker1:/tmp/pycore.1/Attacker1.conf#
```

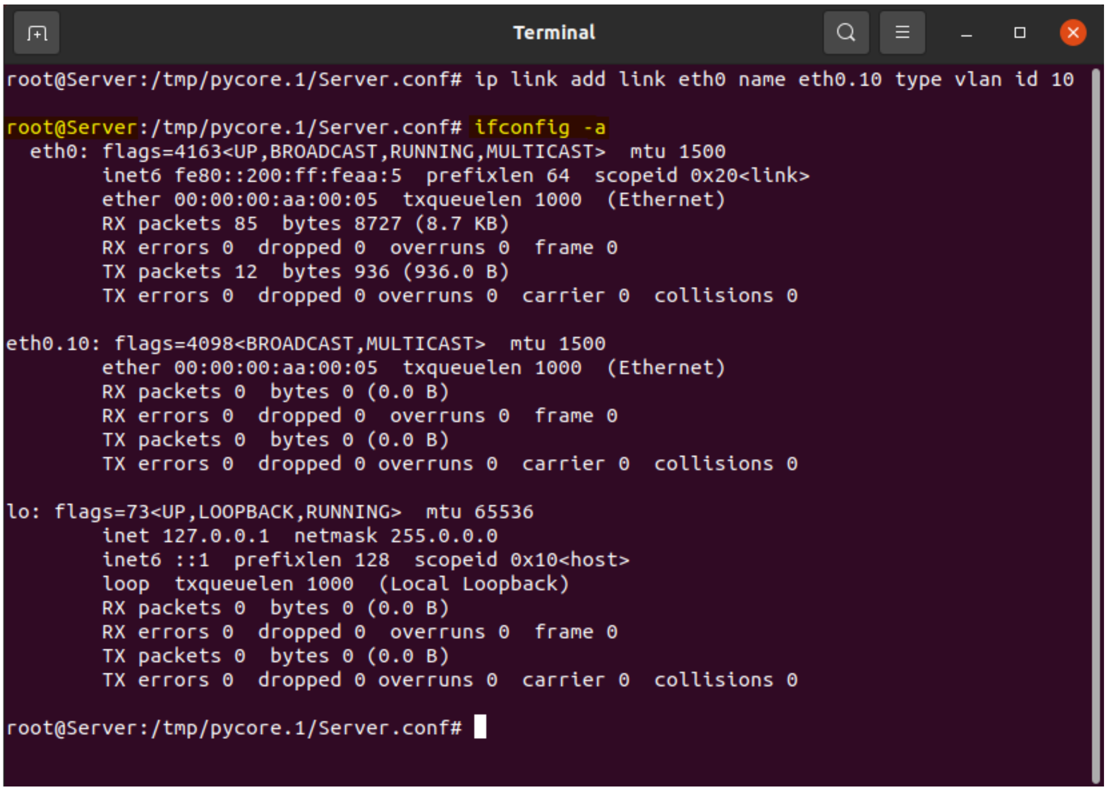
Attacker 2: Adding a new VLAN interface to the Attacker 2 node using the command **ip link add link eth0 name eth0.20 type vlan id 20** as shown below:



```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# ip link add link eth0 name eth0.20 type vlan id 20

root@Attacker2:/tmp/pycore.1/Attacker2.conf#
```

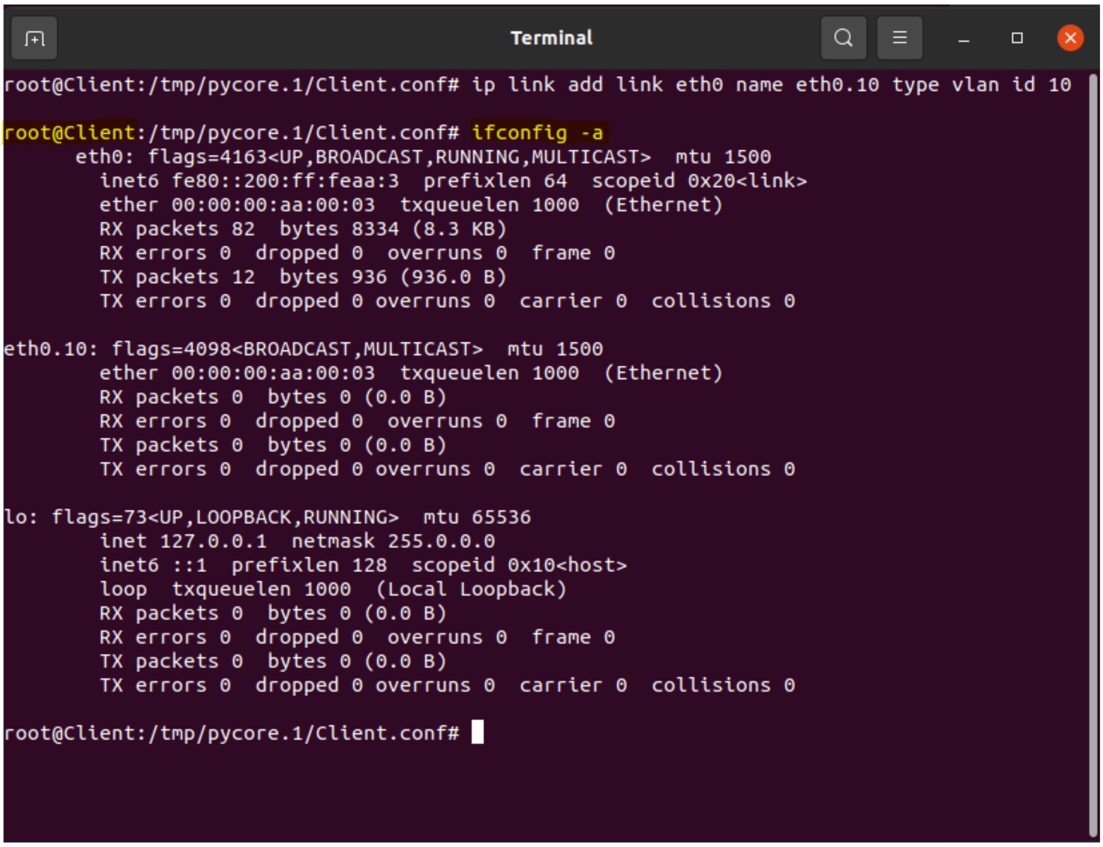
b. **Notice a new interface called *eth0.10.* You can see it by running the following command: *ifconfig –a***

Server: We can see that the Server is connected to VLAN eth0.10 as shown below:



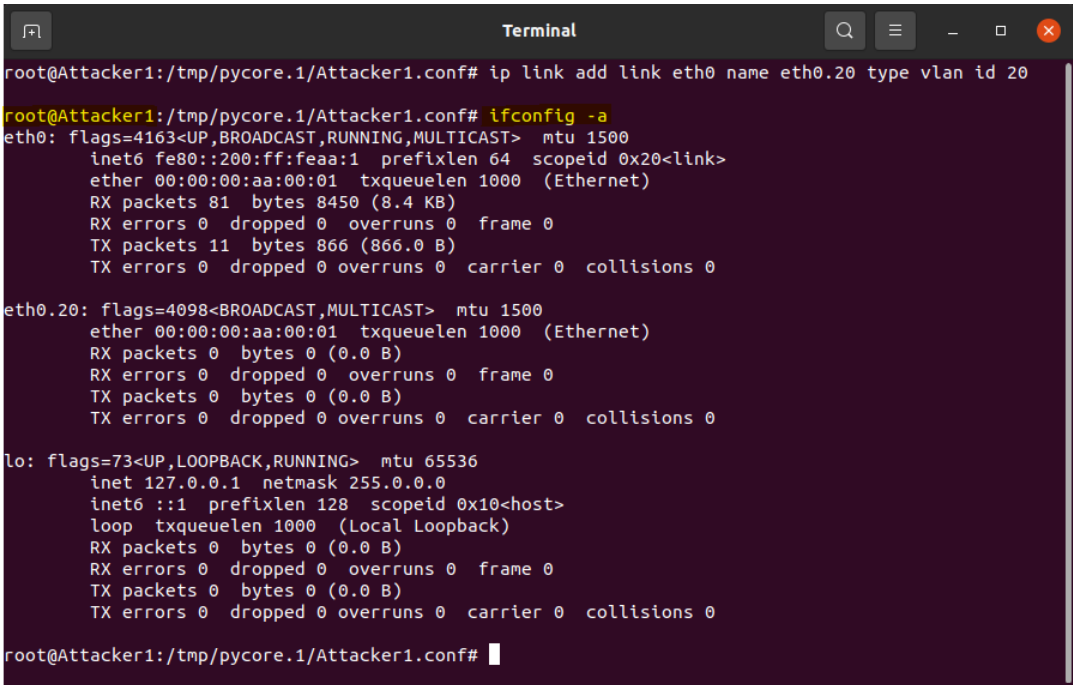
```
root@Server:/tmp/pycore.1/Server.conf# ip link add link eth0 name eth0.10 type vlan id 10

root@Server:/tmp/pycore.1/Server.conf# ifconfig -a
  eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:5  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:05  txqueuelen 1000  (Ethernet)
        RX packets 85  bytes 8727 (8.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.10: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 00:00:00:aa:00:05  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Server:/tmp/pycore.1/Server.conf# 
```

Client: We can see that the Client is connected to VLAN eth0.10 as shown below:



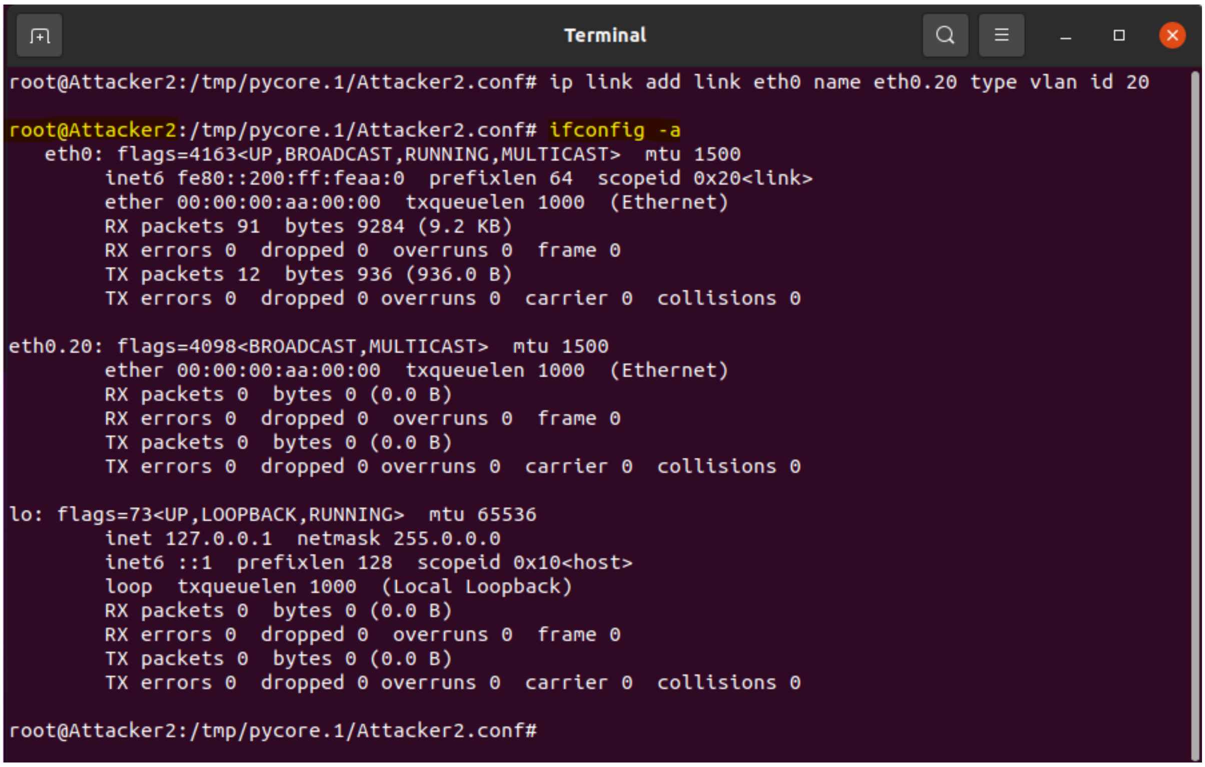
```
root@Client:/tmp/pycore.1/Client.conf# ip link add link eth0 name eth0.10 type vlan id 10

root@Client:/tmp/pycore.1/Client.conf# ifconfig -a
      eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:3  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 82  bytes 8334 (8.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.10: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Client:/tmp/pycore.1/Client.conf# 
```

Attacker 1: We can see that the Attacker 1 is connected to VLAN eth0.20 as shown:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# ip link add link eth0 name eth0.20 type vlan id 20

root@Attacker1:/tmp/pycore.1/Attacker1.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:1  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:01  txqueuelen 1000  (Ethernet)
        RX packets 81  bytes 8450 (8.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.20: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 00:00:00:aa:00:01  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Attacker1:/tmp/pycore.1/Attacker1.conf# 
```

Attacker 2: We can see that the Attacker 2 is connected to VLAN eth0.20 as shown:



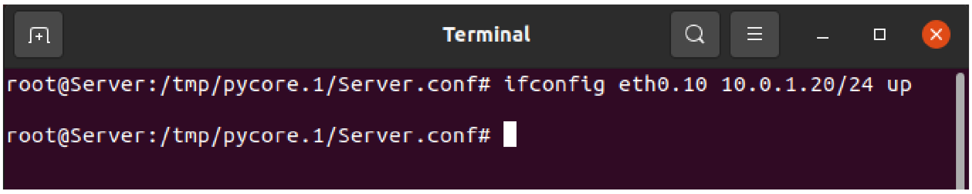
```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# ip link add link eth0 name eth0.20 type vlan id 20

root@Attacker2:/tmp/pycore.1/Attacker2.conf# ifconfig -a
   eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 91  bytes 9284 (9.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.20: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 00:00:00:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Attacker2:/tmp/pycore.1/Attacker2.conf#
```

Here, note that the client and server are connected to the new VLAN interface eth0.10, while Attacker 1 and Attacker 2 are connected to the new VLAN interface eth0.20. This setup differs from the previous network configuration, where all hosts were associated with eth0 (same network).

13. **Assign the addresses from the figure shown in step 1 to the VLAN interfaces on all nodes using the ifconfig command:**

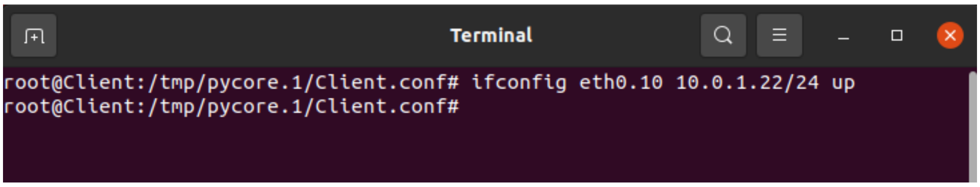
   a. **For instance, on client 1: ifconfig eth0.10 10.0.1.22/24 up**

   Server: Assign the Server IP address to the new VLAN interface eth0.10 on the server node as shown below:



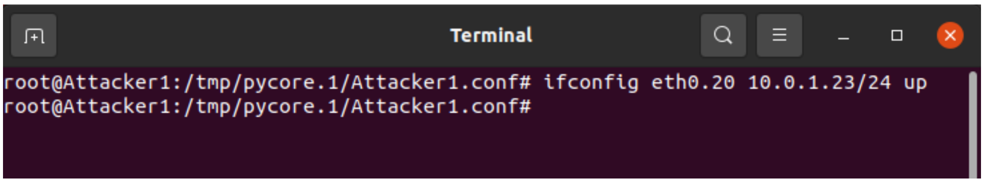
```
root@Server:/tmp/pycore.1/Server.conf# ifconfig eth0.10 10.0.1.20/24 up

root@Server:/tmp/pycore.1/Server.conf# 
```

   Client: Assign the Client IP address to the new VLAN interface eth0.10 on the client node as shown below:



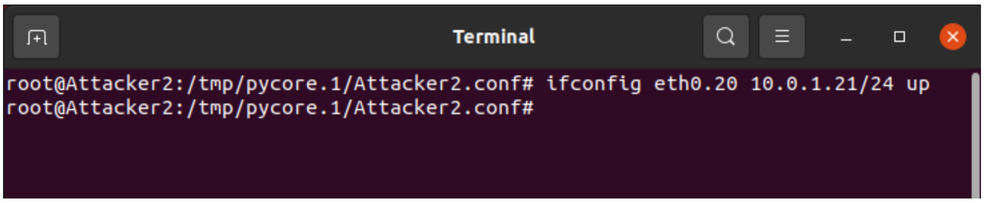
```
root@Client:/tmp/pycore.1/Client.conf# ifconfig eth0.10 10.0.1.22/24 up
root@Client:/tmp/pycore.1/Client.conf#
```

   Attacker 1: Assign the Attacker 1 IP address to the new VLAN interface eth0.20 on the Attacker 1 node as shown below:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# ifconfig eth0.20 10.0.1.23/24 up
root@Attacker1:/tmp/pycore.1/Attacker1.conf#
```

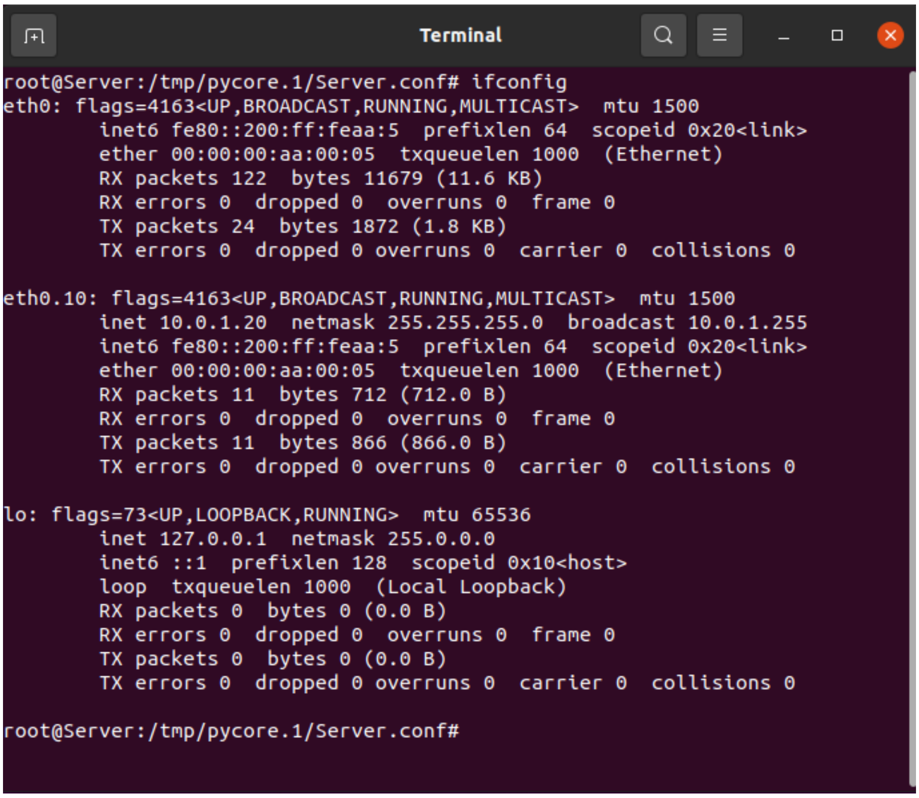
   Attacker 2: Assign the Attacker 2 IP address to the new VLAN interface eth0.20 on the Attacker 2 node as shown below:



```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# ifconfig eth0.20 10.0.1.21/24 up
root@Attacker2:/tmp/pycore.1/Attacker2.conf#
```

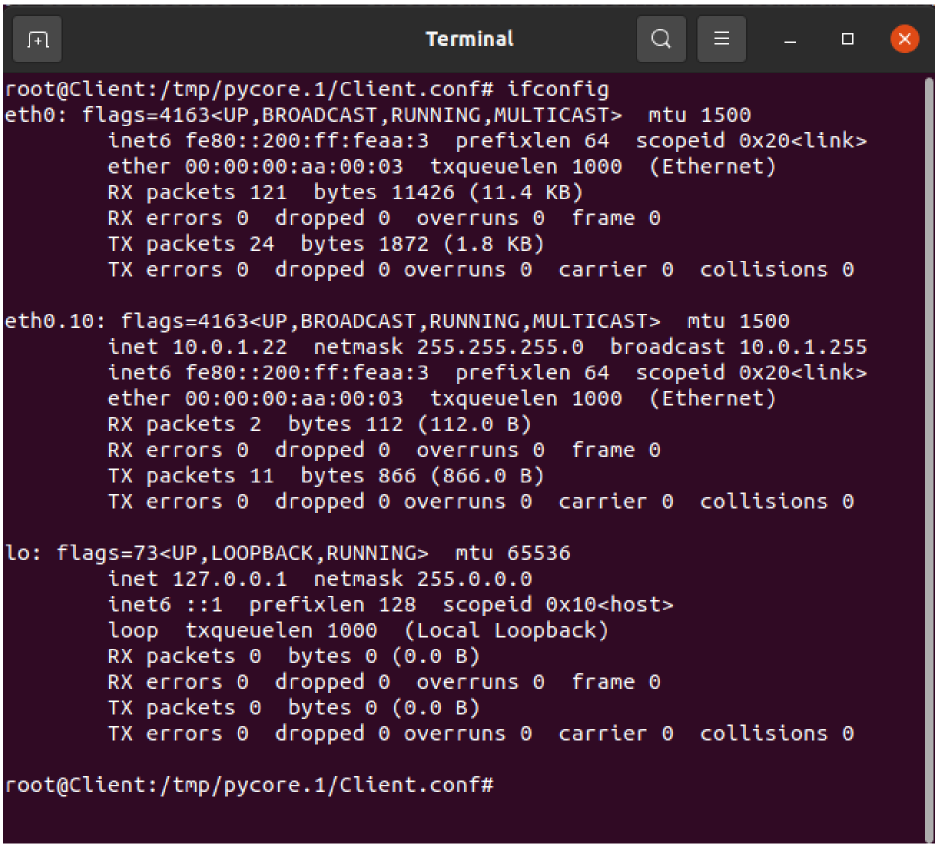
14. **(1 point) Run the command *ifconfig* on each of the 4 nodes and show a screenshot of the results.**

Server: Running the command **ifconfig** on server node to observe that it is connected to eth0.10 VLAN interface as shown below:



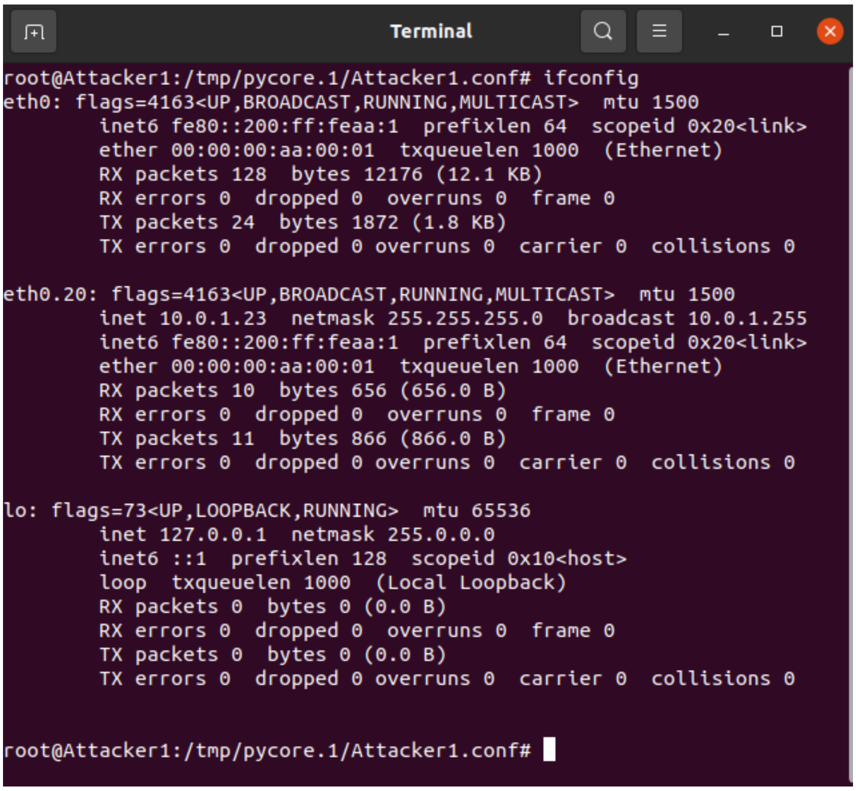
```
root@Server:/tmp/pycore.1/Server.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:5  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:05  txqueuelen 1000  (Ethernet)
        RX packets 122  bytes 11679 (11.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1872 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.20  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::200:ff:feaa:5  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:05  txqueuelen 1000  (Ethernet)
        RX packets 11  bytes 712 (712.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Server:/tmp/pycore.1/Server.conf#
```

Client: Running the command **ifconfig** on client node to observe that it is connected to eth0.10 VLAN interface as shown below:



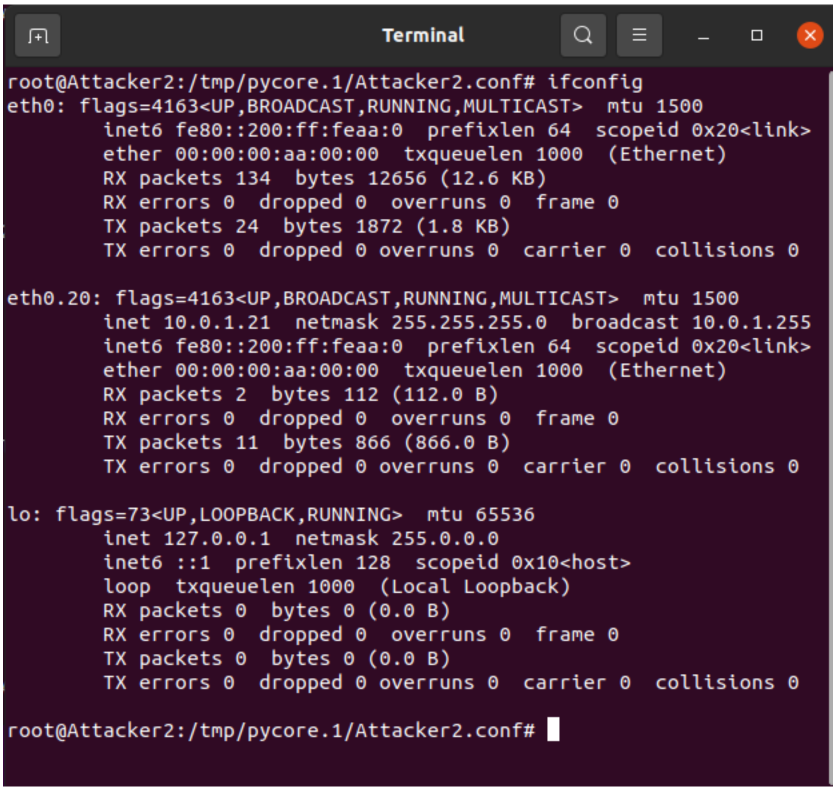
```
root@Client:/tmp/pycore.1/Client.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:3  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 121  bytes 11426 (11.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1872 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.22  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::200:ff:feaa:3  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:03  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 112 (112.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Client:/tmp/pycore.1/Client.conf#
```

Attacker 1: Running the command **ifconfig** on Attacker 1 node to observe that it is connected to eth0.20 VLAN interface as shown below:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:1  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:01  txqueuelen 1000  (Ethernet)
        RX packets 128  bytes 12176 (12.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1872 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.20: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.23  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::200:ff:feaa:1  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:01  txqueuelen 1000  (Ethernet)
        RX packets 10  bytes 656 (656.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


root@Attacker1:/tmp/pycore.1/Attacker1.conf# 
```

Attacker 2: Running the command **ifconfig** on Attacker 2 node to observe that it is connected to eth0.20 VLAN interface as shown below:

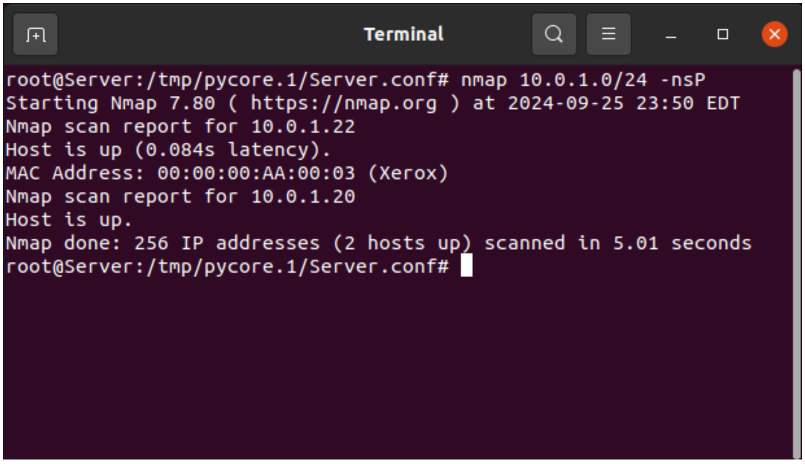


```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::200:ff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 134  bytes 12656 (12.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1872 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0.20: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.21  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::200:ff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 2  bytes 112 (112.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 866 (866.0 B)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Attacker2:/tmp/pycore.1/Attacker2.conf# 
```

15. (2.5 point) **Repeat steps 5 through 10 and record your answer.**

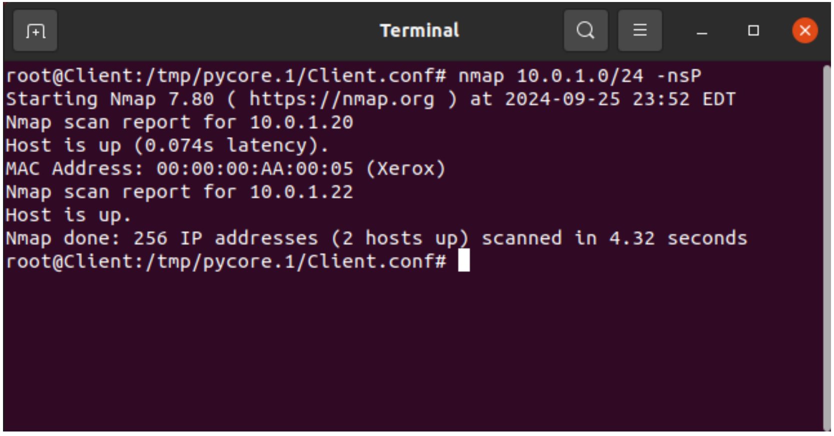
Command used: **nmap 10.0.1.0/24 -nsP**

Nmap from Server: The Nmap scan conducted on the Server after adding the VLAN interface reported that Server can now see only 2 active hosts – Server and the Client, which are up and running as shown below:



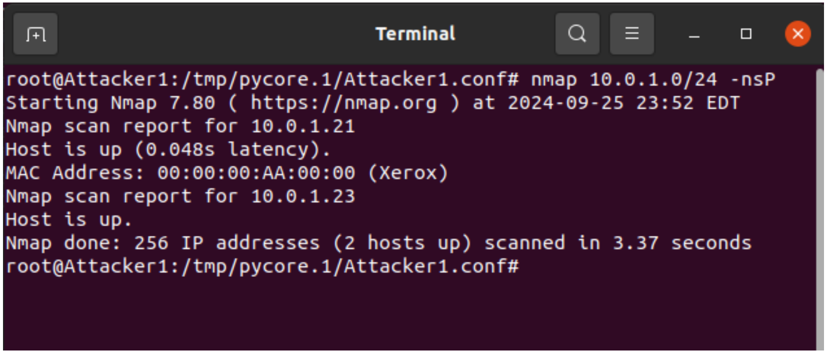
```
root@Server:/tmp/pycore.1/Server.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 23:50 EDT
Nmap scan report for 10.0.1.22
Host is up (0.084s latency).
MAC Address: 00:00:00:AA:00:03 (Xerox)
Nmap scan report for 10.0.1.20
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 5.01 seconds
root@Server:/tmp/pycore.1/Server.conf# 
```

Nmap from Client: The Nmap scan conducted on the Client after adding the VLAN interface reported that Client can now see only 2 active hosts – Client and the Server, which are up and running as shown below:



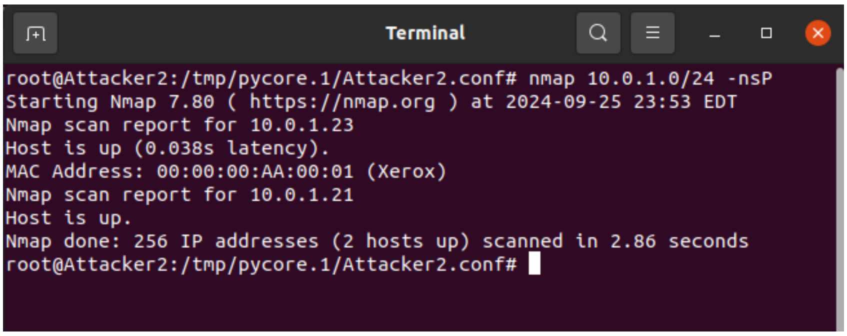
```
root@Client:/tmp/pycore.1/Client.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 23:52 EDT
Nmap scan report for 10.0.1.20
Host is up (0.074s latency).
MAC Address: 00:00:00:AA:00:05 (Xerox)
Nmap scan report for 10.0.1.22
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 4.32 seconds
root@Client:/tmp/pycore.1/Client.conf# 
```

Nmap from Attacker 1: The Nmap scan conducted on the Attacker 1 after adding the VLAN interface reported that Attacker 1 can now see only 2 active hosts –Attacker 1 and the Attacker 2, which are up and running as shown below:



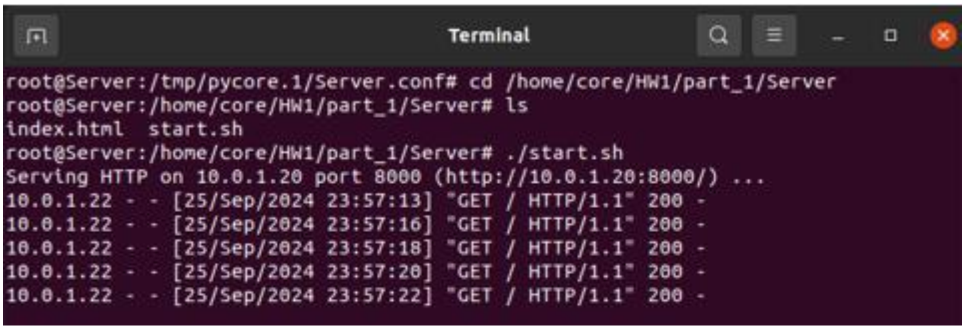
```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 23:52 EDT
Nmap scan report for 10.0.1.21
Host is up (0.048s latency).
MAC Address: 00:00:00:AA:00:00 (Xerox)
Nmap scan report for 10.0.1.23
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.37 seconds
root@Attacker1:/tmp/pycore.1/Attacker1.conf#
```

Nmap from Attacker 2: The Nmap scan conducted on the Attacker 2 after adding the VLAN interface reported that Attacker 2 can now see only 2 active hosts – Attacker 2 and the Attacker 1, which are up and running as shown below:



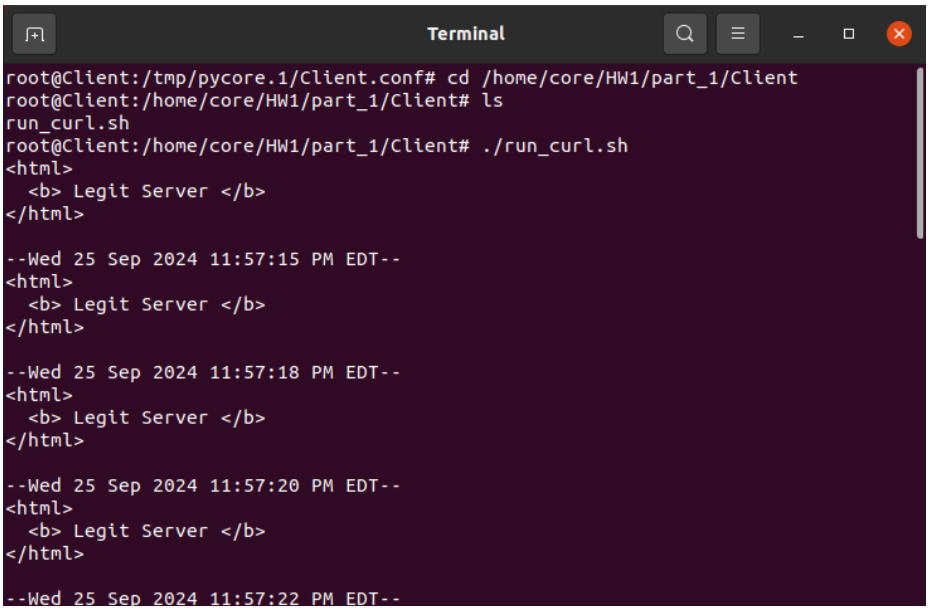
```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# nmap 10.0.1.0/24 -nsP
Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-25 23:53 EDT
Nmap scan report for 10.0.1.23
Host is up (0.038s latency).
MAC Address: 00:00:00:AA:00:01 (Xerox)
Nmap scan report for 10.0.1.21
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 2.86 seconds
root@Attacker2:/tmp/pycore.1/Attacker2.conf#
```

Run HTTP Server from HW 1 on Server: Navigate to start.sh script from HW 1 in the Server node and host the HTTP server using **./start.sh** command. Here, we can see that the server is hosting the HTTP Server as shown below:



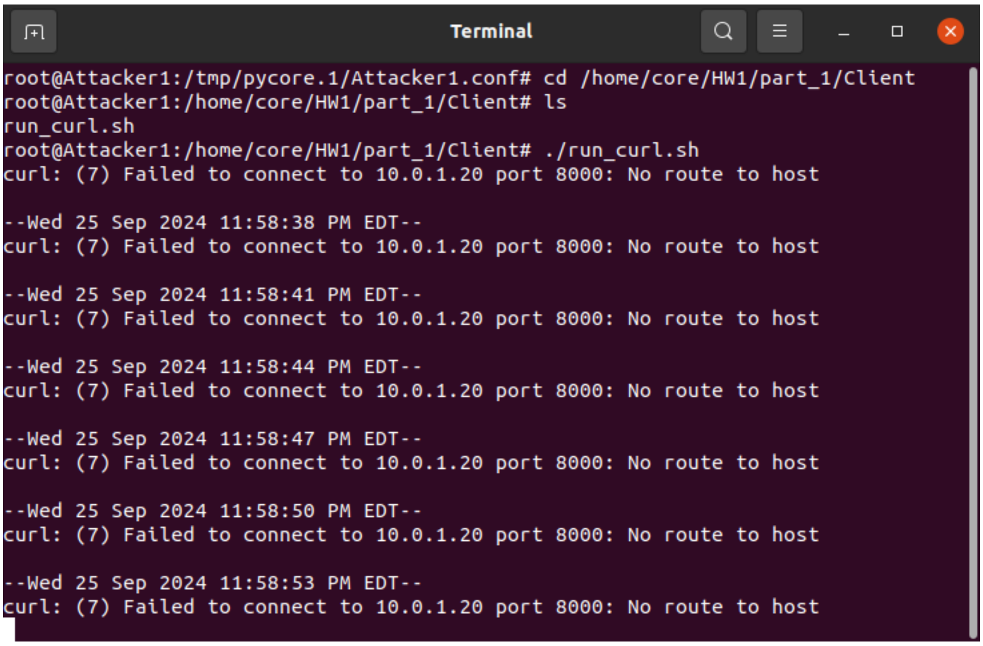
```
root@Server:/tmp/pycore.1/Server.conf# cd /home/core/HW1/part_1/Server
root@Server:/home/core/HW1/part_1/Server# ls
index.html  start.sh
root@Server:/home/core/HW1/part_1/Server# ./start.sh
Serving HTTP on 10.0.1.20 port 8000 (http://10.0.1.20:8000/) ...
10.0.1.22 - - [25/Sep/2024 23:57:13] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 23:57:16] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 23:57:18] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 23:57:20] "GET / HTTP/1.1" 200 -
10.0.1.22 - - [25/Sep/2024 23:57:22] "GET / HTTP/1.1" 200 -
```

Run run_curl.sh from HW 1 on Client: Navigate to run_curl.sh script from HW 1 in the Client node and run the client script using the **./run_curl.sh** command. Here, we can see that the Client is running the curl script and receiving HTTP content from the server as shown below:



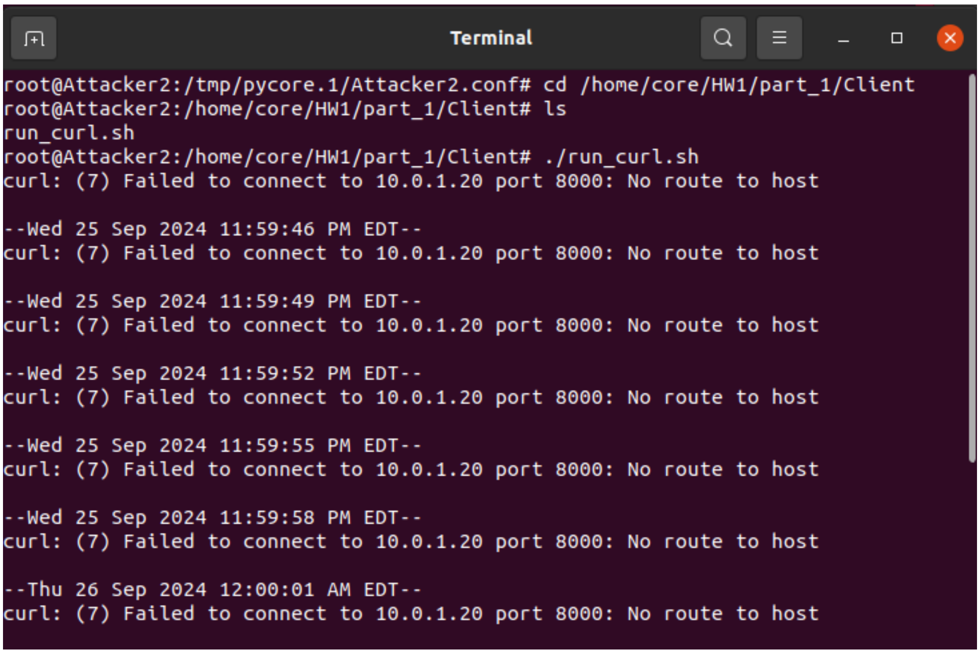
```
root@Client:/tmp/pycore.1/Client.conf# cd /home/core/HW1/part_1/Client
root@Client:/home/core/HW1/part_1/Client# ls
run_curl.sh
root@Client:/home/core/HW1/part_1/Client# ./run_curl.sh
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 11:57:15 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 11:57:18 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 11:57:20 PM EDT--
<html>
  <b> Legit Server </b>
</html>

--Wed 25 Sep 2024 11:57:22 PM EDT--
```

Attacker 1 trying to connect to client: Navigate to run_curl.sh script from HW 1 in the Attacker 1 node and run the client script using the **./run_curl.sh** command. Here, we can see that the attacker 1 is running the curl script but cannot connect to the server since it is in a different VLAN eth0.20 interface than the server which is in VLAN eth0.10 interface as shown:



```
root@Attacker1:/tmp/pycore.1/Attacker1.conf# cd /home/core/HW1/part_1/Client
root@Attacker1:/home/core/HW1/part_1/Client# ls
run_curl.sh
root@Attacker1:/home/core/HW1/part_1/Client# ./run_curl.sh
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:38 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:41 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:44 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:47 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:50 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:58:53 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host
```

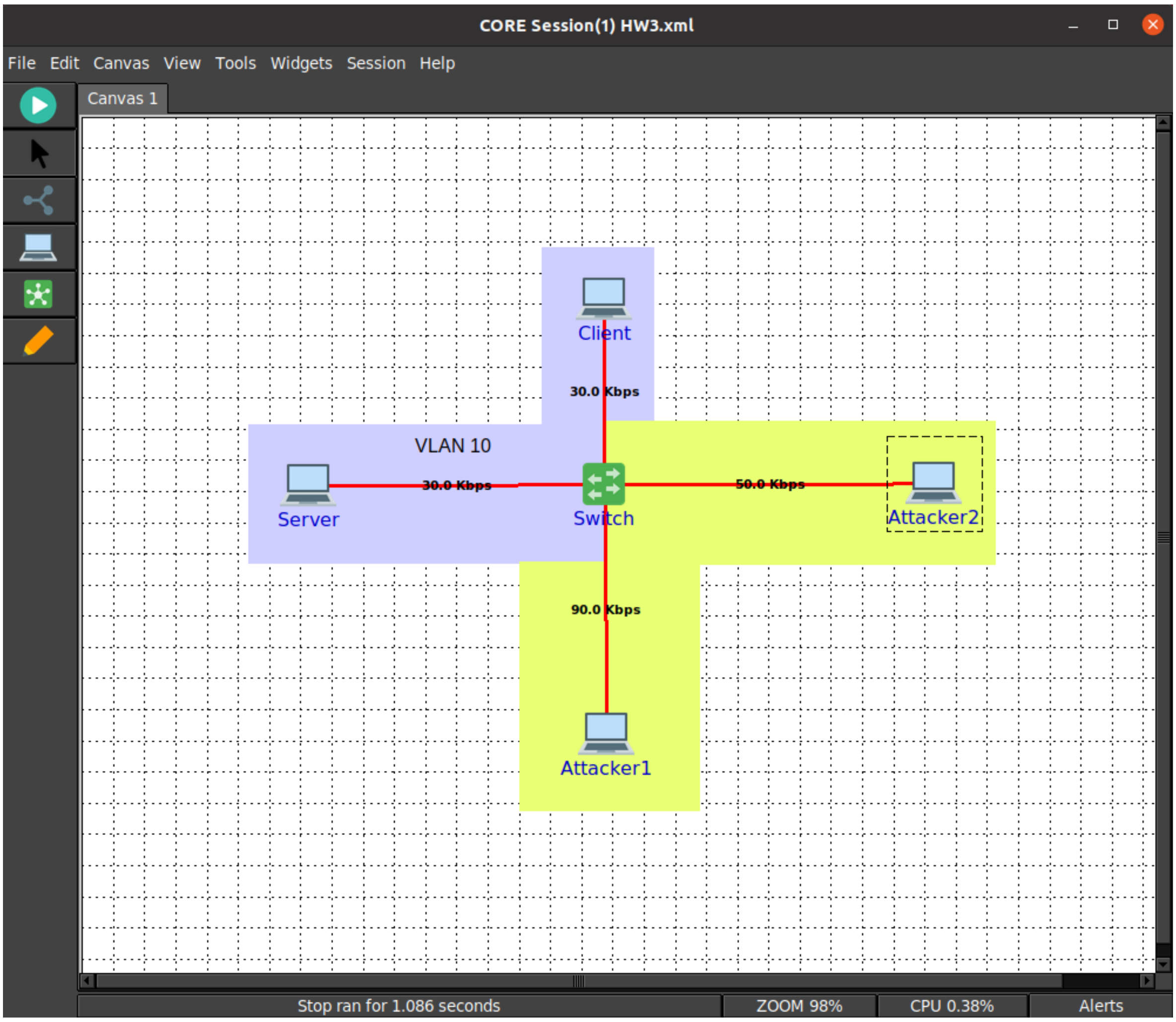
Attacker 2 trying to connect to client: Navigate to run_curl.sh script from HW 1 in the Attacker 2 node and run the client script using the **./run_curl.sh** command. Here, we can see that the attacker 2 is running the curl script but cannot connect to the server since it is in a different VLAN eth0.20 interface than the server which is in VLAN eth0.10 interface as shown:



```
root@Attacker2:/tmp/pycore.1/Attacker2.conf# cd /home/core/HW1/part_1/Client
root@Attacker2:/home/core/HW1/part_1/Client# ls
run_curl.sh
root@Attacker2:/home/core/HW1/part_1/Client# ./run_curl.sh
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:59:46 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:59:49 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:59:52 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:59:55 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Wed 25 Sep 2024 11:59:58 PM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host

--Thu 26 Sep 2024 12:00:01 AM EDT--
curl: (7) Failed to connect to 10.0.1.20 port 8000: No route to host
```

Here, Attacker 1 and Attacker 2 both fail to connect to the client since they are from different VLAN interface. This confirms that 2 separate VLAN interfaces are set up correctly and the two subnets have been successfully created.

Stopping the core scenario:



Stop ran for 1.086 seconds