Group 91 Lab 8: Firewall Evasion

TASK 0

> IP address 10.8.0 does not have an interface name of eth0, will rename to br-b2b21ba3ae94. Conversely, IP address 192.168.20 does not have an interface name of eth1, will rename to br-58463cbfc40a

```
[11/30/24]seed@VM:~/.../Labsetup$ ip -br address
lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24 fd00::c17e:9c05:6072:f
3e3/64 fd00::e6e6:ad58:2fca:e494/64 fe80::1253:d4d4:e652:dde0/64
docker0          DOWN           172.17.0.1/16
br-b2b21ba3ae94  DOWN           10.8.0.1/24
br-ecc6968f3729  DOWN           192.168.50.1/24
br-15511391c179  DOWN           10.9.0.1/24
br-58463cbfc40a  DOWN           192.168.20.1/24
br-98f5bb75a5c4  DOWN           192.168.60.1/24
[11/30/24]seed@VM:~/.../Labsetup$
```

> Adding two more IP addresses to block:

www.instagram.com - 157.240.22.174

https://www.youtube.com/ - 142.251.46.174

```
command: bash -c "
               ip route del default  &&
               ip route add default via 10.8.0.1 &&
               iptables -t nat -A POSTROUTING ! -d 10.8.0.0/24 -j MASQUERADE -o br-b2b21ba3ae94 &&
               iptables -A FORWARD -i br-b2b21ba3ae94 -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT &&
               iptables -A FORWARD -i br-b2b21ba3ae94 -p tcp --dport 22 -j ACCEPT &&
               iptables -A FORWARD -i br-b2b21ba3ae94 -p tcp -j DROP &&
               iptables -A FORWARD -i br-58463cbfc40a  -d 93.184.216.0/24 -j DROP &&

               iptables -A FORWARD -i br-58463cbfc40a -d 142.251.46.174 -j DROP &&
               iptables -A FORWARD -i br-58463cbfc40a -d 157.240.22.174 -j DROP &&

               /etc/init.d/openbsd-inetd start  &&
               tail -f /dev/null
          "
```

TASK 1

> SSH-ing to create a static port forwarding tunnel from host A to host B

ssh -4NT -L 10.8.0.99:7000:192.168.20.5:23 seed@192.168.20.99

```
root@98bc81f83766:/# ssh -4NT -L 10.8.0.99:7000:192.168.20.5:23 seed@192.168.20.99
The authenticity of host '192.168.20.99 (192.168.20.99)' can't be established.
ECDSA key fingerprint is SHA256:sNepY8+aL8Ad20SGWEfdE3YvvVJmhVDjjdEaydvNmj8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.20.99' (ECDSA) to the list of known hosts.
seed@192.168.20.99's password:
```

> Given the following list of addresses, we will prove there is a tunnel between the external and internal network by telnetting into server B1 from A, A1 and A2

```
[11/30/24]seed@VM:~/.../Labsetup$ dockps
98bc81f83766  A-10.8.0.99
6a0b6f34028c  router-firewall
8dce387c460c  A2-10.8.0.6
64a793ec129b  B2-192.168.20.6
7f702b589c0a  B-192.168.20.99
d1270108e36b  A1-10.8.0.5
fe23a685c32e  B1-192.168.20.5
```

Telnetting from A to B1

```
[11/30/24]seed@VM:~/.../Labsetup$ docksh 98bc
root@98bc81f83766:/# telnet 192.168.20.5 23
Trying 192.168.20.5...
Connected to 192.168.20.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fe23a685c32e login:
```

Telnetting from A1 to B1

```
[11/30/24]seed@VM:~/.../Labsetup$ docksh d127
root@d1270108e36b:/# telnet 192.168.20.5 23
Trying 192.168.20.5...
Connected to 192.168.20.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fe23a685c32e login:
```

Telnetting from A2 to B1

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh 8dce
root@8dce387c460c:/# telnet 192.168.20.5 23
Trying 192.168.20.5...
Connected to 192.168.20.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fe23a685c32e login:
```

As observed the connection from telnetting to B1 from A, A1 and A2 has proven successful, thus implying that there is indeed a tunnel between the external and internal network

Question 1)

By using the command *tcpdump*, it is evident that there are two TCP connections involved in the process:

- The connection of the client to the port forwarder, shown with:
  - *192.168.20.5.telnet > 8dce387c460c.40674t* and vice versa
- The connection of the port forwarder to the destination, show with:
  - *8dce387c460c.54112 > 10.0.2.3.domain*

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh 8dce
root@8dce387c460c:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
05:32:55.187764 IP 8dce387c460c.40674 > 192.168.20.5.telnet: Flags [S], seq 787696
921, win 64240, options [mss 1460,sackOK,TS val 2337045131 ecr 0,nop,wscale 7], le
ngth 0
05:32:55.188194 IP 192.168.20.5.telnet > 8dce387c460c.40674: Flags [S.], seq 18310
17508, ack 787696922, win 65160, options [mss 1460,sackOK,TS val 457289177 ecr 233
7045131,nop,wscale 7], length 0
05:32:55.188219 IP 8dce387c460c.40674 > 192.168.20.5.telnet: Flags [.], ack 1, win
 502, options [nop,nop,TS val 2337045132 ecr 457289177], length 0
05:32:55.198029 ARP, Request who-has 10.8.0.1 tell 8dce387c460c, length 28
05:32:55.198119 ARP, Reply 10.8.0.1 is-at 02:42:05:1b:29:c4 (oui Unknown), length
28
05:32:55.198135 IP 8dce387c460c.54112 > 10.0.2.3.domain: 8001+ PTR? 5.20.168.192.i
n-addr.arpa. (43)
05:32:55.217143 IP 10.0.2.3.domain > 8dce387c460c.54112: 8001 NXDomain* 0/0/0 (43)
05:32:55.219848 IP 8dce387c460c.38072 > 10.0.2.3.domain: 58111+ PTR? 1.0.8.10.in-a
ddr.arpa. (39)
```

Question 2)

By using a static port forwarding tunnel, we can redirect traffic into a different port, avoiding the firewall rules set up for a specific port, which was evaded by the tunnel. For example, a firewall might block the traffic to IP 108.11.2, but not to IP 12.2.2.1. The port forwarding tunnel would remap the former IP address to the latter, allowing it to avoid the firewall rules.

TASK 2.1

Connecting B to A using ssh.

```
root@0f7b79d3211b:/# ssh -4NT -D 192.168.20.99:7200 seed@10.8.0.99
seed@10.8.0.99's password:
```

Running the curl command on host B while in B1's root

```
root@f9454c5364c1:/# curl --proxy socks5h://192.168.20.99:7200 www.exa
mple.com
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"
 />
    <meta name="viewport" content="width=device-width, initial-scale=1
" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Se
goe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
```

This results in being able to view the html page for example.com, connection successful.
We test again while in B2's root, resulting in a successful connection.

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh 8f7
root@8f705d5d5ff4:/# curl --proxy socks5h://192.168.20.99:7200 www.exa
mple.com
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"
 />
    <meta name="viewport" content="width=device-width, initial-scale=1
" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
```

Question 1

In the dynamic port forwarding setup, the computer that establishes the actual connection with the intended web server is **host A**. Host A serves as the other end of the SSH tunnel, forwarding the requests received from the internal network to the intended web server.

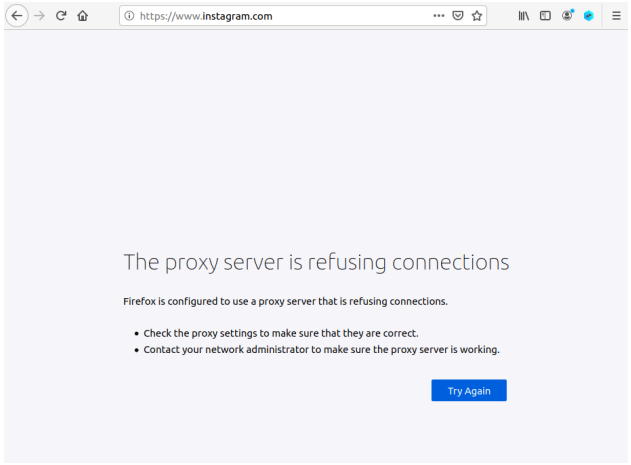
Question 2

Host A determines the intended web server based on the data provided through the SOCKS5 protocol by host B (acting as a proxy). When a client on the internal network (host B, B1, or B2) sends a request using the proxy, the destination details (server's address and port) are included in the SOCKS5 protocol communication. Host A reads these details from the SOCKS5 communication and establishes a connection to the specified destination server.

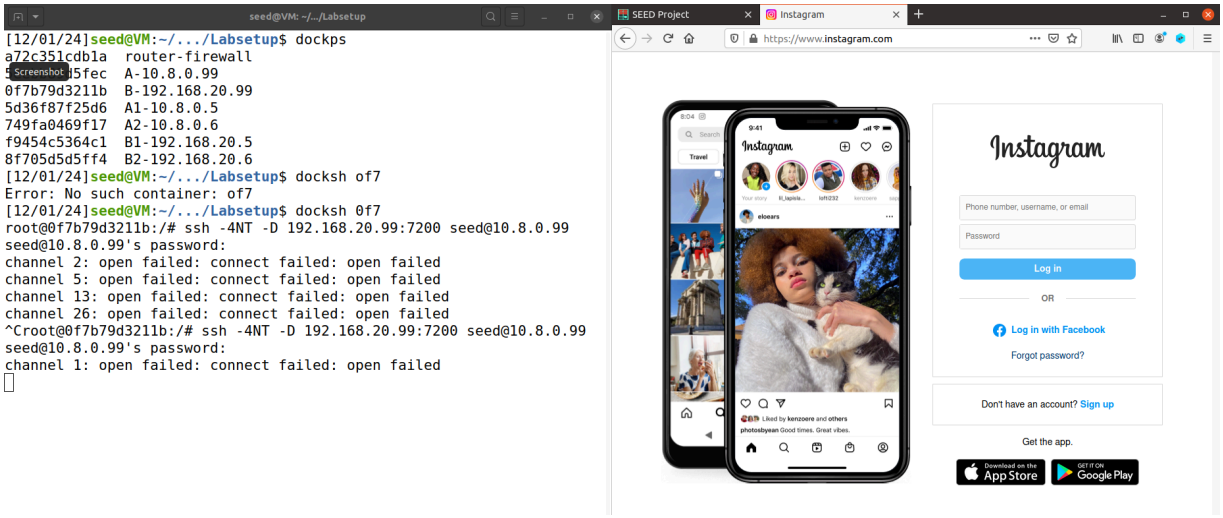
TASK 2.2

I was unable to find anything notable when running tcpdump while in the server-router.

After configuring the browser's proxy, it is no longer able reach the instragram connection as show below



This is not the case when we ssh



TASK 2.3

```
#!/bin/env python3
```

Screenshot

```
import socks

s = socks.socksocket()

# Set the proxy
s.set_proxy(socks.SOCKS5, "192.168.20.99", 7200)

# Connect to final destination via the proxy
hostname = "www.example.com"
s.connect((hostname, 80))

request = b"GET / HTTP/1.0\r\nHost: " + hostname.encode('utf-8') + b"\
r\n\r\n"
s.sendall(request)

# Get the response
response = s.recv(2048)
while response:
  print(response.split(b"\r\n"))
  response = s.recv(2048)
~
```

Modified the port to be 7200

After running the provided python file in LabSetup with one modification to the port name, the following is outputted.

This is B1's connection

```
root@f9454c5364c1:/home/seed# ./SocksClient.py
[b'HTTP/1.0 200 OK', b'Age: 482253', b'Cache-Control: max-age=604800',
 b'Content-Type: text/html; charset=UTF-8', b'Date: Mon, 02 Dec 2024 0
4:30:35 GMT', b'Etag: "3147526947+gzip+ident"', b'Expires: Mon, 09 Dec
 2024 04:30:35 GMT', b'Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT',
b'Server: ECAcc (chd/0768)', b'Vary: Accept-Encoding', b'X-Cache: HIT'
, b'Content-Length: 1256', b'Connection: close', b'', b'<!doctype html
>\n<html>\n<head>\n    <title>Example Domain</title>\n\n    <meta char
set="utf-8" />\n    <meta http-equiv="Content-type" content="text/html
; charset=utf-8" />\n    <meta name="viewport" content="width=device-w
idth, initial-scale=1" />\n    <style type="text/css">\n    body {\n
      background-color: #f0f0f2;\n        margin: 0;\n        padding:
 0;\n        font-family: -apple-system, system-ui, BlinkMacSystemFont
, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-se
rif;\n        \n    }\n    div {\n        width: 600px;\n        margi
n: 5em auto;\n        padding: 2em;\n        background-color: #fdfdff
;\n        border-radius: 0.5em;\n        box-shadow: 2px 3px 7px 2px
rgba(0,0,0,0.02);\n    }\n    a:link, a:visited {\n        color: #384
88f;\n        text-decoration: none;\n    }\n    @media (max-width: 70
0px) {\n        div {\n            margin: 0 auto;\n            width:
 auto;\n        }\n    }\n    </style>    \n</head>\n\n<body>\n<div>\n
    <h1>Example Domain</h1>\n    <p>This domain is for use in illustra
tive examples in documents. You may use this\n    domain in literature
 without prior coordination or asking for permission.</p>\n    <p><a h
ref="https://www.iana.org/domains/example">More information...</a></p>
\n</div>\n</body>\n</html>\n']
```

B2's Connection

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh 8f7
root@8f705d5d5ff4:/# cd home/seed
root@8f705d5d5ff4:/home/seed# ls
root@8f705d5d5ff4:/home/seed# nano
root@8f705d5d5ff4:/home/seed# chmod u+x SocksClient.py
root@8f705d5d5ff4:/home/seed# ./SocksClient.py
[b'HTTP/1.0 200 OK', b'Age: 386299', b'Cache-Control: max-age=604800',
 b'Content-Type: text/html; charset=UTF-8', b'Date: Mon, 02 Dec 2024 0
4:53:22 GMT', b'Etag: "3147526947+ident"', b'Expires: Mon, 09 Dec 2024
 04:53:22 GMT', b'Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT', b'Ser
ver: ECAcc (chd/0757)', b'Vary: Accept-Encoding', b'X-Cache: HIT', b'C
ontent-Length: 1256', b'Connection: close', b'', b'<!doctype html>\n<h
tml>\n<head>\n    <title>Example Domain</title>\n\n    <meta charset="
utf-8" />\n    <meta http-equiv="Content-type" content="text/html; cha
rset=utf-8" />\n    <meta name="viewport" content="width=device-width,
 initial-scale=1" />\n    <style type="text/css">\n    body {\n
 background-color: #f0f0f2;\n        margin: 0;\n        padding: 0;\n
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Se
goe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;\
n        \n    }\n    div {\n        width: 600px;\n        margin: 5e
m auto;\n        padding: 2em;\n        background-color: #fdfdff;\n
      border-radius: 0.5em;\n        box-shadow: 2px 3px 7px 2px rgba(
0,0,0,0.02);\n    }\n    a:link, a:visited {\n        color: #38488f;\
n        text-decoration: none;\n    }\n    @media (max-width: 700px)
{\n        div {\n            margin: 0 auto;\n            width: auto
;\n        }\n    }\n    </style>    \n</head>\n\n<body>\n<div>\n    <
h1>Example Domain</h1>\n    <p>This domain is for use in illustrative
examples in documents. You may use this\n    domain in literature with
out prior coordination or asking for permission.</p>\n    <p><a href="
https://www.iana.org/domains/example">More information...</a></p>\n</d
iv>\n</body>\n</html>\n']
```

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh 0f7
root@0f7b79d3211b:/# cd home/seed
root@0f7b79d3211b:/home/seed# nano
root@0f7b79d3211b:/home/seed# chmod u+x SocksCLient.py
root@0f7b79d3211b:/home/seed# ./SocksCLient.py
[b'HTTP/1.0 200 OK', b'Age: 386405', b'Cache-Control: max-age=604800',
 b'Content-Type: text/html; charset=UTF-8', b'Date: Mon, 02 Dec 2024 0
4:55:08 GMT', b'Etag: "3147526947+ident"', b'Expires: Mon, 09 Dec 2024
 04:55:08 GMT', b'Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT', b'Ser
ver: ECAcc (chd/0757)', b'Vary: Accept-Encoding', b'X-Cache: HIT', b'C
ontent-Length: 1256', b'Connection: close', b'', b'<!doctype html>\n<h
tml>\n<head>\n    <title>Example Domain</title>\n\n    <meta charset="
utf-8" />\n    <meta http-equiv="Content-type" content="text/html; cha
rset=utf-8" />\n    <meta name="viewport" content="width=device-width,
 initial-scale=1" />\n    <style type="text/css">\n    body {\n
 background-color: #f0f0f2;\n        margin: 0;\n        padding: 0;\n
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Se
goe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;\
n        \n    }\n    div {\n        width: 600px;\n        margin: 5e
m auto;\n        padding: 2em;\n        background-color: #fdfdff;\n
      border-radius: 0.5em;\n        box-shadow: 2px 3px 7px 2px rgba(
0,0,0,0.02);\n    }\n    a:link, a:visited {\n        color: #38488f;\
n        text-decoration: none;\n    }\n    @media (max-width: 700px)
{\n        div {\n            margin: 0 auto;\n            width: auto
;\n        }\n    }\n    </style>    \n</head>\n\n<body>\n<div>\n    <
h1>Example Domain</h1>\n    <p>This domain is for use in illustrative
examples in documents. You may use this\n    domain in literature with
out prior coordination or asking for permission.</p>\n    <p><a href="
```

← B's Connection

All files output the raw html file that comes from example.com, which means that the connection has been successful.

Please note that I found success when manually creating new SocksClient.py files in each of the different dockershells. You can achieve this by using *docksh <name of the shell you want to enter>* and copying and pasting the provided python file into a new file located in the seed directory. This can be found by inputting the following command:

cd home/seed

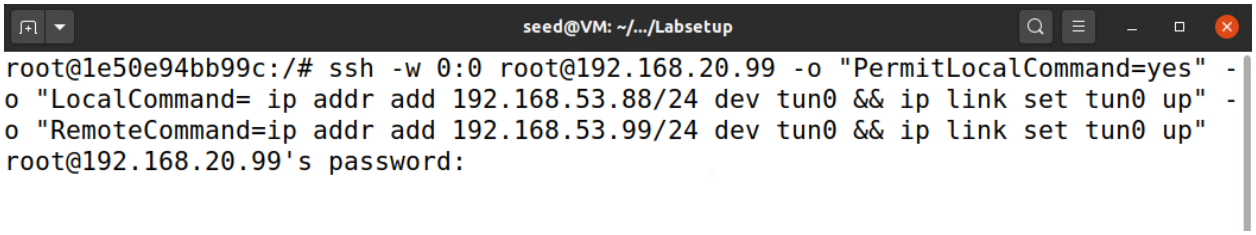
Remember to change the file permissions as well.

chmod u+x <file>

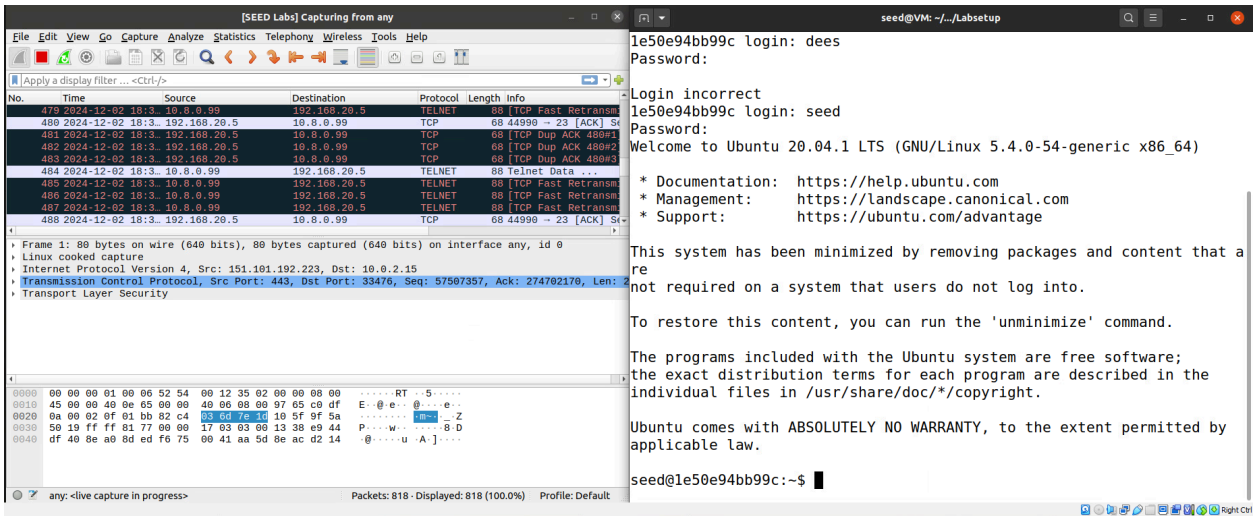
Task 3:

Task 3.1:

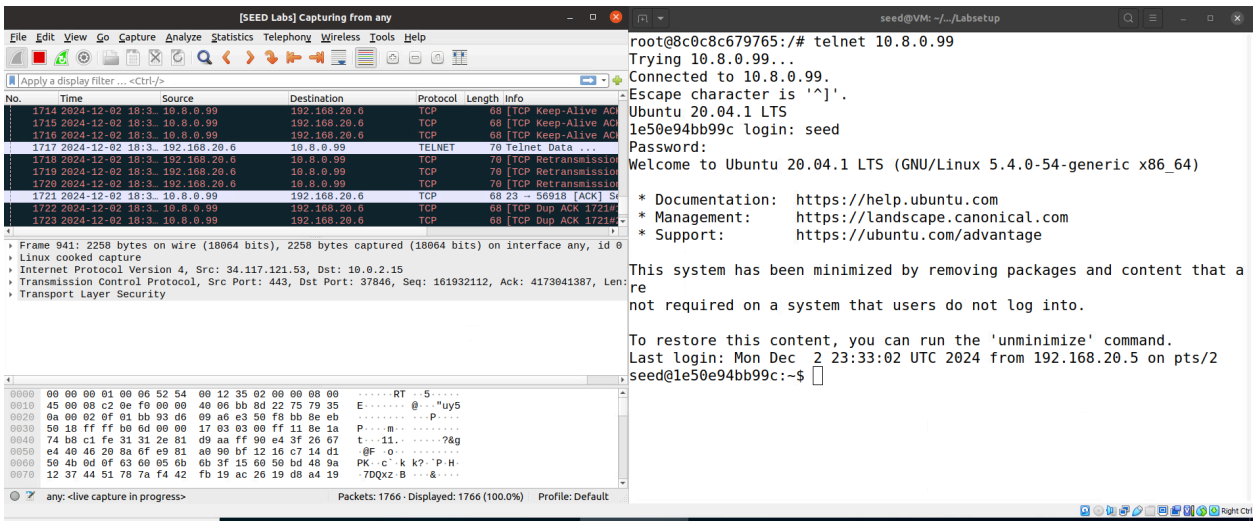
Creation and configuration of the tunnel on host A and host B (server). We enter the password "dees" and successfully connect to server B from A, the external network.



We open Wireshark and run Telnet 10.8.0.99 on the B1 host (192.168.20.5). We notice from the Wireshark data that A can reach B1 (source IP 10.8.0.99 → destination 192.168.20.5 with TELNET protocol).



Now we do the same for B2, by running Telnet 10.8.0.99 from the B2 host. The data sent passes through the VPN tunnel as shown on Wireshark.



The incoming firewall blocks TCP connections outside of the SSH port, but it cannot see the traffic once it is encapsulated in the SSH tunnel.
By creating a VPN tunnel over SSH, all network traffic is sent through this secure tunnel. The firewall cannot identify this traffic because it is treated as standard SSH packets, so no other connections are blocked.
The VPN tunnel hides the true destination and nature of the traffic, allowing access to services that would otherwise be blocked by the firewall.

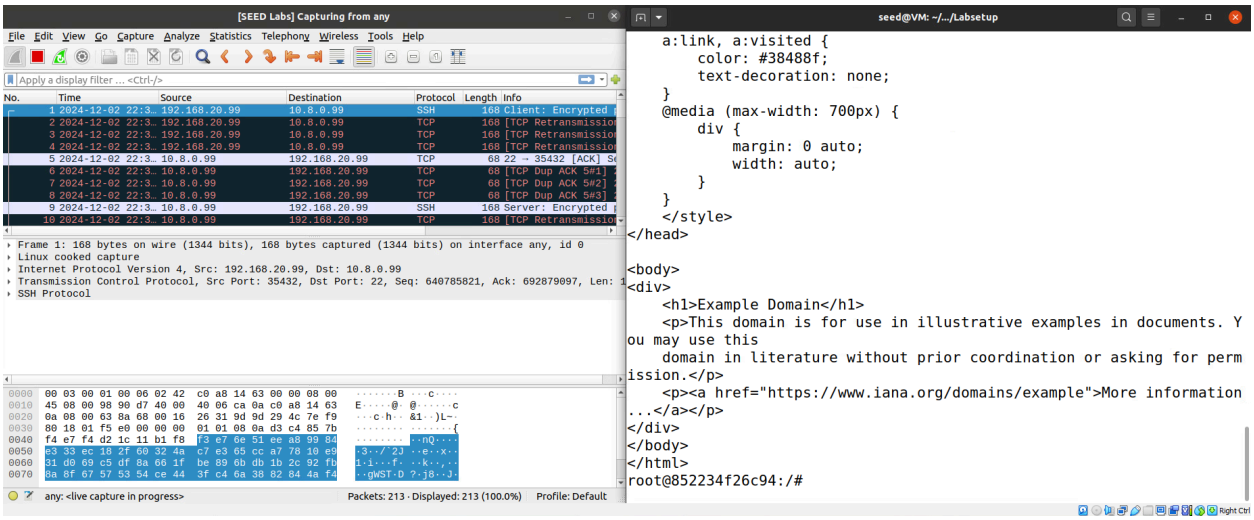
Task 3.2 :

Creating the dynamic tunnel between A and B: This time, A is the server and B is the client; the command is the same as in Task 3.1, but 88 and 99 are swapped.
Then, on A's terminal, the command *iptables -t nat -A POSTROUTING -j MASQUERADE -o eth0* is run to configure NAT on A.
This ensures that packets sent to example.com by B (via the tunnel) are correctly routed and accepted by example.com. The response from example.com will be sent to A, and then forwarded to B via the tunnel.

This command ssh -4NT -D 192.168.53.88:1080 root@10.8.0.99 is used to establish a dynamic SSH tunnel with A (the remote host), creating a SOCKS5 proxy on the address 192.168.53.88 and port 1080. This allows the local host (B) to route its network traffic through A, thereby bypassing the outgoing firewall restrictions.

After executing the curl command, we see that the traffic successfully passes through the SSH tunnel, allowing access to the externally blocked site by the outgoing firewall. This demonstrates that the SOCKS5 proxy is working as intended, redirecting B's network traffic through A (the VPN server), thus bypassing the restrictions imposed by the firewall.

The packets captured during the execution of the curl command show that the HTTP traffic is routed through the VPN tunnel and the SOCKS5 proxy, with the source address of the packets masked by A's IP address. The packets are encrypted, which prevents B's firewall from blocking them, allowing B to access blocked sites. The responses from external servers are correctly sent back to B via the tunnel without being filtered.



Task 4 :

The SOCKS5 Proxy is a lightweight and fast solution for bypassing geographical restrictions and masking the user's IP address. It allows traffic to be redirected without encrypting the data, meaning it performs better in terms of speed but is less secure. Without encryption, the exchanged data can be easily intercepted, making it less suitable for confidentiality needs.

The VPN, on the other hand, provides a higher level of security by encrypting all traffic between the user and the server, thus ensuring the protection of data against interception. This encryption enhances privacy but can slightly impact performance due to the resources required to encrypt and decrypt the data. Additionally, setting up a VPN is often more complex, which can pose a challenge for less experienced users.

In summary, the choice between a SOCKS5 Proxy and a VPN primarily depends on the user's priorities. If performance and simplicity are essential, the SOCKS5 proxy would be the better choice. However, if security, privacy, and complete traffic protection are priorities, the VPN is a much more suitable solution.