CPS 633 Section 09

Firewall Evasion Lab

Group 18

Roxie Reginold (501087897)
Hetu Virajkumar Patel (501215707)
Sayyada Aisha Mehvish (501106795)

Task 0: Get Familiar with the Lab Setup

```
[11/29/24]seed@VM:~/.../Labsetup$ dcup --remove-orphans
Removing orphan container "host-192.168.60.6"
Removing orphan container "client-10.9.0.5"
Removing orphan container "host-192.168.60.5"
Starting B-192.168.20.99 ... done
Starting A-10.8.0.99     ... done
Starting router-firewall ... done
Starting A2-10.8.0.6     ... done
Starting B2-192.168.20.6 ... done
Starting B1-192.168.20.5 ... done
Starting A1-10.8.0.5     ... done
Attaching to B-192.168.20.99, A2-10.8.0.6, B1-192.168.20.5, A1-10.8.0.5, A-10.8.
0.99, B2-192.168.20.6, router-firewall
A-10.8.0.99 |  * Starting internet superserver inetd                    [ OK ]
A1-10.8.0.5 |  * Starting internet superserver inetd                    [ OK ]
A2-10.8.0.6 |  * Starting internet superserver inetd                    [ OK ]
B1-192.168.20.5 |  * Starting internet superserver inetd                [ OK ]
B-192.168.20.99 |  * Starting internet superserver inetd                [ OK ]
B-192.168.20.99 |  * Starting OpenBSD Secure Shell server sshd          [ OK ]
B2-192.168.20.6 |  * Starting internet superserver inetd                [ OK ]
A-10.8.0.99 |  * Starting OpenBSD Secure Shell server sshd             [ OK ]
router-firewall |  * Starting internet superserver inetd                [ OK ]
```

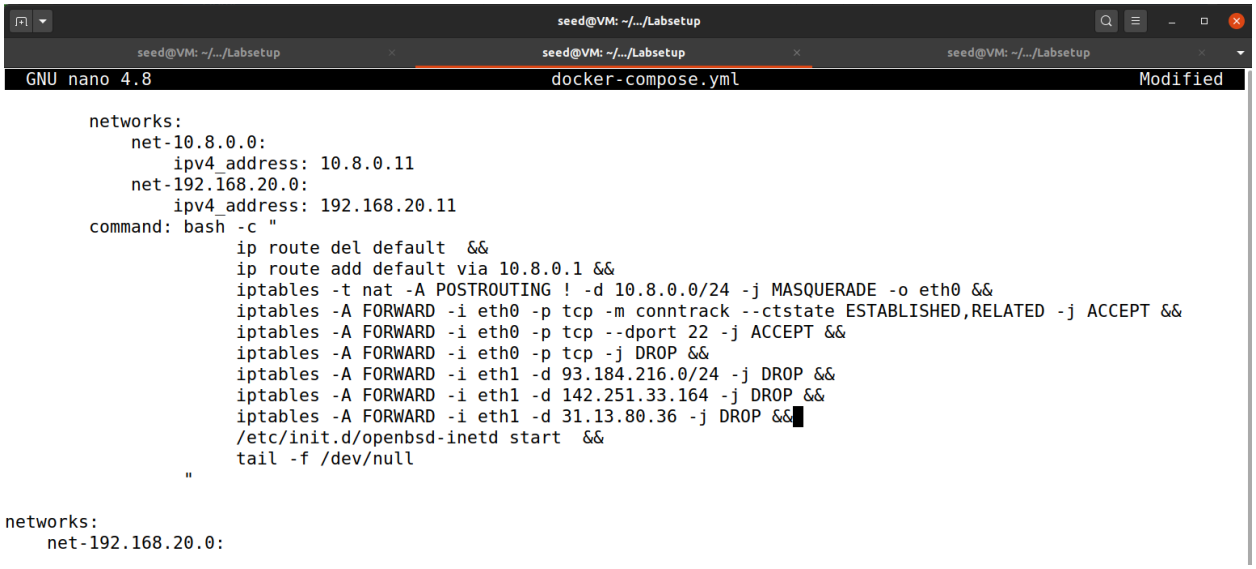
```
root@a732d2924642:/#  ip -br address
lo                UNKNOWN        127.0.0.1/8
eth0@if39         UP             10.8.0.11/24
eth1@if43         UP             192.168.20.11/24
```

```
root@a732d2924642:/# iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 DROP       tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            93.184.216.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Lab task. Please block two more websites and add the firewall rules to the setup files. The choice of websites is up to you. We will use them in one of the tasks. Keep in mind that most popular websites have multiple IP addresses that can change from time to time. After adding the rules, start the containers, and verify that all the ingress and egress firewall rules are working as expected.

```
root@a732d2924642:/# nslookup www.google.com
Server:         127.0.0.11
Address:        127.0.0.11#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.251.33.164
Name:   www.google.com
Address: 2607:f8b0:400b:802::2004

root@a732d2924642:/# nslookup www.facebook.com
Server:         127.0.0.11
Address:        127.0.0.11#53

Non-authoritative answer:
www.facebook.com        canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 31.13.80.36
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f10e:83:face:b00c:0:25de


root@a732d2924642:/# iptables -A FORWARD -i eth1 -d 142.251.33.164 -j DROP
root@a732d2924642:/# iptables -A FORWARD -i eth1 -d 31.13.80.36 -j DROP
root@a732d2924642:/# iptables -L FORWARD -n -v
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0            ctstate RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0            tcp dpt:22
    0     0 DROP       tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            93.184.216.0/24
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            142.251.33.164
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            31.13.80.36
```

Modify the docker file to add those firewall rules before we restart the containers:



```
[11/29/24]seed@VM:~/.../Labsetup$ dcup
Creating network "net-10.8.0.0" with the default driver
Creating network "net-192.168.20.0" with the default driver
Creating B1-192.168.20.5 ... done
Creating A-10.8.0.99     ... done
Creating A1-10.8.0.5     ... done
Creating B2-192.168.20.6 ... done
Creating B-192.168.20.99 ... done
Creating A2-10.8.0.6     ... done
Creating router-firewall ... done
Attaching to A-10.8.0.99, B1-192.168.20.5, B2-192.168.20.6, A1-10.8.0.5, A2-10.8.0.6, router-firewall, B-192.168.20.99
A-10.8.0.99 |  * Starting internet superserver inetd                    [ OK ]
A-10.8.0.99 |  * Starting OpenBSD Secure Shell server sshd               [ OK ]
A1-10.8.0.5 |  * Starting internet superserver inetd                    [ OK ]
A2-10.8.0.6 |  * Starting internet superserver inetd                    [ OK ]
B1-192.168.20.5 |  * Starting internet superserver inetd                [ OK ]
B2-192.168.20.6 |  * Starting internet superserver inetd                [ OK ]
router-firewall |  * Starting internet superserver inetd                [ OK ]
B-192.168.20.99 |  * Starting internet superserver inetd                [ OK ]
B-192.168.20.99 |  * Starting OpenBSD Secure Shell server sshd           [ OK ]
```

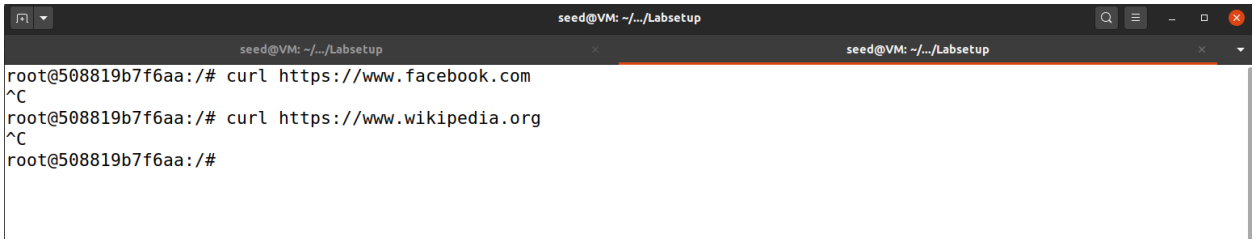
We enter into container B to test the internal network:

```
root@b2ace954eb63:/# curl https://www.facebook.com
c^C
root@b2ace954eb63:/# curl https://www.google.com
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-CA"><head><meta content="text/html; ch
arset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" it
emprop="image"><title>Google</title><script nonce="geul0LA_zcluGrzysaTDKQ">(function(){var _g={kEI:'MNlJZ5inCqXW5NoPy7
Ca4QE',kEXPI:'0,793344,2906978,620,442,538661,2872,2891,8349,34679,30022,54138,40185,236970,18648,10960,23351,22435,97
79,38677,23980,92025,1804,7734,13277,14257,2414,9400,1633,29278,27083,11206181,2842058,44,1,6,3,6,1,6,1,16,4,17,44,4,2
,2,2,2,14,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,1,1,1,11,1,1,1,4,2,1,1,1,1,2,1,1,1,2,1,1,1,1,1,2,1,1,1,3,2,1,1,1,4,1,1,2,1,1
,2,2,2,1,2,3,1,1,1,1,1,1,1,1,5,2,1,1,1,1,1,1,89,20,22,32,3,19,3,3,6,3,3,3,3,22,1,6,5,10,66,11,1,1,1,1,1,1,1,1,27978377,1
6673,25199209,8163,4636,14986,1450,84045,11643,10979,15165,8181,5941,43489,5967,15708,6749,22594,1285,9138,4600,328,44
```

Seems like google returned something because of its use of dynamic and multiple IP ranges. We will replace google.com with www.wikipedia.org

Modifying the docker file and restarting the containers:

```
                    ip route add default via 10.8.0.1 &&
                    iptables -t nat -A POSTROUTING ! -d 10.8.0.0/24 -j MASQUERADE -o eth0 &&
                    iptables -A FORWARD -i eth0 -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT &&
                    iptables -A FORWARD -i eth0 -p tcp --dport 22 -j ACCEPT &&
                    iptables -A FORWARD -i eth0 -p tcp -j DROP &&
                    iptables -A FORWARD -i eth1 -d 93.184.216.0/24 -j DROP &&
                    iptables -A FORWARD -i eth1 -d 208.80.154.224 -j DROP &&
                    iptables -A FORWARD -i eth1 -d 31.13.80.36 -j DROP &&
                    /etc/init.d/openbsd-inetd start  &&
                    tail -f /dev/null
                "
```



Observation: Currently, when we attempt to ping, nothing is displayed in the terminal because the default firewall is blocking the requests.

Testing the Ingress Rules:
Test SSH Traffic

```
[11/29/24]seed@VM:~/.../Labsetup$ docksh A-10.8.0.99
root@08789cd1407b:/# ssh 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:SWOfIaxcHX2QJXJJOUqir1sz+P84sduAtwuEfKhAyJI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.20.99' (ECDSA) to the list of known hosts.
root@192.168.20.99's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)
```

Test Non-SSH Traffic
Sending traffic from A to B doing a port not 22

```
[11/29/24]seed@VM:~/.../Labsetup$ docksh A-10.8.0.99
root@08789cd1407b:/# telnet 192.168.20.99 80
Trying 192.168.20.99...
```

Fails because for Port 80 is not stated as allowed by the given firewall rules, so the connection fails

Test Established/Related Connections
From B I will establish a connection with A.

```
root@508819b7f6aa:/# ping 10.8.0.99
PING 10.8.0.99 (10.8.0.99) 56(84) bytes of data.
64 bytes from 10.8.0.99: icmp_seq=1 ttl=63 time=0.097 ms
64 bytes from 10.8.0.99: icmp_seq=2 ttl=63 time=0.175 ms
64 bytes from 10.8.0.99: icmp_seq=3 ttl=63 time=0.168 ms
64 bytes from 10.8.0.99: icmp_seq=4 ttl=63 time=0.332 ms
64 bytes from 10.8.0.99: icmp_seq=5 ttl=63 time=0.132 ms
64 bytes from 10.8.0.99: icmp_seq=6 ttl=63 time=0.100 ms
64 bytes from 10.8.0.99: icmp_seq=7 ttl=63 time=0.122 ms
64 bytes from 10.8.0.99: icmp_seq=8 ttl=63 time=0.081 ms
64 bytes from 10.8.0.99: icmp_seq=9 ttl=63 time=0.101 ms
64 bytes from 10.8.0.99: icmp_seq=10 ttl=63 time=0.258 ms
64 bytes from 10.8.0.99: icmp_seq=11 ttl=63 time=0.117 ms
64 bytes from 10.8.0.99: icmp_seq=12 ttl=63 time=0.096 ms
64 bytes from 10.8.0.99: icmp_seq=13 ttl=63 time=0.160 ms
64 bytes from 10.8.0.99: icmp_seq=14 ttl=63 time=0.539 ms
64 bytes from 10.8.0.99: icmp_seq=15 ttl=63 time=0.080 ms
```

```
root@08789cd1407b:/# ping 192.168.20.99
PING 192.168.20.99 (192.168.20.99) 56(84) bytes of data.
64 bytes from 192.168.20.99: icmp_seq=1 ttl=63 time=0.093 ms
64 bytes from 192.168.20.99: icmp_seq=2 ttl=63 time=0.166 ms
64 bytes from 192.168.20.99: icmp_seq=3 ttl=63 time=0.118 ms
64 bytes from 192.168.20.99: icmp_seq=4 ttl=63 time=0.175 ms
64 bytes from 192.168.20.99: icmp_seq=5 ttl=63 time=0.814 ms
64 bytes from 192.168.20.99: icmp_seq=6 ttl=63 time=0.133 ms
64 bytes from 192.168.20.99: icmp_seq=7 ttl=63 time=0.118 ms
64 bytes from 192.168.20.99: icmp_seq=8 ttl=63 time=0.247 ms
64 bytes from 192.168.20.99: icmp_seq=9 ttl=63 time=0.459 ms
64 bytes from 192.168.20.99: icmp_seq=10 ttl=63 time=0.177 ms
64 bytes from 192.168.20.99: icmp_seq=11 ttl=63 time=0.154 ms
64 bytes from 192.168.20.99: icmp_seq=12 ttl=63 time=0.105 ms
^C
```

Testing an unrelated connection

```
root@08789cd1407b:/# telnet 192.168.20.99 80
Trying 192.168.20.99...
```

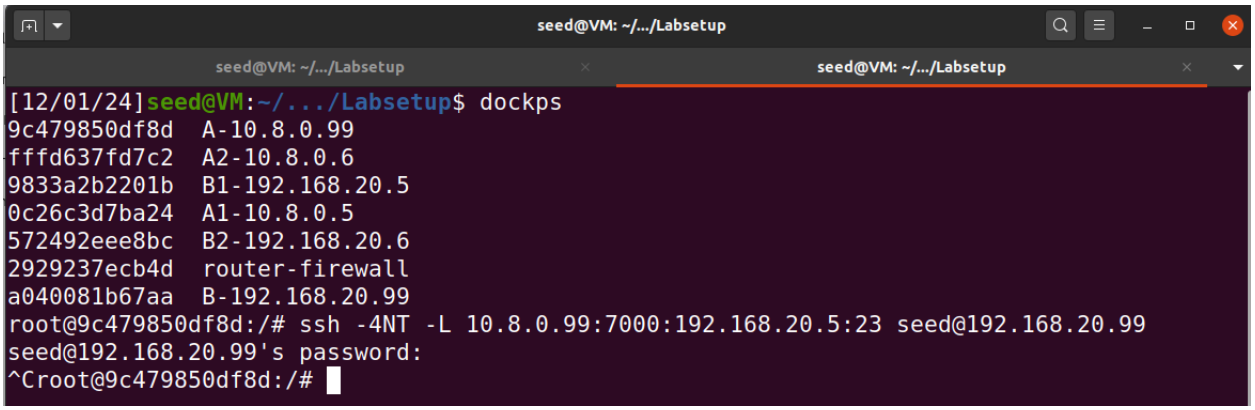
This does not match --ctstate ESTABLISHED,RELATED rule or the SSH rule (port 22).

Task 1: Static Port Forwarding

Create the SSH Static Port Forwarding Tunnel: Run the following command to forward traffic from A to B1:

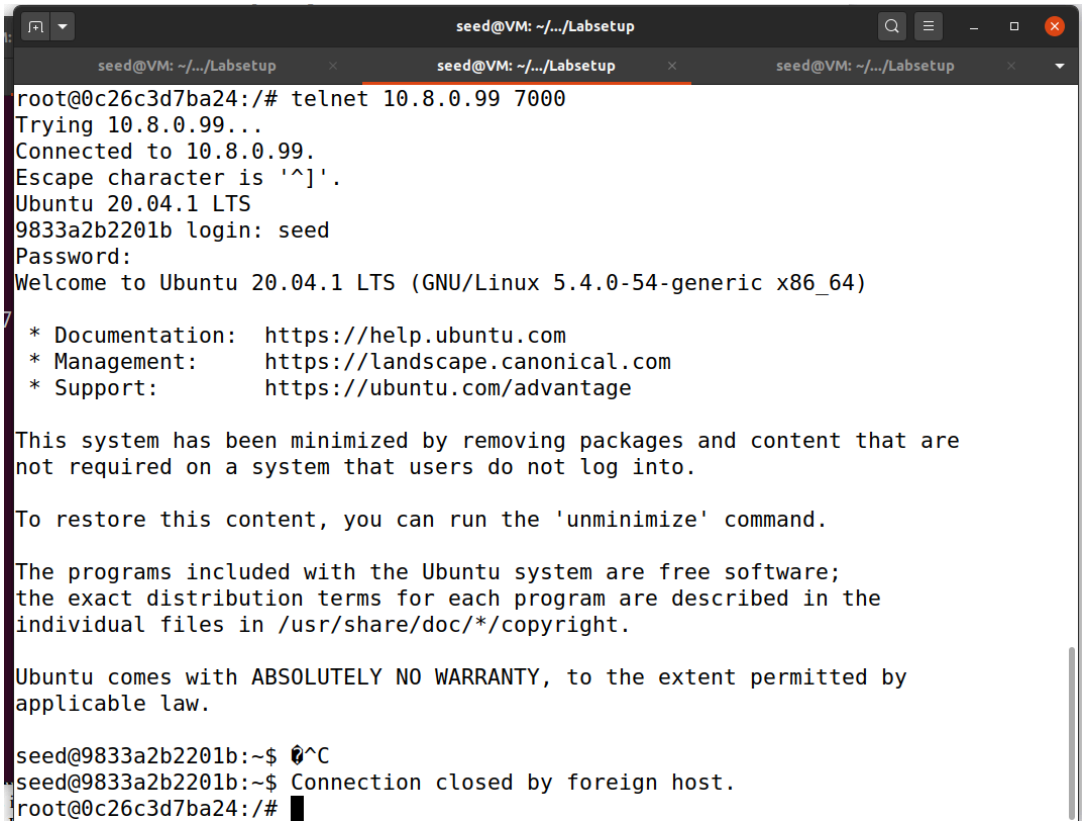
On docker container A-10.8.0.99:

- **We create a tunnel to B**
- **Set out target to machine B1 using port 7000 and port 23**



We will leave that ssh session running to maintain the tunnel.

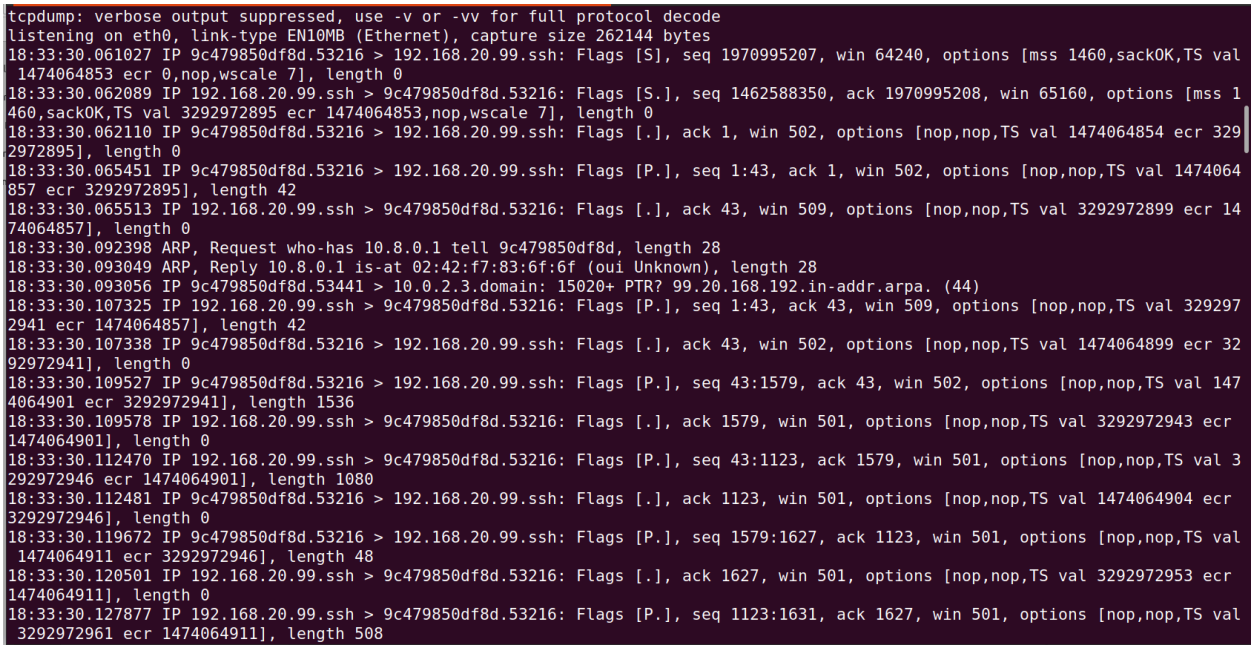
On docker container A1-10.8.0.5:



On docker container A2-10.8.0.6:

Since we logged in, the connection was successful.

Now on the router's container, we will analyze traffic using tcpdump:



```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:33:30.061027 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [S], seq 1970995207, win 64240, options [mss 1460,sackOK,TS val
 1474064853 ecr 0,nop,wscale 7], length 0
18:33:30.062089 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [S.], seq 1462588350, ack 1970995208, win 65160, options [mss 1
460,sackOK,TS val 3292972895 ecr 1474064853,nop,wscale 7], length 0
18:33:30.062110 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [.], ack 1, win 502, options [nop,nop,TS val 1474064854 ecr 329
2972895], length 0
18:33:30.065451 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [P.], seq 1:43, ack 1, win 502, options [nop,nop,TS val 1474064
857 ecr 3292972895], length 42
18:33:30.065513 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [.], ack 43, win 509, options [nop,nop,TS val 3292972899 ecr 14
74064857], length 0
18:33:30.092398 ARP, Request who-has 10.8.0.1 tell 9c479850df8d, length 28
18:33:30.093049 ARP, Reply 10.8.0.1 is-at 02:42:f7:83:6f:6f (oui Unknown), length 28
18:33:30.093056 IP 9c479850df8d.53441 > 10.0.2.3.domain: 15020+ PTR? 99.20.168.192.in-addr.arpa. (44)
18:33:30.107325 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [P.], seq 1:43, ack 43, win 509, options [nop,nop,TS val 329297
2941 ecr 1474064857], length 42
18:33:30.107338 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [.], ack 43, win 502, options [nop,nop,TS val 1474064899 ecr 32
92972941], length 0
18:33:30.109527 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [P.], seq 43:1579, ack 43, win 502, options [nop,nop,TS val 147
4064901 ecr 3292972941], length 1536
18:33:30.109578 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [.], ack 1579, win 501, options [nop,nop,TS val 3292972943 ecr
1474064901], length 0
18:33:30.112470 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [P.], seq 43:1123, ack 1579, win 501, options [nop,nop,TS val 3
292972946 ecr 1474064901], length 1080
18:33:30.112481 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [.], ack 1123, win 501, options [nop,nop,TS val 1474064904 ecr
3292972946], length 0
18:33:30.119672 IP 9c479850df8d.53216 > 192.168.20.99.ssh: Flags [P.], seq 1579:1627, ack 1123, win 501, options [nop,nop,TS val
 1474064911 ecr 3292972946], length 48
18:33:30.120501 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [.], ack 1627, win 501, options [nop,nop,TS val 3292972953 ecr
1474064911], length 0
18:33:30.127877 IP 192.168.20.99.ssh > 9c479850df8d.53216: Flags [P.], seq 1123:1631, ack 1627, win 501, options [nop,nop,TS val
 3292972961 ecr 1474064911], length 508
```

Here we can see that telnet connection is successfully established between internal host and external host.

1. How many TCP connections are involved in this entire process. You should run wireshark or tcpdump to capture the network traffic, and then point out all the involved TCP connections from the captured traffic
    a. There are 3 TCP connections;
        i. SSH tunnel from A to B
        ii. Telnet connection from A1/A2 to A
        iii. Telnet connection from A to target B1

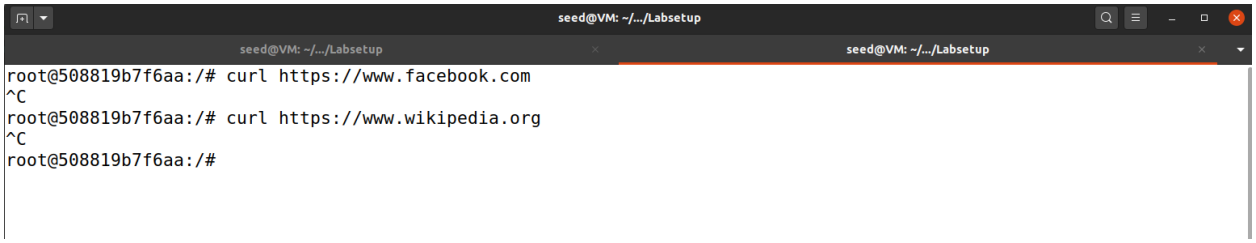
2. Why can this tunnel successfully help users evade the firewall rule specified in the lab setup?

The SSH tunnel uses **static port forwarding and** allows users to evade the firewall by forwarding traffic from A to B through an encrypted connection. These bypassing rules block direct access to port 23. The firewall permits outgoing SSH traffic on port 22, and since all Telnet traffic is encapsulated within the secure SSH tunnel, it cannot inspect or block it. From the perspective of A1 and A2, connections to A appear legitimate to the firewall, as they do not directly target the restricted destination. This end-to-end encapsulation ensures the Telnet traffic bypasses firewall restrictions unnoticed.

Task 2: Dynamic Port Forwarding

Dynamic port forwarding leverages a proxy setup to redirect traffic to various destinations. This allows a single SSH tunnel to act as a gateway for multiple blocked websites. Dynamic port forwarding uses the SOCKS (Socket Secure) protocol to enable applications to send their traffic through a proxy.

From task 0, we blocked www.facebook.com and www.wikipedia.org by adding these lines in our docker file:
iptables -A FORWARD -i eth1 -d 208.80.154.224 -j DROP && ← Wikipedia
iptables -A FORWARD -i eth1 -d 31.13.80.36 -j DROP && ← Facebook



Task 2.1: Setting Up Dynamic Port Forwarding

We establish a dynamic port forwarding tunnel using SSH, allowing you to route traffic through a proxy to access blocked websites. After the tunnel is established we test it using curl with the SOCKS proxy.

On Host B, we created a SOCKS proxy using:
ssh -4NT -D 0.0.0.0:1080 seed@10.8.0.99

```
root@508819b7f6aa:/# ssh -4NT -D 0.0.0.0:1080 seed@10.8.0.99
seed@10.8.0.99's password:
```

We will runt the following for Host B, B1 and B2 to show that we can have access to the blocked sites:
curl --proxy socks5h://192.168.20.99:1080 https://www.wikipedia.org
curl --proxy socks5h://192.168.20.99:1080 https://www.facebook.com

Sh from another shell, we test on Host B the blocked websites:

```
root@508819b7f6aa:/# curl --proxy socks5h://192.168.20.99:1080 https://www.wikipedia.org
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="utf-8">
<title>Wikipedia</title>
<meta name="description" content="Wikipedia is a free online encyclopedia, created and edited by volunteers around the world and hoste
d by the Wikimedia Foundation.">
<script>
document.documentElement.className = document.documentElement.className.replace( /(^|\s)no-js(\s|$)/, "$1js-enabled$2" );
</script>
<meta name="viewport" content="initial-scale=1,user-scalable=yes">
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png">
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico">
<link rel="license" href="//creativecommons.org/licenses/by-sa/4.0/">
<style>
.sprite{background-image:linear-gradient(transparent,transparent),url(portal/wikipedia.org/assets/img/sprite-de847d1a.svg);background-
repeat:no-repeat;display:inline-block;vertical-align:middle}.svg-Commons-logo_sister{background-position:0 0;width:47px;height:47px}.s
vg-MediaWiki-logo_sister{background-position:0 -47px;width:42px;height:42px}.svg-Meta-Wiki-logo_sister{background-position:0 -89px;wid
th:37px;height:37px}.svg-Wikibooks-logo_sister{background-position:0 -126px;width:37px;height:37px}.svg-Wikidata-logo_sister{backgroun
d-position:0 -163px;width:49px;height:49px}.svg-Wikifunctions-logo_sister{background-position:0 -212px;width:50px;height:50px}.svg-Wik
```

```
root@508819b7f6aa:/# curl --proxy socks5h://192.168.20.99:1080 https://www.facebook.com
<!DOCTYPE html>
<html lang="en" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="referrer" content="default" id="meta_referrer" /><script nonce="vcHskqme">function envFlush(
a){function b(b){for(var c in a)b[c]=a[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{},b(window.Env))}e
nvFlush({"useTrustedTypes":false,"isTrustedTypesReportOnly":false,"ajaxpipe_token":"AXh91_ATFFv_eMJ8vW8","stack_trace_limit":30,"times
liceBufferSize":5000,"show_invariant_decoder":false,"compat_iframe_token":"AUWMlDm5cp_UPh-hOpbu6tvGb68","isCQuick":false,"brsid":"7443
461531903686993"});</script><script nonce="vcHskqme">window.openDatabase&&(window.openDatabase=function(){throw new Error()});</script
><script nonce="vcHskqme">_btldr={};</script><script nonce="vcHskqme">function parentIsNotHeadNorBody(a){return a.parentElement!==docu
ment.body&&a.parentElement!==document.head}function isTagSupported(a){return a.nodeName==="SCRIPT"||a.nodeName==="LINK"&&((a=getNodeDa
taSet(a))==null?void 0:a.asyncCss)}function getNodeDataSet(a){return!(a.dataset instanceof window.DOMStringMap)?null:a.dataset}functio
n addLoadEventListeners(a){var b;try{if(a.nodeType!==Node.ELEMENT_NODE)return}catch(a){return}if(parentIsNotHeadNorBody(a)||!isTagSupp
orted(a))return;var c=(b=getNodeDataSet(a))==null?void 0:b.bootloaderHash;if(c!=null&&c!==""){var d=null,e=function(){window._btldr[c]
=1,d==null?void 0:d()};d=function(){a.removeEventListener("load",e),a.removeEventListener("error",e)};a.addEventListener("load",e);a.a
ddEventListener("error",e)}}(function(){Array.from(document.querySelectorAll('script,link[data-async-css="1"]')).forEach(function(a){r
eturn addLoadEventListeners(a)});var a=new MutationObserver(function(a,b){a.forEach(function(a){a.type==="childList"&&Array.from(a.add
edNodes).forEach(function(a){addLoadEventListeners(a)})})});a.observe(document.getElementsByTagName("html")[0],{attributes:!1,childLis
t:!0,subtree:!0})})();</script><style nonce="vcHskqme">/*</style><script nonce="vcHskqme">__DEV__=0;</script><noscript><meta http-equiv=
"refresh" content="0; URL=/?_fb_noscript=1" /></noscript><link rel="manifest" id="MANIFEST_LINK" href="/data/manifest/" crossorigin="u
se-credentials" /><title id="pageTitle">Facebook - log in or sign up</title><meta name="bingbot" content="noarchive" /><meta property=
"og:site_name" content="Facebook" /><meta property="og:url" content="https://www.facebook.com/" /><meta property="og:image" content="h
ttps://www.facebook.com/images/fb_icon_325x325.png" /><meta property="og:locale" content="en_US" /><link rel="alternate" media="only s
creen and (max-width: 640px)" href="https://m.facebook.com/" /><link rel="alternate" media="handheld" href="https://m.facebook.com/" />
<meta name="description" content="Log into Facebook to start sharing and connecting with your friends, family, and people you know."
/><script type="application/ld+json" nonce="vcHskqme">{"\u0040context":"http:\/\/schema.org","\u0040type":"WebSite","name":"Facebook",
"url":"https:\/\/www.facebook.com\/"}</script><link rel="canonical" href="https://www.facebook.com/" /><link rel="icon" href="https://
static.xx.fbcdn.net/rsrc.php/yB/r/2sFJRNmJ5OP.ico" /><link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php
```

From Host B1:

```
root@2677cf5b04f3:/# curl --proxy socks5h://192.168.20.99:1080 https://www.wikipedia.org
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="utf-8">
<title>Wikipedia</title>
<meta name="description" content="Wikipedia is a free online encyclopedia, created and edited by volunteers around the world and hoste
d by the Wikimedia Foundation.">
<script>
document.documentElement.className = document.documentElement.className.replace( /(^|\s)no-js(\s|$)/, "$1js-enabled$2" );
</script>
<meta name="viewport" content="initial-scale=1,user-scalable=yes">
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png">
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico">
<link rel="license" href="//creativecommons.org/licenses/by-sa/4.0/">
<style>
.sprite{background-image:linear-gradient(transparent,transparent),url(portal/wikipedia.org/assets/img/sprite-de847d1a.svg);background-
repeat:no-repeat;display:inline-block;vertical-align:middle}.svg-Commons-logo_sister{background-position:0 0;width:47px;height:47px}.s
vg-MediaWiki-logo_sister{background-position:0 -47px;width:42px;height:42px}.svg-Meta-Wiki-logo_sister{background-position:0 -89px;wid
th:37px;height:37px}.svg-Wikibooks-logo_sister{background-position:0 -126px;width:37px;height:37px}.svg-Wikidata-logo_sister{backgroun
d-position:0 -163px;width:49px;height:49px}.svg-Wikifunctions-logo_sister{background-position:0 -212px;width:50px;height:50px}.svg-Wik
imedia-logo_black{background-position:0 -262px;width:42px;height:42px}.svg-Wikipedia_wordmark{background-position:0 -304px;width:176px
;height:32px}.svg-Wikiquote-logo_sister{background-position:0 -336px;width:42px;height:42px}.svg-Wikisource-logo_sister{background-pos
ition:0 -378px;width:39px;height:39px}.svg-Wikispecies-logo_sister{background-position:0 -417px;width:42px;height:42px}.svg-Wikiversit
```

```
root@2677cf5b04f3:/# curl --proxy socks5h://192.168.20.99:1080 https://www.facebook.com
<!DOCTYPE html>
<html lang="en" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="referrer" content="default" id="meta_referrer" /><script nonce="bK7K3vO5">function envFlush(
a){function b(b){for(var c in a)b[c]=a[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{},b(window.Env))}e
nvFlush({"useTrustedTypes":false,"isTrustedTypesReportOnly":false,"ajaxpipe_token":"AXh91_ATFFv_eMJ87_o","stack_trace_limit":30,"times
liceBufferSize":5000,"show_invariant_decoder":false,"compat_iframe_token":"AUWMlDm5cp_UPh-hOpbu6tvGrVM","isCQuick":false,"brsid":"7443
463557051587824"});</script><script nonce="bK7K3vO5">window.openDatabase&&(window.openDatabase=function(){throw new Error()});</script
><script nonce="bK7K3vO5">_btldr={};</script><script nonce="bK7K3vO5">function parentIsNotHeadNorBody(a){return a.parentElement!==docu
ment.body&&a.parentElement!==document.head}function isTagSupported(a){return a.nodeName==="SCRIPT"||a.nodeName==="LINK"&&((a=getNodeDa
taSet(a))==null?void 0:a.asyncCss)}function getNodeDataSet(a){return!(a.dataset instanceof window.DOMStringMap)?null:a.dataset}functio
n addLoadEventListeners(a){var b;try{if(a.nodeType!==Node.ELEMENT_NODE)return}catch(a){return}if(parentIsNotHeadNorBody(a)||!isTagSupp
orted(a))return;var c=(b=getNodeDataSet(a))==null?void 0:b.bootloaderHash;if(c!=null&&c!=="")(var d=null,e=function(){window._btldr[c]
=1,d==null?void 0:d()};d=function(){a.removeEventListener("load",e),a.removeEventListener("error",e)};a.addEventListener("load",e);a.a
ddEventListener("error",e)}}(function(){Array.from(document.querySelectorAll('script,link[data-async-css="1"]')).forEach(function(a){r
eturn addLoadEventListeners(a)});var a=new MutationObserver(function(a,b){a.forEach(function(a){a.type==="childList"&&Array.from(a.add
edNodes).forEach(function(a){addLoadEventListeners(a)})})});a.observe(document.getElementsByTagName("html")[0],{attributes:!1,childLis
t:!0,subtree:!0})})();</script><style nonce="bK7K3vO5"></style><script nonce="bK7K3vO5">__DEV__=0;</script><noscript><meta http-equiv=
"refresh" content="0; URL=/?_fb_noscript=1" /></noscript><link rel="manifest" id="MANIFEST_LINK" href="/data/manifest/" crossorigin="u
se-credentials" /><title id="pageTitle">Facebook - log in or sign up</title><meta name="bingbot" content="noarchive" /><meta property=
"og:site_name" content="Facebook" /><meta property="og:url" content="https://www.facebook.com/" /><meta property="og:image" content="h
ttps://www.facebook.com/images/fb_icon_325x325.png" /><meta property="og:locale" content="en_US" /><link rel="alternate" media="only s
```

From Host B2:

```
root@425a3a7e6b60:/# curl --proxy socks5h://192.168.20.99:1080 https://www.wikipedia.org
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="utf-8">
<title>Wikipedia</title>
<meta name="description" content="Wikipedia is a free online encyclopedia, created and edited by volunteers around the world and hoste
d by the Wikimedia Foundation.">
<script>
document.documentElement.className = document.documentElement.className.replace( /(^|\s)no-js(\s|$)/, "$1js-enabled$2" );
</script>
<meta name="viewport" content="initial-scale=1,user-scalable=yes">
<link rel="apple-touch-icon" href="/static/apple-touch/wikipedia.png">
<link rel="shortcut icon" href="/static/favicon/wikipedia.ico">
<link rel="license" href="//creativecommons.org/licenses/by-sa/4.0/">
<style>
.sprite{background-image:linear-gradient(transparent,transparent),url(portal/wikipedia.org/assets/img/sprite-de847d1a.svg);background-
repeat:no-repeat;display:inline-block;vertical-align:middle}.svg-Commons-logo_sister{background-position:0 0;width:47px;height:47px}.s
vg-MediaWiki-logo_sister{background-position:0 -47px;width:42px;height:42px}.svg-Meta-Wiki-logo_sister{background-position:0 -89px;wid
th:37px;height:37px}.svg-Wikibooks-logo_sister{background-position:0 -126px;width:37px;height:37px}.svg-Wikidata-logo_sister{backgroun
d-position:0 -163px;width:49px;height:49px}.svg-Wikifunctions-logo_sister{background-position:0 -212px;width:50px;height:50px}.svg-Wik
```

```
root@425a3a7e6b60:/# curl --proxy socks5h://192.168.20.99:1080 https://www.facebook.com
<!DOCTYPE html>
<html lang="en" id="facebook" class="no_js">
<head><meta charset="utf-8" /><meta name="referrer" content="default" id="meta_referrer" /><script nonce="EWIZF1wb">function envFlush(
a){function b(b){for(var c in a)b[c]=a[c]}window.requireLazy?window.requireLazy(["Env"],b):(window.Env=window.Env||{},b(window.Env))}e
nvFlush({"useTrustedTypes":false,"isTrustedTypesReportOnly":false,"ajaxpipe_token":"AXh91_ATFFv_eMJ8GxQ","stack_trace_limit":30,"times
liceBufferSize":5000,"show_invariant_decoder":false,"compat_iframe_token":"AUWMlDm5cp_UPh-hOpbu6tvGjU8","isCQuick":false,"brsid":"7443
464257334139066"});</script><script nonce="EWIZF1wb">window.openDatabase&&(window.openDatabase=function(){throw new Error()});</script
><script nonce="EWIZF1wb">_btldr={};</script><script nonce="EWIZF1wb">function parentIsNotHeadNorBody(a){return a.parentElement!==docu
ment.body&&a.parentElement!==document.head}function isTagSupported(a){return a.nodeName==="SCRIPT"||a.nodeName==="LINK"&&((a=getNodeDa
taSet(a))==null?void 0:a.asyncCss)}function getNodeDataSet(a){return!(a.dataset instanceof window.DOMStringMap)?null:a.dataset}functio
n addLoadEventListeners(a){var b;try{if(a.nodeType!==Node.ELEMENT_NODE)return}catch(a){return}if(parentIsNotHeadNorBody(a)||!isTagSupp
orted(a))return;var c=(b=getNodeDataSet(a))==null?void 0:b.bootloaderHash;if(c!=null&&c!=="")(var d=null,e=function(){window._btldr[c]
=1,d==null?void 0:d()};d=function(){a.removeEventListener("load",e),a.removeEventListener("error",e)};a.addEventListener("load",e);a.a
ddEventListener("error",e)}}(function(){Array.from(document.querySelectorAll('script,link[data-async-css="1"]')).forEach(function(a){r
eturn addLoadEventListeners(a)});var a=new MutationObserver(function(a,b){a.forEach(function(a){a.type==="childList"&&Array.from(a.add
edNodes).forEach(function(a){addLoadEventListeners(a)})})});a.observe(document.getElementsByTagName("html")[0],{attributes:!1,childLis
t:!0,subtree:!0})})();</script><style nonce="EWIZF1wb"></style><script nonce="EWIZF1wb">__DEV__=0;</script><noscript><meta http-equiv=
"refresh" content="0; URL=/?_fb_noscript=1" /></noscript><link rel="manifest" id="MANIFEST_LINK" href="/data/manifest/" crossorigin="u
se-credentials" /><title id="pageTitle">Facebook - log in or sign up</title><meta name="bingbot" content="noarchive" /><meta property=
"og:site_name" content="Facebook" /><meta property="og:url" content="https://www.facebook.com/" /><meta property="og:image" content="h
ttps://www.facebook.com/images/fb_icon_325x325.png" /><meta property="og:locale" content="en_US" /><link rel="alternate" media="only s
creen and (max-width: 640px)" href="https://m.facebook.com/" /><link rel="alternate" media="handheld" href="https://m.facebook.com/" /
```

(1) Which computer establishes the actual connection with the intended web server?

Host A establishes the actual connection with the intended web servers (www.wikipedia.org, www.facebook.com). Host A serves as the endpoint of the SSH tunnel and routes the traffic to the target web servers.

(2) How does this computer know which server it should connect to?

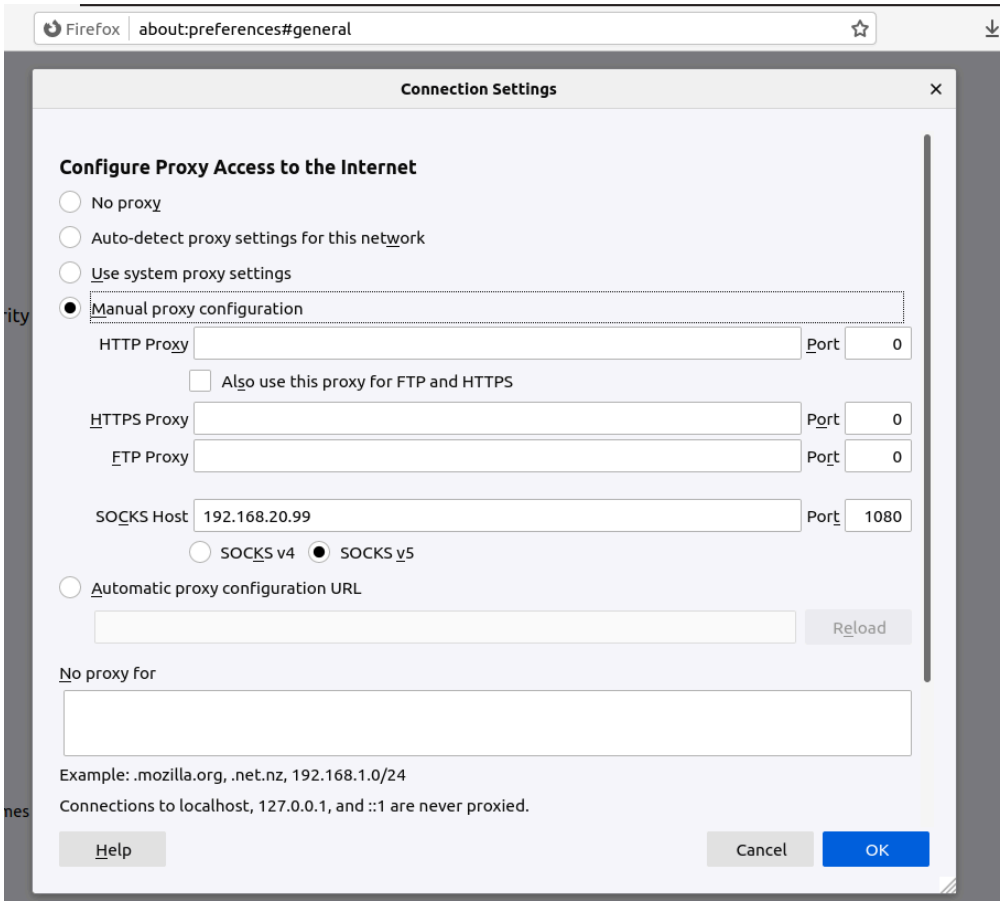
Host A knows the target server because the SOCKS protocol (SOCKS5) embeds the destination hostname or IP address in the traffic. The curl command specifies the target website

in its HTTP request, and the SOCKS proxy on Host B forwards this information through the SSH tunnel to Host A, which decodes and connects to the intended server.

Task 2.2: Testing the Tunnel Using Browser

Host B's IP address: 192.168.20.99

We configured Firefox to route its traffic through the SOCKS5 proxy. This was done by accessing Firefox's proxy settings and specifying the proxy. By doing this, all HTTP requests from Firefox were forwarded through the dynamic tunnel, enabling us to test access to blocked websites.

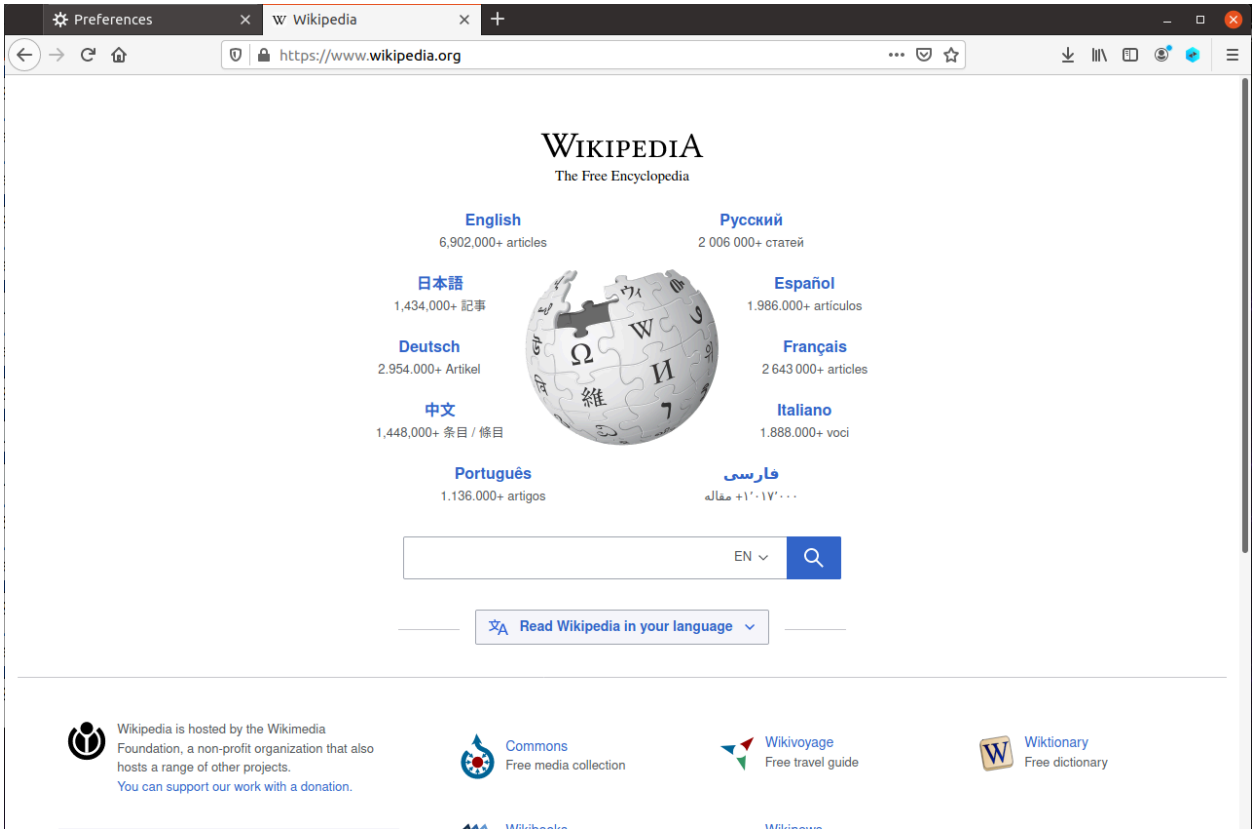


Running tcpdump on the router/firewall to monitor the traffic:

```
root@f0b53f7100a9:/#  tcpdump -i eth1 host 192.168.20.99
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

This allowed us to verify that traffic was indeed being forwarded through the SSH tunnel to the destination website.

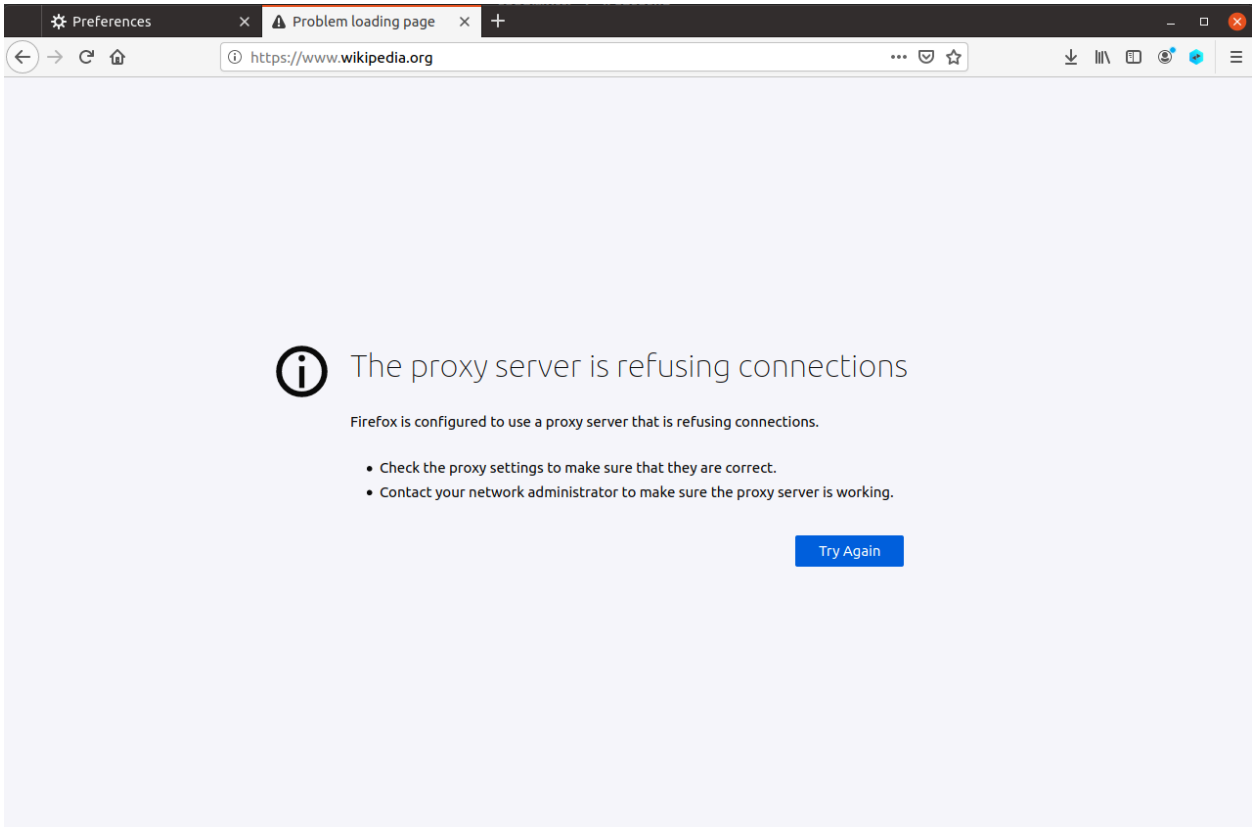
We try to access wikipedia.org




```
root@f0b53f7100a9:/#  tcpdump -i eth1 host 192.168.20.99
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
15:59:24.924229 ARP, Request who-has B-192.168.20.99.net-192.168.20.0 tell 192.168.20.1, length 28
15:59:24.924807 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 2068872782:2068872882, ack 4
097053753, win 501, options [nop,nop,TS val 364183062 ecr 2837824981], length 100
15:59:24.924915 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.46674: Flags [.], ack 100, win 501, options [nop,no
p,TS val 2838171206 ecr 364183062], length 0
15:59:24.957746 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.46674: Flags [P.], seq 1:45, ack 100, win 501, opti
ons [nop,nop,TS val 2838171239 ecr 364183062], length 44
15:59:24.957843 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 45, win 501, options [nop,nop
,TS val 364183095 ecr 2838171239], length 0
15:59:24.958533 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 100:472, ack 45, win 501, op
tions [nop,nop,TS val 364183096 ecr 2838171239], length 372
15:59:24.992880 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.46674: Flags [P.], seq 45:257, ack 472, win 501, op
tions [nop,nop,TS val 2838171274 ecr 364183096], length 212
15:59:24.992925 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 257, win 501, options [nop,no
p,TS val 364183130 ecr 2838171274], length 0
15:59:24.994712 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 472:508, ack 257, win 501, o
ptions [nop,nop,TS val 364183132 ecr 2838171274], length 36
15:59:25.007795 IP B-192.168.20.99.net-192.168.20.0.46674 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 508:608, ack 257, win 501, o
ptions [nop,nop,TS val 364183145 ecr 2838171274], length 100
```

We will kill the ssh tunnel to see if the website is still accessible once refreshed.

```
root@508819b7f6aa:/# ssh -4NT -D 0.0.0.0:1080 seed@10.8.0.99
seed@10.8.0.99's password:
^Croot@508819b7f6aa:/#
```



Since the traffic is no longer tunnelied the browser fails to load the website.

After completing the tests, we ensured to remove the proxy configuration in Firefox by selecting the "No proxy" option, thus ensuring that the proxy settings wouldn't interfere with future tasks or browsing sessions.

Task 2.3: Writing a SOCKS Client Using Python

To do this task, we modified the docker file so it included the exact IP address of www.example.com

```
[12/01/24]seed@VM:~/.../Labsetup$ nslookup www.example.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.example.com
Address: 93.184.215.14
Name:   www.example.com
Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c
```

Creating the program:

From container B, we run the following to set up the SSH dynamic port forwarding tunnel between B and A:

```
root@4459d390c81f:/# ssh -4NT -D 0.0.0.0:1080 seed@10.8.0.99
The authenticity of host '10.8.0.99 (10.8.0.99)' can't be established.
ECDSA key fingerprint is SHA256:SWOfIaxcHX2QJXJJOUqir1sz+P84sduAtwuEfKhAyJI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.8.0.99' (ECDSA) to the list of known hosts.
seed@10.8.0.99's password:
```

Running the program on B in a different shell:

```
root@4459d390c81f:/# ./socks_client.py
HTTP/1.0 200 OK
Age: 402724
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sun, 01 Dec 2024 18:36:17 GMT
Etag: "3147526947+ident"
Expires: Sun, 08 Dec 2024 18:36:17 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECAcc (dcd/7D11)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
Connection: close

<!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">
```

The HTTP response was retrieved successfully from www..example.com

From running tcpdump on the router/firewall container:

```
root@492633afbdb2:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:36:16.995706 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 33028740:33028
840, ack 3432385297, win 501, options [nop,nop,TS val 2962739909 ecr 3679801308], length 100
18:36:16.995737 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [.], ack 100, win 501, o
ptions [nop,nop,TS val 3679848306 ecr 2962739909], length 0
18:36:17.043741 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1:45, ack 100,
 win 501, options [nop,nop,TS val 3679848354 ecr 2962739909], length 44
18:36:17.044314 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 100:176, ack 4
5, win 501, options [nop,nop,TS val 2962739958 ecr 3679848354], length 76
18:36:17.080017 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 45:1545, ack 1
76, win 501, options [nop,nop,TS val 3679848391 ecr 2962739958], length 1500
18:36:17.080127 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1545, win 501,
options [nop,nop,TS val 2962739994 ecr 3679848391], length 0
18:36:17.080238 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1545:1733, ack
 176, win 501, options [nop,nop,TS val 3679848391 ecr 2962739994], length 188
18:36:17.124918 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1733, win 501,
options [nop,nop,TS val 2962740038 ecr 3679848391], length 0
18:36:17.124961 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1733:1769, ack
 176, win 501, options [nop,nop,TS val 3679848436 ecr 2962740038], length 36
18:36:17.124992 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1769, win 501,
options [nop,nop,TS val 2962740039 ecr 3679848436], length 0
18:36:17.128504 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 176:248, ack 1
769, win 501, options [nop,nop,TS val 2962740042 ecr 3679848436], length 72
18:36:17.128893 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1769:1805, ack
 248, win 501, options [nop,nop,TS val 3679848440 ecr 2962740042], length 36
18:36:17.128998 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1805, win 501,
options [nop,nop,TS val 2962740043 ecr 3679848440], length 0
18:40:00.258441 IP 10.8.0.1.mdns > mdns.mcast.net.mdns: 0 [9q] PTR (QM)? _nfs._tcp.local. PTR (QM)? _ipp._tcp.local. PTR
 (QM)? _ipps._tcp.local. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM
)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _afpovertcp._tcp.local. (141)
18:40:00.261643 ARP, Request who-has 10.8.0.1 tell 492633afbdb2, length 28
18:40:00.261715 ARP, Reply 10.8.0.1 is-at 02:42:b4:2f:6a:a9 (oui Unknown), length 28
18:40:00.261719 IP 492633afbdb2.33078 > 10.0.2.3.domain: 8408+ PTR? 251.0.0.224.in-addr.arpa. (42)
18:40:00.289776 IP 10.0.2.3.domain > 492633afbdb2.33078: 8408 1/0/0 PTR mdns.mcast.net. (70)
18:40:00.290380 IP 492633afbdb2.49265 > 10.0.2.3.domain: 35537+ PTR? 1.0.8.10.in-addr.arpa. (39)
18:40:00.317482 IP 10.0.2.3.domain > 492633afbdb2.49265: 35537 NXDomain 0/1/0 (116)
18:40:00.318679 IP 492633afbdb2.57544 > 10.0.2.3.domain: 39265+ PTR? 3.2.0.10.in-addr.arpa. (39)
18:40:00.331077 IP 10.0.2.3.domain > 492633afbdb2.57544: 39265 NXDomain 0/1/0 (116)
18:40:03.817573 IP6 fe80::42:b4ff:fe2f:6aa9.mdns > ff02::fb.mdns: 0 [9q] PTR (QM)? _nfs._tcp.local. PTR (QM)? _ipp._tcp.
local. PTR (QM)? _ipps._tcp.local. PTR (QM)? _ftp._tcp.local. PTR (QM)? _webdav._tcp.local. PTR (QM)? _webdavs._tcp.loca
l. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _afpovertcp._tcp.local. (141)
18:40:03.818169 IP 492633afbdb2.46792 > 10.0.2.3.domain: 1039+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.2.0.f.f.ip6.arpa. (90)
18:40:03.846106 IP 10.0.2.3.domain > 492633afbdb2.46792: 1039 NXDomain 0/1/0 (154)
18:40:03.846785 IP 492633afbdb2.53790 > 10.0.2.3.domain: 62498+ PTR? 9.a.a.6.f.2.e.f.f.f.4.b.2.4.0.0.0.0.0.0.0.0.0.0.0.0
.0.0.0.8.e.f.ip6.arpa. (90)
```

From B1:

```
root@7620d28a7e64:/# ./socks_clientB1.py
HTTP/1.0 200 OK
Accept-Ranges: bytes
Age: 429080
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sun, 01 Dec 2024 18:57:59 GMT
Etag: "3147526947+gzip"
Expires: Sun, 08 Dec 2024 18:57:59 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECAcc (dcd/7D44)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
Connection: close

<!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, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, A
rif;

    `
```

The result of tcpdump:

```
root@492633afbdb2:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:57:59.120700 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 33029448:33029548, ack 343239
0581, win 501, options [nop,nop,TS val 2964042034 ecr 3681138910], length 100
18:57:59.120722 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [.], ack 100, win 501, options [nop,nop
,TS val 3681150431 ecr 2964042034], length 0
18:57:59.151133 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1:45, ack 100, win 501, optio
ns [nop,nop,TS val 3681150462 ecr 2964042034], length 44
18:57:59.151278 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 45, win 501, options [nop,nop,
TS val 2964042065 ecr 3681150462], length 0
18:57:59.152617 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 100:176, ack 45, win 501, opt
ions [nop,nop,TS val 2964042066 ecr 3681150462], length 76
18:57:59.181314 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 45:1545, ack 176, win 501, op
tions [nop,nop,TS val 3681150492 ecr 2964042066], length 1500
18:57:59.181401 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1545, win 501, options [nop,no
p,TS val 2964042095 ecr 3681150492], length 0
18:57:59.181636 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1545:1757, ack 176, win 501,
options [nop,nop,TS val 3681150492 ecr 2964042095], length 212
18:57:59.181747 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1757, win 501, options [nop,no
p,TS val 2964042095 ecr 3681150492], length 0
18:57:59.182471 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1757:1793, ack 176, win 501,
options [nop,nop,TS val 3681150493 ecr 2964042095], length 36
18:57:59.182496 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1793, win 501, options [nop,no
p,TS val 2964042096 ecr 3681150493], length 0
18:57:59.188181 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 176:248, ack 1793, win 501, o
ptions [nop,nop,TS val 2964042102 ecr 3681150493], length 72
18:57:59.188582 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1793:1829, ack 248, win 501,
options [nop,nop,TS val 3681150499 ecr 2964042102], length 36
18:57:59.188699 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1829, win 501, options [nop,no
p,TS val 2964042102 ecr 3681150499], length 0
18:58:02.248458 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1829:1901, ack 248, win 501,
options [nop,nop,TS val 3681153559 ecr 2964042102], length 72
18:58:02.248502 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1901, win 501, options [nop,no
p,TS val 2964045162 ecr 3681153559], length 0
18:58:02.248659 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 248:284, ack 1901, win 501, o
ptions [nop,nop,TS val 2964045162 ecr 3681153559], length 36
18:58:02.289071 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [.], ack 284, win 501, options [nop,nop
,TS val 3681153600 ecr 2964045162], length 0
```

From B2:

```
root@492633afbdb2:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
19:04:03.089734 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 33030156:33030256, ack 343239
5873, win 501, options [nop,nop,TS val 2964406003 ecr 3681502973], length 100
19:04:03.089797 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [.], ack 100, win 501, options [nop,nop
,TS val 3681514400 ecr 2964406003], length 0
19:04:03.124855 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1:45, ack 100, win 501, optio
ns [nop,nop,TS val 3681514435 ecr 2964406003], length 44
19:04:03.125506 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 100:176, ack 45, win 501, opt
ions [nop,nop,TS val 2964406039 ecr 3681514435], length 76
19:04:03.156885 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 45:1545, ack 176, win 501, op
tions [nop,nop,TS val 3681514468 ecr 2964406039], length 1500
19:04:03.156925 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1545, win 501, options [nop,no
p,TS val 2964406071 ecr 3681514468], length 0
19:04:03.162457 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1545:1733, ack 176, win 501,
options [nop,nop,TS val 3681514473 ecr 2964406071], length 188
19:04:03.208957 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1733, win 501, options [nop,no
p,TS val 2964406123 ecr 3681514473], length 0
19:04:03.209001 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1733:1769, ack 176, win 501,
options [nop,nop,TS val 3681514520 ecr 2964406123], length 36
19:04:03.209026 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1769, win 501, options [nop,no
p,TS val 2964406123 ecr 3681514520], length 0
19:04:03.212888 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 176:248, ack 1769, win 501, o
ptions [nop,nop,TS val 2964406126 ecr 3681514520], length 72
19:04:03.213086 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1769:1805, ack 248, win 501,
options [nop,nop,TS val 3681514524 ecr 2964406126], length 36
19:04:03.213129 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1805, win 501, options [nop,no
p,TS val 2964406127 ecr 3681514524], length 0
```

```
root@da8b1a95ab9b:/# ./socks_clientB2.py
HTTP/1.0 200 OK
Age: 398332
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sun, 01 Dec 2024 19:04:03 GMT
Etag: "3147526947+ident"
Expires: Sun, 08 Dec 2024 19:04:03 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECAcc (dcd/7D34)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
Connection: close

<!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, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-se
rif;

    }
    div {
```

The result of tcpdump:

```
root@492633afbdb2:/# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
19:04:03.089734 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 33030156:33030256, ack 343239
5873, win 501, options [nop,nop,TS val 2964406003 ecr 3681502973], length 100
19:04:03.089797 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [.], ack 100, win 501, options [nop,nop
,TS val 3681514400 ecr 2964406003], length 0
19:04:03.124855 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1:45, ack 100, win 501, optio
ns [nop,nop,TS val 3681514435 ecr 2964406003], length 44
19:04:03.125506 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 100:176, ack 45, win 501, opt
ions [nop,nop,TS val 2964406039 ecr 3681514435], length 76
19:04:03.156885 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 45:1545, ack 176, win 501, op
tions [nop,nop,TS val 3681514468 ecr 2964406039], length 1500
19:04:03.156925 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1545, win 501, options [nop,no
p,TS val 2964406071 ecr 3681514468], length 0
19:04:03.162457 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1545:1733, ack 176, win 501,
options [nop,nop,TS val 3681514473 ecr 2964406071], length 188
19:04:03.208957 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1733, win 501, options [nop,no
p,TS val 2964406123 ecr 3681514473], length 0
19:04:03.209001 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1733:1769, ack 176, win 501,
options [nop,nop,TS val 3681514520 ecr 2964406123], length 36
19:04:03.209026 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1769, win 501, options [nop,no
p,TS val 2964406123 ecr 3681514520], length 0
19:04:03.212888 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 176:248, ack 1769, win 501, o
ptions [nop,nop,TS val 2964406126 ecr 3681514520], length 72
19:04:03.213086 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.45726: Flags [P.], seq 1769:1805, ack 248, win 501,
options [nop,nop,TS val 3681514524 ecr 2964406126], length 36
19:04:03.213129 IP B-192.168.20.99.net-192.168.20.0.45726 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1805, win 501, options [nop,no
p,TS val 2964406127 ecr 3681514524], length 0
```

Here we used tcpdump to capture network traffic and analyse the packets during the dynamic port forwarding process. It helped to get clear evidence that the data was being routed through the SSH tunnel from HostB, bypassing the firewall restrictions and allowing access to blocked websites. The tool effectively helped monitor the success of dynamic port forwarding and its ability to evade firewall rules.

Task 3: Virtual Private Network (VPN)

Task 3.1: Bypassing Ingress Firewall

Creating the VPN tunnel From A to B with password dees: (VPN Tunnel Setup)

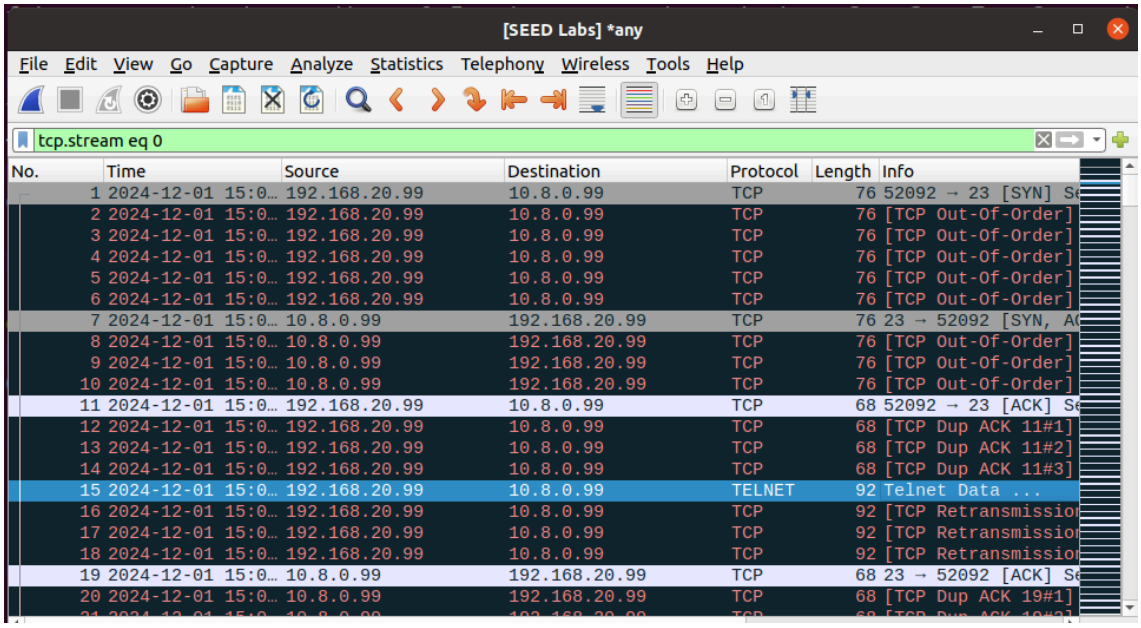
```
[12/01/24]seed@VM:~/.../Labsetup$ dockps
9c479850df8d  A-10.8.0.99
fffd637fd7c2  A2-10.8.0.6
9833a2b2201b  B1-192.168.20.5
0c26c3d7ba24  A1-10.8.0.5
572492eee8bc  B2-192.168.20.6
2929237ecb4d  router-firewall
a040081b67aa  B-192.168.20.99
[12/01/24]seed@VM:~/.../Labsetup$ docksh A-10.8.0.99
root@9c479850df8d:/# ssh -w 0:0 root@192.168.20.99 \
> -o "PermitLocalCommand=yes" \
> -o "LocalCommand= ip addr add 192.168.53.88/24 dev tun0 && \
> ip link set tun0 up" \
> -o "RemoteCommand=ip addr add 192.168.53.99/24 dev tun0 && \
> ip link set tun0 up"
root@192.168.20.99's password:
```

Here, I set up the VPN tunnel using the ssh -w 0:0 command to create a TUN interface (tun0) on both Host A and Host B, enabling encrypted communication between the external and internal networks.

Container B (Telnet Demonstration)

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh B-192.168.20.99
root@a040081b67aa:/# telnet 10.8.0.99
Trying 10.8.0.99...
Connected to 10.8.0.99.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9c479850df8d login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sun Dec  1 19:59:17 UTC 2024 from 192.168.20.5 on pts/3
seed@9c479850df8d:~$
```

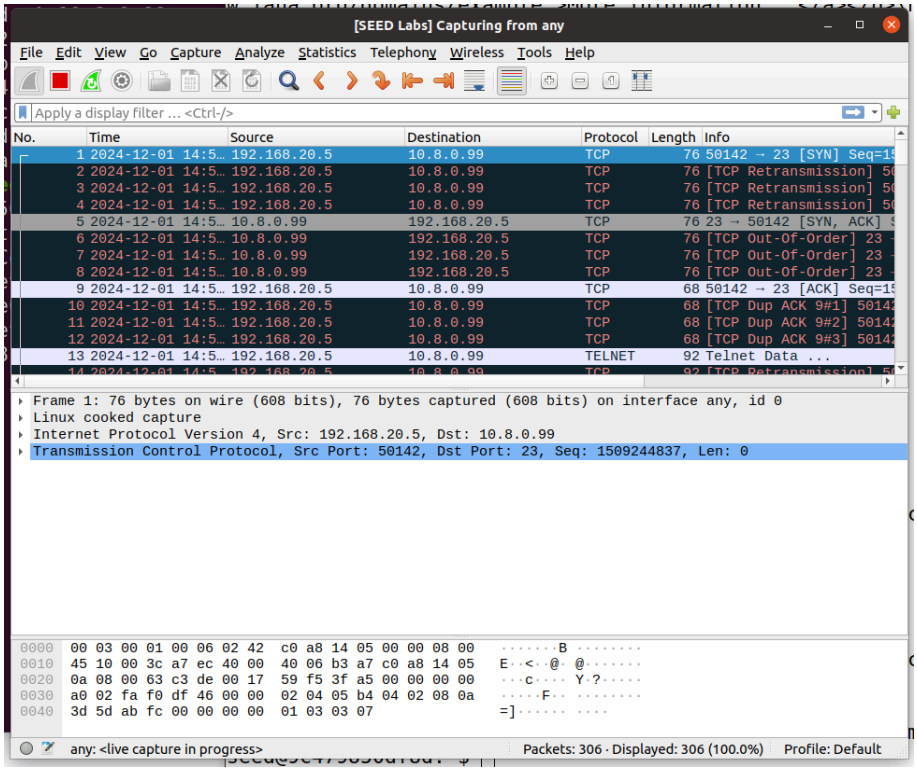
Packet Trace:

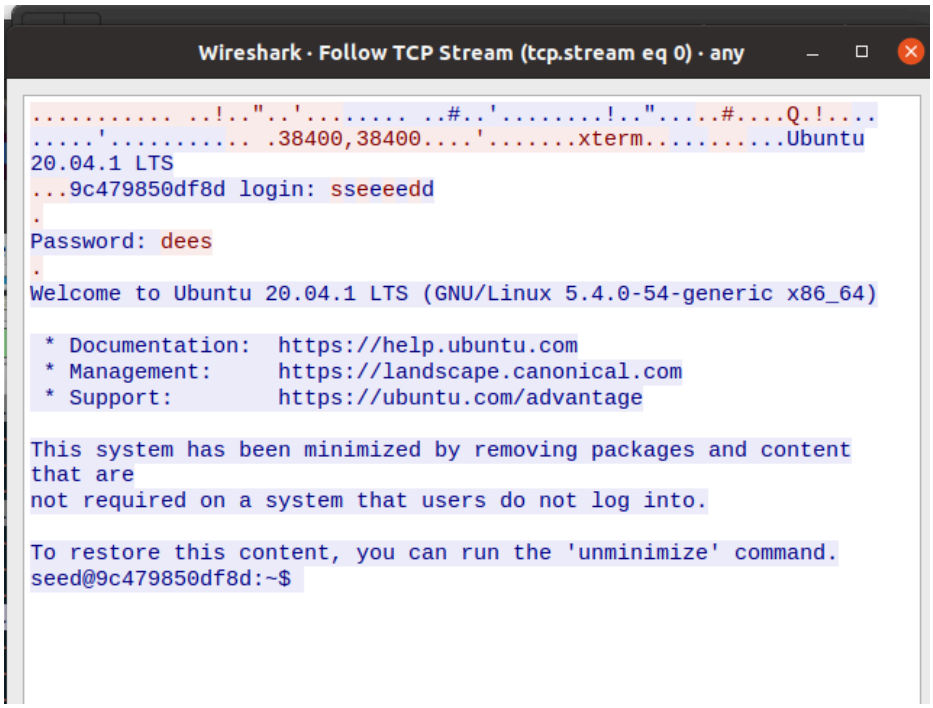


Container B1 (Telnet Demonstration)

```
[12/01/24]seed@VM:~/.../Labsetup$ dockps
9c479850df8d  A-10.8.0.99
fffd637fd7c2  A2-10.8.0.6
9833a2b2201b  B1-192.168.20.5
0c26c3d7ba24  A1-10.8.0.5
572492eee8bc  B2-192.168.20.6
2929237ecb4d  router-firewall
a040081b67aa  B-192.168.20.99
[12/01/24]seed@VM:~/.../Labsetup$ docksh B1-192.168.20.5
root@9833a2b2201b:/# telnet 10.8.0.99
Trying 10.8.0.99...
Connected to 10.8.0.99.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
9c479850df8d login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
seed@9c479850df8d:~$
```

Packet Trace:

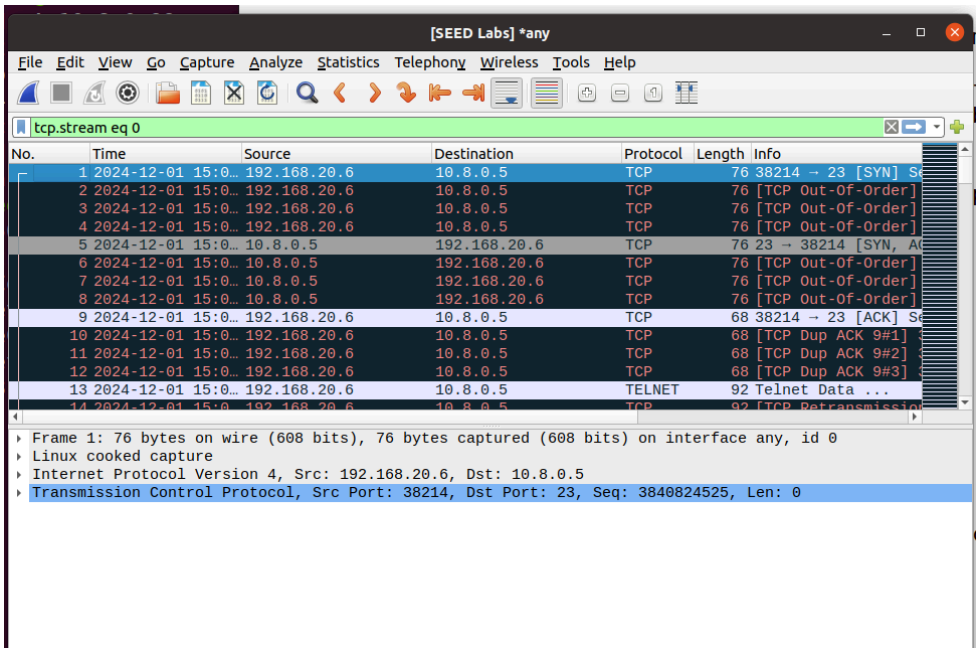


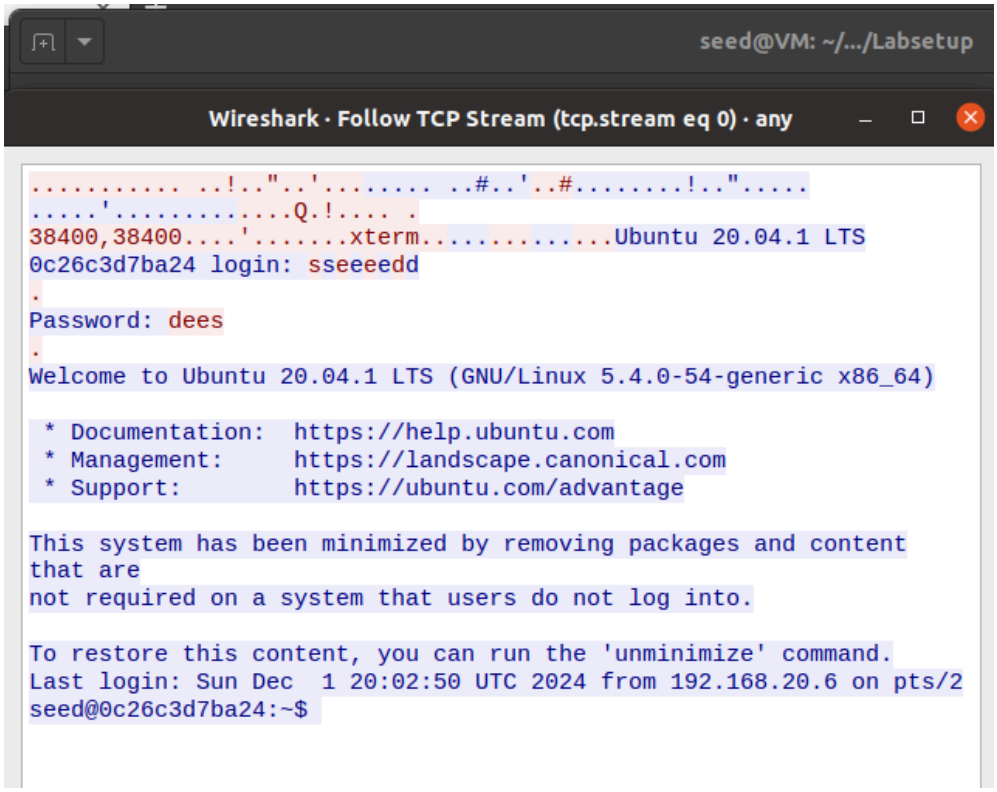


Container B2 (Telnet Demonstration)

```
root@572492eee8bc:/# telnet 10.8.0.5
Trying 10.8.0.5...
Connected to 10.8.0.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
0c26c3d7ba24 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sun Dec  1 20:02:50 UTC 2024 from 192.168.20.6 on pts/2
seed@0c26c3d7ba24:~$
```

Packet Trace:



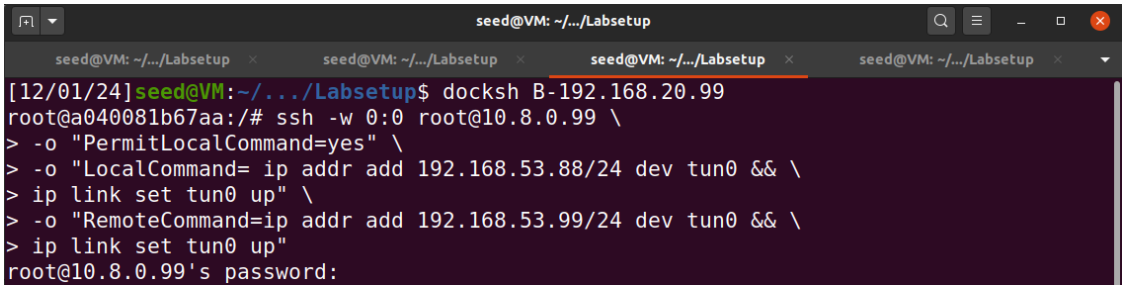


The above screenshots show a successful telnet connection from Host A to Hosts B, B1, and B2, demonstrating that the VPN tunnel bypasses ingress firewall restrictions. The packet trace shows encapsulated traffic from Host A to Host B over the VPN tunnel and forwarded traffic within the internal network to B1 and B2.

The VPN tunnel uses SSH to encapsulate packets from Host A to Host B. This creates an encrypted connection, making the traffic appear as standard SSH traffic. The ingress firewall rules allow SSH packets on port 22 but block other direct TCP connections. Since the actual traffic is encapsulated within SSH, the firewall does not inspect or block it, effectively bypassing the restrictions.

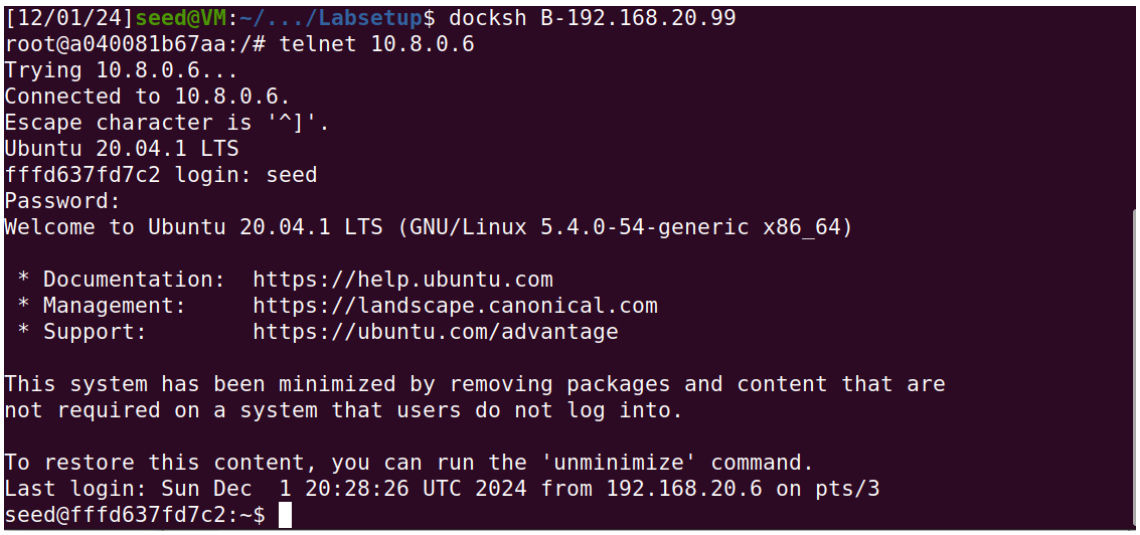
Task 3.2: Bypassing Egress Firewall

Creating the VPN tunnel From B to A with password dees: (VPN Tunnel Setup)

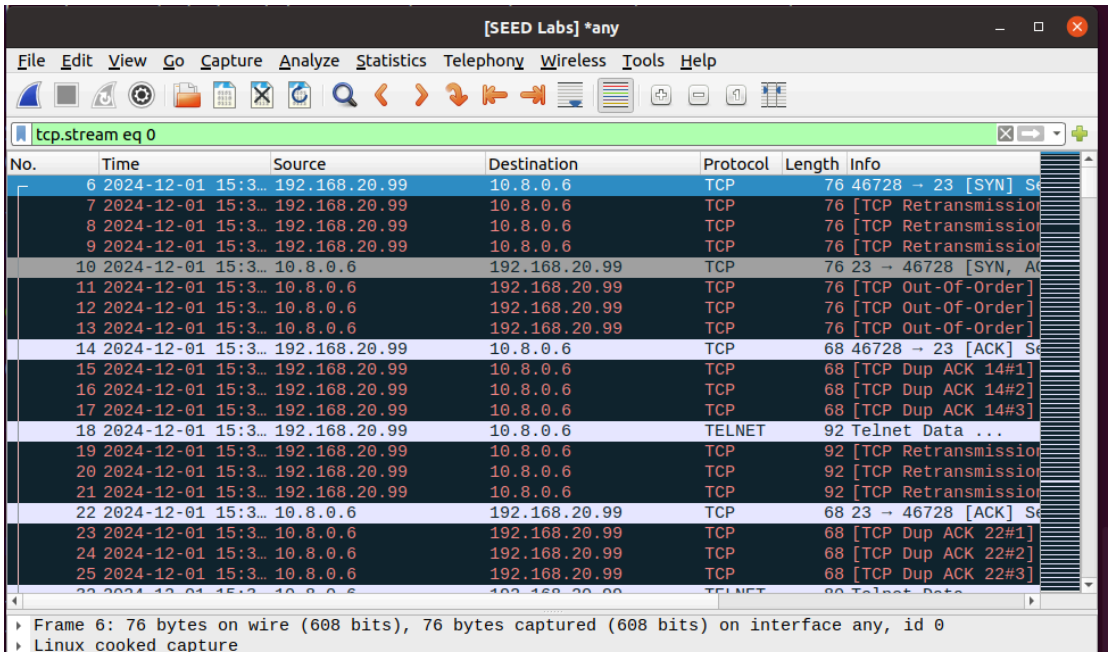


Here, we set up the VPN tunnel between Host A and Host B, with NAT on Host A (iptables -t nat -A POSTROUTING) to allow traffic from **192.168.53.0/24** to reach external websites.

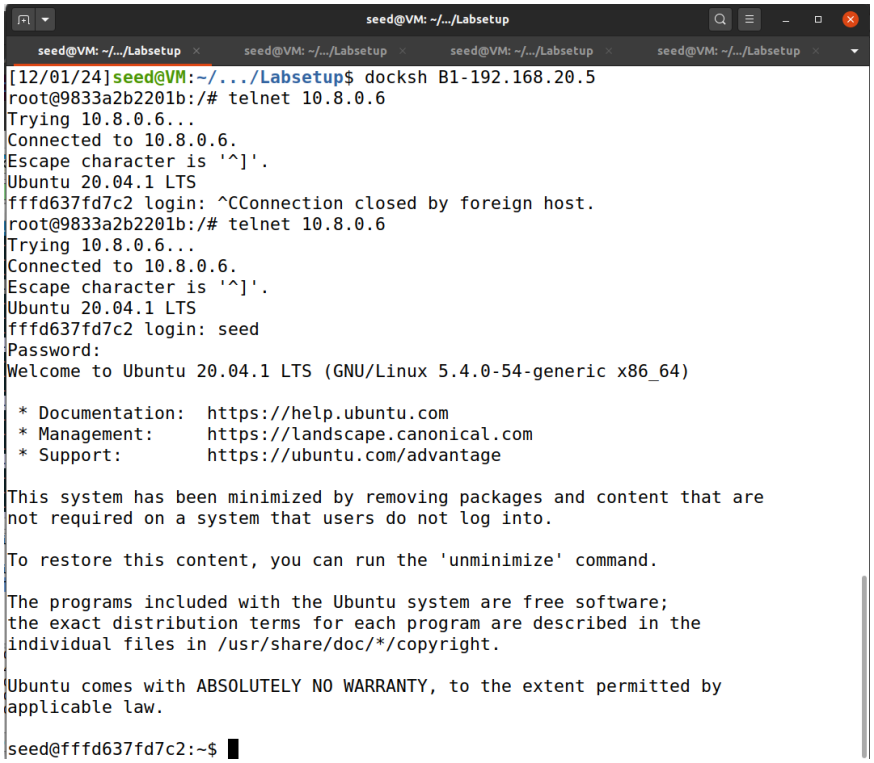
Container B (Telnet Demonstration)



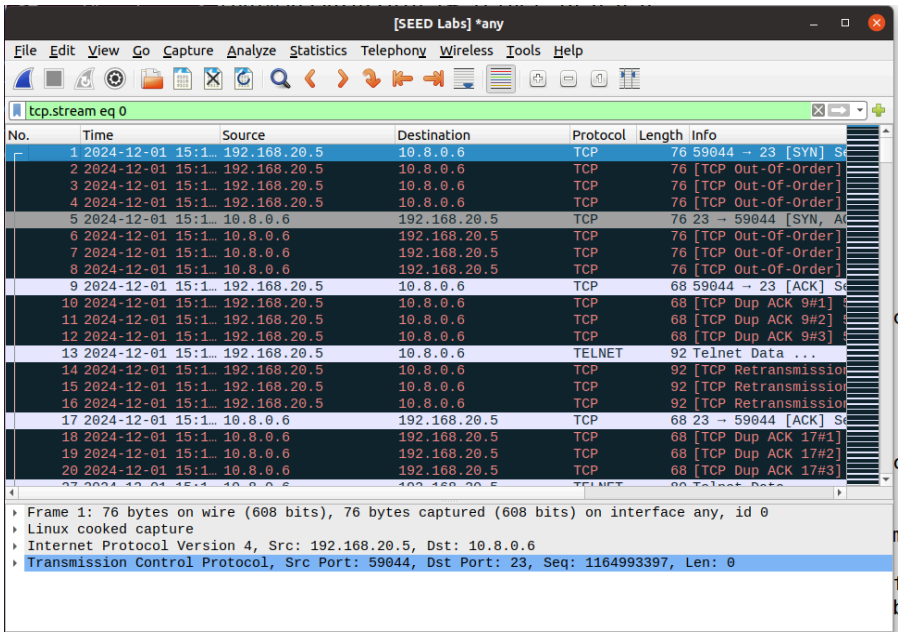
Packet Trace:

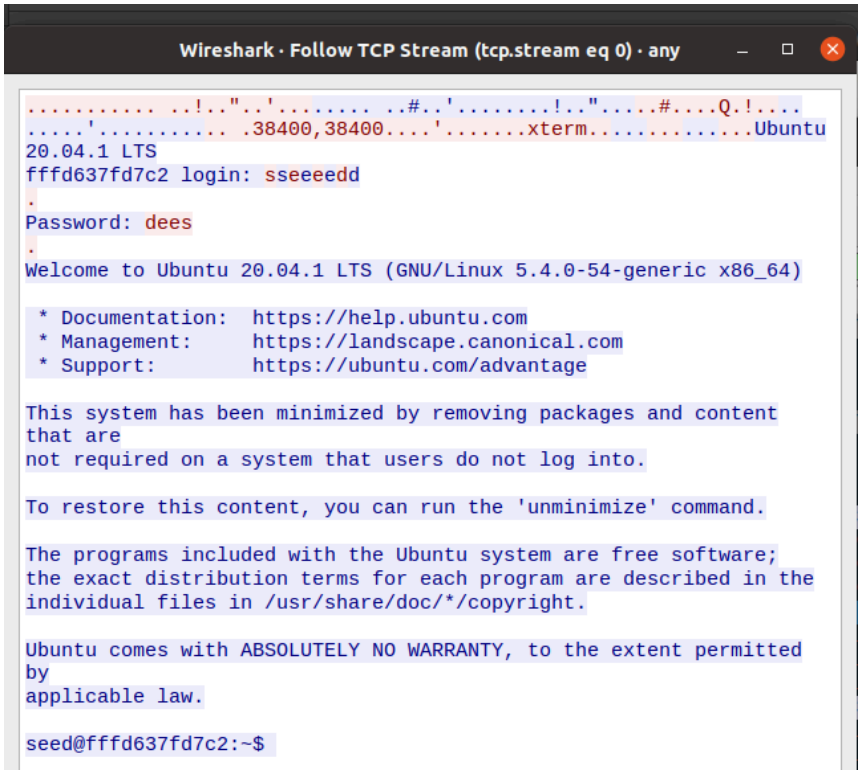


Container B1 (Telnet Demonstration)



Packet Trace:

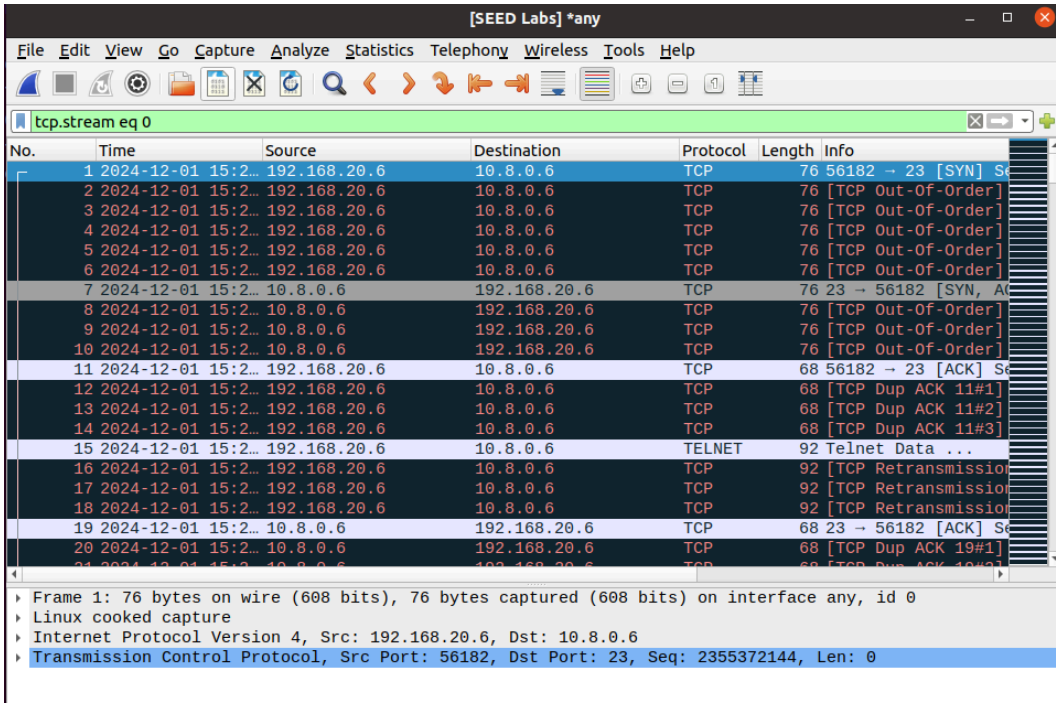




Container B2 (Telnet Demonstration)

```
[12/01/24]seed@VM:~/.../Labsetup$ docksh B2-192.168.20.6
root@572492eee8bc:/# telnet 10.8.0.6
Trying 10.8.0.6...
Connected to 10.8.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
fffd637fd7c2 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sun Dec  1 20:14:55 UTC 2024 from 192.168.20.5 on pts/2
seed@fffd637fd7c2:~$
```

Packet Trace:



The above screenshots shows how traffic from B, B1, and B2 to the blocked websites is routed through the VPN tunnel and NATed at Host A, bypassing egress firewall restrictions. We also

observe that in Wireshark traffic originating from Container B is sent through the tunnel to Host A. NAT at Host A changes the source IP to 10.8.0.99 before forwarding packets to 10.8.0.6.

NAT on Host A modifies the source IP of VPN traffic originating from the internal network, replacing it with Host A's external IP. This makes the traffic appear as if it originates from Host A, which is allowed by the egress firewall. As a result, packets destined for blocked websites are not identified as coming from restricted internal hosts and pass through the firewall unblocked.

Task 4: Comparing SOCKS5 Proxy and VPN

In this lab, we compared the SOCKS5 proxy (via dynamic port forwarding) and VPN (using SSH tunneling) to bypass firewalls and enhance privacy. A SOCKS5 proxy routes traffic through a proxy server, hiding IP addresses but not providing encryption. This was demonstrated in Task 2, where dynamic port forwarding allowed us to configure specific applications, such as browsers or telnet clients, to bypass restrictions. In contrast, a VPN establishes a system-wide encrypted tunnel at the network layer, routing all traffic securely through the tunnel without requiring individual application configuration, as seen in Tasks 3.1 and 3.2.

SOCKS5 proxies are lightweight and introduce minimal latency because they lack encryption, making them faster for tasks like streaming or general browsing. However, this simplicity comes with drawbacks: they only apply to configured applications, offer limited privacy, and require manual setup for each application. VPNs, on the other hand, encrypt all transmitted data, providing robust security and privacy, as demonstrated in the lab when VPNs bypassed ingress and egress firewalls by encapsulating traffic in encrypted tunnels. However, this encryption adds overhead, resulting in slightly reduced speeds, and VPNs require more complex setup, including creating TUN interfaces and configuring routing and NAT rules.

Overall, SOCKS5 proxies are ideal for scenarios where speed and simplicity are prioritized over security, such as bypassing geographical restrictions for streaming services. VPNs, however, are better suited for comprehensive security, privacy, and bypassing restrictions on all network traffic, as demonstrated in the lab. The choice between the two depends on specific use cases: SOCKS5 for lightweight, application-specific use, and VPNs for robust, system-wide encryption and privacy.