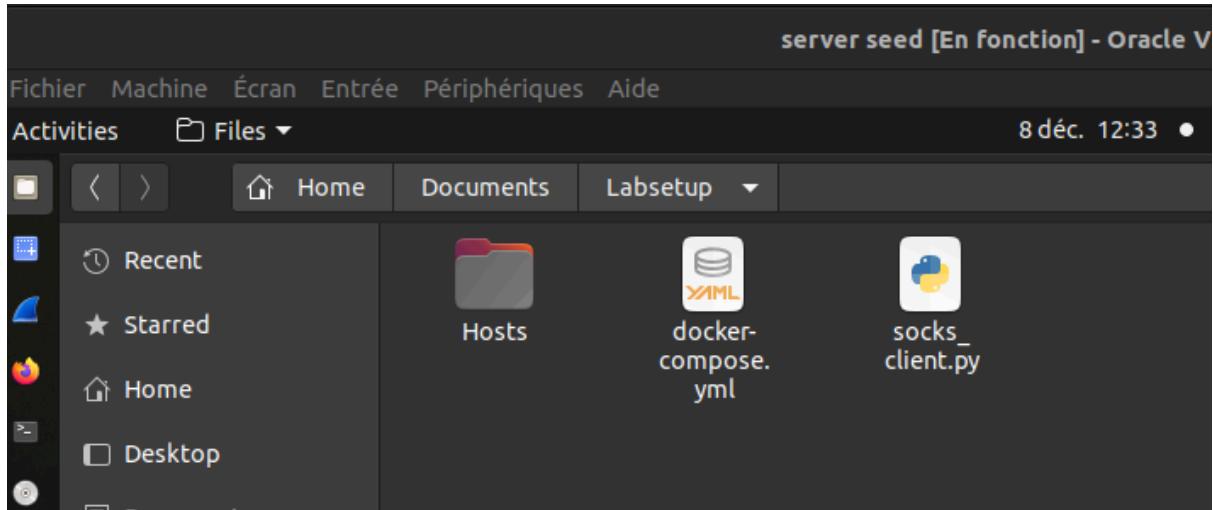


# SEED : FIREWALL EVASION LAB

## Task 0: Get Familiar with the Lab Setup

On télécharge le lab setup



On lance le docker-compose.yml avec la commande `docker-compose up`

```
server seed [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Activities Terminal 8 déc. 12:34 •
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
--> 34329c9573cc

Successfully built 34329c9573cc
Successfully tagged seed-image-ubuntu-hosts:latest
● WARNING: Image for service hostA was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Creating B2-192.168.20.6 ... done
Creating A1-10.8.0.5 ... done
Creating A-10.8.0.99 ... done
Creating router-firewall ... done
Creating B-192.168.20.99 ... done
Creating B1-192.168.20.5 ... done
Creating A2-10.8.0.6 ... done
Attaching to B2-192.168.20.6, A-10.8.0.99, B1-192.168.20.5, A2-10.8.0.6, A1-10.8.0.5, route
-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 ]
B1-192.168.20.5 | * Starting internet superserver inetd [ OK ]
B2-192.168.20.6 | * Starting internet superserver inetd [ OK ]
A2-10.8.0.6 | * Starting internet superserver inetd [ OK ]
A1-10.8.0.5 | * Starting internet superserver inetd [ OK ]
B-192.168.20.99 | * Starting internet superserver inetd [ OK ]
router-firewall | * Starting internet superserver inetd [ OK ]
B-192.168.20.99 | * Starting OpenBSD Secure Shell server sshd [ OK ]
```

On va bloquer l'ip de facebook (185.60.219.35) et twitter (104.244.42.193) (ip donnée par nslookup)

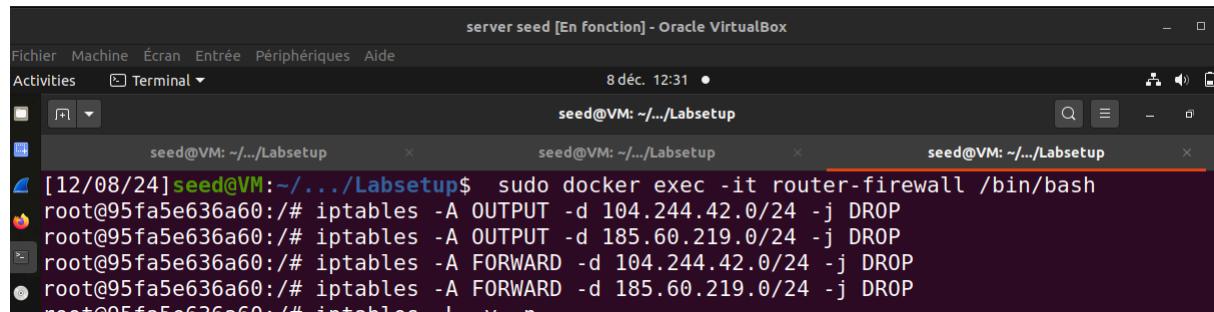
```
[12/08/24] seed@VM:~$ nslookup facebook.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:  facebook.com
Address: 185.60.219.35
Name:  facebook.com
Address: 2a03:2880:f17b:88:face:b00c:0:25de

[12/08/24] seed@VM:~$ nslookup twitter.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:  twitter.com
Address: 104.244.42.193
Name:  twitter.com
Address: 64:ff9b::68f4:2ac1
```

Dans le firewall docker on bloque l'ip de facebook et de twitter (185.60.219.0/24 et 104.244.42.0/24 )



```
server seed [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Activités Terminal 8 déc. 12:31 •
seed@VM: ~.../Labsetup
seed@VM: ~.../Labsetup
seed@VM: ~.../Labsetup
[12/08/24] seed@VM:~.../Labsetup$ sudo docker exec -it router-firewall /bin/bash
root@95fa5e636a60:/# iptables -A OUTPUT -d 104.244.42.0/24 -j DROP
root@95fa5e636a60:/# iptables -A OUTPUT -d 185.60.219.0/24 -j DROP
root@95fa5e636a60:/# iptables -A FORWARD -d 104.244.42.0/24 -j DROP
root@95fa5e636a60:/# iptables -A FORWARD -d 185.60.219.0/24 -j DROP
root@95fa5e636a60:/# iptables -L
```

pareil pour le site parrot.live 206.189.36.145/24 (site qui affiche un perroquet sur terminal apres un curl) **iptables -A FORWARD -d 206.189.36.145/24 -j DROP**

On drop les paquets si le routeur lui même ou les paquets qu'il forward ont pour destination les ip de facebook ou twitter

```
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
  0    0 ACCEPT      tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0
  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
  9  756 DROP        all   --  *      *       0.0.0.0/0            104.244.42.0/24
  4  336 DROP        all   --  *      *       0.0.0.0/0            185.60.219.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            104.244.42.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            104.244.42.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            185.60.219.0/24

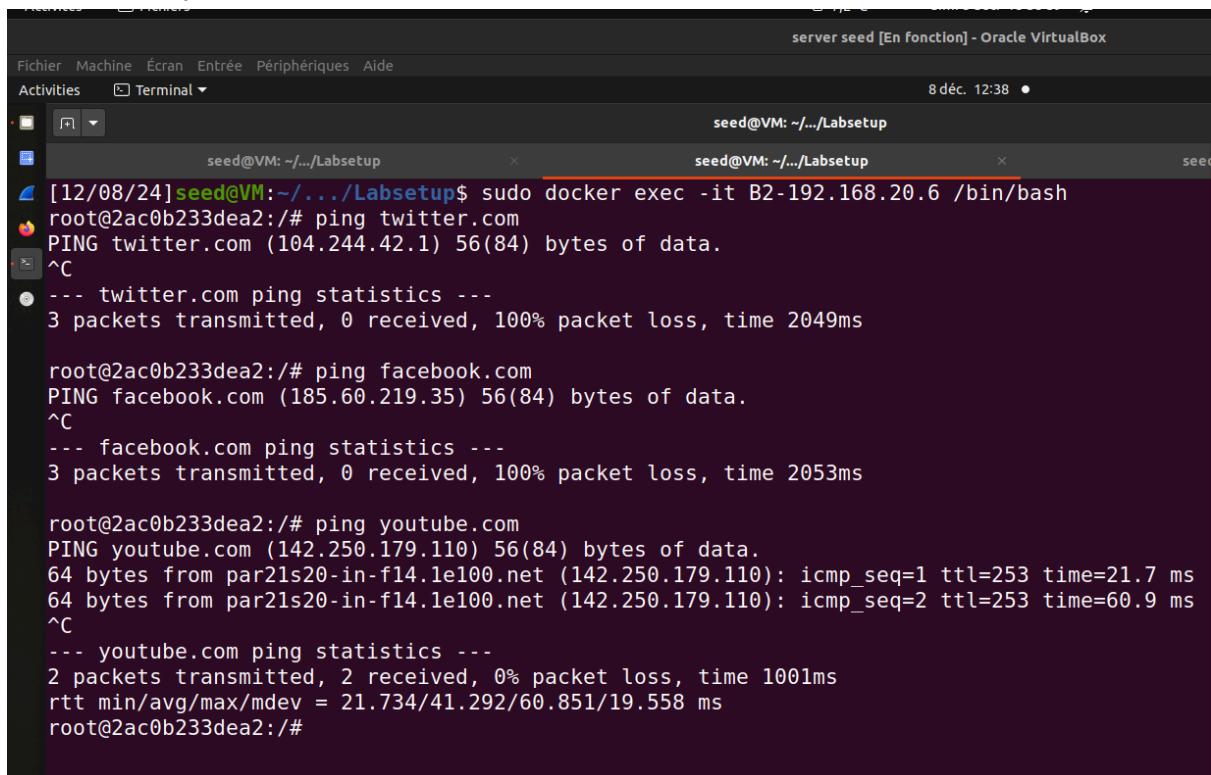
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in      out      source               destination
  2  168 DROP        all   --  *      *       0.0.0.0/0            104.244.42.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            185.60.219.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            104.244.42.0/24
  0    0 DROP        all   --  *      *       0.0.0.0/0            185.60.219.0/24
root@95fa5e636a60:/#
```

comme on peut le voir le firewall ne peux plus ping les ip et nom de domaine (et peux pour google.com)

```
root@95fa5e636a60:/# ping twitter.com
PING twitter.com (104.244.42.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- twitter.com ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1035ms

root@95fa5e636a60:/# ping google.com
PING google.com (142.250.201.174) 56(84) bytes of data.
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=1 ttl=254 time=30.1 ms
64 bytes from par21s23-in-f14.1e100.net (142.250.201.174): icmp_seq=2 ttl=254 time=39.2 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 30.089/34.658/39.227/4.569 ms
root@95fa5e636a60:/#
```

essayons sur les machines du groupe B , elles peuvent pas ping twitter et facebook (et peuvent pour youtube)



```
Fichier Machine Écran Entrée Périphériques Aide
Activities Terminal 8 déc. 12:38 •
server seed [En fonction] - Oracle VirtualBox
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
[12/08/24] seed@VM:~/.../Labsetup$ sudo docker exec -it B2-192.168.20.6 /bin/bash
root@2ac0b233dea2:/# ping twitter.com
PING twitter.com (104.244.42.1) 56(84) bytes of data.
^C
--- twitter.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2049ms

root@2ac0b233dea2:/# ping facebook.com
PING facebook.com (185.60.219.35) 56(84) bytes of data.
^C
--- facebook.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2053ms

root@2ac0b233dea2:/# ping youtube.com
PING youtube.com (142.250.179.110) 56(84) bytes of data.
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=1 ttl=253 time=21.7 ms
64 bytes from par21s20-in-f14.1e100.net (142.250.179.110): icmp_seq=2 ttl=253 time=60.9 ms
^C
--- youtube.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 21.734/41.292/60.851/19.558 ms
root@2ac0b233dea2:/#
```

## Task 1: Static Port Forwarding

- Vérifions le docker container :

| CONTAINER ID | IMAGE                              | COMMAND                  | CREATED       | STATUS       | PORTS | NAMES           |
|--------------|------------------------------------|--------------------------|---------------|--------------|-------|-----------------|
| 53937483e8f3 | handsongsecurity/seed-ubuntu:large | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | router-firewall |
| 479b58ef1362 | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | A1-10.8.0.5     |
| 89e1bf75c4bc | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | B-192.168.20.99 |
| d68c40f1e269 | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | A2-10.8.0.6     |
| c213d293b1e1 | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | A-10.8.0.99     |
| 61e7d6b76f58 | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | B2-192.168.20.6 |
| 2df36540ed32 | seed-image-ubuntu-hosts            | "bash -c ' ip route ..." | 3 minutes ago | Up 3 minutes |       | B1-192.168.20.5 |

| INTERFACE        | STATE | IP ADDRESS                                 |
|------------------|-------|--|
| docker0          | DOWN  | 172.17.0.1/16 fe80::42:4ff:fe92:bd13/64    |
| br-67cf9d7ca6e1  | UP    | 10.8.0.1/24 fe80::42:6eff:feaf:93f5/64     |
| br-503426c03eeef | UP    | 192.168.20.1/24 fe80::42:41ff:fe3a:d027/64 |
| veth1df8cb3@if10 | UP    | fe80::acd3:47ff:fe58:5453/64               |
| veth0d4fd07@if12 | UP    | fe80::bc23:b2ff:fe25:6436/64               |
| veth732a49d@if14 | UP    | fe80::14d9:cbff:fe07:2544/64               |
| veth776fa34@if16 | UP    | fe80::ccc8:83ff:fe63:6b1a/64               |
| vethb8e54df@if18 | UP    | fe80::8c14:aff:fe78:5c2e/64                |
| veth214395f@if20 | UP    | fe80::48f5:a9ff:fe69:30f1/64               |
| veth268cff7@if22 | UP    | fe80::683d:ddff:fefc:855/64                |
| veth3732707@if24 | UP    | fe80::b41a:1aff:feef:18f0/64               |

- \*\*External Network:\*\* `10.8.0.0/24`
- \*\*Internal Network:\*\* `192.168.20.0/24`
- Log in en routeur:

```
(yassine@vbox)-[/media/sf_shared/Labsetup]
$ sudo docker exec -it router-firewall bash
```

root@53937483e8f3:/#

- installons et lançons ssh:

```
root@53937483e8f3:/# apt install openssh-server -y
Reading package lists... 0%          compose.yml
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@53937483e8f3:/#
root@53937483e8f3:/# service ssh start
 * Starting OpenBSD Secure Shell server sshd                                         [ OK ]
root@53937483e8f3:/# netstat -tuln | grep 22
tcp        0      0 0.0.0.0:22              0.0.0.0:*                  LISTEN
tcp6       0      0 ::1:22                 ::*                      LISTEN
root@53937483e8f3:/#
```

- Configurons le mot de passe du routeur:

```
(yassine@vbox)-[/media/sf_shared/Labsetup]
$ sudo docker exec -it router-firewall bash
```

root@ab42586b0fd9:/# passwd
 new password:
 retype new password:
passwd: password updated successfully

- et finalement on met le port forwarding:

```
(yassine@vbox)-[~/media/sf_shared/Labsetup]
$ sudo ssh -4NT -L 0.0.0.0:8080:192.168.20.5:23 root@10.8.0.11
root@10.8.0.11's password:
```

1. Nombre de connexions TCP :
  - Au moins deux :
    - Entre Kali (Host A) et le routeur (Host B) pour SSH.
    - Entre le routeur (Host B) et le serveur interne (Host B1) pour Telnet.
2. Pourquoi cela contourne-t-il le pare-feu ?
  - Le pare-feu bloque les connexions directes aux serveurs internes.
  - Le tunnel SSH encapsule le trafic Telnet, le rendant "invisible" au pare-feu.

## Task 2: Dynamic Port Forwarding

### Task 2.1: Setting Up Dynamic Port Forwarding

`ssh -4NT -D 192.168.20.99:9999 seed@10.8.0.99`

cette commande établit un tunnel SSH et configure un proxy SOCKS sur l'adresse `192.168.20.99:9999`. Et on peut voir que l'on peut curl [www.example.com](http://www.example.com) et parrot.live grâce au tunnel ssh (et que sans tunnel on peut pas) depuis la machine `B1-192.168.20.5` avec la commande `curl -proxy socks5h://192.168.20.99:9999 www.example.com (ou parrot.live)`

```
Activities Terminal
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
[12/08/24] seed@VM:~/.Labsetup$ sudo docker exec -it B-192.168.20.99 /bin/bash
root@78e4cf6ed97b:/# ssh -4NT -D 192.168.20.99:9999 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 SHA256:81031FAgDNA/WmQJmFal8s7VWydYdszu8xDz
99w8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? ye
s
Warning: Permanently added '10.8.0.99' (ECDSA) to the list of known hosts.
seed@10.8.0.99's password:
```

```
[12/08/24] seed@VM:~$ sudo docker exec -it B-192.168.20.5 /bin/bash
root@cc277b328cf31:/# curl --proxy socks5h://192.168.20.99:9999 www.example.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, "Segoe UI", "Open Sans", "Helvetica Neue", Arial, sans-serif;
    }
    div {
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
```

Fichier Machine Écran Entrée Périphériques Aide  
Activities Terminal

server.seed [En Fonction] - Oracle VirtualBox

seed@VM: ~\$ sudo docker exec -it B-192.168.20.99 /bin/bash

root@78e4cf6ed07b:/# ssh -4NT -D 192.168.20.99:9999 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 SHA256:8l031FaGnA/WmQPJmFal8s7vgWydYdszu8x/Dz9w8.

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:

.....  
.ckKxodoox00dcc.  
.ccloo'....';coo!.  
.loc;;;cllllc;;;;;;.  
.c:'.;okd;;cdo::::cl..oc  
.io:';okx';;;:;:;:;:;.  
co..ckkkkkddkc,cclll:..c:;:o:  
co..ckkkkkkkk;.cllli:..ckd:.';c.  
.;;:okkkkkkkk;.cclll;.ckkkd:;o:  
cNo..ckkkkkkkkkko,,;loc,.ckkkkkc.oc  
.dd:;ckkkkkkkkRx;:;:;,'lkkkkko,.,.  
.;;:ckkkkkkkkkkkc.....;ldkkkkk:;,'.  
.dc:'.okkkkkkkkkkxoc;cxkkkkkkkc;.,.  
kNo:'.llllldkkkkkkkkkkkkkkkkkkdce;,il.  
K0c,c:'';';';lldkkkkkkkkkkkkkkkkc:.;lc.  
xx:':';';';';';cllllllllllllc:'.;od,  
cNo.....oc  
^C

root@c277b328cf31:# ping 206.189.36.145  
PING 206.189.36.145 (206.189.36.145) 56(84) bytes of data.  
^C  
--- 206.189.36.145 ping statistics ---  
3 packets transmitted, 0 received, 100% packet loss, time 2136ms  
root@c277b328cf31:# curl parrot.live  
^C  
root@c277b328cf31:#

Essayon depuis B2-192.168.20.6 pour prouver que c'est bien dynamique avec la commande `curl --proxy socks5h://192.168.20.99:9999` [www.example.com](http://www.example.com) (ou [www.parrot.live](http://www.parrot.live)) (qui affiche le html de la page)

```
ficher Machine Ecran Entre Périphériques Aide
Activities Terminal server seed [En fonction] - Oracle VirtualBox
seed@VM: ~]$ ./Labsetup
seed@VM: ~]$ ./Labsetup
seed@VM: ~]$ ./Labsetup
[12/08/24] seed@VM:~$ sudo docker exec -it B-192.168.20.99/bin/bash
root@2a0b23de2a:/# curl -x socks5h://192.168.20.99:9999 www.example.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: #f0f0f0;
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
div {
    width: 600px;
    margin: 5em auto;
    padding: 2em;
    background-color: #fdfdff;
    border-radius: 0.5em;
    box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
    color: #38488f;
    text-decoration: none;
}
@media (max-width: 700px) {
```

## Quel ordinateur établit la connexion avec le serveur web ? :

C'est l'ordinateur distant A (IP : 10.8.0.99) qui établit la connexion au serveur web cible.

### Comment sait-il à quel serveur se connecter ?:

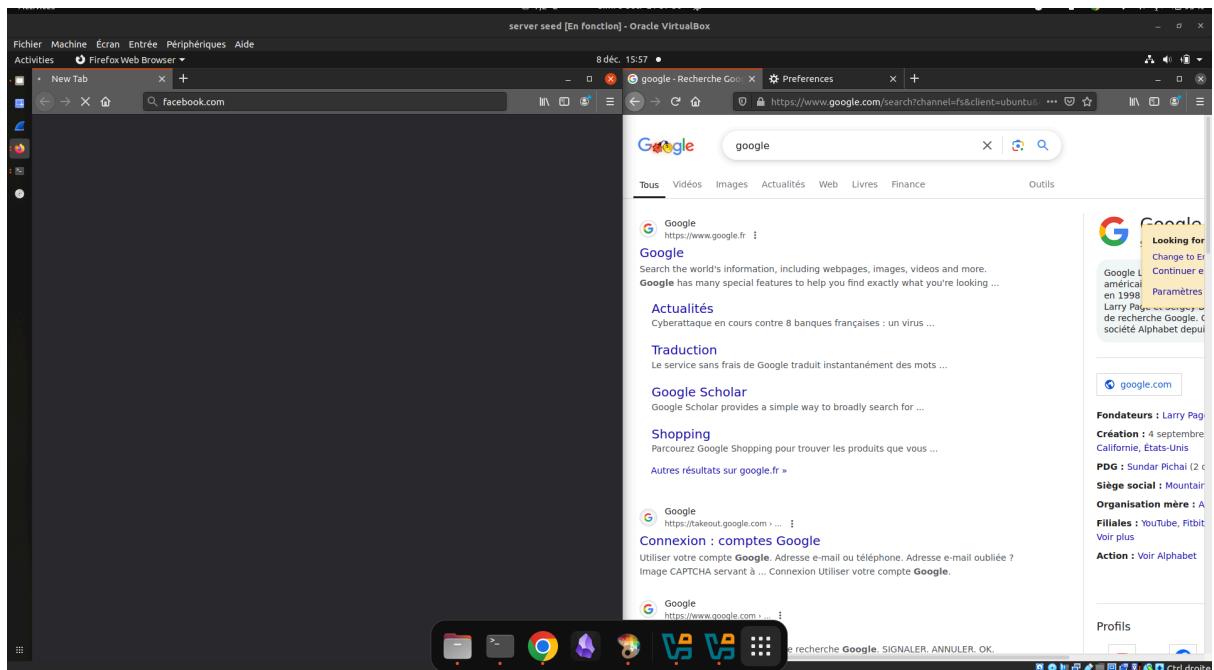
L'ordinateur A reçoit l'information sur le serveur cible via le protocole SOCKS, qui inclut l'adresse et le port dans chaque requête.

## Task 2.2: Testing the Tunnel Using Browser

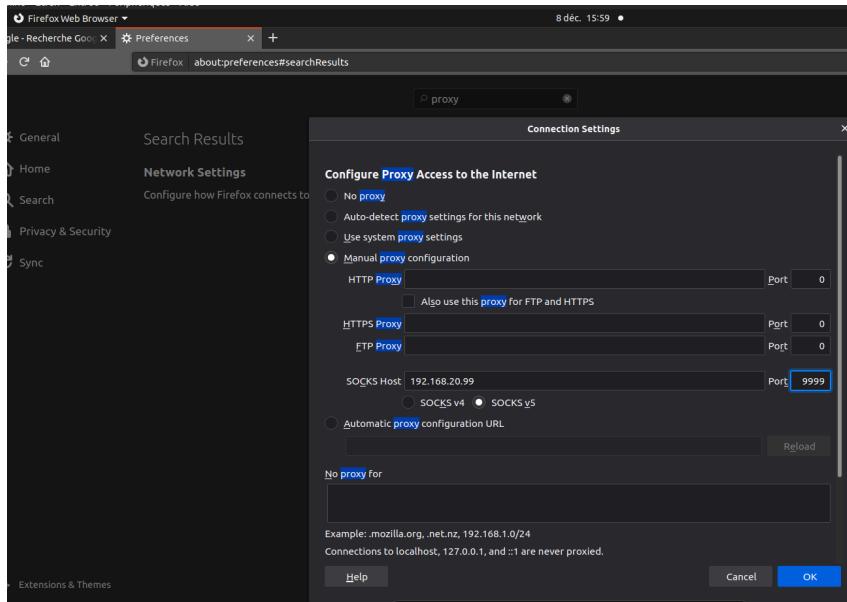
On reste connecté au serveur proxy ssh -4NT -D 192.168.20.99:9999 seed@10.8.0.99

Cette commande établit un **proxy SOCKS5** sur **192.168.20.99:9999** via une connexion SSH sécurisée au serveur **10.8.0.99**, permettant de rediriger le trafic réseau à travers le tunnel SSH.

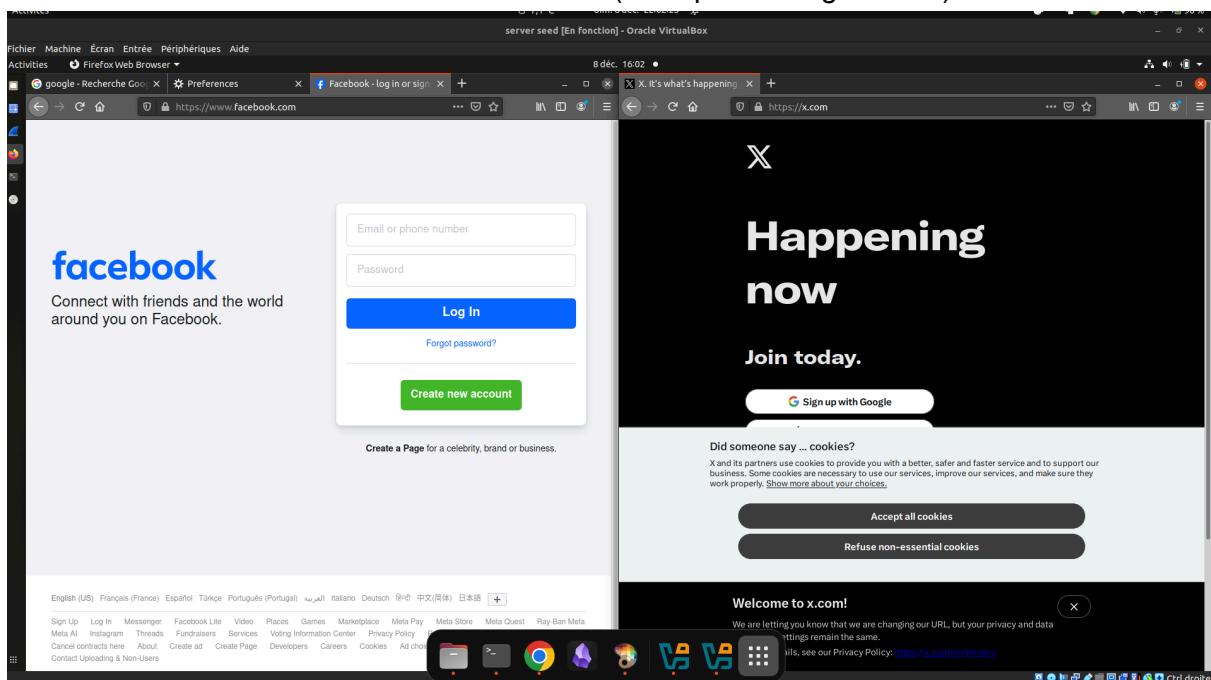
Comme on peut voir on ne peux pas accéder depuis le browser à facebook.com (a gauche) mais on peut accéder à google.com (a droite)



On va donc ajouter un proxy sur firefox (192.168.20.99:9999)



Et maintenant on a accès à facebook et twitter (example.com également)



On peut également voir le tcpdump avant et après la désactivation du proxy

```

Fichier Machine Écran Entrée Périphériques Aide
Activités Terminal
server seed [En fonction] - Oracle VirtualBox
8 déc. 16:17 •
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
21:15:37.935154 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1201225, win 2548, options [nop,nop,TS val 306193286 ecr 35134364481], length 0
21:15:37.936538 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 35724:35800, ack 1201225, win 2548, options [nop,nop,TS val 306193288 ecr 3513436448], length 76
21:15:37.936654 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [.], ack 35800, win 554, options [nop,nop,TS val 3513436450 ecr 3061932881], length 0
21:15:38.041651 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 35800:35948, ack 1201225, win 2548, options [nop,nop,TS val 306193393 ecr 3513436456], length 148
21:15:38.041869 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [.], ack 35948, win 554, options [nop,nop,TS val 3513436555 ecr 306193393], length 0
21:15:38.087839 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [P.], seq 1201225:1201941, ack 35948, win 554, options [nop,nop,TS val 3513436601 ecr 306193393], length 716
21:15:38.130352 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1201941, win 2548, options [nop,nop,TS val 306193482 ecr 3513436601], length 0
21:15:38.130420 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [P.], seq 1201941:1202017, ack 35948, win 554, options [nop,nop,TS val 3513436644 ecr 306193482], length 76
21:15:38.130469 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1202017, win 2548, options [nop,nop,TS val 306193482 ecr 3513436644], length 0
21:15:38.131520 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 35948:36024, ack 1202017, win 2548, options [nop,nop,TS val 306193483 ecr 3513436644], length 76
21:15:38.131679 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [.], ack 36024, win 554, options [nop,nop,TS val 3513436645 ecr 306193483], length 0
21:15:38.139514 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 36024:36084, ack 1202017, win 2548, options [nop,nop,TS val 306194091 ecr 3513436645], length 60
21:15:38.139575 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [.], ack 36084, win 554, options [nop,nop,TS val 3513437253 ecr 306194091], length 0
21:15:38.740308 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [F.], seq 36084, ack 1202017, win 2548, options [nop,nop,TS val 306194092 ecr 3513437253], length 0
21:15:38.745089 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.56208: Flags [F.], seq 1202017, ack 36085, win 554, options [nop,nop,TS val 3513437258 ecr 306194092], length 0
21:15:38.745148 IP B-192.168.20.99.net-192.168.20.0.56208 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 1202018, win 2548, options [nop,nop,TS val 306194096 ecr 3513437258], length 0

```

### (1) Analyse du trafic avec tcpdump

Les paquets capturés montrent un échange entre B (192.168.20.99) et A (10.8.0.99) via le port SSH (22).

A (10.8.0.99) établit les connexions vers les serveurs web cibles, transmettant les données encapsulées de B.

### (2) Rompre le tunnel SSH

Lorsque le tunnel SSH est coupé, tout le trafic vers les sites bloqués cesse complètement. Les requêtes de B échouent car elles ne peuvent plus passer par A. Le pare-feu bloque directement l'accès aux sites comme Facebook ou Twitter.

## Task 2.3: Writing a SOCKS Client Using Python

On reste toujours connecté au serveur proxy avec `ssh -4NT -D 192.168.20.99:9999`  
`seed@10.8.0.99`

Cette commande crée un **proxy SOCKS** sur **192.168.20.99:9999**, en passant par une connexion SSH au serveur distant **10.8.0.99** avec l'utilisateur **seed**. Le trafic réseau peut être acheminé via ce proxy pour contourner les restrictions réseau.

Voici le code python fournis adapté, On va tester sur l'host B1-192.168.20.5 et on voit bien

que ça fonctionne (on a une redirection https car on demande un http pour facebook)

The screenshot shows a Linux desktop environment with a terminal window and a browser window. The terminal window has two tabs: 'code.py' and 'seed@VM: ~'. The 'code.py' tab contains Python code for connecting to a SOCKS5 proxy and sending an HTTP GET request to www.facebook.com. The 'seed@VM: ~' tab shows the output of the command 'python3 code.py', which includes a response from the server indicating a permanent redirect to https://www.facebook.com. The browser window shows the Facebook login page.

```
#!/bin/env python3
import socks
s = socks.socksocket()
s.set_proxy(socks.SOCKS5, "192.168.20.99", 9999)
s.connect(("185.60.219.35", 80))
hostname = "www.facebook.com"
req = b"GET / HTTP/1.0\r\nHost: " + hostname.encode('utf-8') + b"\r\n\r\n"
s.sendall(req)
response = s.recv(2048)
while response:
    print(response.split(b"\r\n"))
    response = s.recv(2048)
```

Ce code Python permet de se connecter à un site web via un proxy SOCKS5. Voici une explication concise :

1. **Création d'un socket avec proxy SOCKS5 :**
  - o Le proxy est configuré sur **192.168.20.99** au port **9000**.
2. **Connexion à la destination finale :**
  - o Le script se connecte au site [www.example.com](http://www.example.com) sur le port HTTP (**80**) via le proxy SOCKS.
3. **Envoi d'une requête HTTP :**
  - o Une requête HTTP GET est formatée pour récupérer la page d'accueil du site.
4. **Réception et affichage de la réponse :**
  - o Le contenu reçu du serveur est lu par morceaux de **2048 octets** et affiché ligne par ligne.

Pareil pour B et B2 on a la redirection donc tout fonctionne

The screenshot shows a Linux terminal window with three tabs. The first tab shows the user navigating to a directory. The second tab shows the user running the Python script 'code.py'. The third tab shows the output of the command 'sudo docker exec -it B-192.168.20.6 /bin/bash', which includes a response from the server indicating a permanent redirect to https://www.facebook.com.

```
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
[12/08/24]seed@VM:~/.../Labsetup$ sudo docker exec -it B-192.168.20.6 /bin/bash
root@78e4cf6ed07b:/# python3 /home/seed/code.py
[b'HTTP/1.1 301 Moved Permanently', b'Location: https://www.facebook.com/',
 b'Content-Type: text/plain', b'Server: proxygen-bolt', b'Date: Sun, 08 Dec 2024 22:08:55 GMT', b'Connection: close', b'Content-Length: 0', b'']
root@78e4cf6ed07b:/#
```

## Task 3: Virtual Private Network (VPN)

### Task 3.1: Bypassing Ingress Firewall

On se commence avec préparer le vpn de A à B avec la commande

```
ssh -w 0:0 root@\ -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"
```

Ce code configure un tunnel VPN entre **A** et **B** à l'aide de SSH, en utilisant une interface virtuelle **tun0** sur chaque machine :

1. **192.168.53.88/24** : Adresse IP locale assignée à **A** sur l'interface tun0.
2. **192.168.53.99/24** : Adresse IP assignée à **B** sur l'interface tun0.
3. **Tunnel VPN actif** : Le trafic entre **A** et **B** passe par cette connexion chiffrée.

Cela permet de contourner les restrictions d'un pare-feu en encapsulant les paquets dans le tunnel VPN.

4o

The screenshot shows a terminal window titled 'seed@VM: ~.../Labsetup'. The terminal displays the following command being run:

```
[12/21/24]seed@VM:~/.../Labsetup$ sudo docker exec -it A-10.8.0.99 /bin/bash  
root@49be7a9ecdd8:/# 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"  
The authenticity of host '192.168.20.99 (192.168.20.99)' can't be established.  
ECDSA key fingerprint is SHA256:8l031FAgDnA/WmQPJmFal8s7VgWydYdszu8x/Dz90w8.  
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.  
root@192.168.20.99's password:
```

Maintenant on attend une connection avec telnet de B1  
a gauche on a le serveur et a droit le client telnet et on peut voir que l'on a réussi la

## connection

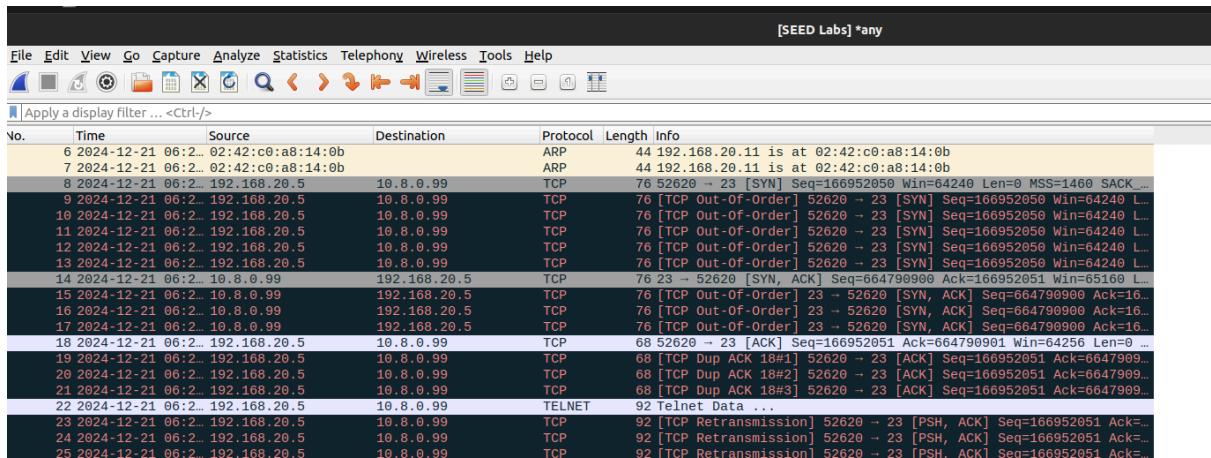
```
[12/21/24]seed@VM:~/.Labsetup$ sudo docker exec -it A-10.8.0.99 /bin/bash
in/bash
root@49be7a9ecdd8:/# ssh -w 0:0 root@192.168.20.99 \
> -o "PermitLocalCommand=yes" \
> -o "LocalCommand=ip addr add 192.168.53.88/24 dev tun0 && \
> ip link set tun0 up" \
> -o "RemoteCommand=ip addr add 192.168.53.99/24 dev tun0 && \
> ip link set tun0 up"
The authenticity of host '192.168.20.99 (192.168.20.99)' can't be established.
ECDSA key fingerprint is SHA256:8l031FAgDnA/WmQPJmFal8s7VgWydYdszu8x/Dz90w8.
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.
root@192.168.20.99's password:
[12/21/24]seed@VM:~/.Labsetup$ sudo docker exec -it B1-192.168.20.5 /bin/bash
root@c277b328cf31:# telnet 10.8.0.99
Trying 10.8.0.99...
Connected to 10.8.0.99.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
49be7a9ecdd8 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@49be7a9ecdd8:~$ ls
seed@49be7a9ecdd8:~$
```

Comme on peut le voir le telnet n'est pas bloqué par le firewall voici la capture du paquet wireshark



Dans la capture, les paquets Telnet apparaissent comme venant de **192.168.20.5** (B1) vers **10.8.0.99** (A). ces paquets transitent par le tunnel VPN et ne sont donc pas bloqué. Les paquets traversent le pare-feu car ils sont encapsulés dans le tunnel SSH via l'interface TUN, ce qui masque leur contenu réel et leur destination, le pare-feu ne voyant que du trafic SSH (port 22)

on fait pareil avec B2 et on voit que ca marche

```

[12/24/24]seed@VM:~/.../Labsetup$ sudo docker exec -it A-10.8.0.99 /bin/bash
root@49be7a9ecdd8:/# 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:

```

[12/24/24]seed@VM:~/.../Labsetup\$ sudo docker exec -it B-192.168.20.6 /bin/bash  
root@2ac0b233dea2:/# telnet 10.8.0.99  
Trying 10.8.0.99...  
Connected to 10.8.0.99.  
Escape character is '^].  
Ubuntu 20.04.1 LTS  
49be7a9ecdd8 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: Sat Dec 21 11:25:05 UTC 2024 from 192.168.20.5 on pts/2  
seed@49be7a9ecdd8:~\$

on peut également curl example.com

```

root@49be7a9ecdd8:/# exit
exit
[12/24/24]seed@VM:~/.../Labsetup$ sudo docker exec -it A-10.8.0.99 /bin/bash
root@49be7a9ecdd8:/# 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:

```

```

q=2 ttl=254 time=269 ms
^C
--- example.com ping statistics ---
3 packets transmitted, 2 received, 33.3333% packet loss, time 2025ms
rtt min/avg/max/mdev = 269.236/678.365/1087.494/409.129 ms, pipe 2
seed@49be7a9ecdd8:~$ curl example.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, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
        }
        div {
            width: 600px;
            margin: 5em auto;
            padding: 2em;
        }
    </style>

```

On voit bien le flux tcp et le mot de passe

```

Wireshark - Follow TCP Stream (tcp.stream eq 1) · any
..... .!..". ..... .#..'. ....!.."....#.....'.....4.....
.38400,38400...xterm.....Ubuntu 20.04.1 LTS
..49be7a9ecdd8 login: dd.. .sseeedd
.
Password: dees
.
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: Sat Dec 21 11:25:05 UTC 2024 from 192.168.20.5 on pts/2
seed@49be7a9ecdd8:~$
```

17 client pkts, 19 server pkts, 25 turns.

Entire conversation (703 bytes) Show and save data as ASCII Stream 1

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

## Task 3.2: Bypassing Egress Firewall

On se connecte a l'host B-192.168.20.99 et on execute cette commande cette fois

```
ssh -w 0:0 root@10.8.0.99 \
-o "PermitLocalCommand=yes" \
-o "LocalCommand= ip addr add 192.168.53.99/24 dev tun0 && \
ip link set tun0 up" \
-o "RemoteCommand=ip addr add 192.168.53.88/24 dev tun0 && \
ip link set tun0 up"
```

Ce code établit un tunnel VPN entre le client (tun0: 192.168.53.99) et le serveur (tun0: 192.168.53.88) via SSH, configurant les interfaces TUN de chaque côté pour permettre une communication réseau chiffrée.

```

seed@VM: ~/.../Labsetup
12/24/24]seed@VM:~/.../Labsetup$ sudo docker exec -it B-192.168.20.99 /bin/bash
root@78e4cf6ed07b:/# ssh -w 0:0 root@10.8.0.99 \
-o "PermitLocalCommand=yes" \
-o "LocalCommand= ip addr add 192.168.53.99/24 dev tun0 && \
ip link set tun0 up" \
-o "RemoteCommand=ip addr add 192.168.53.88/24 dev tun0 && \
ip link set tun0 up"
root@10.8.0.99's password:

root@2ac0b233dea2:/# exit
exit
[12/24/24]seed@VM:~/.../Labsetup$ sudo docker exec -it B-192.168.20.6 /bin/bash
root@2ac0b233dea2:/# telnet 10.8.0.99
Trying 10.8.0.99...
Connected to 10.8.0.99.
Escape character is '^].
Ubuntu 20.04.1 LTS
49be7a9ecdd8 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: Tue Dec 24 20:28:46 UTC 2024 from 192.16
8.20.6 on pts/2
seed@49be7a9ecdd8:~$ █

```

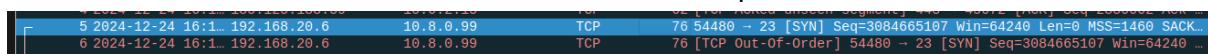
On peut voir que l'on peut pinger les sites bloqués

```

seed@49be7a9ecdd8:~$ ping example.com
PING example.com (93.184.215.14) 56(84) bytes of data.
64 bytes from 93.184.215.14 (93.184.215.14): icmp_seq=1 ttl=254 time=286 ms
^C
--- example.com ping statistics ---
2 packets transmitted, 1 received, 50% packet loss, time 1002ms
rtt min/avg/max/mdev = 286.398/286.398/286.398/0.000 ms
seed@49be7a9ecdd8:~$ ping twitter.com
PING twitter.com (104.244.42.193) 56(84) bytes of data.
64 bytes from 104.244.42.193: icmp_seq=1 ttl=254 time=79.5 ms
^C
--- twitter.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 79.515/79.515/79.515/0.000 ms
seed@49be7a9ecdd8:~$ ping facebook.com
PING facebook.com (185.60.219.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-01-cdg4.facebook.com (185.60.219.35): icmp_seq=1 ttl=254 time=46.9 ms
^C
--- facebook.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 46.945/46.945/46.945/0.000 ms
seed@49be7a9ecdd8:~$ █

```

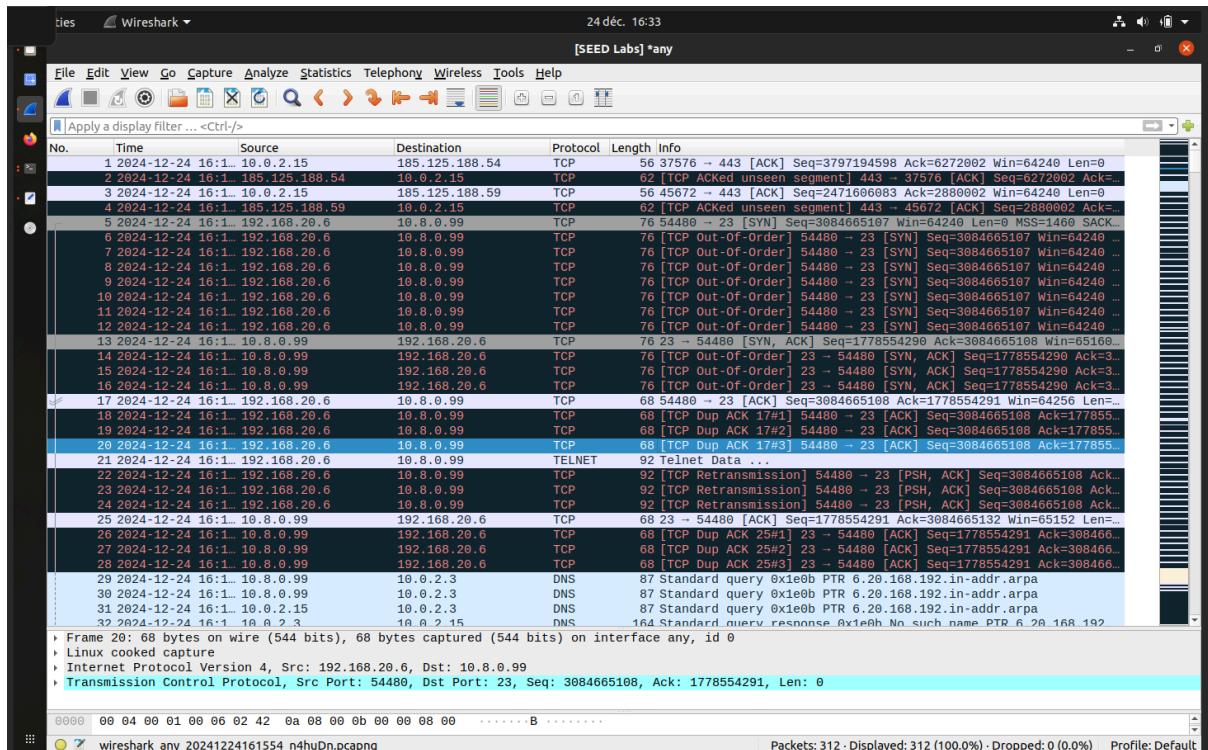
et voici sur wireshark la connection entre l'host B2 et A apres le telnet



et comme on peut voir on peut curl example.com

Les paquets ne sont pas bloqués par le pare-feu car ils sont encapsulés dans le tunnel VPN. Le NAT configuré sur le serveur VPN (Machine A) remplace l'adresse source par celle de l'interface eth0 du serveur VPN (10.8.0.99). Ainsi, les paquets sortants semblent provenir directement du serveur VPN, contournant les restrictions de pare-feu.

commentaire de notre capture wireshark



Dans Wireshark, on observe que les paquets émis depuis le client (192.168.20.6) vers le serveur VPN (10.8.0.99) passent par le tunnel VPN. Grâce à la règle NAT (**MASQUERADE**) sur le serveur VPN, les paquets sortants ont leur adresse source remplacée par celle du serveur VPN (10.8.0.99), ce qui leur permet de contourner les restrictions du pare-feu. Les réponses reviennent via le serveur VPN et sont réencapsulées pour être transmises au client, confirmant le fonctionnement du tunnel VPN et du NAT.

## Task 4: Comparing SOCKS5 Proxy and VPN

Le **Proxy SOCKS5** et le **VPN** sont des technologies utilisées pour créer des tunnels afin de contourner les pare-feux et protéger les communications. Voici une comparaison basée sur l'expérience de ce laboratoire :

### 1. Proxy SOCKS5

#### Description :

- Un protocole de proxy léger qui gère le trafic réseau au niveau de l'application.
- Redirige uniquement le trafic des applications configurées via un serveur proxy.

#### Avantages :

- **Efficacité** : Plus rapide, car il fonctionne au niveau de l'application et n'ajoute pas de surcharge liée au chiffrement par défaut.

- **Flexibilité** : Prend en charge différents types de trafic (TCP/UDP) et peut utiliser une authentification.
- **Simplicité de configuration** : Nécessite seulement une commande simple (par exemple, `ssh -D`) pour établir un proxy dynamique.

#### Inconvénients :

- **Portée limitée** : Seul le trafic des applications configurées pour utiliser le proxy est acheminé via le tunnel.
  - **Sécurité plus faible** : Le trafic n'est pas chiffré par défaut, sauf si des mesures supplémentaires (comme SSH) sont utilisées.
  - **Pas de tunnel au niveau réseau** : Ne permet pas de rediriger tout le trafic d'un appareil.
- 

## 2. VPN (Réseau Privé Virtuel)

#### Description :

- Une solution robuste qui crée un tunnel sécurisé et chiffré entre deux réseaux ou appareils.
- Fonctionne au niveau réseau et achemine tout le trafic via le tunnel VPN.

#### Avantages :

- **Protection complète** : Chiffre tout le trafic réseau, offrant une meilleure sécurité.
- **Portée globale** : Tout le trafic de l'appareil passe par le tunnel, indépendamment des applications utilisées.
- **Accès réseau étendu** : Permet une intégration transparente entre deux réseaux, simulant une connexion locale.

#### Inconvénients :

- **Plus lent** : Le chiffrement et la gestion au niveau réseau ajoutent une surcharge.
- **Configuration complexe** : Nécessite des priviléges root et une configuration réseau supplémentaire.
- **Consommation de ressources** : Utilise plus de ressources système que SOCKS5.