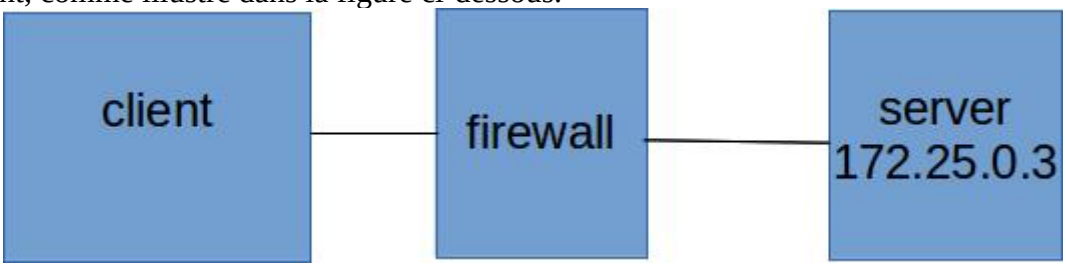
Lab Iptables

Présentation
Cet exercice Labtainer illustre l'utilisation d'iptables sur un pare-feu pour limiter l'accès réseau à un serveur à partir d'un client, comme illustré dans la figure ci-dessous.



Lorsqu'il est correctement configuré, le pare-feu n'autorisera que le trafic sélectionné du client vers le serveur.

La limitation des types de trafic réseau envoyés à un serveur peut contribuer à protéger le serveur contre les accès non autorisés.
Par exemple, si le serveur contient un service non sécurisé disponible par le biais de son interface réseau, l'exploitation de ce service est plus difficile si quelque chose bloque le trafic destiné à ce service.
Il existe une variété de techniques et de produits différents dans le but de limiter le trafic du réseau IP entre des ordinateurs.
Dans cette activité, vous limiterez le trafic IP grâce à l'utilisation d'iptables sous Linux.

Prérequis
L'étudiant est censé avoir appris séparément l'utilisation d'iptables pour bloquer sélectivement le trafic réseau. Le composant pare-feu inclut un exemple de script de configuration de pare-feu auquel vous pouvez vous référer.
La page de manuel d'iptables peut être consultée sur le terminal du pare-feu en utilisant :
man iptables
man iptables-extensions

Les étudiants doivent avoir une connaissance de base de la ligne de commande Linux ainsi que la capacité d'éditer des fichiers et d'exécuter des scripts shell simples. Une certaine expérience avec Wireshark est également présumée, par exemple à travers l'étude du laboratoire d'introduction à wireshark ***wireshark-intro***.

Démarrer le laboratoire
Le laboratoire est lancé à partir du répertoire de travail labtainer sur votre hôte sur votre hôte ou votre machine virtuelle Linux. Exécutez la commande :

```
labtainer iptables2
```

Les terminaux virtuels résultants comprennent : un terminal (**shell bash**) connecté à un ordinateur **client** "MyComputer" et un terminal (shell bash) connecté à un Firewall.

Client <===> [Firewall]<===> serveur (nom server, adresse IP 172.25.0.3)

Tâches

1 Exploration

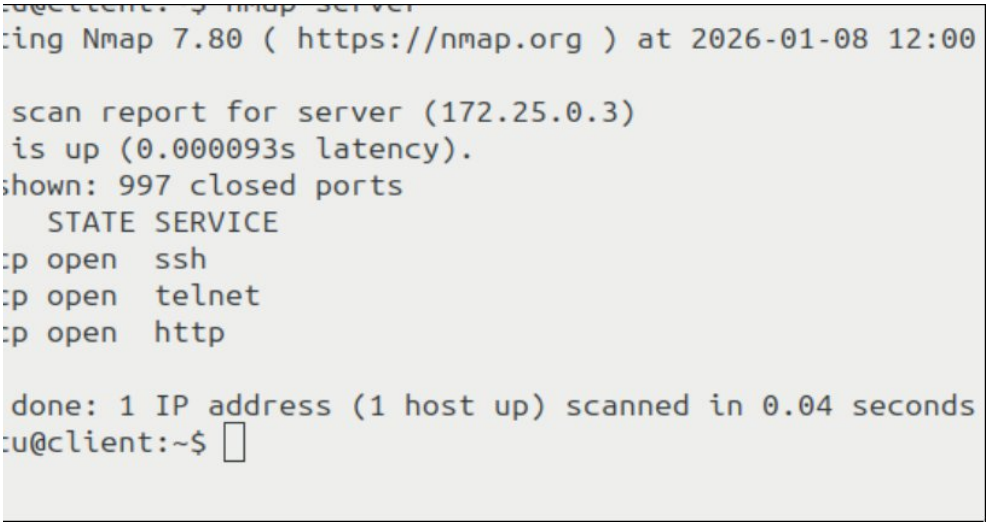
L'utilitaire Wireshark est installé sur le pare-feu. Utilisez-le pour afficher le trafic réseau à travers le pare-feu et pour déboguez vos règles de pare-feu. Démarrez-le à partir du terminal du pare-feu :

```
wireshark &
```

Sélectionnez ensuite l'interface **eth0**.

Sur le terminal client, utilisez l'utilitaire nmap pour répertorier (certains des) ports ouverts sur le serveur :

```
nmap server ou nmap 172.25.0.3
```

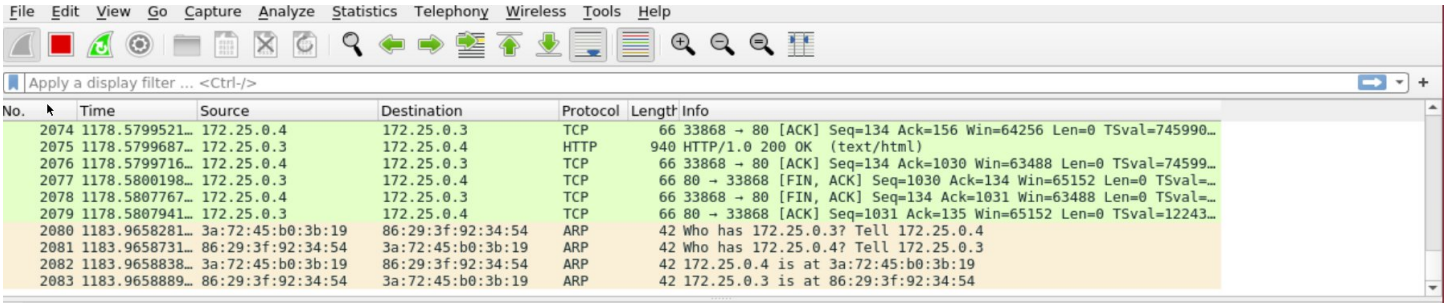


Résultat du scan nmap depuis le client vers le serveur (172.25.0.3) montrant les ports SSH (22), Telnet (23) et HTTP (80) ouverts.

Utilisez `wget` pour confirmer que le serveur répond aux requêtes **HTTP** , utilisez **ctrl C** pour quitter une fois que vous obtenez une réponse du serveur.

```
wget server &
```

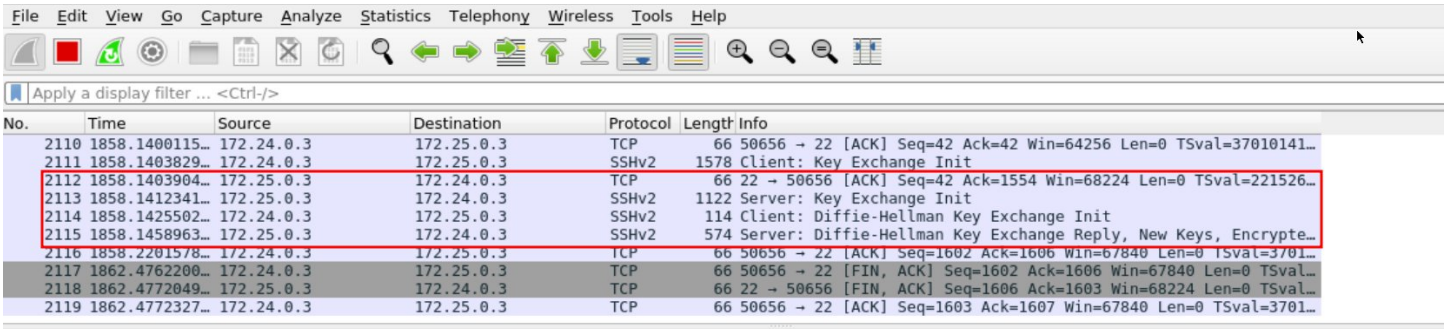
Vérifiez qu'un service ssh est proposé sur le serveur - vous n'avez pas besoin de vous connecter lorsque vous y êtes invité, répondez « no » ou utilisez ctrl C pour quitter une fois que vous obtenez une réponse du serveur.



Capture Wireshark montrant l'échange de paquets HTTP entre le client et le serveur après l'envoi d'une requête avec wget, confirmant que le service web répond correctement.

`ssh server`

Vérifiez qu'un service telnet est proposé sur le serveur - vous n'avez pas besoin de vous connecter lorsque vous y êtes invité, utilisez également ctrl C pour quitter une fois que vous obtenez une réponse du serveur.

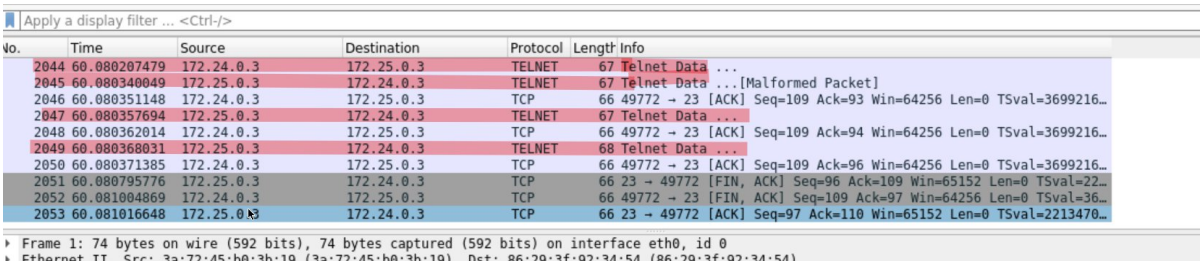


Capture Wireshark montrant une tentative de connexion SSH depuis le poste client vers le serveur. Les échanges de paquets confirment que le service SSH est bien accessible sur le serveur.

`telnet server`

Observez le trafic dans wireshark, notez les adresses IP sources et les ports de destination utilisés par le client lors de la connexion au serveur.

IP source : 172.24.0.3 – IP destination : 172.25.0.3 – Port de destination : 23 (Telnet)



Capture Wireshark montrant une tentative de connexion Telnet depuis le poste client vers le serveur. Les paquets échangés indiquent que la connexion utilise le port 23, ce qui confirme que le service Telnet est actif sur le serveur.

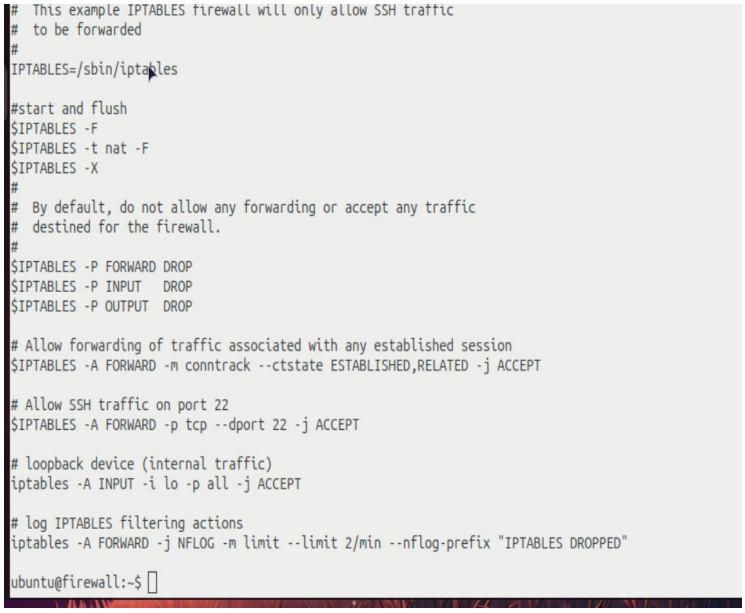
2. Utiliser iptables pour limiter le trafic

L'utilitaire iptables est installé sur le composant « firewall ». Utilisez-le pour empêcher le pare-feu de transférer tout trafic vers le serveur autre que **SSH** et **HTTP**.

Vous pouvez vous référer à l'exemple de script de pare-feu qui se trouve sur le composant de pare-feu dans le répertoire *home* et l'expérimenter.

Consultez le contenu du script exemple **`example_fw.sh`** pour comprendre ce qu'il fait.

Notez que la dernière ligne du script **`example fw.sh`** demande à iptables d'enregistrer les paquets bloqués dans les logs d'iptables **`/var/log/iptables.log`**.



Extrait du script de pare-feu example_fw.sh montrant la configuration d'iptables. Le script autorise uniquement le trafic SSH et enregistre les paquets bloqués dans le fichier /var/log/iptables.log, ce qui permet de suivre les tentatives de connexion refusées.

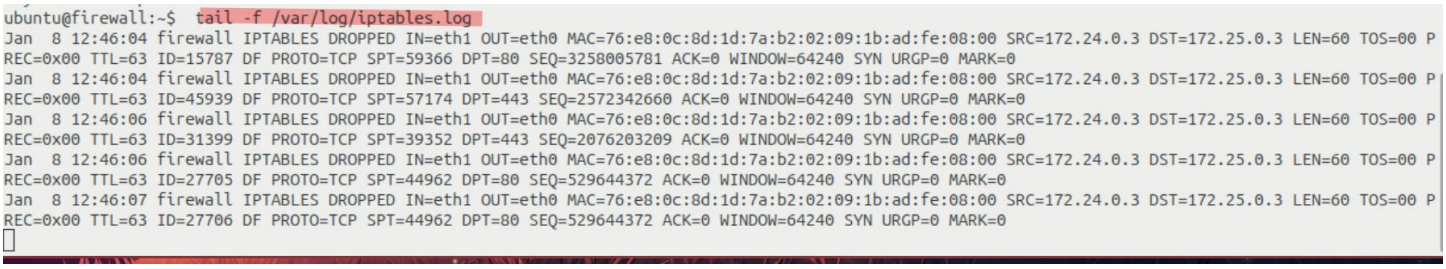
Pour exécuter le script **`example_fw.sh`**, utilisez :

`sudo ./example_fw.sh`

Vous pouvez visualiser les logs enregistrés à partir de l'un des onglets du terminal du pare-feu via :

`tail -f /var/log/iptables.log`

Après avoir modifié votre configuration iptables, utilisez les applications sur le client pour démontrer que le pare-feu n'autorise que le trafic souhaité.
Surveillez le trafic dans wireshark pour voir que la négociation TCP échoue lorsque vous essayez de vous connecter aux ports filtrés.



Extrait du fichier /var/log/iptables.log montrant des paquets bloqués par le pare-feu.

Utilisez nmap pour confirmer la bonne configuration :

```
nmap server
```

NB : Pensez à mettre vos commandes iptables dans un script, il est ainsi plus facile de tester et de reconfigurer iptables si vous redémarrez le laboratoire.

```
an  8 12:58:31 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60046 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:58:32 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60047 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:58:33 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60048 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:58:34 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60049 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:58:35 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60050 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:59:07 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60055 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
an  8 12:59:40 firewall IPTABLES DROPPED IN=eth1 OUT=eth0 MAC=76:e8:0c:8d:1d:7a:b2:02:09:1b:ad:fe:08:00 SRC=172.24.0.3 DST=172.25.0.3 LEN=60 TOS=10
EC=0x00 TTL=63 ID=60056 DF PROTO=TCP SPT=53648 DPT=23 SEQ=1823394965 ACK=0 WINDOW=64240 SYN URGP=0 MARK=0
```

Blocage du port 23= telnet

3. Ouvrir un nouveau service sur un port

L'ordinateur client comprend un programme **wizbang** que vous devez maintenant autoriser à envoyer du trafic vers le serveur.

Wizbang filtrer par le pare-feu

```
    9 6.626173182   172.25.0.3      172.24.0.3      TCP      54 443 → 50840 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   10 6.626187967   172.24.0.3      172.25.0.3      TCP      66 59654 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=42166…
```

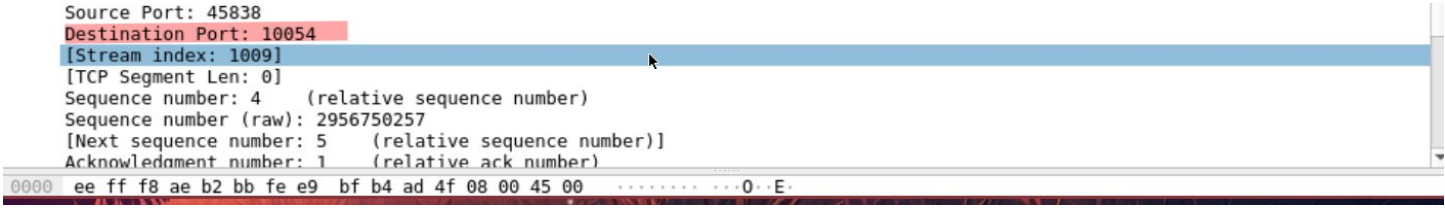
```
ubuntu@client:~$ nmap -Pn server
Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-08 16:43
 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.00028s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 6.68 seconds
ubuntu@client:~$ ./wizbang go
ERROR: [Errno 110] Connection timed out
ubuntu@client:~$
```

Pn = pour qui ping le serveur

Wizbang qui ne répond pas

Exécutez le programme à partir du client et observez le port qu'il tente d'utiliser dans wireshark :

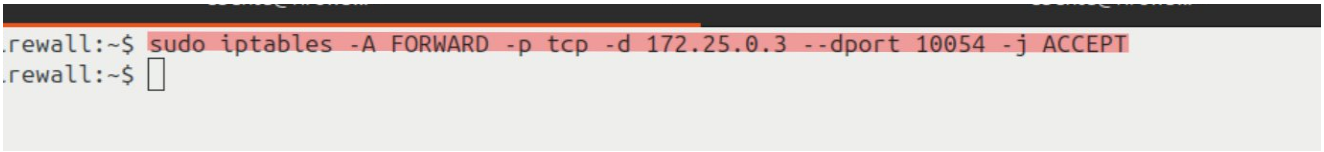
```
./wizbang
```



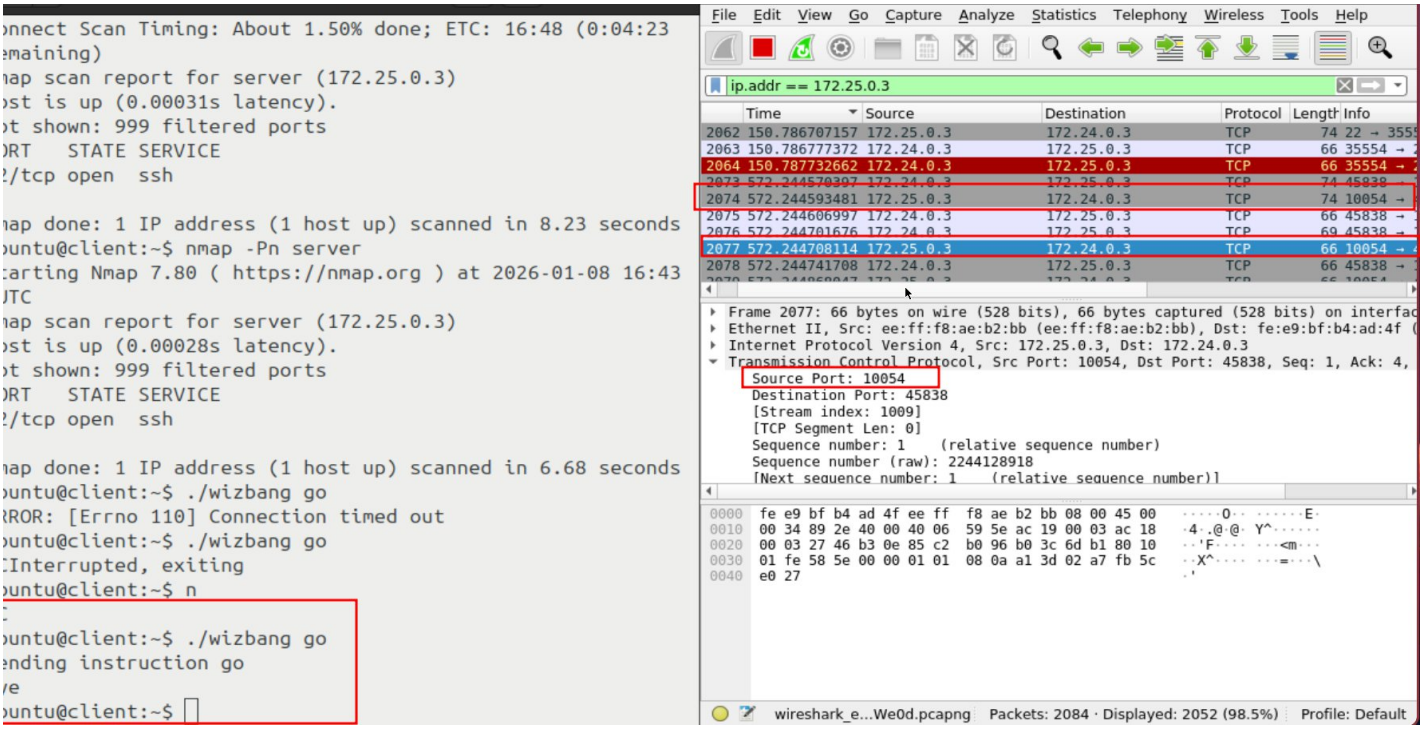
wizbang utilise le port 10054 pour communiquer avec le serveur.

Ensuite, modifiez votre iptables pour autoriser ce service. Après avoir ajusté vos règles iptables, confirmez que vous pouvez exécuter le programme wizbang avec succès.
Utilisez à nouveau nmap pour confirmer la bonne configuration

```
nmap server
```



Règle iptables ajoutée pour autoriser le port 10054 vers le serveur.



Exécution de wizbang réussie après l'ouverture du port par iptables, avec trafic visible dans Wireshark.

Arrêter le labtainer

Lorsque le laboratoire est terminé, ou si vous souhaitez arrêter de travailler pendant un certain temps, dans le terminal qui vous a permis de le lancer, exécutez : stoplab

Vous pouvez toujours redémarrer le Labtainer et continuer votre travail.
Lorsque le Labtainer est arrêté, un fichier zip est créé et copié dans un emplacement affiché par la commande « **stoplab** ».
Une fois le laboratoire terminé, vous pouvez envoyer ce fichier zip au formateur pour correction éventuelle.