

Administration et sécurité d'un réseau local

Document technique du projet final

**UNIVERSITÉ AMADOU MAKHTAR MBOW
ECOLE SUPÉRIEURE POLYTECH DIAMNIADIO
DEPARTEMENT DES SCIENCES TECHNIQUE DE L'INGÉNIEURE**



**Conception et Implémentation d'une Infrastructure Sécurisée pour un
Réseau Local Utilisant Iptables, DMZ, Proxy, et VPN**

PRÉSENTÉ PAR :

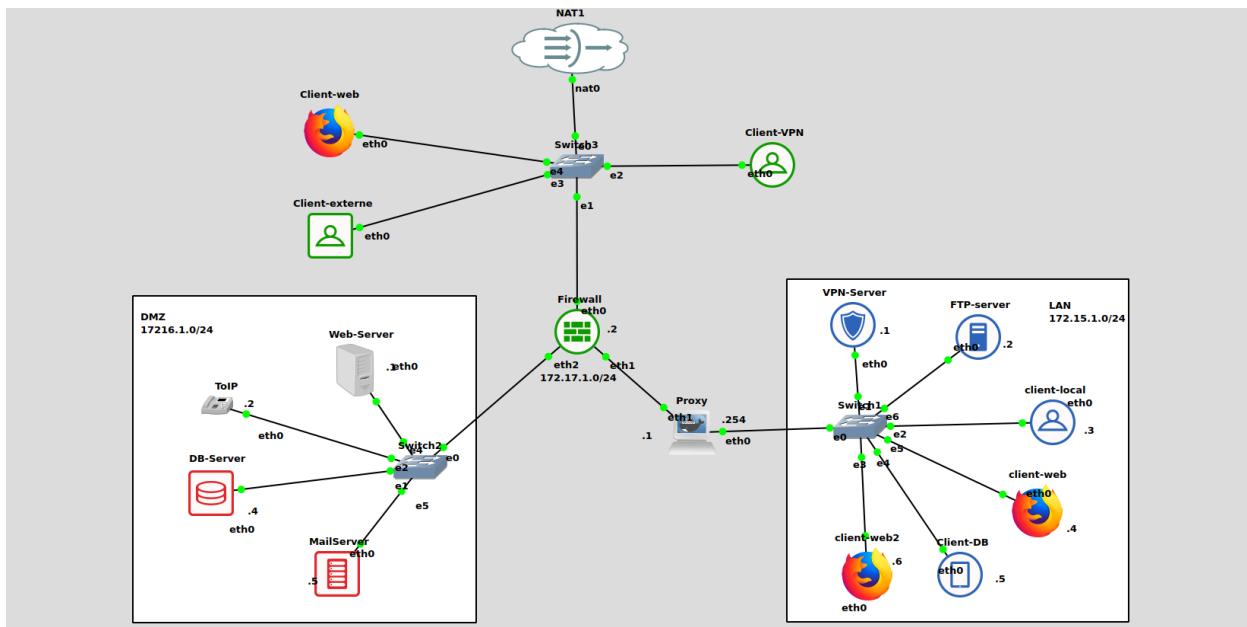
Sidy NDIAYE

Limamoulaye DIA

I. Topologie.....	3
II. Configuration du Pare-Feu et du serveur Proxy.....	3
a. Configuration du Pare-feu.....	3
b. Configuration du serveur Proxy.....	5
1. Test.....	8
III. Configuration LAN.....	10
a. Mise en place Serveur VPN (openvpn).....	10
1. Configuration sur le pare-feu.....	11
2. Configuration du serveur openvpn.....	11
3. Configuration du client openvpn.....	11
4. Test.....	13
b. Configuration des clients.....	15
c. Mise en place du serveur FTP.....	17
IV. Configuration DMZ.....	18
a. Mise en place du serveur-web Apache.....	18
1. Mise en place.....	18
2. Configuration sur le pare-feu.....	19
3. Test.....	19
b. Mise en place du serveur VoIP.....	22
1. Mise en place.....	22
2. Configuration sur le Pare-feu.....	27
3. Test.....	27
b. Mise en place du serveur Base de données.....	29
1. Mise en place.....	29
2. Configuration sur le Pare-feu.....	30
3. Test.....	31
c. Mise en place du serveur Messagerie.....	32
1. Mise en place.....	32
2. Configuration sur le Pare-feu.....	36
3. Test.....	37

I. Topologie

Voici la topologie adoptée pour le projet:



II. Configuration du Pare-Feu et du serveur Proxy

a. Configuration du Pare-feu

Voici l'**edit config** du pare-feu:

```

# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
#auto eth0
 iface eth0 inet static
 #      address 192.168.0.2
 #      netmask 255.255.255.0
 #      gateway 192.168.0.1
 #      up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
auto eth0
iface eth0 inet dhcp
#      hostname debian-1

# Static config for eth1
auto eth1
iface eth1 inet static
#      address 172.17.1.2
#      netmask 255.255.255.0
#      gateway 192.168.1.1
#      up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth1
#auto eth1
#iface eth1 inet dhcp
#      hostname debian-1

# Static config for eth2
auto eth2
iface eth2 inet static
#      address 172.16.1.254
#      netmask 255.255.255.0
#      gateway 192.168.2.1
#      up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth2
#auto eth2
#iface eth2 inet dhcp

```

Refresh Cancel Save

Dans le terminal on a commencé par mettre à jour en tapant la commande **apt update** puis on a installer nano et iptables pour mettre en place des règles de nattage et de filtrage selon le principe de la DMZ.

- `iptables -A FORWARD -s 172.15.1.0/24 -d 172.16.1.0/24 -j ACCEPT`

permet d'ajouter une règle au pare-feu qui permet aux paquets provenant du réseau 172.15.1.0/24 (LAN) d'être transférés vers le réseau 172.16.1.0/24 (DMZ).

- `iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT`

permet de s'assurer que les paquets appartenant à des connexions déjà établies ou liés à des connexions existantes sont autorisés à être transférés par le pare-feu.

- `iptables -A FORWARD -s 172.16.1.0/24 -d 172.15.1.0/24 -j DROP`

permet d'ajouter une règle au pare-feu qui bloque les paquets provenant du réseau DMZ et destinés au réseau LAN.

-
- `iptables -A FORWARD -i eth0 -d 172.15.1.0/24 -j DROP`

permet d'ajouter une règle au pare-feu qui bloque les paquets entrants sur l'interface eth0 s'ils sont destinés au réseau LAN.

- `iptables -A FORWARD -s 172.16.1.0/24 -j ACCEPT`
`iptables -A FORWARD -d 172.16.1.0/24 -j ACCEPT`

permet d'autoriser le trafic entrant et sortant de la DMZ.

On applique des règles pour permettre l'accès à internet aux réseaux

```
iptables -t nat -A POSTROUTING -s 172.15.1.0/24 -o eth0 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -o eth0 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -s 172.17.1.0/24 -o eth0 -j MASQUERADE
```

Installer netfilter-persistent et iptables-persistent puis sauvegarder les règles via la commande `iptables-save > /etc/iptables/rules.v4`

On a fait un routage pour autoriser le réseau LAN à sortir en passant par la passerelle du proxy: `ip route add 172.15.1.0/24 via 172.17.1.1`

On installe procps puis on modifie le fichier fichier sysctl.conf et décommenter la ligne `net.ipv4.forward=1` pour permettre le forwarding des paquets ipv4: `nano /etc/sysctl.conf`.

b. Configuration du serveur Proxy

Voici l'edit config de notre serveur Proxy:

```
Proxy-Server interfaces

# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
auto eth0
iface eth0 inet static
    address 172.15.1.254
    netmask 255.255.255.0
    gateway 172.17.1.1
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#    hostname debian-7

# Static config for eth1
auto eth1
iface eth1 inet static
    address 172.17.1.1
    netmask 255.255.255.0
    gateway 172.17.1.2
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth1
#auto eth1
```

On a choisi de configurer notre serveur avec Squid. Pour ce faire, on a mis les paquets disponibles à jour puis on installe les paquets squid via la commande `apt install squid`

Ensute on navigue vers le fichier de configuration squid et on le modifie:

```
nano /etc/squid/squid.conf
```

```
GNU nano 7.2                                     squid.conf
visible_hostname Limamou
http_port 172.15.1.254:3128

cache_dir ufs /var/spool/squid 100 16 256

acl lan src 172.15.1.0/24
acl safe_ports port 21
acl safe_ports port 80
acl safe_ports port 443

http_access deny !lan
http_port 3128

acl domaine url_regex -i "etc/squid/listenoire.txt"
http_access deny domaine
```

`http_port 172.15.1.254:3128`

Indique l'adresse IP et le port sur lesquels Squid écoute pour les requêtes HTTP. Ici, il écoute sur l'adresse IP **172.15.1.254** et le port 3128.

```
cache_dir ufs /var/spool/squid 100 16 256
```

Définit le répertoire de cache de Squid. Le type de cache est ufs, le répertoire est /var/spool/squid, la taille maximale du cache est de 100 Mo, et il est organisé en 16 répertoires de premier niveau avec 256 sous-répertoires chacun.

```
cl domaine url_regex -i "/etc/squid/listenoire.txt"
```

Définit une ACL domaine pour filtrer les URL correspondant aux expressions régulières listées dans le fichier /etc/squid/listenoire.txt.

```
http_access deny domaine
```

Refuse l'accès HTTP pour toutes les requêtes dont l'URL correspond à l'une des expressions régulières dans domaine.

```
acl lan src 172.15.1.0/24
```

Définit une liste de contrôle d'accès (ACL) nommée lan pour les sources d'adresse IP dans le réseau 172.15.1.0/24.

```
cl safe_ports port 21
```

Définit une ACL safe_ports pour le port 21 (FTP).

```
acl safe_ports port 80
```

Définit une ACL safe_ports pour le port 80 (HTTP).

```
acl safe_ports port 443
```

Définit une ACL safe_ports pour le port 443 (HTTPS).

```
http_access deny !lan
```

Refuse l'accès HTTP à toutes les requêtes qui ne proviennent pas du réseau **lan**.

```
http_port 3128
```

Indique un autre port HTTP pour Squid, ici le port 3128 (peut être redondant avec la ligne 2).

Nous allons optionnellement, depuis notre proxy, interdire les visites vers certains sites. On navigue dans /etc/squid/ puis on crée un fichier du nom de `listenoire.txt`

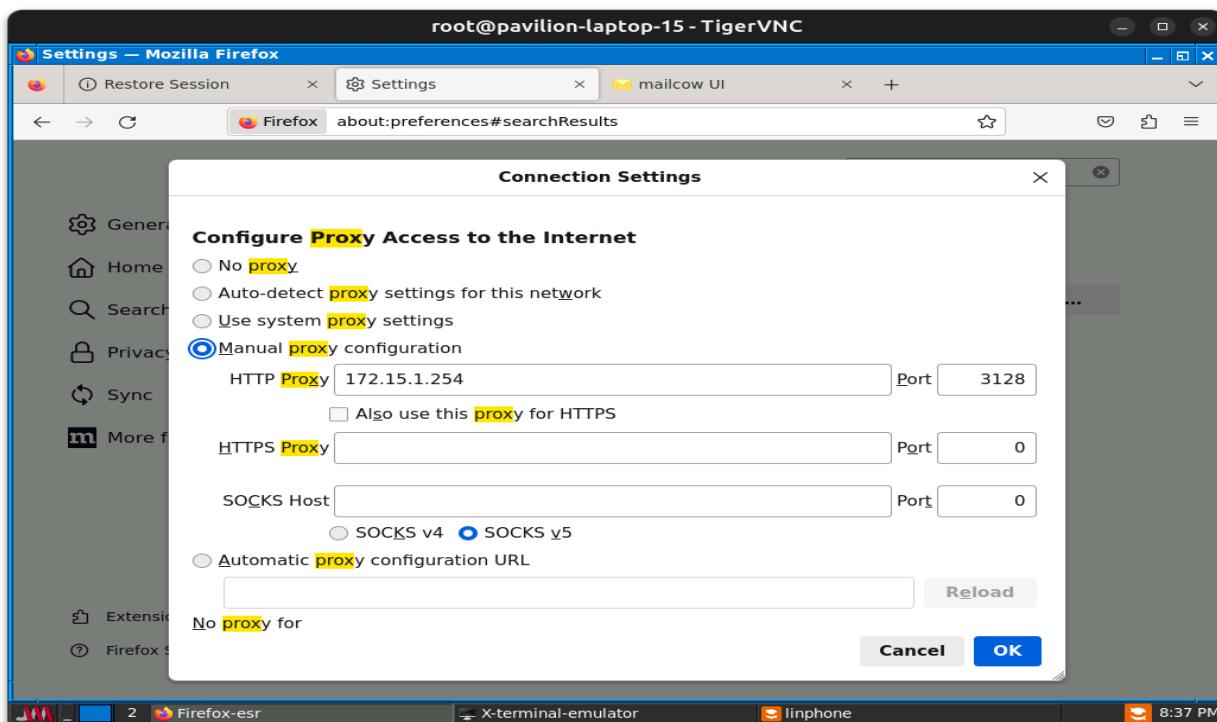
```
nano /etc/squid/listenoire.txt
```

On y définit les URL à filtrer

```
GNU nano 7.2          /etc/squid/listenoire.txt
jam.sn
www.youtube.com
```

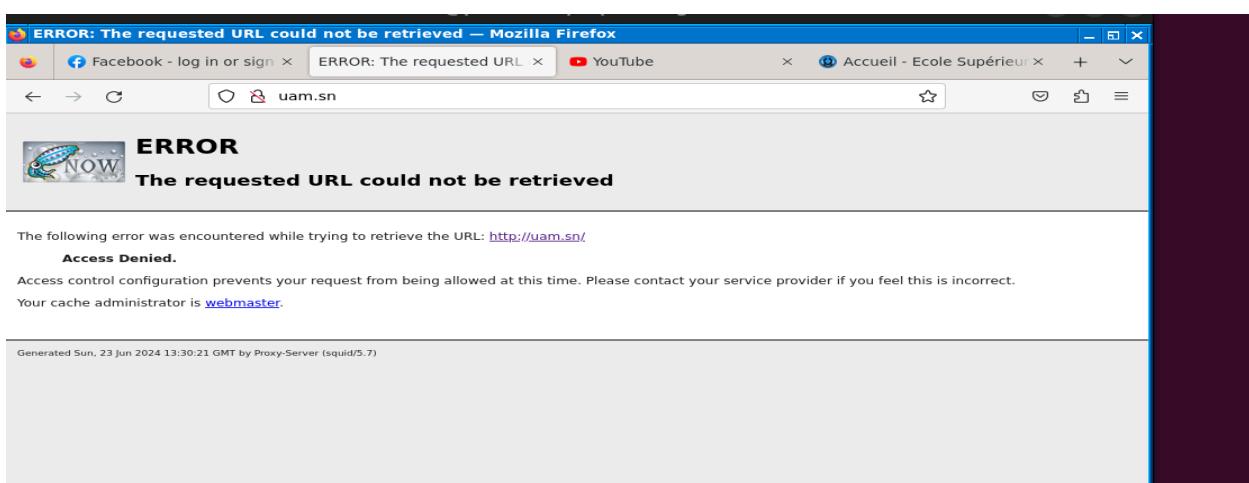
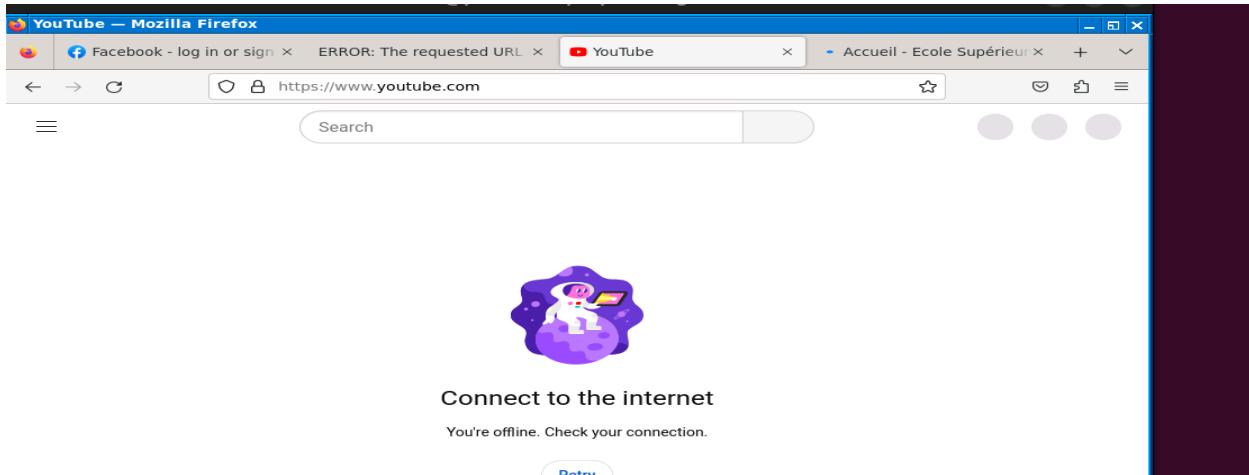
1. Test

On configure la connexion par proxy sur chaque webterm



Ensuite on redémarre les services Squid via la commande `/etc/init.d/squid restart`

On teste la connexion pour les URL filtré depuis un webterm du LAN



On voit bien que le filtrage marche comme sur des roulettes

III. Configuration LAN

Voici l'edit config du serveur vpn:



```
VPN-Server interfaces

# This is a sample network config, please uncomment lines to configure the network
#
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
auto eth0
iface eth0 inet static
    address 172.15.1.1
    netmask 255.255.255.0
    gateway 172.15.1.254
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#    hostname ubuntu-1

# Static config for eth1
#auto eth1
#iface eth1 inet static
```

a. Mise en place Serveur VPN (openvpn)

Pour ce faire, nous mettons d'abord à jour le noyau ainsi que les composants du serveur graphique pour différentes versions d'ubuntu.

Voici les commandes:

```
apt install --install-recommends linux-generic-hwe-24.04
xserver-xorg-hwe-18.04

apt install linux-headers-$(uname -r)
```

On exécute la commande suivante pour générer l'adresse IP du serveur

```
apt install resolvconf
```

Par la suite on clone le repository github de OpenVPN à travers la commande:

```
git clone https://github.com/pivpn/pivpn.git
```

Ensuite on installe l'exécutable openvpn

```
./pivpn/auto_install/install.sh
```

Ensuite on a un menu sur lequel on choisit Pivpn au lieu de Wireguard ainsi que le port 1194 pour openvpn et le DNS public du nom de dsti.sn

Ensuite on crée un utilisateur du nom de naruto via la commande

```
pivpn -a
```

Le fichier généré se trouve sur le répertoire /etc/openvpn/easy-rsa/pki. On modifie le DNS donné lors de l'installation du serveur (dsti.sn) par l'adresse ip de notre serveur.

```
nano /etc/openvpn/easy-rsa/pki/naruto.ovpn
```

```
proto udp
remote 192.168.122.98 1194
resolv-retry infinite
nobind
remote-cert-tls server
tls-version-min 1.2
verify-x509-name VPN-Server_2f1850b0-888f-41f1-b049-2f4d93c214da name
```

1. Configuration sur le pare-feu

- Autoriser tous les paquets UDP destinés à l'adresse IP 172.15.1.1 sur le port 1194

```
iptables -A FORWARD -d 172.15.1.1 -p udp --dport 1194 -j ACCEPT
```

- redirige tous les paquets UDP entrant sur l'interface eth0 et destinés au port 1194 vers l'adresse IP 172.15.1.1

```
iptables -t nat -A PREROUTING -i eth0 -p udp --dport 1194 -j DNAT
--to-destination 172.15.1.1
```

2. Configuration du serveur openvpn

Effectuer la translation d'adresses source (SNAT) pour tous les paquets provenant du tunnel VPN 10.137.57.0/24 et sortant par l'interface eth0

```
iptables -t nat -A POSTROUTING -s 10.137.57.1/24 -o eth0 -j
MASQUERADE
```

3. Configuration du client openvpn

Voici l'edit config du client vpn:

```

Client-VPN interfaces

# This is a sample network config, please uncomment lines to configure the network
#
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*
#
# Static config for eth0
#auto eth0
#iface eth0 inet static
#       address 192.168.0.2
#       netmask 255.255.255.0
#       gateway 192.168.0.1
#       up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
auto eth0
iface eth0 inet dhcp
#       hostname ubuntu-1

# Static config for eth1
#auto eth1
#iface eth1 inet static
#       address 192.168.1.2

```

On installe sur le client vpn openssh-server pour permettre l'envoie du fichier naruto.ovpn par scp. Ensuite on ajoute un utilisateur via la commande adduser.

```

apt install openssh-server
adduser gaara

```

L'utilisateur créé est gaara. On se connecte par ssh via la commande ssh gaara@ip_address_client.

Ensuite on effectue l'envoie par scp sur le serveur openvpn via la commande:

```
scp naruto.ovpn gaara@192.168.122.1:/home/gaara
```

```

[gaara@192.168.122.1 ~] $ scp naruto.ovpn gaara@192.168.122.1:/home/gaara
gaara@192.168.122.1's password:
naruto.ovpn                                              100% 2680      3.6MB/s   00:00
[gaara@192.168.122.1 ~] $

```

On installe screen qui est un utilitaire permettant d'exécuter des commandes en arrière plan puis on se connecte sur le serveur vpn via la commande:

```
openvpn -config /home/gaara/naruto.ovpn
```

```

28 juin 23:47
VPN-Client
VPN-Client
VPN-Server

2024-06-28 23:47:40 VERIFY OK: depth=1, CN=Easy-RSA CA
2024-06-28 23:47:40 VERIFY KU OK
2024-06-28 23:47:40 Validating certificate extended key usage
2024-06-28 23:47:40 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2024-06-28 23:47:40 VERIFY EKU OK
2024-06-28 23:47:40 VERIFY X509NAME OK: CN=VPN-Server_0fcfd4608-88f4-42db-90de-534311cb0c1
2024-06-28 23:47:40 VERIFY OK: depth=0, CN=VPN-Server_0fcfd4608-88f4-42db-90de-534311cb0c1
2024-06-28 23:47:40 Control Channel: TLSv1.3, cipher TLSv1.3_AES_256_GCM_SHA384, peer certificate: 256 bit ECprime256v1, signature: ecdsa-with-SHA256
2024-06-28 23:47:40 [VPN-Server_0fcfd4608-88f4-42db-90de-534311cb0c1] Peer Connection Initiated with [AF_INET]192.168.122.218:1194
2024-06-28 23:47:40 TLS: move_session: dest_TM_ACTIVE src_TM_INITIAL reincit_src=1
2024-06-28 23:47:40 TLS: tls_multi_process: initial untrusted session promoted to trusted
2024-06-28 23:47:40 PUSH: Received control message: "PUSH_REPLY,dhcp-option DNS 10.17.65.1,block-outside-dns,redirect-gateway def1,route-gateway 10.17.65.1,to-ping subnet,ping 15,ping-restart 120,ifconfig 10.17.65.2 255.255.255.0,peer-id 0,cipher AES-256-GCM,protocol-flags cc-exit tls-ekm dyn-tls-crypt,tun-ntu 150 0"
2024-06-28 23:47:40 Options error: Unrecognized option or missing or extra parameter(s) in [PUSH-OPTIONS]:2: block-outside-dns (2.6.3)
2024-06-28 23:47:40 OPTIONS IMPORT: --ifconfig/up options modified
2024-06-28 23:47:40 OPTIONS IMPORT: route options modified
2024-06-28 23:47:40 OPTIONS IMPORT: route-related options modified
2024-06-28 23:47:40 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
2024-06-28 23:47:40 OPTIONS IMPORT: tun-mtu set to 1500
2024-06-28 23:47:40 net_route_v4_best_gw query: dst 0.0.0.0
2024-06-28 23:47:40 net_route_v4_best_gw result: via 192.168.122.1 dev eth0
2024-06-28 23:47:40 ROUTE_GATEWAY 192.168.122.1/255.255.255.0 IFACE=eth0 HWADDR=a6:d4:c4:5b:af:ee
2024-06-28 23:47:40 TUN/TAP device tun0 opened
2024-06-28 23:47:40 net_iface_mtu_set: mtu 1500 for tun0
2024-06-28 23:47:40 net_iface_up: set tun0 up
2024-06-28 23:47:40 net_addr_v4_add: 10.17.65.2/24 dev tun0
2024-06-28 23:47:40 net_route_v4_add: 192.168.122.218/32 via 192.168.122.1 dev eth0 table 0 metric -1
2024-06-28 23:47:40 net_route_v4_add: 0.0.0.0/1 via 10.17.65.1 dev [NULL] table 0 metric -1
2024-06-28 23:47:40 net_route_v4_add: 128.0.0.0/1 via 10.17.65.1 dev [NULL] table 0 metric -1
2024-06-28 23:47:40 Initialization Sequence Completed
2024-06-28 23:47:40 Data Channel: cipher 'AES-256-GCM', peer-id: 0
2024-06-28 23:47:40 Timers: ping 15, ping-restart 120
2024-06-28 23:47:40 Protocol options: protocol-flags cc-exit tls-ekm dyn-tls-crypt

```

4. Test

On remarque qu'une fois déconnecté, le client vpn ne parvient pas à avoir accès au LAN.

```

/ # ping 172.15.1.1
PING 172.15.1.1 (172.15.1.1): 56 data bytes
^C
--- 172.15.1.1 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss

```

Capture wireshark de la liaison entre le switch 3 et l'interface eth0:

Capture en cours de [Switch3 Ethernet2 to Client-VPN eth0]						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
2	1.983975	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
3	3.9686220	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
4	4.897827	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=0/9, ttl=64 (no response found!)	
5	5.897303	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=1/256, ttl=64 (no response found!)	
6	6.897341	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
7	6.897353	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=2/512, ttl=64 (no response found!)	
8	7.897600	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=3/768, ttl=64 (no response found!)	
9	8.897611	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
10	8.897628	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=4/1024, ttl=64 (no response found!)	
11	9.897676	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=5/1280, ttl=64 (no response found!)	
12	9.984098	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
13	10.176641	4a:bd:95:f6:39:06	52:54:00:58:71:6d	ARP	42 who has 192.168.122.1? Tell 192.168.122.179	
14	10.176352	52:54:00:58:71:6d	4a:bd:95:f6:39:06	ARP	42 192.168.122.1 is at 52:54:00:58:71:6d	
15	10.897045	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=6/1536, ttl=64 (no response found!)	
16	11.897975	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=7/1792, ttl=64 (no response found!)	
17	11.988147	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	
18	12.898200	192.168.122.179	172.15.1.1	ICMP	98 Echo (ping) request id=0x004e, seq=8/2048, ttl=64 (no response found!)	
19	14.016302	c6:7e:da:59:69:49	Spanning-tree-(for-> STP	52 Conf.	Root = 32768/0/52:54:00:58:71:6d Cost = 0 Port = 0x8001	

On voit bien que l'accès est refusé au client nomade non connecté.

Mais après la connexion, le client est autorisé à être un client du LAN parvient à faire tout ce que les clients LAN peuvent faire parce qu'on a fait un routage sur le serveur vpn permettant au client

```
29 juin 16:55
VPN-Client
Firewall
VPN-Client
VPN-Server

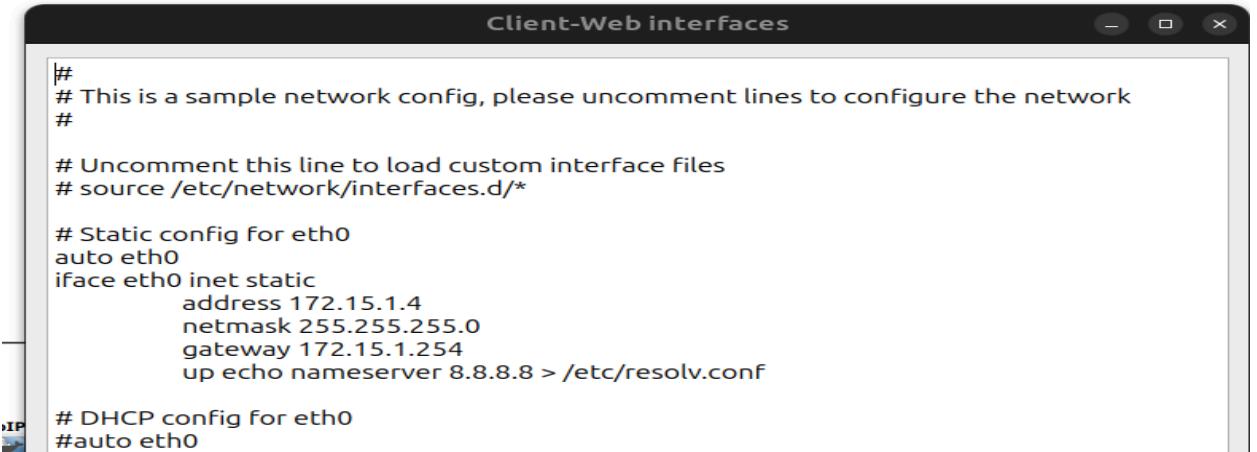
/home/gaara # ping 172.15.1.3
PING 172.15.1.3 (172.15.1.3): 56 data bytes
^C
--- 172.15.1.3 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
/home/gaara # ping 172.15.1.3
PING 172.15.1.3 (172.15.1.3): 56 data bytes
64 bytes from 172.15.1.3: seq=0 ttl=63 time=1.796 ms
64 bytes from 172.15.1.3: seq=1 ttl=63 time=3.080 ms
64 bytes from 172.15.1.3: seq=2 ttl=63 time=2.600 ms
64 bytes from 172.15.1.3: seq=3 ttl=63 time=2.849 ms
^C
--- 172.15.1.3 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.796/2.581/3.080 ms
/home/gaara # ping 172.15.1.2
PING 172.15.1.2 (172.15.1.2): 56 data bytes
64 bytes from 172.15.1.2: seq=0 ttl=63 time=2.093 ms
64 bytes from 172.15.1.2: seq=1 ttl=63 time=2.083 ms
^C
--- 172.15.1.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 2.083/2.088/2.093 ms
/home/gaara #
```

Capture en cours de [Switch1 Ethernet1 to VPN-Server eth0]						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2
2	0.000467	172.15.1.1	34.107.221.82	TCP	66	50460 -> 80 [FIN, ACK] Seq=1 Ack=1 Win=249 Len=0 TSval=3435458696 TSecr=3118657391
3	0.249603	34.107.221.82	172.15.1.1	TCP	66	88 - 50460 [FIN, ACK] Seq=1 Ack=2 Win=261 Len=0 TSval=3118666862 TSecr=3435458696
4	0.249979	172.15.1.1	192.168.122.237	OpenVPN	118	MessageType: P_DATA_V2
5	0.250928	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2
6	0.251118	172.15.1.1	34.107.221.82	TCP	66	50460 -> 80 [ACK] Seq=2 Ack=2 Win=249 Len=0 TSval=3435458948 TSecr=3118666862
7	1.000396	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2
8	1.000643	172.15.1.1	34.107.221.82	TCP	66	50478 -> 80 [FIN, ACK] Seq=1 Ack=1 Win=250 Len=0 TSval=3435459697 TSecr=2828608046
9	1.205773	34.107.221.82	172.15.1.1	TCP	66	88 - 50478 [FIN, ACK] Seq=1 Ack=2 Win=261 Len=0 TSval=2828618126 TSecr=3435459697
10	1.296022	172.15.1.1	192.168.122.237	OpenVPN	118	MessageType: P_DATA_V2
11	1.296990	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2
12	1.297189	172.15.1.1	34.107.221.82	TCP	66	50478 -> 80 [ACK] Seq=2 Ack=2 Win=250 Len=0 TSval=3435459904 TSecr=2828618126
13	5.251699	2a:12:02:27:f7:ae	3e:85:a8:23:5e:1b	ARP	42	Who has 172.15.1.1? Tell 172.15.1.254
14	5.251931	3e:85:a8:23:5e:1b	2a:12:02:27:f7:ae	ARP	42	172.15.1.1 is at 3e:85:a8:23:5e:1b

b. Configuration des clients

Le LAN contient des machines clients permettant de tester le bon fonctionnement des serveurs et la sécurité du réseau. Voici les clients configurés:

- Client-web1



```
# This is a sample network config, please uncomment lines to configure the network #
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*

# Static config for eth0
auto eth0
iface eth0 inet static
    address 172.15.1.4
    netmask 255.255.255.0
    gateway 172.15.1.254
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
```

- Client-web2

```
Clien-Web-2 interfaces
```

```
#  
# This is a sample network config, please uncomment lines to configure the network  
#  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 172.15.1.6  
    netmask 255.255.255.0  
    gateway 172.15.1.254  
    up echo nameserver 8.8.8.8 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0
```

- Client-Local

```
Client-Local interfaces
```

```
#  
# This is a sample network config, please uncomment lines to configure the network  
#  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 172.15.1.3  
    netmask 255.255.255.0  
    gateway 172.15.1.254  
    up echo nameserver 8.8.8.8 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0
```

- Client-DB permettant d'interagir avec le serveur-DB

```
Client-DB interfaces
```

```
#  
# This is a sample network config, please uncomment lines to configure the network  
#  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 172.15.1.5  
    netmask 255.255.255.0  
    gateway 172.15.1.254  
    up echo nameserver 8.8.8.8 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0
```

c. Mise en place du serveur FTP

Pour la mise en place du serveur FTP, on a d'abord mis à jours les paquets disponibles:

```
apt update
```

Puis on a installé les paquets VSFTPD via la commande:

```
apt install vsftpd
```

Ensuite on navigue vers le fichier de configuration

```
nano /etc/vsftpd.conf
```

Puis on le modifie et l'adapte comme suit:

```
# Run scandir.c. vsftpd can run either from an initscript or a
# daemon started from an initscript.
listen=YES
#
# Entries returned by the MAILER-PIPE command are also directed to
# option.
use_localtime=YES
#
```

On décommente la ligne `listen=YES` et la ligne `use_localtime=YES`.

Ensuite on crée un utilisateur

```
adduser naruto
```

On démarre ensuite les services vsftpd:

```
/etc/init.d/vsftpd start
```

On part sur un client local et on installe les paquets FTP:

```
apt install ftp
```

Puis on se connecte au serveur FTP:

```
ftp 172.15.1.2
```

```
/ # ftp 172.15.1.2
Connected to 172.15.1.2.
220 (vsFTPd 3.0.3)
Name (172.15.1.2:root): naruto
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

L'utilité du serveur FTP est de permettre le transfert de fichiers d'un client local à un autre. Donc l'accès est restreint et n'est autorisé qu'aux clients LAN et VPN. Cette restriction est gérée par la règle `iptables` qui refuse tous les paquets entrant dans le LAN.

IV. Configuration DMZ

Comme indiquée sur la topologie, la DMZ est une zone démilitarisée qui abrite des serveurs. Le réseau de la DMZ est **172.16.1.0/24**.

Le principe est de faire de tel sorte que:

- Les connexions depuis internet vers le LAN sont interdites
- Les connexions depuis la DMZ vers le LAN sont interdites
- Les connexions depuis le LAN vers la DMZ sont autorisées
- Les connexions depuis le LAN vers internet sont autorisées
- Les connexions depuis la DMZ vers internet sont autorisées
- Les connexions depuis internet vers la DMZ sont autorisées

Ces règles sont déjà mises en place sur le pare-feu.

a. Mise en place du serveur-web Apache

1. Mise en place

Pour mettre en place le serveur web, il faut mettre à jour les paquets disponibles puis installer les paquets Apache2.

```
apt update
apt install apache2
```

On va appliquer une mise en place par site. Pour ce faire, on navigue vers le dossier

/var/www/html. On renomme le fichier index.html existant en index.html.old puis on crée un autre fichier index.html et y mettre notre code.

```
cd /var/www/html  
mv index.html index.html.old  
nano index.html
```

Ensuite on démarre les services Apache2

```
/etc/init.d/apache2 start
```

2. Configuration sur le pare-feu

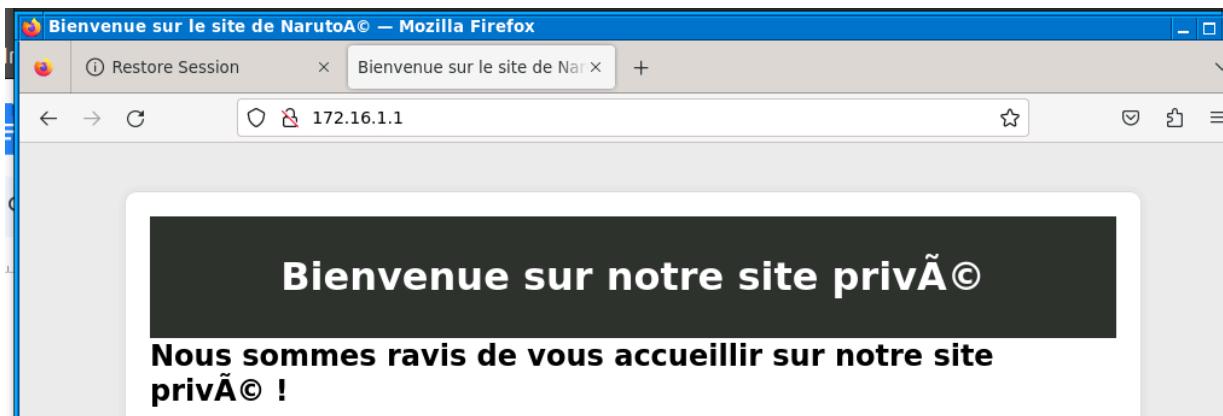
- Redirection des connexions HTTP (8080) et HTTPS (443) vers le serveur de messagerie

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 8080 -j DNAT  
--to-destination 172.16.1.1:8080  
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT  
--to-destination 172.16.1.1:443
```

- Autoriser le forwarding des trafics HTTP et HTTPS

```
iptables -A FORWARD -d 172.16.1.1 -p tcp --dport 8080 -j ACCEPT  
iptables -A FORWARD -d 172.16.1.1 -p tcp --dport 443 -j ACCEPT
```

3. Test



Test avec client externe

On essaye de se connecter avec un client webterm externe. Pour ce faire, on écrit sur la barre recherche web l'adresse du pare-feu. Le port forwarding va permettre au pare-feu de connaître l'adresse du serveur web et nous y rediriger.

6 juil. 02:07
Capture en cours de - [web-Server eth0 to Switch2 Ethernet1]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.122.231	172.16.1.1	TCP	74	47736 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=1666671898 TSecr=0 WS=128
2	0.000202	172.16.1.1	192.168.122.231	TCP	74	80 → 47736 [SYN] Seq=0 Ack=1 Win=31856 Len=0 MSS=1460 SACK_PERM TSval=1997305346 TSecr=0 WS=128
3	0.000659	192.168.122.231	172.16.1.1	TCP	66	47736 → 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=1666671899 TSecr=1997305346
4	0.000799	192.168.122.231	172.16.1.1	HTTP	496	GET / HTTP/1.1
5	0.000951	172.16.1.1	192.168.122.231	TCP	66	80 → 47736 [ACK] Seq=1 Ack=431 Win=31872 Len=0 TSval=1997305347 TSecr=1666671899
6	0.001768	172.16.1.1	192.168.122.231	HTTP	1028	HTTP/1.1 200 OK (text/html)
7	0.002268	192.168.122.231	172.16.1.1	TCP	66	47736 → 80 [ACK] Seq=431 Ack=963 Win=31872 Len=0 TSval=1666671900 TSecr=1997305348
8	0.169689	192.168.122.231	172.16.1.1	HTTP	496	GET / HTTP/1.1
9	0.170067	172.16.1.1	192.168.122.231	HTTP	1027	HTTP/1.1 200 OK (text/html)
10	0.170248	192.168.122.231	172.16.1.1	TCP	66	47736 → 80 [ACK] Seq=861 Ack=1924 Win=31872 Len=0 TSval=1666672068 TSecr=1997305516
11	0.398760	192.168.122.231	172.16.1.1	HTTP	496	GET / HTTP/1.1
12	0.399254	172.16.1.1	192.168.122.231	HTTP	1027	HTTP/1.1 200 OK (text/html)
13	0.399451	192.168.122.231	172.16.1.1	TCP	66	47736 → 80 [ACK] Seq=1291 Ack=2885 Win=31872 Len=0 TSval=1666672298 TSecr=1997305745
14	1.093608	192.168.122.231	172.16.1.1	HTTP	496	GET / HTTP/1.1
15	1.094371	172.16.1.1	192.168.122.231	HTTP	1027	HTTP/1.1 200 OK (text/html)
16	1.094655	192.168.122.231	172.16.1.1	TCP	66	47736 → 80 [ACK] Seq=1721 Ack=3846 Win=31872 Len=0 TSval=1666672993 TSecr=1997306440
17	5.084675	72:01:08:33:c1:ac	32:98:c1:9a:5e:25	ARP	42	Who has 172.16.1.1? Tell 172.16.1.1

La capture wireshark ci-dessus (liaison entre web-Server et switch2) montre que le client externe (192.168.122.231) parvient à se connecter au serveur web.

Test avec client externe connecté au serveur vpn

Pour ce faire, on a juste écrit l'adresse du serveur web. Etant donné que le client s'est connecté au serveur vpn, les requêtes sont transmises au serveur vpn (172.15.1.1) via le tunnel vpn qui va ensuite les transmettre au serveur web. Voici les images des captures wireshark:

6 juil. 02:07
Capture en cours de - [debian-1 eth0 to Switch3 Ethernet1]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	c6:7e:da:59:69:49	Spanning-tree-(for... STP	52	Conf. Root = 32768/0:52:54:00:58:71:6d Cost = 0 Port = 0x8001	
2	1.983978	c6:7e:da:59:69:49	Spanning-tree-(for... STP	52	Conf. Root = 32768/0:52:54:00:58:71:6d Cost = 0 Port = 0x8001	
3	4.032019	c6:7e:da:59:69:49	Spanning-tree-(for... STP	52	Conf. Root = 32768/0:52:54:00:58:71:6d Cost = 0 Port = 0x8001	
4	5.848283	192.168.122.237	192.168.122.181	OpenVPN	82	MessageType: P_DATA_V2
5	5.849767	192.168.122.181	192.168.122.237	OpenVPN	82	MessageType: P_DATA_V2
6	6.016093	c6:7e:da:59:69:49	Spanning-tree-(for... STP	52	Conf. Root = 32768/0:52:54:00:58:71:6d Cost = 0 Port = 0x8001	
7	7.411291	192.168.122.237	192.168.122.181	OpenVPN	157	MessageType: P_DATA_V2
8	7.412424	192.168.122.181	34.107.243.93	TLSv1.2	105	Application Data
9	7.508729	34.107.243.93	192.168.122.181	TLSv1.2	105	Application Data
10	7.509958	192.168.122.181	192.168.122.237	OpenVPN	157	MessageType: P_DATA_V2
11	7.510637	192.168.122.237	192.168.122.181	OpenVPN	118	MessageType: P_DATA_V2
12	7.511490	192.168.122.181	34.107.243.93	TCP	66	44160 → 443 [ACK] Seq=40 Ack=40 Win=249 Len=0 TSval=3475730203 TSecr=3263652721
13	7.999916	c6:7e:da:59:69:49	Spanning-tree-(for... STP	52	Conf. Root = 32768/0:52:54:00:58:71:6d Cost = 0 Port = 0x8001	
14	9.412318	192.168.122.237	192.168.122.181	OpenVPN	164	MessageType: P_DATA_V2
15	9.413736	192.168.122.181	34.149.100.209	TLSv1.2	112	Application Data
16	9.547993	34.149.100.209	192.168.122.181	TLSv1.2	112	Application Data
17	9.549030	192.168.122.181	192.168.122.237	OpenVPN	164	MessageType: P_DATA_V2
18	9.549101	192.168.122.237	192.168.122.181	OpenVPN	116	MessageType: P_DATA_V2

6 juil. 02:06

Capture en cours de - [proxy eth0 to Switch1 Ethernet0]

Fichier	Editer	Vue	Aller	Capture	Analyser	Statistiques	Telephonie	Wireless	Outils	Aide
No.	Time	Source	Destination	Protocol	Length	Info				
15	21.450847	192.168.122.237	172.15.1.1	OpenVPN	543	MessageType: P_DATA_V2				
16	21.450999	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
17	21.451444	172.16.1.1	172.15.1.1	TCP	66	80 -> 52478 [ACK] Seq=1 Ack=426 Win=31872 Len=0 TStamp=729196972 TSecr=2700041760				
18	21.451582	172.15.1.1	192.168.122.237	OpenVPN	118	MessageType: P_DATA_V2				
19	21.452044	172.16.1.1	172.15.1.1	HTTP	1028	HTTP/1.1 200 OK (text/html)				
20	21.452166	172.15.1.1	192.168.122.237	OpenVPN	1080	MessageType: P_DATA_V2				
21	21.452691	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2				
22	21.452796	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=426 Ack=963 Win=31872 Len=0 TStamp=2700041762 TSecr=729196973				
23	21.871203	192.168.122.237	172.15.1.1	OpenVPN	543	MessageType: P_DATA_V2				
24	21.871591	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
25	21.872419	172.15.1.1	172.16.1.1	HTTP	1027	HTTP/1.1 200 OK (text/html)				
26	21.872508	172.15.1.1	192.168.122.237	OpenVPN	1079	MessageType: P_DATA_V2				
27	21.872856	192.168.122.237	172.15.1.1	OpenVPN	118	MessageType: P_DATA_V2				
28	21.872930	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=851 Ack=1924 Win=31872 Len=0 TStamp=2700042183 TSecr=729197393				
29	22.015742	2a:12:02:27:f7:ae	3e:85:a8:23:5e:1b	ARP	42	Who has 172.15.1.1? Tell 172.15.1.254				
30	22.015918	3e:85:a8:23:5e:1b	2a:12:02:27:f7:ae	ARP	42	172.15.1.1 is at 3e:85:a8:23:5e:1b				
31	22.158926	192.168.122.237	172.15.1.1	OpenVPN	543	MessageType: P_DATA_V2				

6 juil. 02:06

Capture en cours de - [web-Server eth0 to Switch2 Ethernet1]

Fichier	Editer	Vue	Aller	Capture	Analyser	Statistiques	Telephonie	Wireless	Outils	Aide
No.	Time	Source	Destination	Protocol	Length	Info				
1	0.000000	172.15.1.1	172.16.1.1	TCP	74	52478 -> 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1400 SACK_PERM TStamp=2700041758 TSecr=0 WS=128				
2	0.000189	172.16.1.1	172.15.1.1	TCP	74	80 -> 52478 [SYN, ACK] Seq=1 Win=31856 Len=0 MSS=1400 SACK_PERM TStamp=729196971 TSecr=0 WS=128				
3	0.001514	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TStamp=2700041760 TSecr=729196971				
4	0.001700	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
5	0.001811	172.16.1.1	172.15.1.1	TCP	66	80 -> 52478 [ACK] Seq=1 Ack=426 Win=31872 Len=0 TStamp=729196972 TSecr=2700041760				
6	0.002417	172.16.1.1	172.15.1.1	HTTP	1028	HTTP/1.1 200 OK (text/html)				
7	0.003414	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=426 Ack=963 Win=31872 Len=0 TStamp=2700041762 TSecr=729196973				
8	0.422234	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
9	0.422846	172.16.1.1	172.15.1.1	HTTP	1027	HTTP/1.1 200 OK (text/html)				
10	1.423515	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=851 Ack=1924 Win=31872 Len=0 TStamp=2700042183 TSecr=729197393				
11	0.709713	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
12	0.710162	172.16.1.1	172.15.1.1	HTTP	1027	HTTP/1.1 200 OK (text/html)				
13	0.710822	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=1276 Ack=2885 Win=31872 Len=0 TStamp=2700042470 TSecr=729197680				
14	0.914895	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
15	0.915820	172.16.1.1	172.15.1.1	HTTP	1027	HTTP/1.1 200 OK (text/html)				
16	0.918000	172.15.1.1	172.16.1.1	TCP	66	52478 -> 80 [ACK] Seq=1701 Ack=3846 Win=31872 Len=0 TStamp=2700042676 TSecr=729197886				
17	1.089595	172.15.1.1	172.16.1.1	HTTP	491	GET / HTTP/1.1				
18	1.260250	172.16.1.1	172.15.1.1	HTTP	1027	HTTP/1.1 200 OK (text/html)				

Test avec client local

On se connecte depuis un webterm (172.15.1.4) du LAN en écrivant l'adresse du serveur-web

6 juil. 02:52

Capture en cours de - [web-Server eth0 to Switch2 Ethernet1]

Fichier	Editer	Vue	Aller	Capture	Analyser	Statistiques	Telephonie	Wireless	Outils	Aide
No.	Time	Source	Destination	Protocol	Length	Info				
1	0.000000	172.15.1.4	172.16.1.1	TCP	74	52442 -> 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1400 SACK_PERM TStamp=952723307 TSecr=0 WS=128				
2	0.000189	172.16.1.1	172.15.1.4	TCP	74	80 -> 52442 [SYN, ACK] Seq=1 Win=31856 Len=0 MSS=1400 SACK_PERM TStamp=331901090 TSecr=0 WS=128				
3	0.000653	172.15.1.4	172.16.1.1	TCP	66	52442 -> 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TStamp=952723307 TSecr=331901090				
4	0.000804	172.15.1.4	172.16.1.1	HTTP	491	GET / HTTP/1.1				
5	0.000984	172.15.1.4	172.16.1.4	TCP	66	80 -> 52442 [ACK] Seq=1 Ack=426 Win=31872 Len=0 TStamp=331901091 TSecr=952723308				
6	0.001747	172.16.1.1	172.15.1.4	HTTP	1028	HTTP/1.1 200 OK (text/html)				
7	0.002102	172.15.1.4	172.16.1.1	TCP	66	52442 -> 80 [ACK] Seq=426 Ack=963 Win=31872 Len=0 TStamp=952723309 TSecr=331901092				
8	0.687044	172.15.1.4	172.16.1.1	HTTP	491	GET / HTTP/1.1				
9	0.688697	172.16.1.1	172.15.1.4	HTTP	1027	HTTP/1.1 200 OK (text/html)				
10	0.689188	172.15.1.4	172.16.1.1	TCP	66	52442 -> 80 [ACK] Seq=851 Ack=1924 Win=31872 Len=0 TStamp=952723996 TSecr=331901779				
11	1.077969	172.15.1.4	172.16.1.1	HTTP	491	GET / HTTP/1.1				
12	1.078415	172.16.1.1	172.15.1.4	HTTP	1027	HTTP/1.1 200 OK (text/html)				
13	1.078673	172.15.1.4	172.16.1.1	TCP	66	52442 -> 80 [ACK] Seq=1276 Ack=2885 Win=31872 Len=0 TStamp=952724386 TSecr=331902168				
14	1.531022	172.15.1.4	172.16.1.1	HTTP	491	GET / HTTP/1.1				
15	1.531672	172.16.1.1	172.15.1.4	HTTP	1027	HTTP/1.1 200 OK (text/html)				
16	1.532011	172.15.1.4	172.16.1.1	TCP	66	52442 -> 80 [ACK] Seq=1701 Ack=3846 Win=31872 Len=0 TStamp=952724839 TSecr=331902622				
17	1.896826	172.15.1.4	172.16.1.1	HTTP	491	GET / HTTP/1.1				
18	1.896826	172.16.1.1	172.15.1.4	HTTP	1027	HTTP/1.1 200 OK (text/html)				

On constate que les requêtes passent bien.

b. Mise en place du serveur ToIP

1. Mise en place

Pour la mise en place de notre serveur, nous avons utilisé **Asterisk**

Asterisk est un PABX-IP open source créé en 1999 par Mark Spencer fondateur de la société Digium. Asterisk est publié sous licence GPL. Il est aujourd'hui adopté sur une majorité de plateformes VoIP du marché.

Voici l'edit config du serveur ToIP



The screenshot shows a terminal window with the title "ToIP interfaces". The content of the file is as follows:

```
#  
# This is a sample network config, please uncomment lines to configure the network  
#  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 172.16.1.2  
    netmask 255.255.255.0  
    gateway 172.16.1.254  
    up echo nameserver 8.8.8.8 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0  
#iface eth0 inet dhcp  
#    hostname ubuntu-1  
  
# Static config for eth1
```

Pour l'installation, on met d'abord le serveur à jour puis on installe les paquets subversion et wget

```
apt install subversion wget
```

Puis navigue dans /usr/src et on essaye de télécharger et de désarchiver la version 18 d'Asterisk.

```
cd /usr/src
```

```
wget  
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz
```

```
tar -zxvf asterisk-18-current.tar.gz  
cd asterisk-18.*
```

Ensuite on se déplace dans le dossier `asterisk/contrib/script` pour installer les prérequis d' Asterisk avec le script `install_prereq`.

Pour exécuter ce script on tape la commande

```
./install_prereq install
```

On installe aussi le script `get_mp3_source.sh` pour pouvoir jouer du mp3.

```
./get_mp3_source.sh
```

On peut installer les paquets `libopus-dev libvpx-dev` pour la prise en charge des nouveaux codecs opus et vp8.

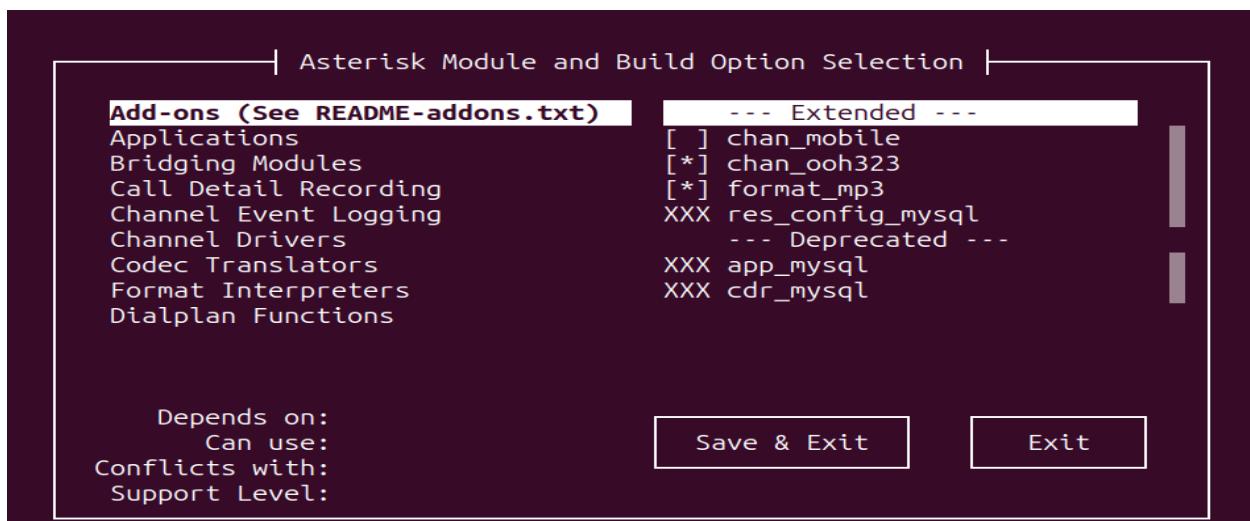
```
apt install libopus-dev libvpx-dev
```

Ensuite on revient dans le dossier source de asterisk `/usr/src/asterisk` et taper les commandes suivantes

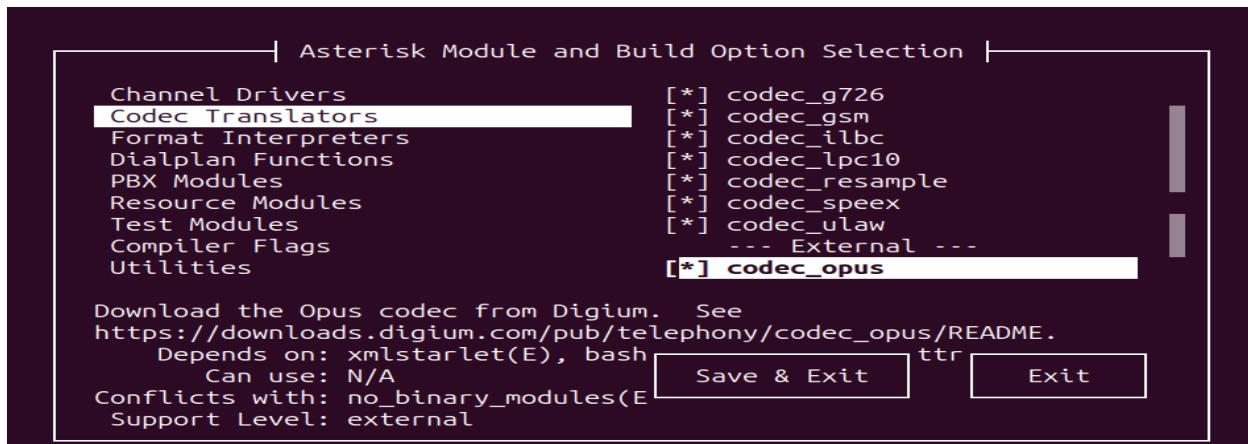
```
./configure --with-jansson-bundled
```

Après on tape la commande `make menuselect` pour choisir les fonctionnalités comme suit:

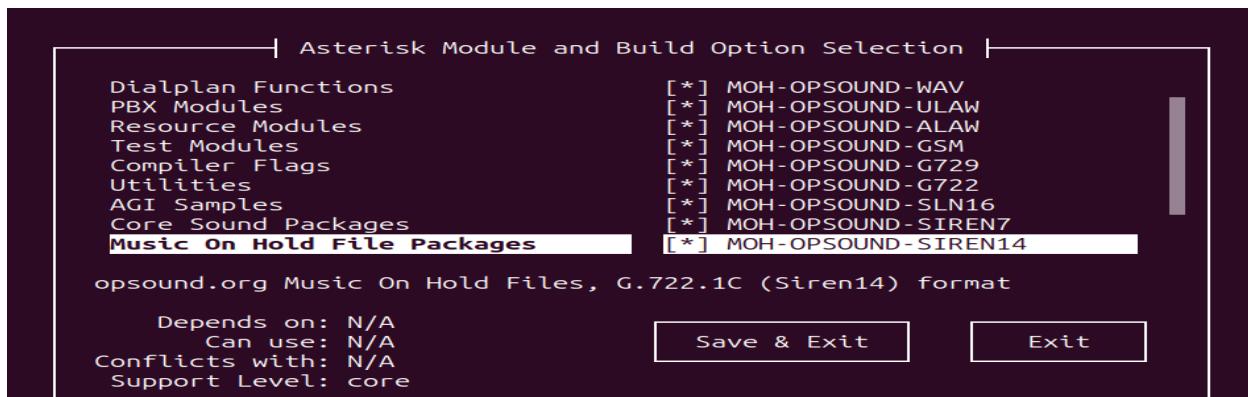
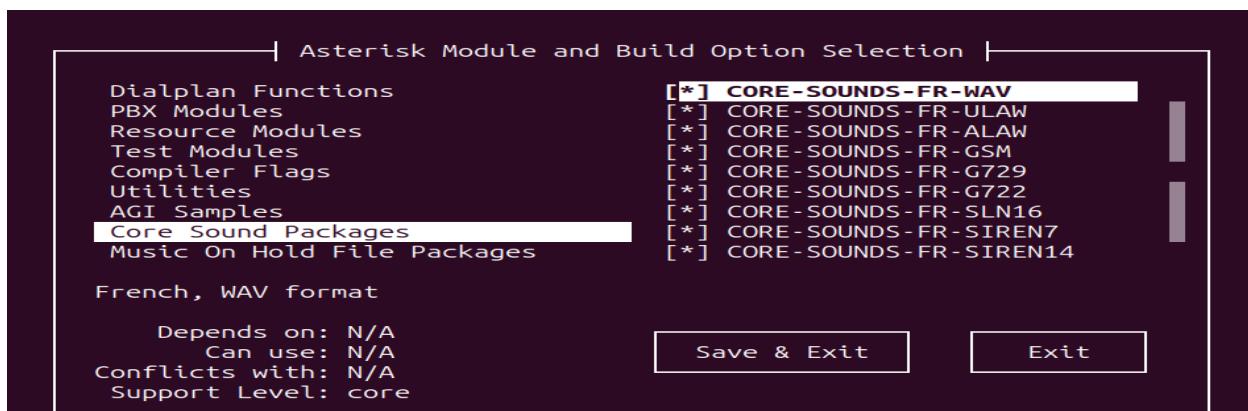
On active la prise en charge du format mp3 pour pouvoir jouer de la musique sur asterisk

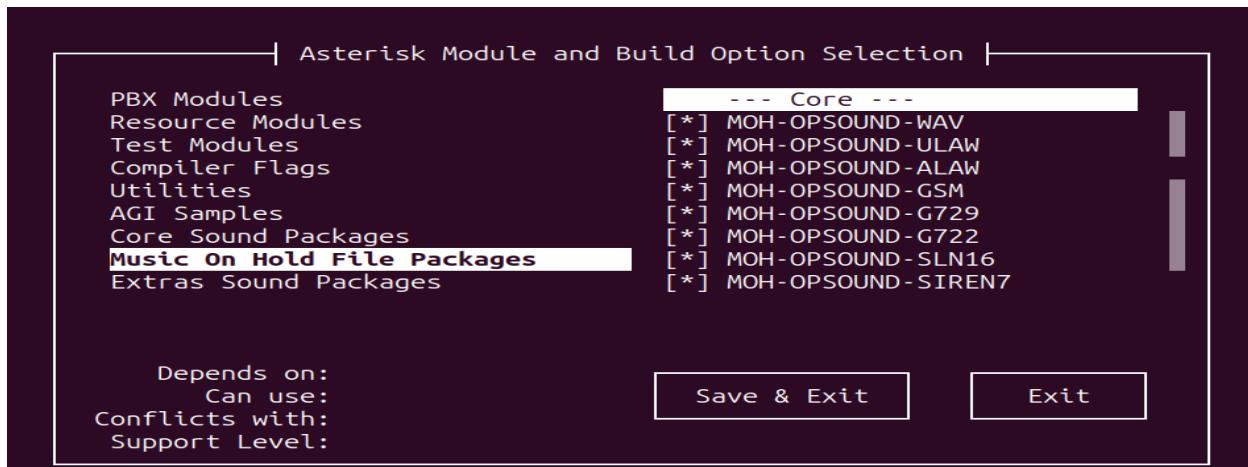


On active aussi la prise en charge du codec audio opus.



On active aussi le son en français comme montre les images ci-dessous.





Ensute on tape les commandes:

- make pour compiler asterisk
- make install pour installer les fonctionnalités
- make samples pour installer les fichiers de configurations
- make config pour générer les scripts de démarrage de asterisk

Maintenant l'installation d' asterisk est terminé on tape la commande :

```
/etc/init.d/asterisk start
```

pour démarrer le serveur et pour entrer dans la console d'asterisk on tape la commande:

```
asterisk -r
```

Par la suite on configure PJSIP et pour se faire on édite le fichier pjsip.conf via la commande:

```
nano /etc/asterisk/pjsip.conf
```

Puis on ajoute les configurations suivantes:

```

[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[1000]
type=endpoint
context=default
disallow=all
allow=ulaw
auth=auth1000
aors=1000

[auth1000]
type=auth
auth_type=userpass
username=1000
password=your_password
■

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute

```

```

[1000]
type=aor
max_contacts=1

[1001]
type=endpoint
context=default
disallow=all
allow=ulaw
auth=auth1001
aors=1001

[auth1001]
type=auth
auth_type=userpass
username=1001
password=another_password

[1001]

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute

```

1000 et 1001 sont deux clients configurés pour utiliser asterisk.

On édite ensuite le fichier extensions.conf via la commande

```
nano /etc/asterisk/extensions.conf
```

Puis on ajoute ces configurations:

```
;#exec "/opt/bin/build-extra-contexts.sh --foo=\"ba
;
[default]
exten => 1000,1,Dial(PJSIP/1000,20)
exten => 1000,n,Voicemail(1000,u)
exten => 1000,n,Hangup

exten => 1001,1,Dial(PJSIP/1001,20)
exten => 1001,n,Voicemail(1001,u)
exten => 1001,n,Hangup
```

Ensute on active les configurations asterisk en tapant les commandes suivantes:

```
asterisk -rx "pjsip reload"
asterisk -rx "dialplan reload"
```

2. Configuration sur le Pare-feu

Pour bien sécuriser le serveur ToIP nous avons eu recours aux règles iptables. Ces règles sont établies sur le pare-feu comme suit:

- Rediriger le trafic SIP vers le serveur ToIP

```
iptables -t nat -A PREROUTING -i eth0 -p udp --dport 5060 -j DNAT
--to-destination 172.16.1.2:5060
```

- Rediriger le trafic RTP vers le serveur ToIP

```
iptables -t nat -A PREROUTING -i eth0 -p udp --dport 10000:20000 -j
DNAT --to-destination 172.16.1.2:10000-20000
```

- Autoriser le forwarding du trafic redirigé

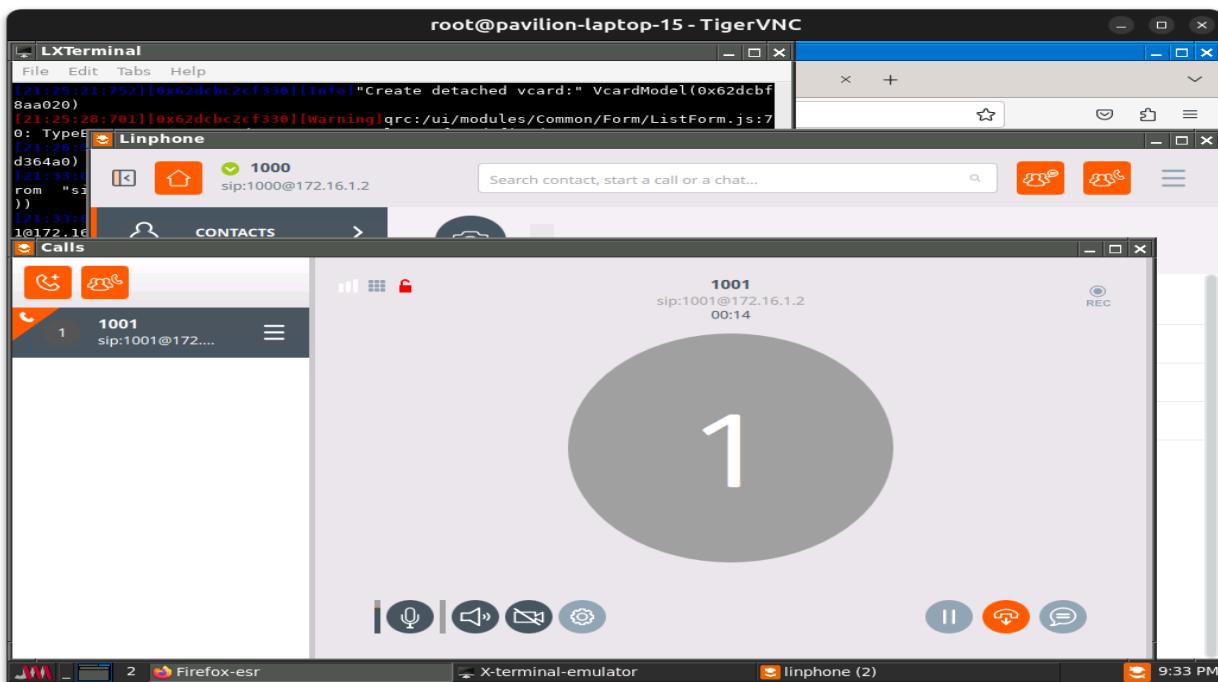
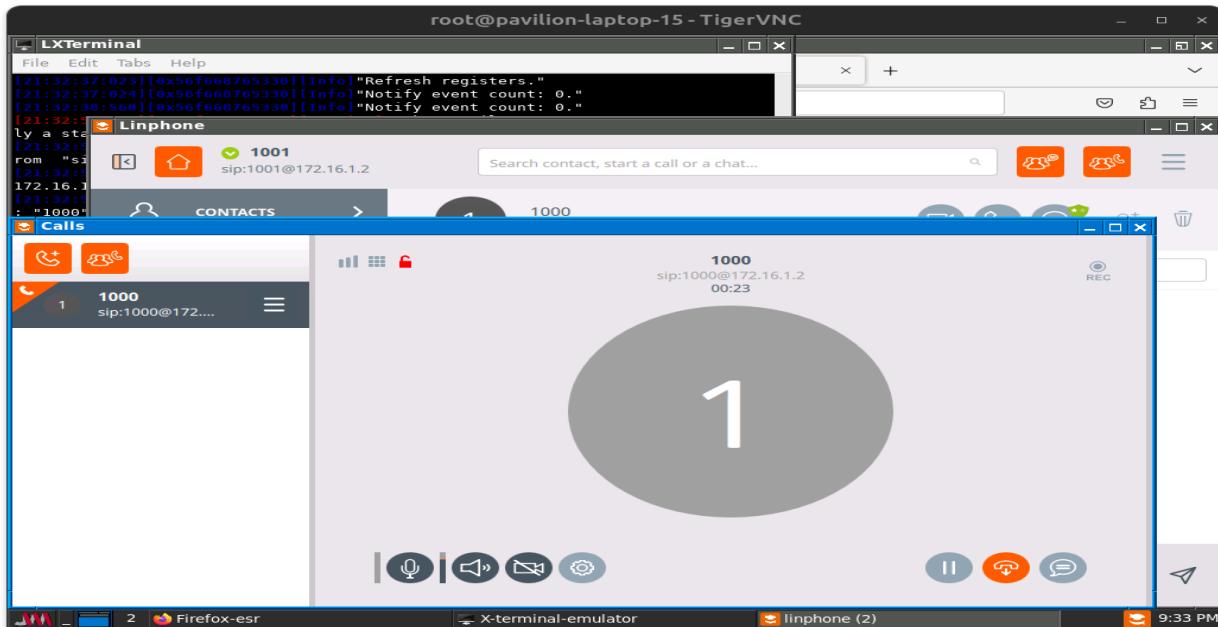
```
iptables -A FORWARD -d 172.16.1.2 -p udp --dport 5060 -j ACCEPT
iptables -A FORWARD -d 172.16.1.2 -p udp --dport 10000:20000 -j
ACCEPT
```

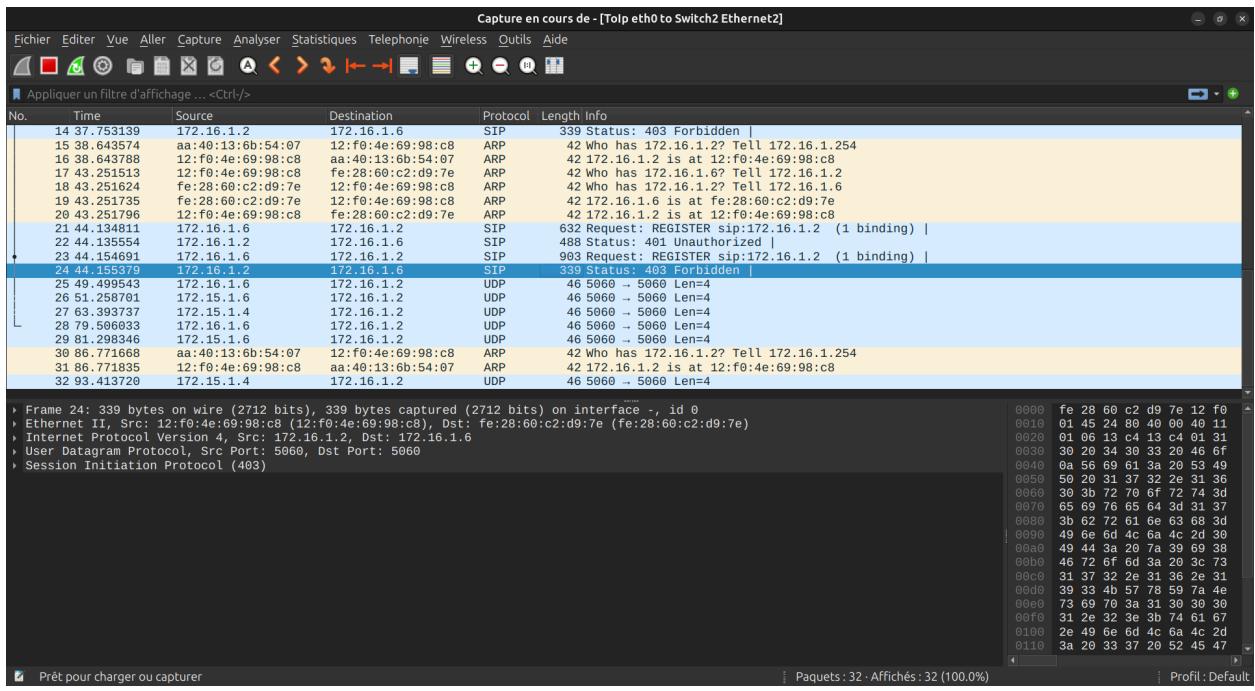
3. Test

Pour effectuer le test, on a installé linphone dans les webterm

```
apt update
apt install linphone-desktop
apt install dbus-x11
```

On se connecte sur linphone en tapant la commande linphone et se connecte avec les identifiants de 1000 et 1001 préenregistrés dans le fichier pjsip.conf puis on effectue des appels





b. Mise en place du serveur Base de données

1. Mise en place

Voici l'edit config du serveur base de données

```
# This is a sample network config, please uncomment lines to configure the network
#
# Uncomment this line to load custom interface files
# source /etc/network/interfaces.d/*
#
# Static config for eth0
auto eth0
iface eth0 inet static
    address 172.16.1.4
    netmask 255.255.255.0
    gateway 172.16.1.254
    up echo nameserver 8.8.8.8 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#    hostname ubuntu-1

# Static config for eth1
#auto eth1
```

On installe le paquet mysql-server sur serveur-db et sur la machine cliente depuis le LAN avec la commande `apt install mysql-server -y`

```
[root@serveur-db ~]# apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

On édite ensuite le fichier de configuration de mysql avec

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

On se déplace ensuite vers la ligne `bind-address = 127.0.0.1` et modifier l'adresse liée au serveur mysql de 127.0.0.1 à l'adresse du serveur-db 172.16.1.4

```
bind-address          = 172.16.1.4
mysqlx-bind-address = 172.16.1.4
```

On redémarre ensuite les services mysql avec la commande:

```
/etc/init.d/mysql start
```

On se connecte donc en tant que root avec la commande

```
mysql -u root -p
```

On crée ensuite un utilisateur du nom de naruto avec comme adresse 172.15.1.% qui représente toutes les adresses IP du réseau LAN incluant le client local et l'interface eth0 du Proxy appartenant dans ce réseau.

```
mysql> create user 'naruto'@'172.15.1.%' identified by 'passer123'
```

2. Configuration sur le Pare-feu

Sur le pare-feu, on établit les règles de filtrage pour bien sécuriser le serveur. Ces règles s'établissent comme suit :

- Autoriser le forwarding du trafic MySQL

-
- ```
iptables -A FORWARD -d 172.16.1.4 -p tcp --dport 3306 -j ACCEPT
```
- Redirection des connexions MySQL vers le serveur de base de données

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 3306 -j DNAT --to-destination 172.16.1.4:3306
```

### 3. Test

On installe mysql puis on essaie de se connecter à partir d'un client local.

```
mysql -h 172.16.1.4 -u naruto -p
```

```
"/ "# mysql -h 172.16.1.4 -u naruto -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.37-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

On voit que le client du LAN parvient à se connecter à partir des identifiants de naruto.

Voici la capture wireshark de la liaison entre le switch2 et le serveur-DB:

Dans l'ensemble, la capture montre une connexion TCP standard entre le client et le serveur, suivie d'une connexion TLS sécurisée et de communications ultérieures.

| Capture en cours de - [Switch2 Ethernet4 to DB-Server eth0] |          |                   |                   |          |        |                                                                                               |
|-------------------------------------------------------------|----------|-------------------|-------------------|----------|--------|-----------------------------------------------------------------------------------------------|
| No.                                                         | Time     | Source            | Destination       | Protocol | Length | Info                                                                                          |
| 1                                                           | 0.000000 | 172.16.1.5        | 172.16.1.4        | TCP      | 74     | 56942 → 3306 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TStamp=3677994271 TSecr=0 WS...   |
| 2                                                           | 0.000220 | 172.16.1.4        | 172.16.1.5        | TCP      | 74     | 3306 → 56942 [SYN, ACK] Seq=0 Ack=1 Win=31856 Len=0 MSS=1460 SACK_PERM TStamp=1070960590 T... |
| 3                                                           | 0.000287 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TStamp=3677994272 TSecr=1070960590             |
| 4                                                           | 0.000333 | 172.16.1.5        | 172.16.1.5        | MySQL    | 140    | 56942 → 3306 [Query] proto=10 Version=5.0.37 -Ubuntu.24.04.1                                  |
| 5                                                           | 0.001723 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [Ack] Seq=1 Ack=90 Win=32128 Len=0 TStamp=3677994273 TSecr=1070960591            |
| 6                                                           | 0.001761 | 172.16.1.5        | 172.16.1.4        | MySQL    | 102    | Login Request user=                                                                           |
| 7                                                           | 0.001858 | 172.16.1.4        | 172.16.1.4        | TCP      | 66     | 3306 → 56942 [ACK] Seq=96 Ack=37 Win=31872 Len=0 TStamp=1070960592 TSecr=3677994273           |
| 8                                                           | 0.007188 | 172.16.1.5        | 172.16.1.4        | TLSv1.3  | 388    | Client Hello                                                                                  |
| 9                                                           | 0.007425 | 172.16.1.4        | 172.16.1.5        | TCP      | 66     | 3306 → 56942 [ACK] Seq=96 Ack=359 Win=31872 Len=0 TStamp=1070960597 TSecr=3677994278          |
| 10                                                          | 0.008533 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 1514   | Server Hello, Change Cipher Spec, Application Data, Application Data                          |
| 11                                                          | 0.008573 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 836    | Application Data, Application Data, Application Data                                          |
| 12                                                          | 0.009095 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [ACK] Seq=359 Ack=2314 Win=31872 Len=0 TStamp=3677994280 TSecr=1070960599        |
| 13                                                          | 0.010331 | 172.16.1.5        | 172.16.1.4        | TLSv1.3  | 174    | 3306 → 56942 [Spec, Application Data, Application Data]                                       |
| 14                                                          | 0.010775 | 172.16.1.5        | 172.16.1.4        | TLSv1.3  | 301    | Application Data                                                                              |
| 15                                                          | 0.011068 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 321    | Application Data                                                                              |
| 16                                                          | 0.011118 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 321    | Application Data                                                                              |
| 17                                                          | 0.011216 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 94     | Application Data                                                                              |
| 18                                                          | 0.011346 | 172.16.1.4        | 172.16.1.5        | TLSv1.3  | 99     | Application Data                                                                              |
| 19                                                          | 0.011506 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [ACK] Seq=704 Ack=2824 Win=31872 Len=0 TStamp=3677994283 TSecr=1070960601        |
| 20                                                          | 0.011784 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [ACK] Seq=704 Ack=2885 Win=31872 Len=0 TStamp=3677994283 TSecr=1070960601        |
| 21                                                          | 0.012106 | 172.16.1.5        | 172.16.1.4        | TLSv1.3  | 127    | Application Data                                                                              |
| 22                                                          | 0.012130 | 172.16.1.5        | 172.16.1.4        | TLSv1.3  | 162    | Application Data                                                                              |
| 23                                                          | 0.053988 | 172.16.1.5        | 172.16.1.4        | TCP      | 66     | 56942 → 3306 [ACK] Seq=765 Ack=2379 Win=31872 Len=0 TStamp=3677994325 TSecr=1070960603        |
| 24                                                          | 5.391642 | 72:30:bc:2e:11:83 | b6:13:0e:db:20:46 | ARP      | 48     | who has 172.16.1.4? Tell 172.16.1.254                                                         |
| 25                                                          | 5.391728 | b6:13:0e:db:20:46 | 72:30:bc:2e:11:83 | ARP      | 48     | who has 172.16.1.254? Tell 172.16.1.4                                                         |
| 26                                                          | 5.391816 | b6:13:0e:db:20:46 | 72:30:bc:2e:11:83 | ARP      | 42     | 172.16.1.4 is at b6:13:0e:db:20:46                                                            |
| 27                                                          | 5.391865 | 72:30:bc:2e:11:83 | b6:13:0e:db:20:46 | ARP      | 42     | 172.16.1.254 is at 72:30:bc:2e:11:83                                                          |

## c. Mise en place du serveur Messagerie

### 1. Mise en place

Voici l'edit config du serveur mail

```
MailServer interfaces

This is a sample network config, please uncomment lines to configure the network
#
Uncomment this line to load custom interface files
source /etc/network/interfaces.d/*

Static config for eth0
auto eth0
iface eth0 inet static
 address 172.16.1.5
 netmask 255.255.255.0
 gateway 172.16.1.254
 up echo nameserver 8.8.8.8 > /etc/resolv.conf

DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
hostname ubuntu-1

Static config for eth1
```

Nous utiliserons Mailcow Dockerized qui est un serveur de courrier entièrement présenté alimenté par Docker.

---

Pour ce faire, nous devons d'abord mettre à jour la liste des paquets disponibles en tapant la commande:

```
apt update
```

Ensuite on tape la commande:

```
apt install apt-transport-https ca-certificates curl gnupg-agent
software-properties-common
```

Cette commande installe plusieurs paquets nécessaires pour l'installation de Docker :

- `apt-transport-https` : Permet à `apt` d'utiliser des dépôts via HTTPS.
- `ca-certificates` : Installe les certificats nécessaires pour vérifier les certificats SSL.
- `curl` : Un outil de ligne de commande pour transférer des données avec des URL.
- `gnupg-agent` : Un outil pour gérer les clés GPG.
- `software-properties-common` : Ajoute des scripts pour gérer les dépôts

Ensuite on tape la commande:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
```

Cette commande télécharge la clé GPG officielle de Docker et l'ajoute au système. La clé GPG est utilisée pour vérifier l'intégrité et l'authenticité des paquets Docker.

On tape ensuite la commande:

```
add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

qui ajoute le dépôt officiel de Docker à la liste des sources de paquets `apt`. `$(lsb_release -cs)` insère automatiquement le nom de code de votre version d'Ubuntu.

On remet à jour les paquets disponibles via `apt update` puis on installe Docker Community Edition (Docker CE), l'interface en ligne de commande Docker (`docker-ce-cli`), et le runtime `containerd` via la commande:

```
apt-get install docker-ce docker-ce-cli containerd.io
```

---

On exécute la commande dockerd pour mettre en marche les services docker.

Maintenant on installe manuellement le plugin Compose.

Pour ce faire on tape les commandes:

```
DOCKER_CONFIG=${DOCKER_CONFIG:-$HOME/.docker}
```

Cette commande définit la variable d'environnement DOCKER\_CONFIG. Si DOCKER\_CONFIG est déjà définie, elle conserve sa valeur. Sinon, elle est définie à \$HOME/.docker, qui est un répertoire dans le répertoire personnel de l'utilisateur. Cela garantit que les configurations Docker sont stockées dans un endroit approprié.

```
mkdir -p $DOCKER_CONFIG/cli-plugins
```

Cette commande crée le répertoire cli-plugins dans le chemin spécifié par DOCKER\_CONFIG. L'option `-p` permet de créer tous les répertoires parents nécessaires sans générer d'erreur si le répertoire existe déjà.

```
curl -SL
https://github.com/docker/compose/releases/download/v2.28.1/docker-compose-linux-x86_64 -o $DOCKER_CONFIG/cli-plugins/docker-compose
```

Après cela on applique des autorisations exécutables au binaire selon la commande:

```
chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
```

On ajoute ensuite notre utilisateur Linux à la dockergroupe

On a avait au préalable ajouter un utilisateur du nom de naruto

```
adduser naruto
usermod -aG docker naruto
usermod -aG docker root
```

L'installation de docker compose est finie. Maintenant on installe mailcow-dockerized.

Pour ce faire, on navigue dans /etc/opt puis on clone le repository github de mailcow:

```
cd /etc/opt/
```

---

```
git clone https://github.com/mailcow/mailcow-dockerized
```

Ensuite on navigue dans /etc/opt/mailcow-dockerized puis on génère le fichier de configuration via la commande

```
./generate_config.sh
```

Par la suite on choisit le Hostname de notre mailserver

```
Mail server hostname (FQDN) - this is not your mail domain, but your mail server's hostname: mail.dsti.sn
Timezone [Africa/Dakar]:
[Ctrl-D] to finish
```

Puis on édite le fichier `mailcow.conf` et on le modifie comme suit:

```
nano mailcow.conf
```

```
SKIP_LETS_ENCRYPT=y
Create generate certificates for all domains - y/n
```

Maintenant on démarre `mailcow` en tapant la commande:

```
docker compose up -d
```

Ceci permet à `mailcow` d'installer ses dépendances.

```

[+] Running 24/19[] Pulling 49.7s B Pulling 49.5s 39MB Pulling 49
[+] Running 24/19[] Pulling 49.8s B Pulling 49.6s 39MB Pulling 49
[+] Running 24/19[] Pulling 49.9s B Pulling 49.7s 39MB Pulling 49
[+] Running 24/19[] Pulling 50.0s B Pulling 49.8s 39MB Pulling 49
[+] Running 25/19[] Pulling 50.1s B Pulling 49.9s 39MB Pulling 49
[+] Running 26/19[] Pulling 50.2s B Pulling 50.0s 39MB Pulling 49
[+] Running 26/19[] Pulling 50.3s B Pulling 50.1s 39MB Pulling 49
[+] Running 26/19[] Pulling 50.4s B Pulling 50.2s 39MB Pulling 49
[+] Running 26/19[] Pulling 50.5s B Pulling 50.3s 39MB Pulling 49
[+] Running 26/19[] Pulling 50.6s B Pulling 50.4s 39MB Pulling 50
[+] Running 26/19[] Pulling 50.7s B Pulling 50.5s 39MB Pulling 50
[+] clamd-mailcow [] Pulling 50.8s B Pulling 50.6s 39MB Pulling 50
[+] redis-mailcow [] Pulling 50.8s B Pulling 50.7s 39MB Pulling 50
[+] olefy-mailcow [██████] 9.741MB / 43.91MB Pulling 50.8s 39MB Pulling 50
[+] solr-mailcow [] Pulling 50.8s MB / 85.39MB Pulling 50.8s 39MB Pulling 50
[+] rspamd-mailcow [] Pulling 50.8s s 85.39MB / 85.39MB Pulling 50
[+] sogo-mailcow [] Pulling 50.8s 85.39MB / 85.39MB Pulling 50
[+] dovecot-mailcow [██████████] 85.39MB / 85.39MB Pulling 50
[+] pv6nat-mailcow [] Pulling 50.7s g 50.6s
[+] ipv6nat-mailcow [] Pulling 50.8s g 50.7s
[+] postfix-mailcow [] Pulling 50.8s
[+] ofelia-mailcow [] Pulling 50.8s 7s 6s .0s
[+] watchdog-mailcow [] Pulling 50.8s 7s .1s
[+] memcached-mailcow [██████] Pulling 50.8s .2s
[+] acme-mailcow [] Pulling 50.8s s .3s
[+] mysql-mailcow [] Pulling 50.8s .4s
[+] nginx-mailcow [] Pulling 50.8s .5s
[+] netfilter-mailcow [] Pulling 50.8s .6s
[+] dockerapi-mailcow [] Pulling 50.8s .7s
[+] php-fpm-mailcow [] Pulling 50.8s
[+] unbound-mailcow [] Pulling 50.8s

```

```

[+] Running 30/19[] Pulling 82.0s B Pulling 82.4s 39MB Pulling 82
[+] Running 30/19[] Pulling 82.7s B Pulling 82.5s 39MB Pulling 82
[+] Running 30/19[] Pulling 82.8s B Pulling 82.6s 39MB Pulling 82
[+] Running 30/19[] Pulling 82.9s B Pulling 82.7s 39MB Pulling 82
[+] Running 30/19[] Pulling 83.0s B Pulling 82.8s 39MB Pulling 82
[+] Running 30/19[] Pulling 83.1s B Pulling 82.9s 39MB Pulling 82
[+] Running 30/19[] Pulling 83.2s B Pulling 83.0s 39MB Pulling 82
[+] Running 30/19[] Pulling 83.3s B Pulling 83.1s 39MB Pulling 82
[+] Running 32/19[] Pulling 83.4s B Pulling 83.2s 39MB Pulling 82
[+] clamd-mailcow [] Pulling 175.4s B Pulling 83.3s 39MB Pulling 82
[+] Running 142/19 [] Pulling 175.4s Pulling 83.4s 39MB Pulling 83
✓ clamd-mailcow Pulled 377.0s 5.88MB Pulling 377.0s MB Pulling 83
✓ redis-mailcow Pulled 389.3s 13.16MB Pulling 389.3s B Pulling 83
✓ olefy-mailcow Pulled 234.2s .91MB Pulling 234.2s 9MB Pulling 83
[+] solr-mailcow [██████...] Pulling 468.8s B / 85.39MB Pulling 83
✓ rspamd-mailcow Pulled 229.0s 72.35MB Pulling 229.0s
✓ sogo-mailcow Pulled 409.5s / 121.2MB Pulling 409.4s
✓ dovecot-mailcow Pulled 83.4s ulling 175.4s
✓ ipv6nat-mailcow Pulled 338.6s 8.368MB Pulling 338.6s
✓ postfix-mailcow Pulled 434.8s 0.4MB / 110.4MB Pulling 434.8s
✓ ofelia-mailcow Pulled 311.3s 47MB Pulling 311.2s
✓ watchdog-mailcow Pulled 366.8s 33.55MB Pulling 366.7s
✓ memcached-mailcow Pulled 57.3s 175.4s .0s
✓ acme-mailcow Pulled 447.6s 42.73MB Pulling 447.6s
[+] mysql-mailcow [██████████] 120.7MB / 120.8MB Pulling 468.8s
✓ nginx-mailcow Pulled 429.7s / 18.38MB Pulling 429.6s
✓ netfilter-mailcow Pulled 234.3s / 39.83MB Pulling 234.3s
[+] dockerapi-mailcow [██████████] 60.37MB / 66.82MB Pulling 468.8s
✓ php-fpm-mailcow Pulled 454.8s .7MB / 104.7MB Pulling 454.7s 5): n
✓ unbound-mailcow Pulled 349.0s 12.55MB Pulling 348.9s

```

## 2. Configuration sur le Pare-feu

- Redirection des connexions SMTP vers le serveur de messagerie (port 25, 465, et 587)

```

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j DNAT
--to-destination 172.16.1.5:25
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 465 -j DNAT
--to-destination 172.16.1.5:465
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 587 -j DNAT
--to-destination 172.16.1.5:587

```

- 
- Redirection des connexions IMAP vers le serveur de messagerie (port 143 et 993)

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 143 -j DNAT
--to-destination 172.16.1.5:143
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 993 -j DNAT
--to-destination 172.16.1.5:993
```

- Redirection des connexions POP3 vers le serveur de messagerie (port 110 et 995)

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 110 -j DNAT
--to-destination 172.16.1.5:110
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 995 -j DNAT
--to-destination 172.16.1.5:995
```

- Autoriser le forwarding du trafic SMTP

```
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 25 -j ACCEPT
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 465 -j ACCEPT
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 587 -j ACCEPT
```

- Autoriser le forwarding du trafic IMAP

```
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 143 -j ACCEPT
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 993 -j ACCEPT
```

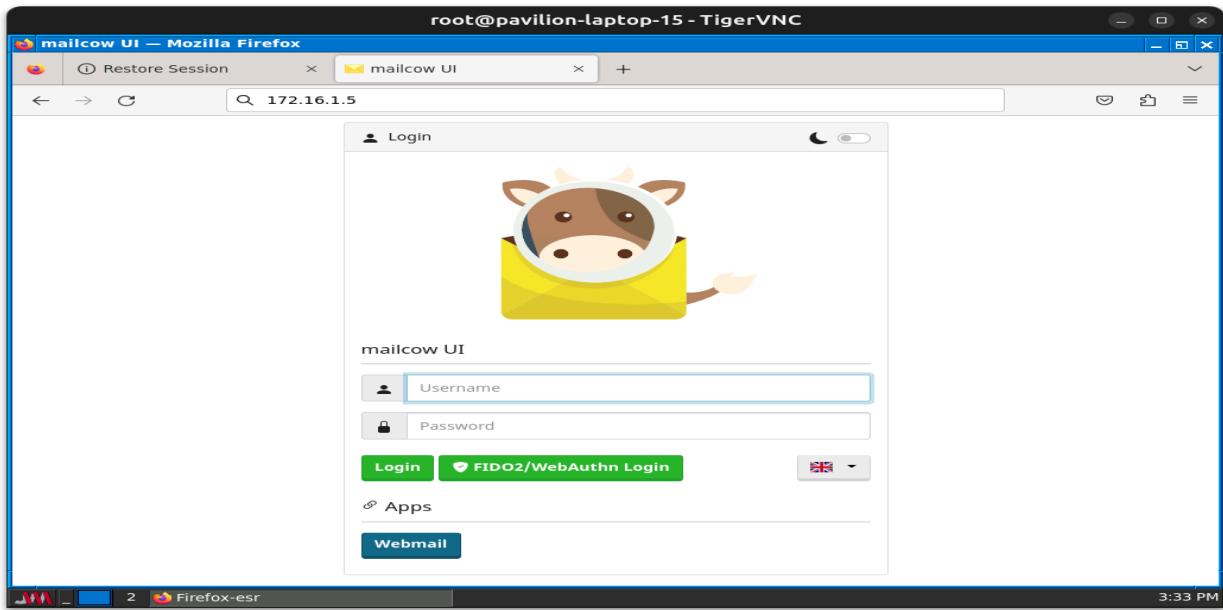
- Autoriser le forwarding du trafic POP3

```
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 110 -j ACCEPT
iptables -A FORWARD -d 172.16.1.5 -p tcp --dport 995 -j ACCEPT
```

### 3. Test

Une fois les installations terminées, on lance un client webterm et on saisit l'adresse IP du serveur de messagerie 172.16.1.5.

On voit l'interface de mailcow



On se connecte avec le nom d'utilisateur admin et le mot de passe moohoo.

On ajoute un domaine dsti.sn

A screenshot of a "Add domain" configuration form. The form has the following fields:

- Domain: dsti.sn
- Description: test mailserver
- Template: Default
- Tags: (empty)
- Max. possible aliases: 400
- Max. possible mailboxes: 10
- Default mailbox quota: 3072

Maintenant on ajoute deux utilisateurs: sidy ndiaye et limamoulaye dia

### Add mailbox

Username (left part of an email address)

Domain

Full name

Password ([generate](#))  Check against haveibeenowned.com  
- Minimum password length is 6

Confirmation password (repeat)

### Add mailbox

Username (left part of an email address)

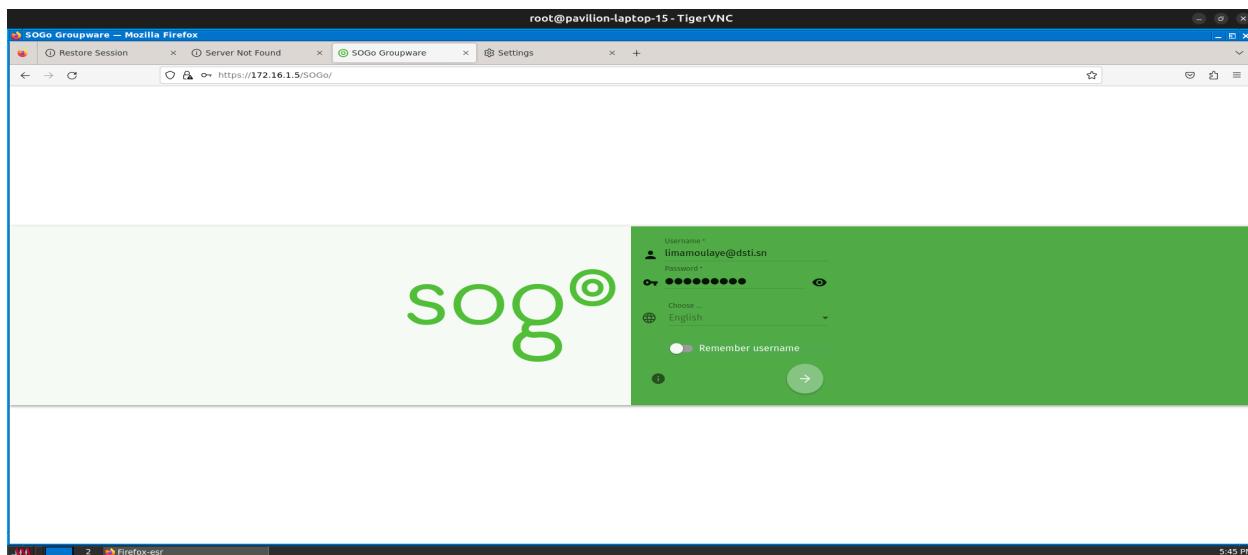
Domain

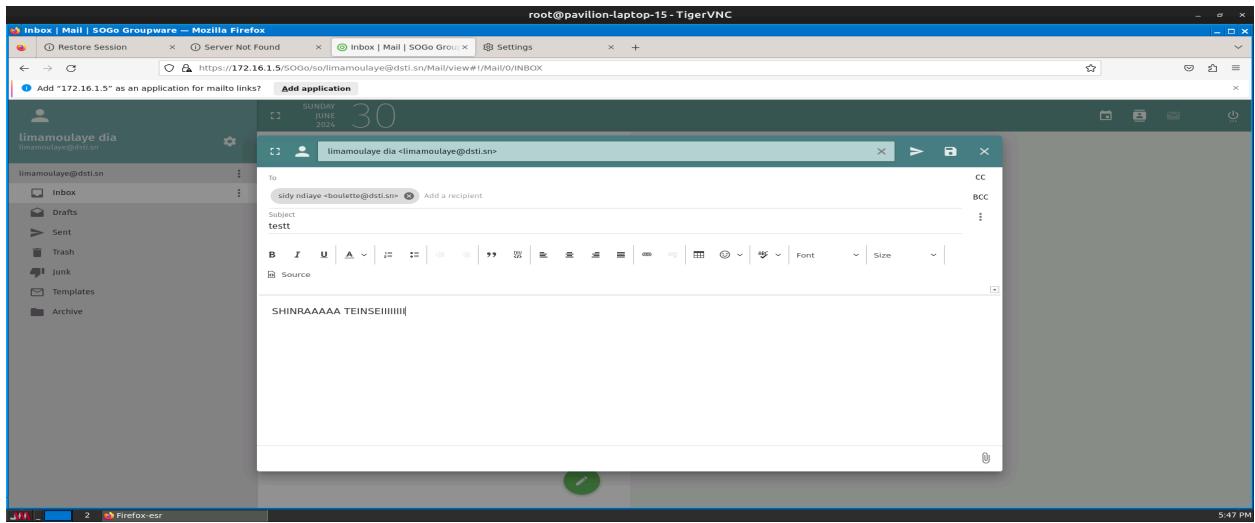
Full name

Password ([generate](#))  Check against haveibeenowned.com  
- Minimum password length is 6

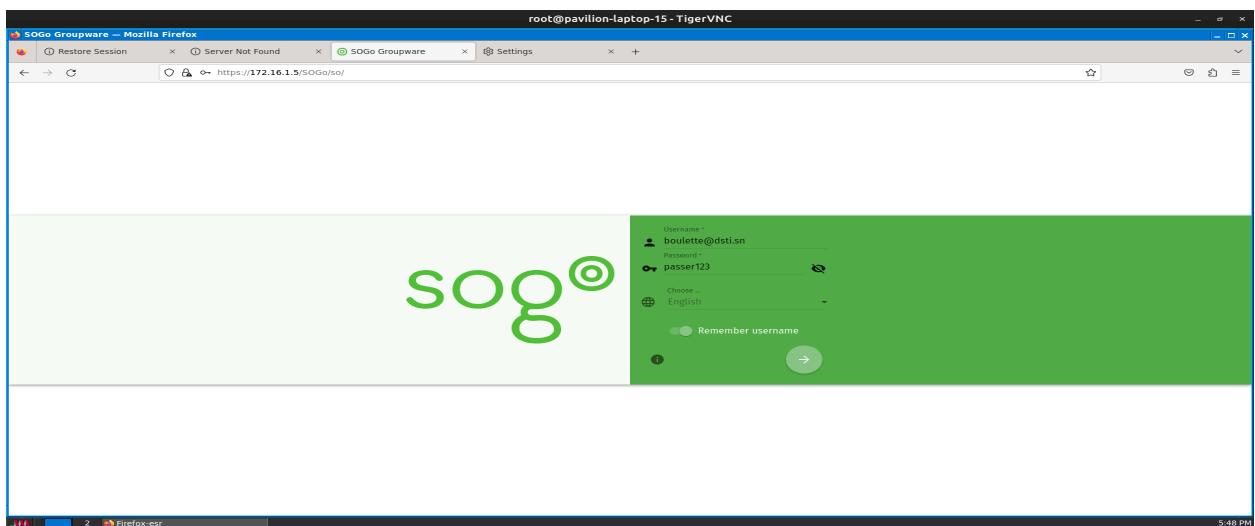
Confirmation password (repeat)

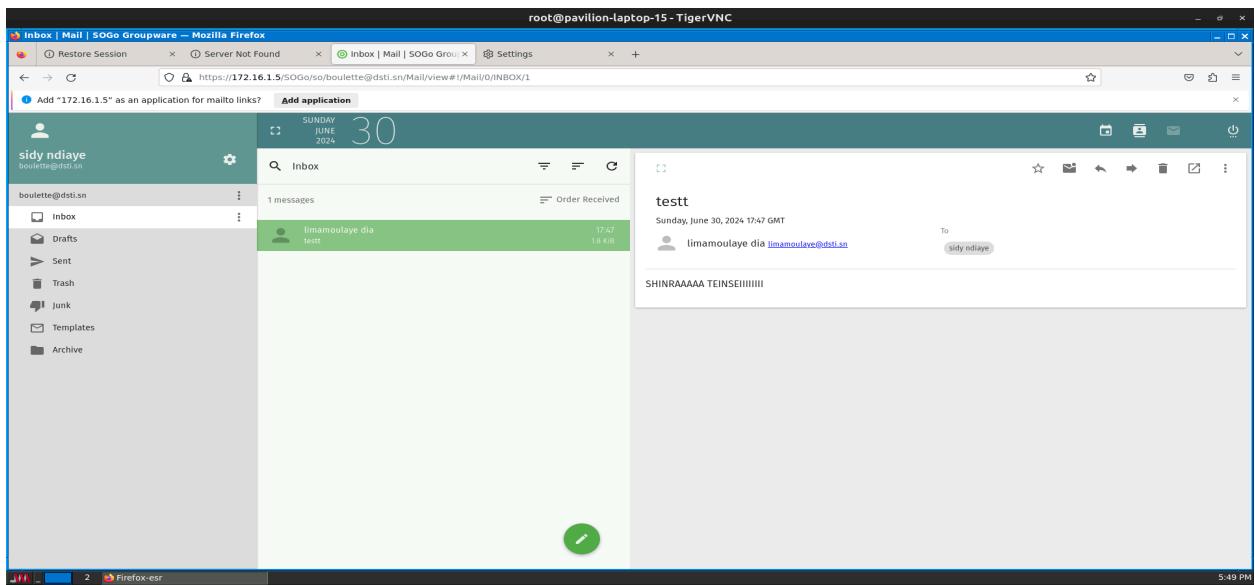
On se connecte ensuite avec les identifiants de limamou pour envoyer un message à sidi





On se connecte avec les identifiants de sidy pour voir si le message est venu





On voit bien que Sidy a reçu le message!