Nom : DOSSA

Prénom : Lophias

Filière : Master Cloud et Infrastructures 1

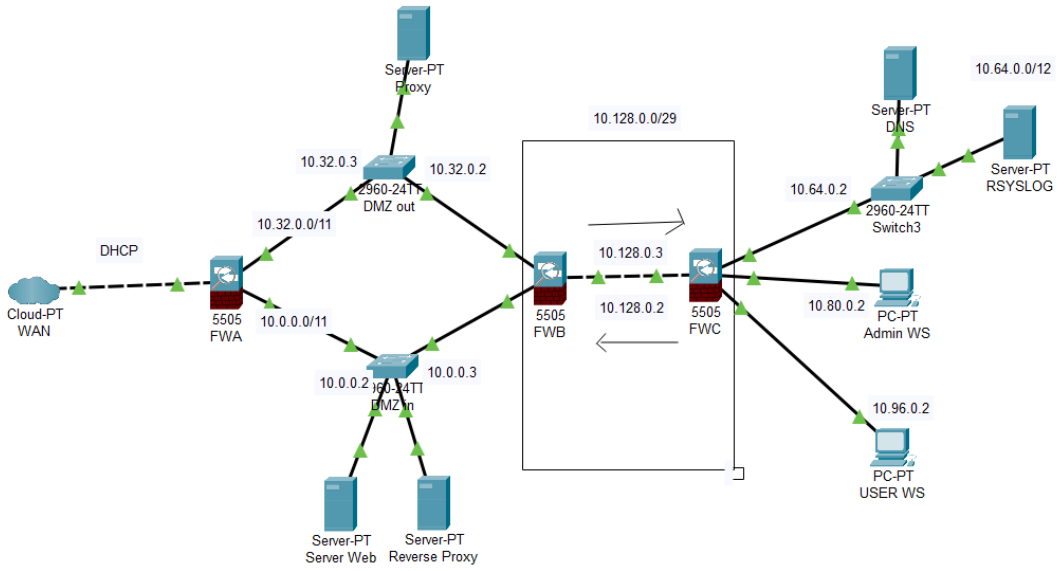
Compte rendu

Sujet : Mise en place d’une infrastructure sécurisée

Année : 2024-2025

Introduction

Pour la réalisation de ce projet, j'ai tout d'abord personnalisé le schéma comme le montre le schéma ci-après :



J'ai donc placé le proxy (Squid) dans la zone DMZ-Out afin de filtrer et contrôler les accès Internet des utilisateurs internes. Le reverse proxy, quant à lui, est positionné dans la DMZ-In pour assurer la protection et le routage des requêtes vers les serveurs internes. Bien évidemment, le serveur web est également situé dans la DMZ-In pour permettre aux utilisateurs externes d'y accéder sans pénétrer dans le réseau interne.

Pour ce TP, j'ai choisi de placer le serveur interne dans la zone "Internal Servers". J'ai délibérément opté pour l'installation de tous les services essentiels sur cette machine — DNS, système de centralisation des logs, serveur web, proxy et reverse proxy — afin de limiter la consommation de ressources sur la machine principale. Nous détaillerons par la suite la configuration de ces différents services.

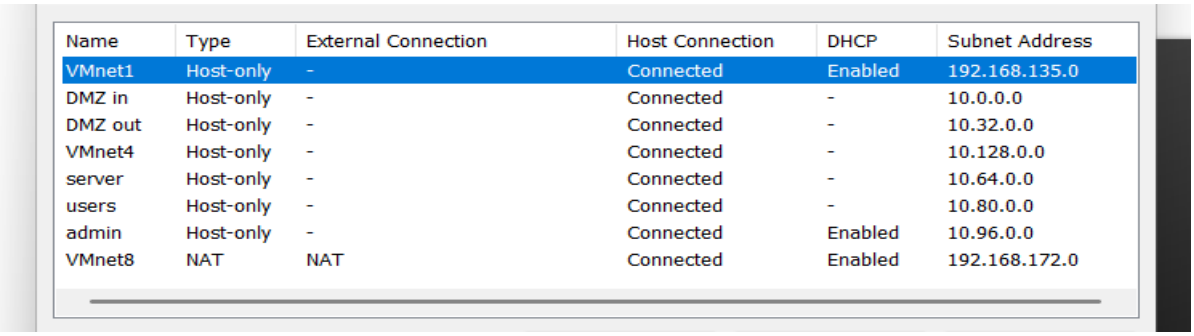
La machine utilisée comme poste client a été installée dans la zone "User WC", conformément aux conventions précisées dans le document.

Résumé : Pour le déploiement de l'infrastructure, j'ai utilisé VMware Workstation et créé un total de cinq machines virtuelles : trois pour les firewalls, une pour le poste client et une pour le serveur.

Mise en œuvre

a- Installation des machines virtuelles

J'ai commencé à créer des VmNet sur VMware pour la connexion de chaque machine au sous-réseau correspondant.



Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.135.0
DMZ in	Host-only	-	Connected	-	10.0.0.0
DMZ out	Host-only	-	Connected	-	10.32.0.0
VMnet4	Host-only	-	Connected	-	10.128.0.0
server	Host-only	-	Connected	-	10.64.0.0
users	Host-only	-	Connected	-	10.80.0.0
admin	Host-only	-	Connected	Enabled	10.96.0.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.172.0

1- Firewall A (fwa)

Ce firewall a trois interfaces :

- La première a le DHCP et est connecté à internet
- La deuxième est connectée au DMZ-in
- La troisième est connectée au DMZ-out

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet dhcp

#Interface DMZ out VMNET3
auto ens35
iface ens35 inet static
        address 10.32.0.3
        netmask 255.224.0.0
        post-up ip route add 10.64.0.0/12 via 10.32.0.10
        post-up ip route add 10.80.0.0/12 via 10.32.0.10
#Interface DMZ in VMNET2
auto ens34
iface ens34 inet static
        address 10.0.0.2
        netmask 255.224.0.0
```

2- Firewall B (fwb)

Nous avons également trois interfaces. Une qui est reliée au DMZ-out, une autre au DMZ-in et la dernière interface au Firewall C

```
iface lo inet loopback

# The primary network interface
#Interface DMZIn
auto ens33
iface ens33 inet static
        address 10.0.0.3
        netmask 255.224.0.0
#InterfaceDMZout
auto ens34
iface ens34 inet static
        address 10.32.0.2
        netmask 255.224.0.0
#Interface fwc
auto ens35
iface ens35 inet static
        address 10.128.0.3
        netmask 255.255.255.248
#route par défaut pour joindre internal defaut, users wc, admin ws
post-up ip route add 10.64.0.0/12 via 10.128.0.3
post-up ip route add 10.80.0.0/12 via 10.128.0.3
post-up ip route add 10.96.0.0/12 via 10.128.0.3
```

3- Firewall C

Nous avons 4 interfaces ici : une pour le firewall b, trois autres pour internal servers, users WC & admin ws.

```
#Interface vers fwb
auto ens33
iface ens33 inet static
        address 10.128.0.2
        netmask 255.255.255.248
        post-up ip route add 10.0.0.0/11 via 10.128.0.3
        post-up ip route add 10.32.0.0/11 via 10.128.0.3
#Interface Internal servers
auto ens34
iface ens34 inet static
        address 10.64.0.2
        netmask 255.240.0.0

#Interface Users WS
auto ens35
iface ens35 inet static
        address 10.80.0.2
        netmask 255.240.0.0

#Interface Admin WS
auto ens36
iface ens36 inet static
        address 10.96.0.2
        netmask 255.240.0.0
```

NB : Des routes ont été ajoutées dans chaque configuration afin de permettre l'accès direct au réseau interne. J'ai veillé à installer les paquets *iptables* sur chacune des machines, car la majorité des règles de filtrage seront définies à leur niveau.

4- Configuration du serveur unique

Comme mentionné précédemment, le serveur est unique et ne possède qu'une seule interface réseau. Pour répondre aux besoins du projet, j'ai configuré trois alias sur cette interface. Chaque alias est activé en fonction de la zone avec laquelle le serveur doit communiquer :

- **DMZ-Out** : pour la gestion du proxy ;
- **DMZ-In** : pour la gestion du site web ;
- **Réseau interne (10.64.0.0/12 - Internal Servers)** : pour la gestion du DNS et la centralisation des logs.

Ainsi, il est possible de basculer entre les différentes configurations à l'aide de commandes comme : ifdown ens33 && ifup ens33 :0 par exemple pour se connecter à la DMZ-in et continuellement. Voici alors le rendu :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#interface dmz-in -pour le server web
iface ens33:0 inet static
        address 10.0.0.10
        netmask 255.224.0.0
        gateway 10.0.0.2
#interface dmzout pour le proxy
iface ens33:1 inet static
        address 10.32.0.10
        netmask 255.224.0.0
        gateway 10.32.0.3
#interface pour dns et syslog
iface ens33:2 inet static
        address 10.64.0.10
        netmask 255.240.0.0
        gateway 10.64.0.2
        ip route add 10.80.0.0/12 via 10.64.0.2
```

Dans ma configuration, trois passerelles (*gateway*) sont définies. Cette configuration est possible car les trois interfaces ne fonctionnent jamais simultanément. Lorsque l'une des interfaces est activée, les autres sont désactivées, car le changement de réseau s'effectue en modifiant le *VmNet* au niveau de VMware.

5- Configuration du poste client

```
# The loopback network interface
auto lo
iface lo inet loopback

#conf pour l'interface du posteclient
auto ens36
iface ens36 inet static
        address 10.80.0.10
        netmask 255.240.0.0
        gateway 10.80.0.2
        dns-nameservers 10.64.0.10 8.8.8.8
        up ip route add 10.64.0.0/12 via 10.80.0.2
```

b- Quelques tests de connectivités

- Depuis le firewall a, je teste juste la connectivité vers le DMZ-in et le DMZ out

```
root@fwa:~# ping 10.32.0.3
PING 10.32.0.3 (10.32.0.3) 56(84) bytes of data.
64 bytes from 10.32.0.3: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 10.32.0.3: icmp_seq=2 ttl=64 time=0.057 ms
64 bytes from 10.32.0.3: icmp_seq=3 ttl=64 time=0.054 ms
^C
--- 10.32.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.054/0.060/0.071/0.007 ms
root@fwa:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.093 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2037ms
rtt min/avg/max/mdev = 0.056/0.074/0.093/0.015 ms
root@fwa:~# _
```

- Depuis le firewall b, je teste la connectivité vers le firewall c

```
root@fwb:~# ping 10.128.0.3
PING 10.128.0.3 (10.128.0.3) 56(84) bytes of data.
64 bytes from 10.128.0.3: icmp_seq=1 ttl=64 time=9.39 ms
64 bytes from 10.128.0.3: icmp_seq=2 ttl=64 time=5.15 ms
64 bytes from 10.128.0.3: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 10.128.0.3: icmp_seq=4 ttl=64 time=0.056 ms
^C
--- 10.128.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.047/3.659/9.387/3.906 ms
```

- Depuis server, je teste la connectivité avec le client

```
PING 10.80.0.10 (10.80.0.10) 56(84) bytes of data.
64 bytes from 10.80.0.10: icmp_seq=1 ttl=63 time=3.11 ms
64 bytes from 10.80.0.10: icmp_seq=2 ttl=63 time=2.80 ms
64 bytes from 10.80.0.10: icmp_seq=3 ttl=63 time=3.80 ms
64 bytes from 10.80.0.10: icmp_seq=4 ttl=63 time=2.92 ms
^X64 bytes from 10.80.0.10: icmp_seq=5 ttl=63 time=2.10 ms
64 bytes from 10.80.0.10: icmp_seq=6 ttl=63 time=2.77 ms
^C
--- 10.80.0.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5013ms
rtt min/avg/max/mdev = 2.101/2.916/3.802/0.503 ms
root@server:/etc/network# _
```

- Depuis le client, je teste la connectivité avec le server

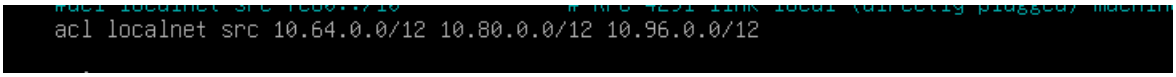
```
PING 10.64.0.10 (10.64.0.10) 56(84) bytes of data.
64 bytes from 10.64.0.10: icmp_seq=1 ttl=63 time=2.83 ms
64 bytes from 10.64.0.10: icmp_seq=2 ttl=63 time=2.34 ms
64 bytes from 10.64.0.10: icmp_seq=3 ttl=63 time=3.70 ms
64 bytes from 10.64.0.10: icmp_seq=4 ttl=63 time=4.31 ms
^C
--- 10.64.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.337/3.292/4.305/0.761 ms
root@client:~#
```

Il reste plusieurs tests de connectivité à réaliser. Par exemple, il est nécessaire de vérifier la connectivité entre le serveur et la DMZ-In, ainsi qu'entre le client et la DMZ-Out, ces zones étant cruciales pour tester le fonctionnement du DNS, du proxy et du serveur web. Ces tests ont été effectués en amont de ce rapport et ont confirmé la bonne connectivité. Je passe donc directement à l'installation des services sur le serveur unique.

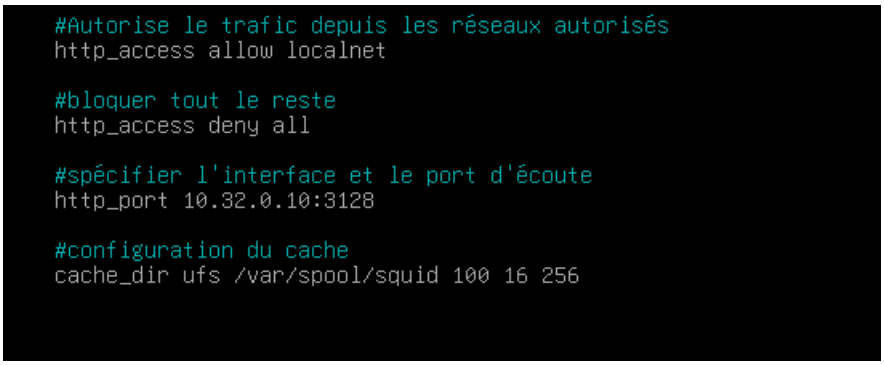
c- Installation des services sur le serveur unique.

1- Squid (Proxy)

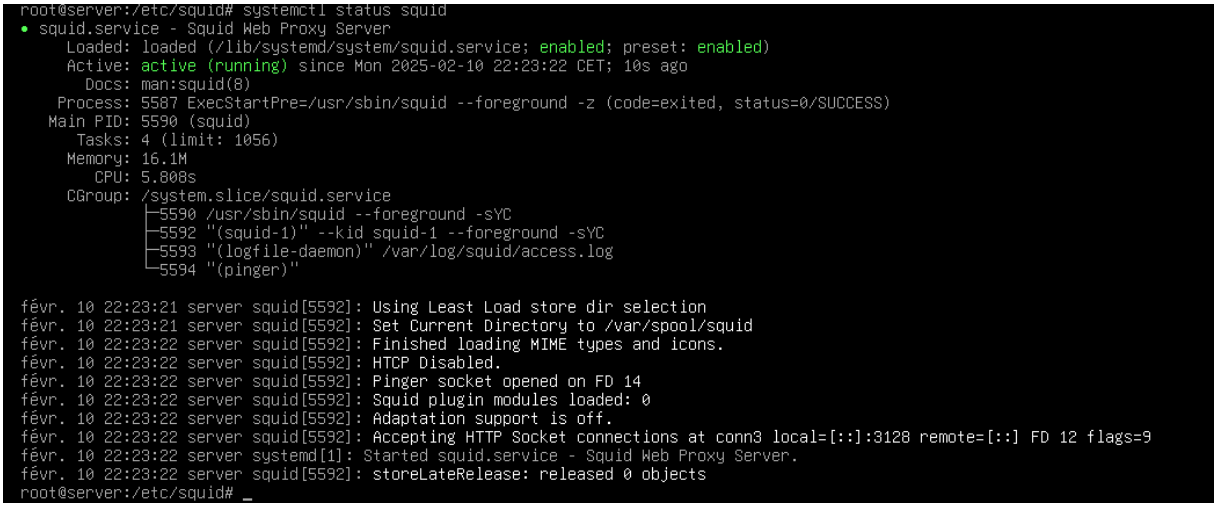
En faisant nano /etc/squid/squid.conf, j'ai rajouté juste cette ligne



Cette ligne demande de filtrer tout ce qui sort du réseau interne. Pour le reste des autres instructions du fichier, j'ai juste décommenté les lignes.



Après enregistrement du fichier, et après avoir démarrer Squid par la commande systemctl restart squid. Celui-ci est bien actif.



2- Rsyslog (pour la centralisation des logs)

RSYSLOG a été choisi comme solution de centralisation des logs pour notre infrastructure réseau. Après son installation, il est maintenant nécessaire de vérifier que celle-ci s'est déroulée correctement.

```
root@server:/# dpkg -l | grep rsyslog
ii  rsyslog                    8.2302.0-1                  amd64        reliable system and kernel logging daemon
root@server:/#
```

Maintenant que la présence de **RSYSLOG** est confirmée, nous allons procéder à quelques configurations supplémentaires. Il est essentiel d'activer l'écoute sur les protocoles **TCP** et **UDP** en modifiant le fichier de configuration **rsyslog.conf**.

```
#Activer l'écoute sur UDP
module(load="imudp")
input(type="imudp" port="514")

#Activer l'écoute sur TCP
module(load="imtcp")
input(type="imtcp" port="514")
```

Activation de RSYSLOG

```
root@server:/etc# systemctl restart rsyslog
root@server:/etc# systemctl status rsyslog
● rsyslog.service - System Logging Service
     Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-02-07 19:05:34 CET; 13s ago
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             man:rsyslog.conf(5)
             https://www.rsyslog.com/doc/
   Main PID: 1188 (rsyslogd)
      Tasks: 10 (limit: 1056)
     Memory: 1.3M
        CPU: 19ms
     CGroup: /system.slice/rsyslog.service
             └─1188 /usr/sbin/rsyslogd -n -iNONE

févr. 07 19:05:34 server systemd[1]: Starting rsyslog.service - System Logging Service...
févr. 07 19:05:34 server systemd[1]: Started rsyslog.service - System Logging Service.
févr. 07 19:05:34 server rsyslogd[1188]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.2302.0]
févr. 07 19:05:34 server rsyslogd[1188]: [origin software="rsyslogd" swVersion="8.2302.0" x-pid="1188" x-info="https://www.rsyslog.com"] start
root@server:/etc#
```

3- <u>Bind9 (DNS)</u>

Après installation du paquet, il faut configurer le fichier named.conf

```
zone "internal.lan" {
        type master;
        file "/etc/bind/db.internal.lan";
};
```

Configuration du fichier db.internal.lan

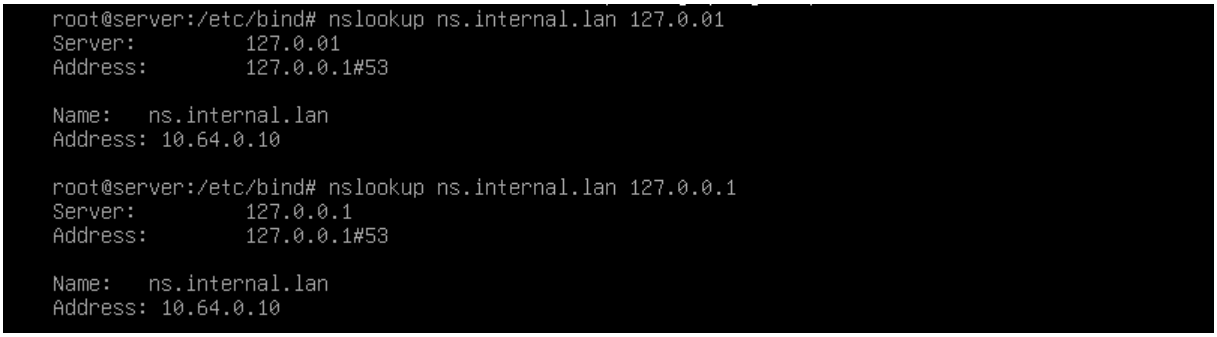
```
$TTL            604800
@               IN              SOA             ns.internal.lan. admin.internal.lan. (
                                        1               ; Serial
                                        604800          ; Refresh
                                        86400           ; Retry
                                        2419200         ; Expire
                                        604800 )        ; Negative Cache TTL
;
@               IN              NS              ns.internal.lan.
ns              IN              A               10.64.0.10
```

Le premier fichier permet de définir les zones dans lesquelles le serveur DNS doit gérer et rechercher les enregistrements.

Le second fichier, **db.internal.lan**, spécifie les correspondances entre les noms de domaine et les adresses IP pour la zone concernée. Dans notre cas, l'adresse **10.64.0.10** est utilisée, car le serveur DNS est situé dans la zone **Internal Servers**.

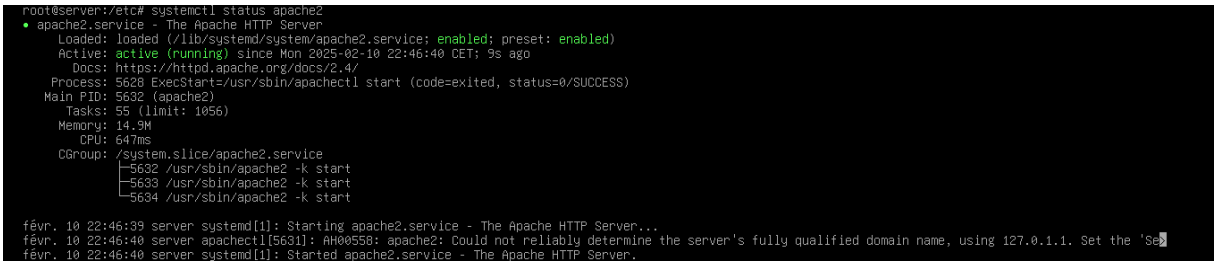
Après l'activation de **bind9**, l'écran suivant s'affiche :

Je teste directement sur le serveur et cela semble concluant et sans erreur.



4- <u>Apache2</u>

Pour la configuration d'Apache, je n'ai pas eu grand-chose à faire à part installer les paquets et à le démarrer.



Afin de m'assurer que la configuration fonctionne comme prévu, j'ai basculé l'interface de mon serveur web sur le réseau **DMZ-In**. J'ai ensuite testé la connectivité en exécutant la commande suivante depuis le serveur : http://10.0.0.10. Voici le résultat obtenu :

```
        <div class="section_header">
            <div id="docroot"></div>
                Document Roots
        </div>

        <div class="content_section_text">
            <p>
                By default, Debian does not allow access through the web browser to
                <em>any</em> file apart of those located in <tt>/var/www</tt>,
                <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">public_html</a>
                directories (when enabled) and <tt>/usr/share</tt> (for web
                applications). If your site is using a web document root
                located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
                document root directory in <tt>/etc/apache2/apache2.conf</tt>.
            </p>
            <p>
                The default Debian document root is <tt>/var/www/html</tt>. You
                can make your own virtual hosts under /var/www. This is different
                to previous releases which provides better security out of the box.
            </p>
        </div>

        <div class="section_header">
          <div id="bugs"></div>
                Reporting Problems
        </div>
        <div class="content_section_text">
          <p>
                Please use the <tt>reportbug</tt> tool to report bugs in the
                Apache2 package with Debian. However, check <a
                href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerged=0"
                rel="nofollow">existing bug reports</a> before reporting a new bug.
          </p>
          <p>
                Please report bugs specific to modules (such as PHP and others)
                to respective packages, not to the web server itself.
          </p>
        </div>




      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>
```

Le retour affiché correspond au contenu de ma page web, ce qui confirme que la configuration fonctionne comme prévu.

5- <u>Configuration du reverse-proxy sur apache2</u>

Apache2 me permet de configurer le reverse proxy. Cette configuration a pour but de masquer l'identité du serveur aux utilisateurs externes et d'optimiser les requêtes, dans le cadre d'une infrastructure sécurisée.

Pour ce faire, j'ai créé le fichier suivant nano /etc/apache2/sites-available/reverse-proxy.conf :

```
<VirtualHost *:8080>

        ProxyPreserveHost On
        ProxyPass "/" http://10.0.0.10/
        ProxyPassReverse "/"  http://10.0.0.10/

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

J’ai réutilisé l’adresse du server car c’est sur celui-ci qu’on configure le reverse-proxy.

<u>NB</u> : Pour des raisons de délais de rendu, je n’ai pas pu tester, le squid, le DNS, et le reverse-proxy. Ce rapport ne pourra donc pas en faire mention.

d- Les iptables

En tenant compte de la matrice des autorisations de trafic ci-dessous, voici toutes les règles iptables qui sont appliquées sur chacun des firewalls. Mise en place dans le Projet (en cours).

4. Matrice des autorisations de trafic

Seules les règles explicites sont mentionnées.

From ↓ To →	wan	dmz in	dmz out	Internal servers	Users WS	Admins WS
wan						
dmz in						
dmz out						
Internal servers						
Users WS						

Règles WAN

Interdire WAN ↔ WAN

iptables -A FORWARD -i enp0s3 -o enp0s3 -j DROP

WAN → DMZ In

iptables -A FORWARD -i enp0s3 -o enp0s8 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i enp0s3 -o enp0s8 -p tcp --dport 443 -j ACCEPT

WAN → DMZ Out (Autoriser accès au proxy Squid)

iptables -A FORWARD -i enp0s3 -o enp0s9 -p tcp --dport 3128 -j ACCEPT

WAN → Internal Servers (Interdire totalement)

iptables -A FORWARD -i enp0s3 -o enp0s10 -j DROP

WAN → Users WS (Interdire totalement)

iptables -A FORWARD -i enp0s3 -o enp0s11 -j DROP

Règles DMZ In

Interdire DMZ In ↔ DMZ In

iptables -A FORWARD -i enp0s8 -o enp0s8 -j DROP

DMZ In → WAN (Interdire tout sauf réponse aux connexions HTTP/HTTPS)

iptables -A FORWARD -i enp0s8 -o enp0s3 -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i enp0s8 -o enp0s3 -j DROP

DMZ In → DMZ Out (Autoriser le serveur web à contacter le proxy)

iptables -A FORWARD -i enp0s8 -o enp0s9 -p tcp --dport 3128 -j ACCEPT

DMZ In → Internal Servers (Autoriser DNS et logs)

iptables -A FORWARD -i enp0s8 -o enp0s10 -p udp --dport 53 -j ACCEPT

iptables -A FORWARD -i enp0s8 -o enp0s10 -p udp --dport 514 -j ACCEPT

DMZ In → Users WS (Interdire totalement)

iptables -A FORWARD -i enp0s8 -o enp0s11 -j DROP

Règles DMZ-out

Interdire DMZ Out ↔ DMZ Out

iptables -A FORWARD -i enp0s9 -o enp0s9 -j DROP

DMZ Out → WAN (Interdire tout sauf le trafic HTTP via le proxy)

iptables -A FORWARD -i enp0s9 -o enp0s3 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i enp0s9 -o enp0s3 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i enp0s9 -o enp0s3 -j DROP

DMZ Out → DMZ In (Interdire sauf communication Proxy → Web Server)

iptables -A FORWARD -i enp0s9 -o enp0s8 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i enp0s9 -o enp0s8 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i enp0s9 -o enp0s8 -j DROP

DMZ Out → Internal Servers (Autoriser accès DNS uniquement)

iptables -A FORWARD -i enp0s9 -o enp0s10 -p udp --dport 53 -j ACCEPT

DMZ Out → Users WS (Interdire totalement)

iptables -A FORWARD -i enp0s9 -o enp0s11 -j DROP

Règles Internal Servers

Interdire Internal Servers ↔ Internal Servers

iptables -A FORWARD -i enp0s10 -o enp0s10 -j DROP

Internal Servers → WAN (Interdire tout accès direct à Internet)

iptables -A FORWARD -i enp0s10 -o enp0s3 -j DROP

Internal Servers → DMZ In (Autoriser DNS et logs)

iptables -A FORWARD -i enp0s10 -o enp0s8 -p udp --dport 53 -j ACCEPT

iptables -A FORWARD -i enp0s10 -o enp0s8 -p udp --dport 514 -j ACCEPT

Internal Servers → DMZ Out (Autoriser requêtes DNS vers le proxy)

iptables -A FORWARD -i enp0s10 -o enp0s9 -p udp --dport 53 -j ACCEPT

Internal Servers → Users WS (Autoriser communication des logs)

iptables -A FORWARD -i enp0s10 -o enp0s11 -p udp --dport 514 -j ACCEPT

Règles Admin WS

Interdire Admin WS ↔ Admin WS

iptables -A FORWARD -i enp0s12 -o enp0s12 -j DROP

Admin WS → WAN (Interdire sauf mises à jour Debian)

iptables -A FORWARD -i enp0s12 -o enp0s3 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i enp0s12 -o enp0s3 -p tcp --dport 443 -j ACCEPT

iptables -A FORWARD -i enp0s12 -o enp0s3 -j DROP

Admin WS → Internal Servers

iptables -A FORWARD -i enp0s12 -o enp0s10 -p tcp --dport 22 -j ACCEPT

iptables -A FORWARD -i enp0s12 -o enp0s10 -p udp --dport 514 -j ACCEPT

Admin WS → DMZ In

iptables -A FORWARD -i enp0s12 -o enp0s8 -p tcp --dport 22 -j ACCEPT

Admin WS → Users WS

iptables -A FORWARD -i enp0s12 -o enp0s11 -p tcp --dport 3389 -j ACCEPT

iptables -A FORWARD -i enp0s12 -o enp0s11 -p tcp --dport 5900 -j ACCEPT

Conclusion

La mise en place de cette infrastructure sécurisée a été un exercice technique approfondi, mobilisant diverses compétences en administration système, réseau et cybersécurité. Le projet a permis de concevoir un environnement cloisonné et sécurisé en appliquant une architecture en zones bien définies (DMZ In, DMZ Out, Internal Servers, etc.), garantissant ainsi un contrôle strict des flux de communication.

L'intégration de plusieurs services essentiels (Squid pour le filtrage, Apache en reverse proxy, Bind9 pour la résolution DNS et Rsyslog pour la centralisation des logs) sur une machine unique, avec un mécanisme d'alias réseau, a permis d'optimiser l'utilisation des ressources tout en maintenant une flexibilité opérationnelle. L'approche adoptée pour la gestion des interfaces et des passerelles a assuré une transition fluide entre les différentes zones, consolidant ainsi la robustesse de l'infrastructure.

Les tests de connectivité réalisés ont validé le bon fonctionnement des configurations mises en place. Toutefois, certains services, comme le proxy Squid et le reverse proxy, mériteraient une validation plus approfondie afin d'en garantir l'efficacité totale.

En somme, cette expérience a démontré la capacité à concevoir, déployer et sécuriser une infrastructure réseau conforme aux bonnes pratiques en matière de cybersécurité. Des améliorations pourraient être envisagées, notamment en automatisant certaines tâches (via des scripts Bash ou Ansible), en intégrant des solutions de monitoring avancées (comme ELK ou Zabbix) et en menant des tests de pénétration pour valider la robustesse du système face aux menaces externes.

Ce projet constitue ainsi une base solide pour des infrastructures de production sécurisées et modulables, et ouvre la voie à des déploiements encore plus complexes dans un cadre professionnel.