SAÉ 3.01B - Pare-feu réseaux

Équipe SABY

Ylian FATMI - Bayram GOKCEN -
Anthony DENISE - Samuel RIGAUD
G4A

iut du Limousin

Université de Limoges

Présentation de la SAÉ

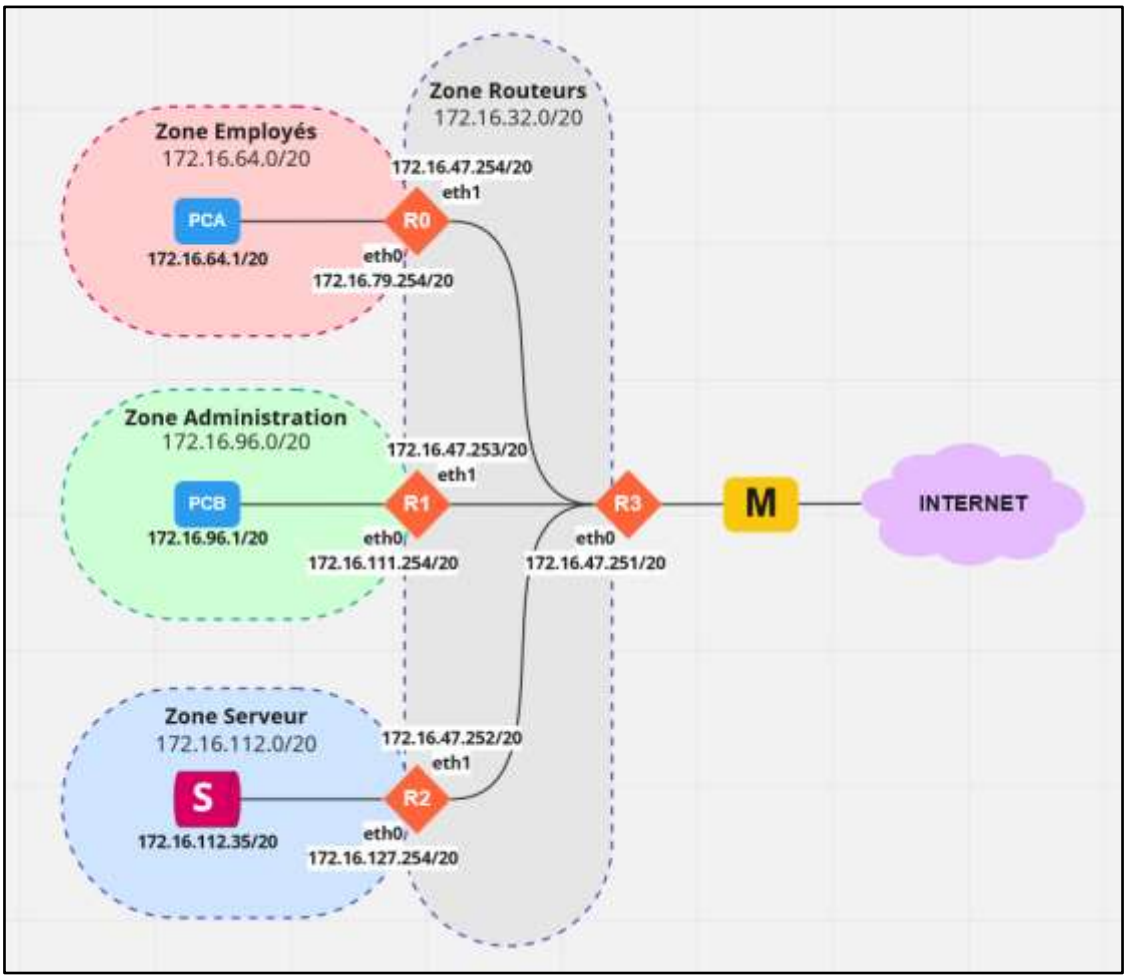
Dans cette SAÉ, nous avons comme objectif de répondre aux besoins de l'entreprise InnovT3ch, qui est une société spécialisée dans la cyber-santé. L'entreprise est divisée en plusieurs équipes qui s'occupent chacune d'un projet différent. Ces équipes doivent se déplacer constamment pour mettre en place les solutions développées pour les clients, et doivent utiliser une application pour chacun de leurs déplacements : **D-PAR**.

Les données du serveur hébergeant l'application D-PAR sont donc confidentielles et sensibles.

Notre but sera alors de simuler dans Kathará le réseau de l'entreprise grâce aux informations mises à notre disposition, puis d'en assurer la sécurité en contrôlant l'accès à internet sur les machines et en mettant en place un filtrage sur le serveur D-PAR pour protéger ses données.

Schéma de l'architecture

Pour construire notre architecture, nous devons d'abord définir les réseaux et adresses IP que nous donnerons aux machines.
Voici notre schéma du réseau après interprétation du sujet :



Configuration des réseaux

Nous pouvons donc voir sur le schéma (qui représente l'architecture réseaux) que chaque zone de travail se retrouve derrière un routeur (**R0**,**R1** et **R2**). De plus, on a un routeur **R3** qui fait le lien entre Internet et chacun des routeurs (donc chacune des zones).

Les routeurs ont comme adresse la dernière adresse de leur sous-réseau et nous avons donné à nos machines la première adresse de leur sous-réseau, ce qui nous donne :

- *Zone des employés :*
  - **R0** : 172.16.79.254/20
  - **PCA** : 172.16.64.1/20
- *Zone administration :*
  - **R1** : 172.16.111.254/20
  - **PCB** : 172.16.96.1/20
- *Zone serveur :*
  - **R2** : 172.16.127.254/20
  - **S** : 172.16.112.35/20
- *Zone des routeurs :*
  - **Adresse réseau** : 172.16.32.0/20
  - **Tous les routeurs** : (adresses sur le schéma)

Tout d'abord, nous avons configuré notre lab.conf de sorte à respecter la configuration précédente :

```
pca[0]=net0
r0[0]=net0
r0[1]=net3
pcb[0]=net1
r1[0]=net1
r1[1]=net3
s[0]=net2
r2[0]=net2
r2[1]=net3
r3[0]=net3
r3[bridged]=true
```

Zone Employés

Zone Administration

Zone Serveur

- **net3 représente la zone routeurs**

Pour chaque machine **PCA**, **PCB** et **S,** nous leur avons défini des routes par défaut vers leur routeur correspondant, afin que les machines puissent communiquer entre elles.

Finalement, on a créé les routes qui lient chaque routeur à chaque zone de travail autre que le sien, y compris **R3** pour qu'ils aient accès à Internet. (**R3** étant le lien entre tout le réseau interne et l'extérieur (la **machine hôte** donc **l'Internet**).

Configuration du filtrage

Nous avons décidé de mettre en place un pare-feu sur les routeurs qui gèrent les différents sous-réseaux (grâce à la chaîne FORWARD). Alors les flux dans les sous-réseaux sont tous filtrés.

- Le routeur **R3** gère tout le réseau (c'est donc lui qui va filtrer l'internet)
- Le routeur **R2** permet d'isoler notre serveur S.
- Les routeurs **R0** et **R1** pourraient servir à filtrer la zone employés ou administration mais il n'y a rien à faire de plus. (**R3** suffit pour ces zones)

Accès sécurisé à Internet des machines de l'entreprise

Dans un premier temps, tous les paquets qui vont vers les sous-réseaux sont bloqués (Sur le routeur **R3**, toutes les données sont **DROP** par défaut).

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP

iptables -P FORWARD DROP
```

Il s'agit de la base pour sécuriser un sous-réseau, on autorisera par la suite les données essentielles une par une (**ACCEPT**).

En commençant par un accès à Internet :

On permet alors aux paquets des protocoles essentiels de se déplacer vers les sous-réseaux : **HTTP** (port **80**) et **HTTPS** (port **443**). On permet aussi la transmission des flux des protocoles **DNS** (port **53**) qui sont presque indispensables pour naviguer sur internet (pour reconnaitre « www.google.fr » etc..).

```
iptables -I FORWARD -p tcp -m multiport --dport 80,443,53 -s 172.16.64.0/18 -m conntrack --ctstate NEW -j ACCEPT
iptables -I FORWARD -p udp --dport 53 -s 172.16.64.0/18 -m conntrack --ctstate NEW -j ACCEPT

iptables -I FORWARD -p tcp -m multiport --sport 80,443,53 -s 172.16.64.0/18 -m conntrack --ctstate NEW -j ACCEPT
iptables -I FORWARD -p udp --sport 53 -s 172.16.64.0/18 -m conntrack --ctstate NEW -j ACCEPT

iptables -I FORWARD -p tcp -m multiport --dport 80,443,53 -m conntrack --ctstate ESTABLISHED -j ACCEPT
iptables -I FORWARD -p udp --dport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT

iptables -I FORWARD -p tcp -m multiport --sport 80,443,53 -m conntrack --ctstate ESTABLISHED -j ACCEPT
iptables -I FORWARD -p udp --sport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

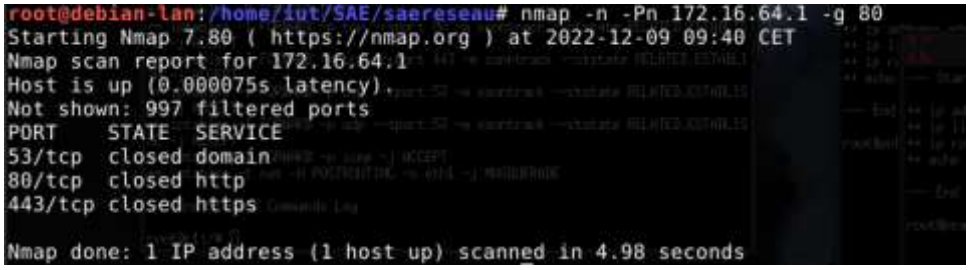
Finalement, pour configurer ce pare-feu, nous avons utilisé les outils **conntrack** qui nous permettent de sécuriser encore plus l'environnement.

En effet, nous avons configuré un pare-feu avec état, il est alors possible de n'accepter que des paquets spécifiques (déjà établi ou en relation par exemple).

Ici, SI la source est un des sous-réseaux (**172.16.64.0/18** englobe tous les sous-réseaux), on autorise la transmission des paquets (des protocoles cités ci-dessus) **MEME** s'il s'agit d'un paquet nouveau, donc les sous-réseaux peuvent parler à Internet.

Cependant, pour les autres cas, (donc si la source vient de l'extérieur) il faut que la transmission de ce paquet de données soit déjà établie, cela veut dire qu'il ne peut s'agir que de réponses ou de feed-back.

Sans état, si nous essayons de scanner le réseau au port 80 (qui est autorisé), nous le pouvons, ce qui pose un problème de sécurité.



Alors qu'avec état, il nous est impossible de scanner le sous-réseau, ce qui permet de garantir une meilleure sécurité.

On peut alors surfer de façon sécurisée sur Internet depuis les machines grâce au routeur R3.

Isolation de la machine S (routeur R2)

Pour isoler la machine S, dans un premier temps tous les paquets qui passent par R2 sont bloqués.

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP

iptables -P FORWARD DROP
```

On doit alors autoriser les paquets seulement essentiels.

Connexion à l'application (depuis port 443 de la machine S)

Le port qui gère la connexion de l'application D-PAR est le port 443, alors on autorise tous les sous-réseaux (grâce à l'adresse 172.16.64.0/18) à communiquer avec la machine S sur ce port.

```
iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --dport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --dport 443,22,25 -j ACCEPT

iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --sport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --sport 443,22,25 -j ACCEPT
```

La communication se faisant dans les deux sens, il faut autoriser l'aller mais aussi le retour.

Alors, il est possible de se connecter à l'application depuis toutes les machines du sous-réseaux 172.16.64.0/18 (donc depuis toutes les zones de travail).

Connexion au SFTP D-PAR (hébergé sur le serveur S)

Une des fonctionnalités que l'on doit implémenter est une possibilité de connexion **SFTP** sur le serveur S, permettant le transfert de fichier de manière sécurisée avec SSH, partageant le même port avec ce dernier (port **22**).

```
iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --dport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --dport 443,22,25 -j ACCEPT

iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --sport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --sport 443,22,25 -j ACCEPT
```

En permettant la transmission des flux TCP de port 22, il est possible pour les machines des sous-réseaux de se connecter en SFTP au serveur S.

Test (on n'était pas obligé de le faire, mais il fallait combler notre curiosité)

Nous avons, pour vérifier les flux, installé le serveur SFTP sur S et cela fonctionne :

Serveur S :

```
[ ok ] Starting OpenBSD Secure Shell server: sshd.
```

Machine :

```
admin@172.16.112.35's password:
Connected to admin@172.16.112.35.
sftp>
```

La connexion fonctionne (s'il y a bien un serveur SFTP sur le serveur S) !

Système de courriels (mail)

Le système de courriels est géré par le protocole SMTP qui permet d'envoyer des mails. Le port 25 est le port le plus utilisé pour cette fonctionnalité (sinon il y a les ports 25,465,587 etc... qui fonctionnent aussi)

```
iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --dport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --dport 443,22,25 -j ACCEPT

iptables -A FORWARD -p tcp -s 172.16.64.0/18 -d 172.16.112.0/20 -m multiport --sport 443,22,25 -j ACCEPT
iptables -A FORWARD -p tcp -s 172.16.112.0/20 -d 172.16.64.0/18 -m multiport --sport 443,22,25 -j ACCEPT
```

Alors, l'application peut envoyer et recevoir des mails depuis les espaces de travails.

Protocole ICMP (ping)

Une des fonctionnalités demandées est le fonctionnement du ping (le serveur doit pouvoir ping les machines), il faut donc permettre aux paquets du protocole ICMP de se déplacer dans les sous-réseaux.

```
iptables -I FORWARD -p icmp -j ACCEPT
```

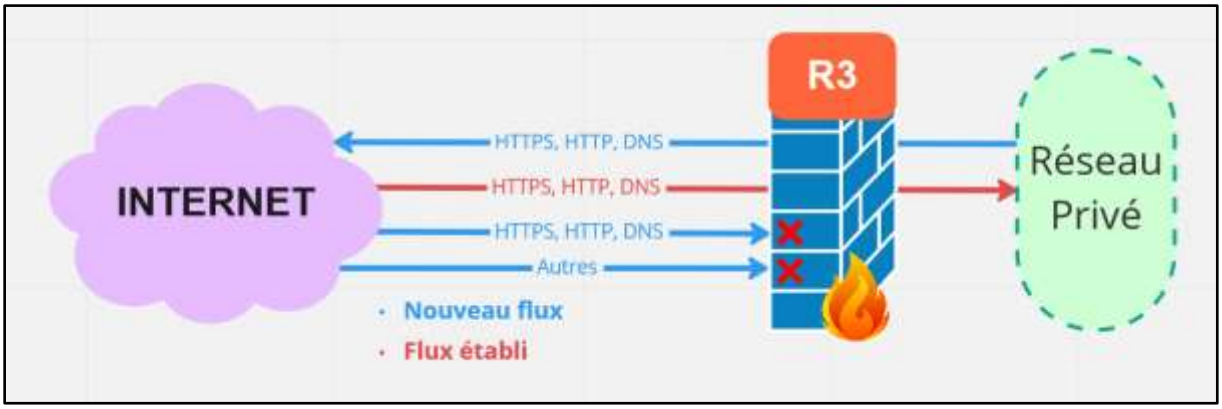
Alors il est possible de ping depuis la machine S, les machines des autres zones. Alors qu'il n'est pas possible de ping des machines extérieurs.

Finalement la machine S est bien isolée, seuls les flux les plus importants traversent le routeur R2.

Diagramme du pare-feu :

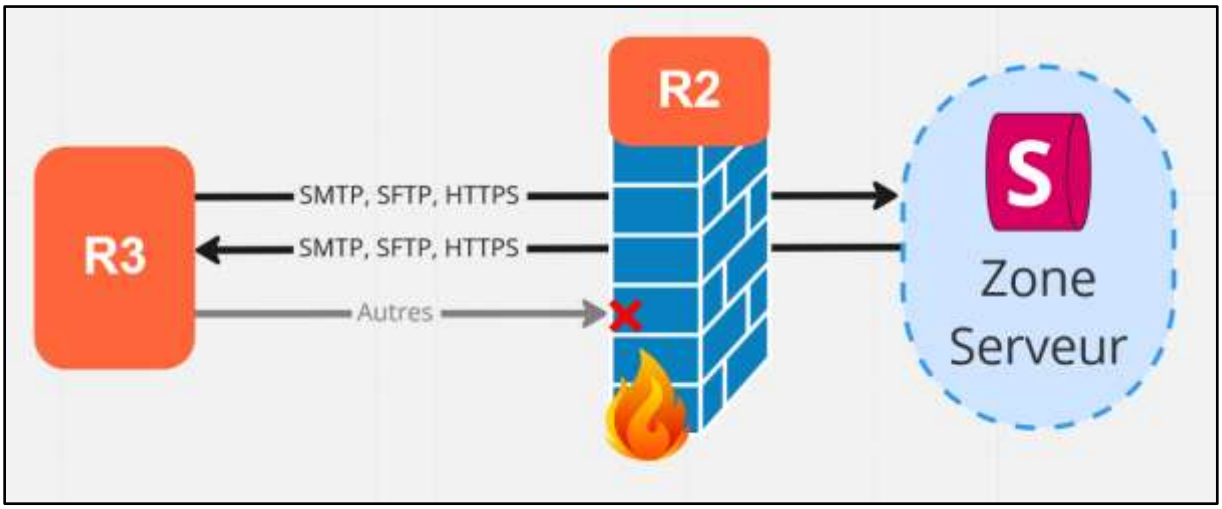
Nous pouvons montrer sous forme graphique (plus parlant) ce que notre filtrage permet de faire :

Pour le premier filtrage Internet grâce à R3 :



Ce filtrage prend donc effet sur toutes les machines du réseau (PCA,PCB et S).

Pour l'isolation de la machine S grâce à R2 :



Ce filtrage prend donc effet sur le serveur S qui doit être isolé.

Conclusion

Finalement, notre simulation permet de réaliser toutes les tâches demandées par l'entreprise InnovT3ch, tout en assurant la sécurité des informations de l'entreprise, en établissant un filtrage des flux de données et en isolant le serveur S.

Le filtrage a été réalisé de manière que les machines de l'entreprise puissent communiquer et aient un accès contrôlé à internet, grâce au routeur R3.

La machine S quant à elle est isolée dans le réseau de manière à protéger ses données tout en pouvant utiliser l'application D-PAR, grâce au routeur R2.