Table of Contents

TP2—Firewall

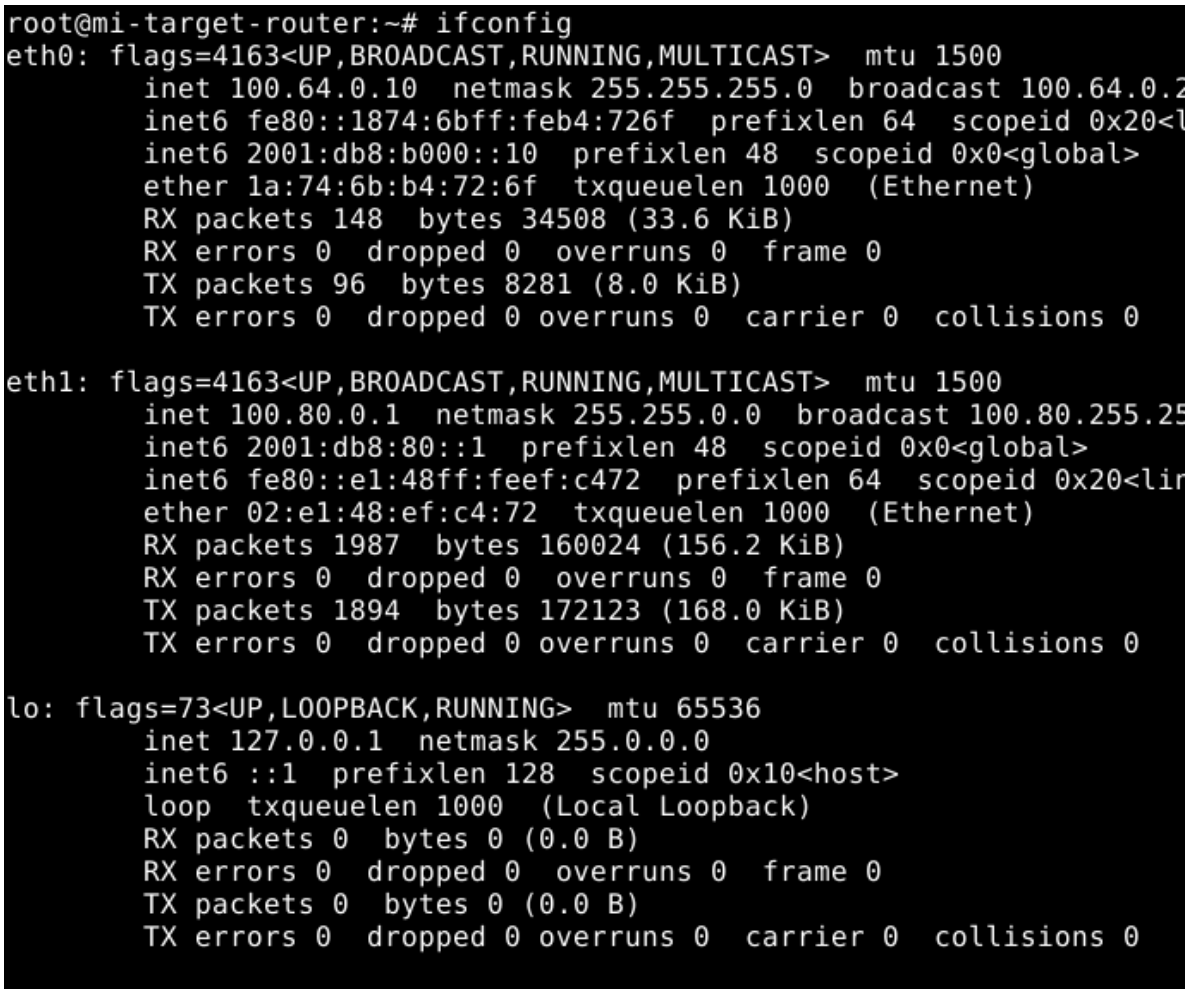
Groupe : Chafai, Hamad, Fauvart, Delanghe, Tonnerre

Topologie du routeur

Question 1

Après avoir connecté à la machine "target-router" via `./mi-lxc.py attach target-router` et examiné les interfaces avec `ip addr`, nous avons identifié:

- **eth0**: Interface externe (WAN) - 100.64.0.10/24
  - Cette interface est connectée à Internet (côté ISP)
  - Elle gère le trafic venant de l'extérieur de l'entreprise
- **eth1**: Interface interne (LAN) - 100.80.0.1/16
  - Cette interface est connectée au réseau local de l'entreprise
  - Elle gère le trafic interne de l'entreprise



ac2c7b6e c2fe 497a b977 e07ac2529ab2

Protection de la machine firewall

Question 2

Pour interdire les connexions SSH (port 22) sur la machine target-router, nous avons appliqué la règle suivante :

```
iptables -A INPUT -p tcp --dport 22 -j DROP
```

Cette règle :

- S'applique à la chaîne INPUT (trafic entrant sur le routeur)
- Filtre le protocole TCP

- Cible spécifiquement le port de destination 22 (SSH)
- Utilise l'action DROP pour ignorer silencieusement les paquets

```
root@mi-target-router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             100.80.1.2
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

9e54566e 306b 4ef6 8ab8 fd2bcac8bb24

Après l'ajout de ma règle DROP sur SSH:

```
root@mi-target-router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:ssh

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             100.80.1.2
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

385c4582 abf9 4b8f a133 9d098185186e

Question 3

Le client SSH met un certain temps à répondre car l'action DROP fait que les paquets sont simplement ignorés sans notification. Le client continue donc d'envoyer des paquets et attend une réponse jusqu'à expiration du délai (timeout). Sans réponse explicite, le client SSH doit attendre que son propre mécanisme de temporisation se déclenche.

```
root@mi-isp-a-hacker:~# ssh root@100.64.0.10
ssh: connect to host 100.64.0.10 port 22: Connection timed out
```

ae9ad2c4 abcf 4a1b a963 e81a5ad0ad63

Question 4

Après avoir remplacé DROP par REJECT :

```
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

```
root@mi-target-router:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp dpt:ssh reject-with icmp-port-unreachable

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             100.80.1.2
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

f56f1dd8 2a3c 4241 b789 66b152b20329

La différence observée est que le client SSH reçoit immédiatement une notification d'échec, sous forme d'un paquet ICMP "port unreachable" ou d'un paquet TCP RST.

En utilisant `tcpdump -i eth0 port 22`, nous observons que :

- Avec DROP : pas de réponse aux tentatives de connexion
- Avec REJECT : des paquets ICMP "port unreachable" sont renvoyés au client

Cette notification explicite permet au client de savoir immédiatement que la connexion est refusée plutôt que d'attendre un timeout.

Avec DROP:

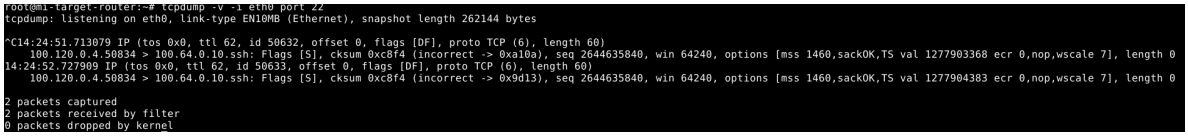
```
root@mi-target-router:~# tcpdump -i eth0 port 22
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C14:18:04.468997 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277496124 ecr 0,nop,wscale 7], length 0
14:18:05.495618 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277497151 ecr 0,nop,wscale 7], length 0
14:18:07.512050 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277499167 ecr 0,nop,wscale 7], length 0
14:18:11.671740 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277503327 ecr 0,nop,wscale 7], length 0
14:18:19.863746 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277511519 ecr 0,nop,wscale 7], length 0
14:18:35.991618 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277527647 ecr 0,nop,wscale 7], length 0
14:19:08.503663 IP 100.120.0.4.50828 > 100.64.0.10.ssh: Flags [S], seq 2824609490, win 64240, options [mss 1460,sackOK,TS val 1277560159 ecr 0,nop,wscale 7], length 0

7 packets captured
7 packets received by filter
0 packets dropped by kernel
```

5f05d7d3 5d7a 4c19 856f b3b265b81b55

```
root@mi-isp-a-hacker:~# ssh root@100.64.0.10
ssh: connect to host 100.64.0.10 port 22: Connection timed out
```

ce3ae40c 51ab 4db1 81d0 e8c2336080de

Avec REJECT:



2af73d6d 6100 4068 a8bc 2f718e2a9655

e55654f7 7d54 444f 8461 35cbd8135fcc

Priorité des règles

Question 5

Pour démontrer que l'ordre des règles est important, voici un exemple :

```
# Première règle : accepter tout le trafic SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Deuxième règle : rejeter tout le trafic SSH
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

Dans ce cas, la première règle sera appliquée et le trafic SSH sera accepté. La deuxième règle ne sera jamais atteinte pour le trafic SSH.

Si nous inversons l'ordre :

```
# D'abord supprimer les règles existantes
iptables -F INPUT

# Première règle : rejeter tout le trafic SSH
iptables -A INPUT -p tcp --dport 22 -j REJECT

# Deuxième règle : accepter tout le trafic SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Dans ce cas, tout le trafic SSH sera rejeté car la première règle sera appliquée et la seconde ne sera jamais atteinte.

Si on accepte puis rejette:

```
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -j REJECT
```

d8964e06 8d95 43e1 8432 aa6f4d460a63

```
root@mi-isp-a-hacker:~# ssh root@100.64.0.10
root@100.64.0.10's password:
Linux mi-target-router 5.10.0-14-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 22 13:59:22 2025 from 100.120.0.4
root@mi-target-router:~#
```

7d693f97 9c1b 41c4 9079 70deb65aa04c

Si on rejette puis accepte:

```
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -j REJECT
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

e864852a bea3 4f0b b6e9 d4f60a9d1d13

```
root@mi-isp-a-hacker:~# ssh root@100.64.0.10
ssh: connect to host 100.64.0.10 port 22: Connection refused
```

2fea9754 80cb 4d29 b762 af83d430d41b

Question 6

Pour autoriser SSH sur le routeur uniquement depuis le LAN interne, nous avons mis en place ces règles :

```
# Nettoyer les règles existantes
iptables -F INPUT

# Autoriser SSH depuis le réseau interne (100.80.0.0/16)
iptables -A INPUT -p tcp --dport 22 -s 100.80.0.0/16 -j ACCEPT

# Rejeter toutes les autres tentatives SSH
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

Ces règles permettent aux machines du réseau interne (comme target-admin) d'accéder au routeur via SSH, tout en bloquant les tentatives depuis l'extérieur (comme depuis isp-a-hacker).

Règles appliquées:

```
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -s 100.80.0.0/16 -j ACCEPT
root@mi-target-router:~# iptables -A INPUT -p tcp --dport 22 -j REJECT
```

71e712c8 1121 4d58 b325 a4d2b20265d8

Test de connexion SSH depuis le réseau interne (succès):

```
root@mi-target-admin:~# ssh root@100.64.0.10
root@100.64.0.10's password:
Linux mi-target-router 5.10.0-14-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 22 14:46:41 2025 from 100.80.0.4
root@mi-target-router:~#
```

3ef82730 8fa8 4974 acf8 d5cb1d5a4b3c

Test de connexion SSH depuis l'extérieur (échec):

```
root@milxc-vm:~/mi-lxc# ./mi-lxc.py attach isp-a-hacker

  Attaching to isp-a-hacker as user root

root@mi-isp-a-hacker:~# ssh root@100.64.0.10
ssh: connect to host 100.64.0.10 port 22: Connection refused
```

5444ec94 8082 44c0 a243 ea051d7e0dd0

Modules iptables

Question 7

Pour autoriser uniquement les réponses aux connexions SSH entrantes, après avoir défini la politique par défaut de OUTPUT à DROP :

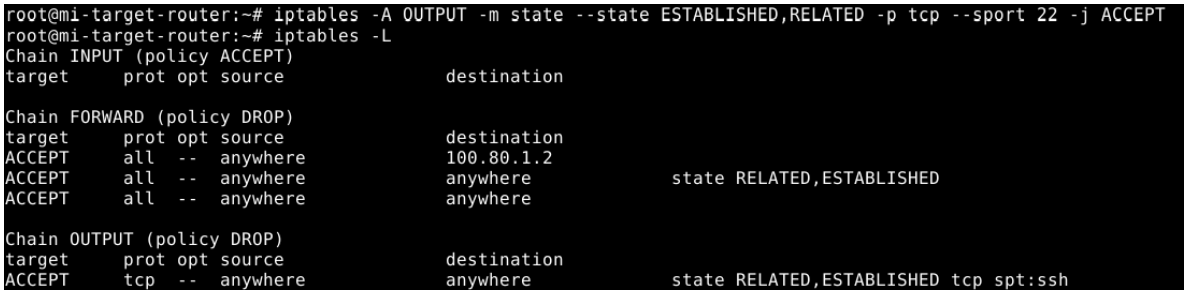
```
iptables -P OUTPUT DROP
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -p tcp --sport
22 -j ACCEPT
```

Cette règle :

- Utilise le module "state" pour identifier l'état des connexions
- Autorise uniquement les paquets correspondant à des connexions déjà établies (ESTABLISHED) ou liées (RELATED)
- S'applique uniquement au protocole TCP
- Filtre sur le port source 22 (réponses du service SSH)
- Permet au trafic correspondant de sortir (ACCEPT)

Ainsi, seules les réponses aux connexions SSH entrantes seront autorisées à sortir du firewall, tandis que les nouvelles connexions sortantes sont bloquées.

Application des règles avec module state:



283aba46 6bed 4ce5 a7f1 560fa53ec95f

Test montrant que seules les réponses SSH sont autorisées:

```
root@mi-target-admin:~# ssh root@100.64.0.10
root@100.64.0.10's password:
Linux mi-target-router 5.10.0-14-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 22 15:08:02 2025 from 100.80.0.4
root@mi-target-router:~# ping 8.8.8.8
ping: connect: Network is unreachable
root@mi-target-router:~#
```

f7c0acd4 5593 4c47 86ad f239c62eda87

```
root@mi-target-router:~# iptables -L OUTPUT -v
Chain OUTPUT (policy DROP 546 packets, 44566 bytes)
 pkts bytes target     prot opt in     out     source               destination

  304 44032 ACCEPT     tcp  --  any    any     anywhere             anywhere
        state RELATED,ESTABLISHED tcp spt:ssh
root@mi-target-router:~# iptables -L OUTPUT -v
Chain OUTPUT (policy DROP 549 packets, 44746 bytes)
 pkts bytes target     prot opt in     out     source               destination

  321 48008 ACCEPT     tcp  --  any    any     anywhere             anywhere
        state RELATED,ESTABLISHED tcp spt:ssh
root@mi-target-router:~# iptables -L OUTPUT -v
Chain OUTPUT (policy DROP 549 packets, 44746 bytes)
 pkts bytes target     prot opt in     out     source               destination

  351 53076 ACCEPT     tcp  --  any    any     anywhere             anywhere
        state RELATED,ESTABLISHED tcp spt:ssh
```

4e15b08b c009 4a59 b214 0aaeb5d2ba09

Mise en place d'une politique de sécurité réseau

Question 8

Matrice de flux pour le SI de l'entreprise, basée sur l'analyse des services actifs :

Matrice de flux par zones

Source \ Destination	Internet	DMZ	LAN	ADMIN	SERVICES
Internet	X	HTTP, HTTPS	X	X	X
DMZ	HTTP, HTTPS	X	X	X	LDAP (→ target-ldap uniquement)
LAN	HTTP, HTTPS (→ Internet)	HTTP, HTTPS (→ DMZ)	X	SSH (→ ADMIN)	LDAP, FTP, HTTP (→ intranet)
ADMIN	HTTP, HTTPS (→ Internet)	SSH, HTTP, HTTPS (→ DMZ)	SSH, FTP (→ LAN)	X	LDAP, SSH, FTP, HTTP (→ SERVICES)
SERVICES	X	X	X	SSH (← ADMIN)	X

Répartition des machines par zone

- **Internet**: Extérieur (machines de test comme `isp-a-hacker`, `isp-a-home`)
- **DMZ**: `target-dmz` (100.80.1.2)
- **LAN**: `target-commercial` (100.80.0.2), `target-dev` (100.80.0.3)
- **ADMIN**: `target-admin` (100.80.2.2)
- **SERVICES**: `target-ldap` (100.80.3.2), `target-filer` (100.80.3.3), `target-intranet` (100.80.3.4)

Légende des services

- SSH : TCP/22
- HTTP : TCP/80
- HTTPS : TCP/443
- DNS : UDP/53, TCP/53
- SMTP : TCP/25
- IMAP : TCP/143
- IMAPS : TCP/993
- LDAP : TCP/389
- SMB/CIFS : TCP/445
- FTP : TCP/21 (et TCP/20, ports passifs)

Cette matrice se base sur l'analyse des services en cours d'exécution sur chaque machine (commande `netstat -laptn`). On remarque notamment :

- La DMZ expose plusieurs services vers Internet (HTTP, HTTPS, DNS, IMAP, SMTP)
- Le serveur LDAP est accessible depuis toutes les machines internes mais pas depuis Internet
- L'administrateur (target-admin) a accès à toutes les machines du réseau
- Le développeur (target-dev) doit pouvoir accéder au serveur intranet pour les déploiements
- Les communications entre zones sont strictement limitées aux services nécessaires

Question 9

Pour segmenter le réseau et implémenter la politique de sécurité :

1. **Segmentation du réseau**:

   Modification du fichier `global.json`:

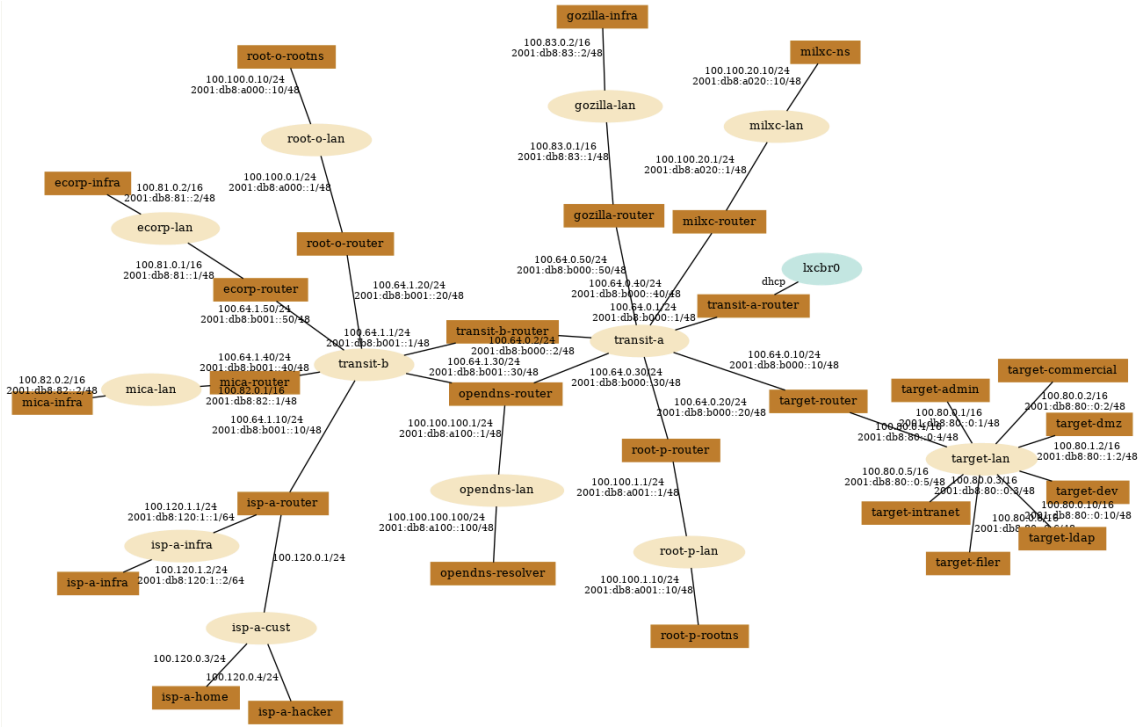
```
{
  "target": {
    "interfaces": [
      {"bridge": "transit-a", "ip": "100.64.0.10/24", "gw":
"100.64.0.1"},
      {"bridge": "target-lan", "ip": "100.80.0.1/24"},
      {"bridge": "target-dmz", "ip": "100.80.1.1/24"},
      {"bridge": "target-admin", "ip": "100.80.2.1/24"},
      {"bridge": "target-services", "ip": "100.80.3.1/24"}
    ],
    "asdev": "eth0;eth1;eth2;eth3;eth4"
  }
}
```

Modification du fichier `groups/target/local.json` pour adapter les interfaces des machines internes :

```
"target-admin": {
  "interfaces": [
    {"bridge": "admin", "ip": "100.80.2.2/24", "gw": "100.80.2.1"}
  ]
},
"target-commercial": {
  "interfaces": [
    {"bridge": "lan", "ip": "100.80.0.2/24", "gw": "100.80.0.1"}
  ]
},
"target-dev": {
  "interfaces": [
    {"bridge": "lan", "ip": "100.80.0.3/24", "gw": "100.80.0.1"}
  ]
},
"target-dmz": {
  "interfaces": [
    {"bridge": "dmz", "ip": "100.80.1.2/24", "gw": "100.80.1.1"}
  ]
},
```

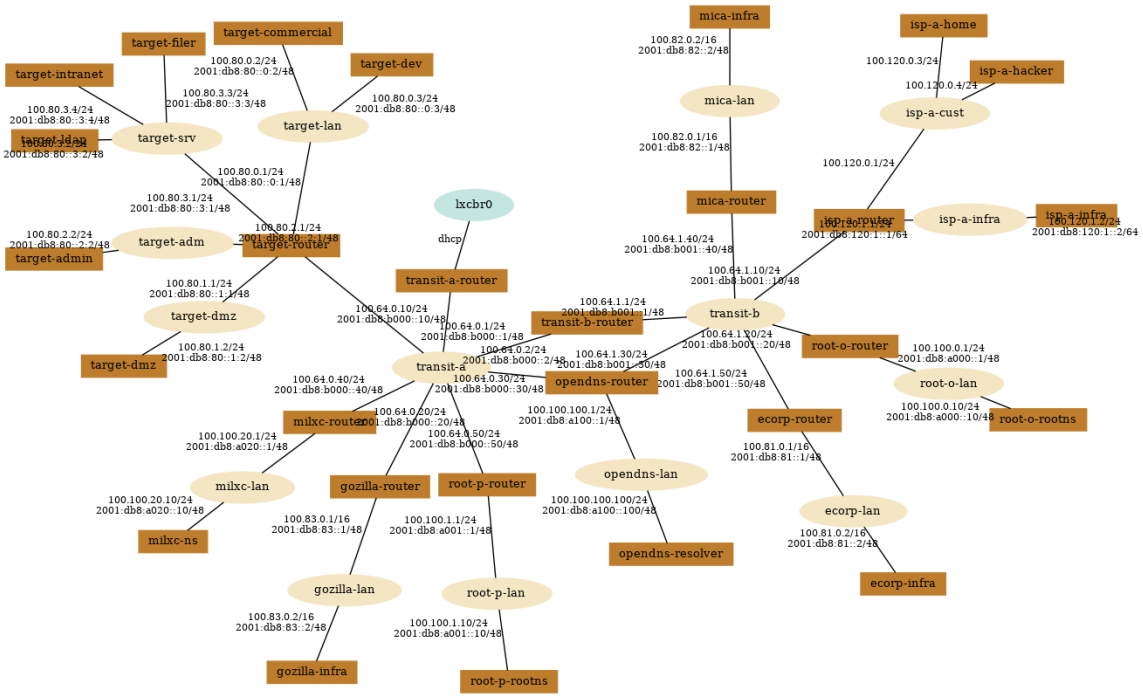
```
"target-ldap": {
  "interfaces": [
    {"bridge": "services", "ip": "100.80.3.2/24", "gw": "100.80.3.1"}
  ]
},
"target-filer": {
  "interfaces": [
    {"bridge": "services", "ip": "100.80.3.3/24", "gw": "100.80.3.1"}
  ]
},
"target-intranet": {
  "interfaces": [
    {"bridge": "services", "ip": "100.80.3.4/24", "gw": "100.80.3.1"}
  ]
}
```

Sortie de la commande `./mi-lxc.py print` montrant l'ancienne topologie



38958ad2 9ddb 4a49 9588 b67fcce00f12

Sortie de la commande `./mi-lxc.py print` montrant la nouvelle topologie



76de94b9 d1e0 4634 9bbe 68b3f14746ea

2. **Script de règles iptables:**

```
#!/bin/bash

# Nettoyage des règles existantes
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

# Politiques par défaut
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Autoriser le trafic loopback
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A OUTPUT -o lo -j ACCEPT

# Autoriser les connexions établies et liées
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Autoriser SSH depuis le réseau admin uniquement
iptables -A INPUT -p tcp -s 100.80.2.0/24 --dport 22 -j ACCEPT

# Règles de routage entre les zones
# Internet vers DMZ
iptables -A FORWARD -i eth0 -o eth2 -p tcp -m multiport --dports
80,443,25,143,21 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth2 -p udp --dport 53 -j ACCEPT

# DMZ vers Internet
iptables -A FORWARD -i eth2 -o eth0 -p tcp -m multiport --dports
80,443,53 -j ACCEPT
iptables -A FORWARD -i eth2 -o eth0 -p udp --dport 53 -j ACCEPT

# DMZ vers Services (uniquement LDAP)
iptables -A FORWARD -i eth2 -o eth4 -d 100.80.3.2 -p tcp --dport 389 -j
ACCEPT

# LAN vers DMZ
iptables -A FORWARD -i eth1 -o eth2 -p tcp -m multiport --dports 80,443
-j ACCEPT

# LAN vers Services
iptables -A FORWARD -i eth1 -o eth4 -p tcp -m multiport --dports
389,445,80,443 -j ACCEPT

# Admin vers tous les réseaux
iptables -A FORWARD -i eth3 -j ACCEPT

# Dev vers Intranet (SSH pour déploiement)
iptables -A FORWARD -i eth1 -o eth4 -s 100.80.0.3 -d 100.80.3.4 -p tcp -
-dport 22 -j ACCEPT
```

```
# Services vers Services (communication interne)
iptables -A FORWARD -i eth4 -o eth4 -j ACCEPT

# Activer le masquerading (NAT) pour permettre aux machines internes
d'accéder à Internet
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Journalisation des paquets rejetés
iptables -A FORWARD -j LOG --log-prefix "IPTABLES FORWARD REJECT: "
```

Sortie de la commande `iptables-save` montrant les règles appliquées:

```
root@mi-target-router:~# ./rules-iptables.sh
root@mi-target-router:~# iptables-save
# Generated by iptables-save v1.8.7 on Tue Apr 22 16:27:15 2025
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 100.80.2.0/24 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i eth0 -o eth2 -p tcp -m multiport --dports 80,443,25,143,21 -j ACCEPT
-A FORWARD -i eth0 -o eth2 -p udp -m udp --dport 53 -j ACCEPT
-A FORWARD -i eth2 -o eth0 -p tcp -m multiport --dports 80,443,53 -j ACCEPT
-A FORWARD -i eth2 -o eth0 -p udp -m udp --dport 53 -j ACCEPT
-A FORWARD -d 100.80.3.2/32 -i eth2 -o eth4 -p tcp -m tcp --dport 389 -j ACCEPT
-A FORWARD -i eth1 -o eth2 -p tcp -m multiport --dports 80,443 -j ACCEPT
-A FORWARD -i eth1 -o eth4 -p tcp -m multiport --dports 389,445,80,443 -j ACCEPT
-A FORWARD -i eth3 -j ACCEPT
-A FORWARD -s 100.80.0.3/32 -d 100.80.3.4/32 -i eth1 -o eth4 -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -i eth4 -o eth4 -j ACCEPT
-A FORWARD -j LOG --log-prefix "IPTABLES FORWARD REJECT: "
-A OUTPUT -o lo -j ACCEPT
COMMIT
# Completed on Tue Apr 22 16:27:15 2025
# Generated by iptables-save v1.8.7 on Tue Apr 22 16:27:15 2025
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Tue Apr 22 16:27:15 2025
```

6097382f aaf7 436d 950c d9a835e0d3b4

Tests de connectivité entre les différentes zones:

- target-ldap :

```
root@mi-target-ldap:~# ldapsearch -x -h 100.80.3.2 -b "dc=target,dc=milxc"
# extended LDIF
#
# LDAPv3
# base <dc=target,dc=milxc> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# target.milxc
dn: dc=target,dc=milxc
objectClass: top
objectClass: dcObject
objectClass: organization
o: target.milxc
dc: target

# People, target.milxc
dn: ou=People,dc=target,dc=milxc
objectClass: organizationalUnit
objectClass: top
ou: People

# Group, target.milxc
dn: ou=Group,dc=target,dc=milxc
objectClass: organizationalUnit
objectClass: top
ou: Group

# employees, Group, target.milxc
dn: cn=employees,ou=Group,dc=target,dc=milxc
objectClass: top
objectClass: posixGroup
gidNumber: 1001
cn: employees

# mail, Group, target.milxc
dn: cn=mail,ou=Group,dc=target,dc=milxc
objectClass: top
objectClass: posixGroup
gidNumber: 8
cn: mail
memberUid: commercial

# commercial, People, target.milxc
dn: uid=commercial,ou=People,dc=target,dc=milxc
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: commercial
uid: commercial
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/commercial
loginShell: /bin/bash
gecos: commercial
```

3886f6a4 d707 4c92 b8fd b89e1e0eceb0

```
root@mi-target-ldap:~# curl http://100.80.3.4
<html><head><meta charset="UTF-8"><title>Événements à venir</title></head>

<body>

        Clients :<br>

        Fichier de clients

        <br><br>Compta :<br>
        Fichier de comptes
<br><br>
Événements à venir :

<form action="index.php" method="POST">
Filtre : <input type=text name=filter value=""></input><input type=submit value="Filtrer"></input></form>


<table border=1>
<tr><th>Nom de la formation</th><th>Participants</th></tr>

<tr><td>Sortie ski</td><td>Achille, Manon</td></tr><tr><td>Cinema</td><td>Eric, MC</td></tr></table>

<br>
Hints :<br> La requête SQL était : SELECT name,participants FROM events<br>Cette requête a été exécutée avec succès !</body>
</html>
```

a794577b 4f33 4395 99d4 953f6c27f436

- target-dev :

```
root@mi-target-dev:~# curl http://100.80.3.4
<html><head><meta charset="UTF-8"><title>Événements à venir</title></head>

<body>

        Clients :<br>

        Fichier de clients

        <br><br>Compta :<br>
        Fichier de comptes
<br><br>
Événements à venir :

<form action="index.php" method="POST">
Filtre : <input type=text name=filter value=""></input><input type=submit value="Filtrer"></input></form>


<table border=1>
<tr><th>Nom de la formation</th><th>Participants</th></tr>

<tr><td>Sortie ski</td><td>Achille, Manon</td></tr><tr><td>Cinema</td><td>Eric, MC</td></tr></table>

<br>
Hints :<br> La requête SQL était : SELECT name,participants FROM events<br>Cette requête a été exécutée avec succès !</body>
</html>
root@mi-target-dev:~# ping -c 2 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1013ms
```

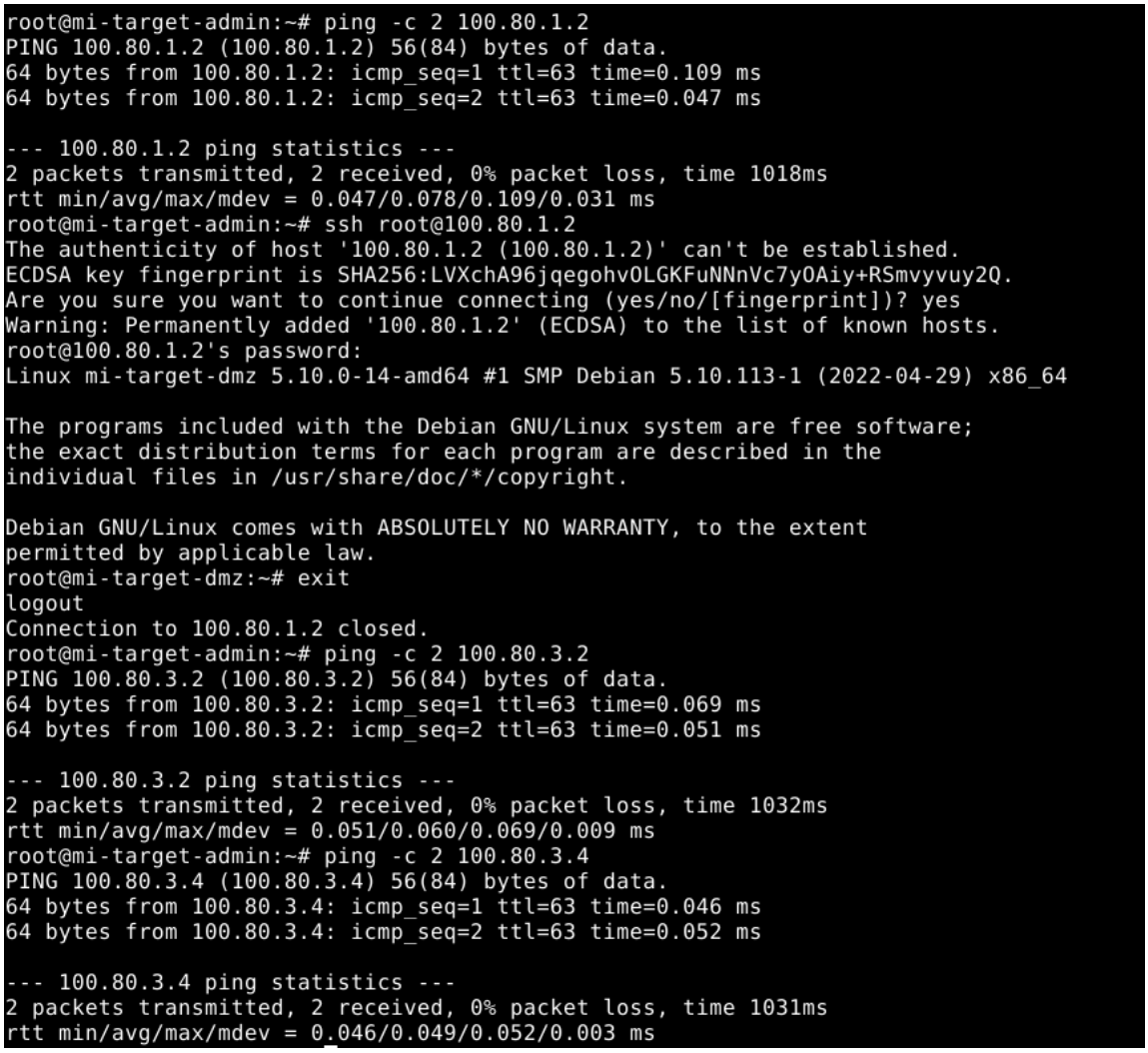
aea51999 c148 479c b2fa af264ddfd6d0

- isp-a-hacker :

```
root@mi-isp-a-hacker:~# ping -c 2 100.80.1.2
PING 100.80.1.2 (100.80.1.2) 56(84) bytes of data.

--- 100.80.1.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1011ms

root@mi-isp-a-hacker:~# curl -v http://100.80.1.2
*   Trying 100.80.1.2:80...
* Connected to 100.80.1.2 (100.80.1.2) port 80 (#0)
> GET / HTTP/1.1
> Host: 100.80.1.2
> User-Agent: curl/7.74.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 302 Found
< Date: Wed, 23 Apr 2025 06:05:50 GMT
< Server: Apache/2.4.53 (Debian)
< Location: doku.php
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 100.80.1.2 left intact
root@mi-isp-a-hacker:~# ping -c 2 100.80.0.2
PING 100.80.0.2 (100.80.0.2) 56(84) bytes of data.

--- 100.80.0.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1022ms

root@mi-isp-a-hacker:~# ping -c 2 100.80.2.2
PING 100.80.2.2 (100.80.2.2) 56(84) bytes of data.

--- 100.80.2.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1008ms

root@mi-isp-a-hacker:~# ping -c 2 100.80.3.2
PING 100.80.3.2 (100.80.3.2) 56(84) bytes of data.

--- 100.80.3.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1025ms
```

d04dd7f6 0f49 4bc2 8ca8 aded8e7e5cce

- target-admin :



90d200d5 0d85 4479 81e6 ab2686486849

Après avoir créé ce script, nous l'avons exécuté et avons vérifié que les règles étaient correctement appliquées avec `iptables-save`. Nous avons également testé les connexions pour confirmer que notre politique fonctionnait comme prévu.

Contournement de la politique

Question 10

Le tunnel netcat entre target-dev et isp-a-home fonctionne selon le schéma suivant :

```
External Client        isp-a-home              target-dev
target-intranet
(isp-a-hacker)         (100.120.0.3)           (100.80.0.3)
```

```
(100.80.0.5)
      |                      |                        |
|
      | HTTP Request         |                        |
|
      | (port 8080)          |                        |
|
      |--------------------->|                        |
|
      |                      | forwarded via          |
|
      |                      | netcat tunnel          |
|
      |                      |----------------------->|
|
      |                      |                        | HTTP Request
|
      |                      |                        |-----------------
----->|
      |                      |                        |
|
      |                      |                        | HTTP Response
|
      |                      |                        |<----------------
------|
      |                      | forwarded via          |
|
      |                      | netcat tunnel          |
|
      |                      |<-----------------------|
|
      | HTTP Response        |                        |
|
      | (port 8080)          |                        |
|
      |<---------------------|                        |
|
```

Configuration du tunnel netcat sur isp-a-home:

```
root@mi-isp-a-home:~# while true; do nc -v -l -p 80 -c "nc -l -p 8080"; done
listening on [any] 80 ...
100.120.0.4: inverse host lookup failed: Unknown host
connect to [100.120.0.3] from (UNKNOWN) [100.120.0.4] 59224
listening on [any] 80 ...
100.64.0.10: inverse host lookup failed: Unknown host
connect to [100.120.0.3] from (UNKNOWN) [100.64.0.10] 34924
listening on [any] 80 ...
```

e09370e8 2dce 421b b013 d530f6a70053

Configuration du tunnel netcat sur target-dev:

```
root@mi-target-dev:~# while true; do nc -v 100.120.0.3 80 -c "nc 100.80.3.4 80";
 sleep 2; done
100.120.0.3: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [100.120.0.3] 80 (http) open
100.120.0.3: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [100.120.0.3] 80 (http) open
100.120.0.3: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [100.120.0.3] 80 (http) open
100.120.0.3: inverse host lookup failed: Unknown host
(UNKNOWN) [100.120.0.3] 80 (http) open
100.120.0.3: inverse host lookup failed: Host name lookup failure
(UNKNOWN) [100.120.0.3] 80 (http) open
```

2944cbc6 e058 4958 88a0 4ce5059d43e0

Accès au serveur intranet depuis isp-a-hacker via le tunnel:

```
root@mi-isp-a-hacker:~# curl http://100.120.0.3:8080
<html><head><meta charset="UTF-8"><title>Événements à venir</title></head>

<body>

        Clients :<br>

        Fichier de clients

        <br><br>Compta :<br>
        Fichier de comptes
<br><br>
Événements à venir :

<form action="index.php" method="POST">
Filtre : <input type=text name=filter value=""></input><input type=submit value="Filtrer"></input></form>


<table border=1>
<tr><th>Nom de la formation</th><th>Participants</th></tr>

<tr><td>Sortie ski</td><td>Achille, Manon</td></tr><tr><td>Cinema</td><td>Eric, MC</td></tr></table>

<br>
Hints :<br> La requête SQL était : SELECT name,participants FROM events<br>Cette requête a été exécutée avec succès !</body>
</html>
```

b80e581f e5e0 4f81 b474 96587f8b2545

Ce tunnel contourne la politique de sécurité parce que :

1. La connexion sortante depuis target-dev vers isp-a-home est autorisée par la politique de filtrage (le développeur peut accéder à Internet)

2. Une fois ce tunnel établi, il crée un canal de communication qui n'est pas inspecté par le firewall
3. Tout le trafic passant par ce tunnel est encapsulé dans la connexion autorisée
4. Le firewall ne voit qu'une connexion TCP normale entre target-dev et isp-a-home, sans pouvoir inspecter le contenu

Ce type de contournement est difficile à détecter car :

- Il utilise des ports autorisés
- Il n'utilise pas de protocoles facilement identifiables
- Le trafic peut être chiffré (avec SSH par exemple)
- Il ressemble à une connexion légitime

Pour se protéger contre ce type d'attaque, il faudrait :

- Limiter strictement les connexions sortantes
- Utiliser une inspection approfondie des paquets (DPI)
- Monitorer les connexions prolongées ou inhabituelles
- Mettre en place des solutions EDR sur les postes clients

Bonus

FTP

Pour permettre l'usage du protocole FTP depuis l'extérieur vers le serveur FTP de la DMZ, nous avons ajouté les règles suivantes :

```
# Autoriser le port FTP contrôle
iptables -A FORWARD -i eth0 -o eth2 -p tcp --dport 21 -j ACCEPT

# Autoriser le port FTP données (mode actif)
iptables -A FORWARD -i eth0 -o eth2 -p tcp --dport 20 -j ACCEPT
```

```
# Autoriser le mode passif (ports éphémères)
iptables -A FORWARD -i eth0 -o eth2 -p tcp --dport 1024:65535 -m state --state RELATED -j ACCEPT
```

Ajout des règles FTP:

```
root@mi-target-router:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  100.80.2.0/24        anywhere             tcp dpt:ssh

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             multiport dports http,https,smtp,imap2,ftp
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere             multiport dports http,https,domain
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain
ACCEPT     tcp  --  anywhere             100.80.3.2           tcp dpt:ldap
ACCEPT     tcp  --  anywhere             anywhere             multiport dports http,https
ACCEPT     tcp  --  anywhere             anywhere             multiport dports ldap,microsoft-ds,http,https
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  100.80.0.3           100.80.3.4           tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
LOG        all  --  anywhere             anywhere             LOG level warning prefix "IPTABLES FORWARD REJECT: "
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ftp-data
ACCEPT     tcp  --  anywhere             anywhere             tcp dpts:1024:65535 state RELATED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

4941c4f7 3d5a 4fa1 ba27 3663c6841740

Test de connexion FTP depuis l'extérieur:

```
root@mi-isp-a-hacker:~# ftp -nv 100.80.1.2
Connected to 100.80.1.2.
220 ProFTPD Server (Debian) [::ffff:100.80.1.2]
```

77410063 ea9c 454d a620 febc2b0aab60

Capture tcpdump montrant le trafic FTP passant par le firewall:

```
root@mi-target-router:~# tcpdump -i eth0 -nn -s 0 -vv \
  'tcp port 21 or tcp port 20 or (tcp portrange 1024-65535)'
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C09:56:42.828907 IP (tos 0xc0, ttl 1, id 38198, offset 0, flags [DF], proto TCP (6), length 71)
    100.64.0.10.59593 > 100.64.0.1.179: Flags [P.], cksum 0xc8c4 (incorrect -> 0xcb6a), seq 2019041400:2019041419, ack 1535947582, win 501, options [nop,nop,TS val 1173159178 ecr 427683607],
 length 19: BGP
        Keepalive Message (4), length: 19
09:56:42.829130 IP (tos 0xc0, ttl 1, id 13304, offset 0, flags [DF], proto TCP (6), length 52)
    100.64.0.1.179 > 100.64.0.10.59593: Flags [.], cksum 0xc8b1 (incorrect -> 0x8316), seq 1, ack 19, win 510, options [nop,nop,TS val 427703165 ecr 1173159178], length 0
09:56:51.096267 IP6 (class 0xc0, flowlabel 0xb6c55, hlim 1, next-header TCP (6) payload length: 51) 2001:db8:b000::1.179 > 2001:db8:b000::10.45453: Flags [P.], cksum 0xbbbd (incorrect -> 0x1
1ef), seq 1571045218:1571045237, ack 1604305418, win 503, options [nop,nop,TS val 3302190698 ecr 4217963404], length 19: BGP
        Keepalive Message (4), length: 19
09:56:51.096286 IP6 (class 0xc0, flowlabel 0xe0cf1, hlim 1, next-header TCP (6) payload length: 32) 2001:db8:b000::10.45453 > 2001:db8:b000::1.179: Flags [.], cksum 0xbbaa (incorrect -> 0xcf
09), seq 1, ack 19, win 507, options [nop,nop,TS val 421798144 8 ecr 3302190698], length 0
09:57:00.335863 IP (tos 0x0, ttl 62, id 3587, offset 0, flags [DF], proto TCP (6), length 60)
    100.120.0.4.49116 > 100.80.1.2.21: Flags [S], cksum 0xc9fc (incorrect -> 0x7423), seq 961476366, win 64240, options [mss 1460,sackOK,TS val 533462787 ecr 0,nop,wscale 7], length 0
09:57:00.335902 IP (tos 0x0, ttl 63, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    100.80.1.2.21 > 100.120.0.4.49116: Flags [S.], cksum 0xc9fc (incorrect -> 0xbfd9), seq 707274490, ack 961476367, win 65160, options [mss 1460,sackOK,TS val 1384975601 ecr 533462787,nop,w
scale 7], length 0
09:57:00.335921 IP (tos 0x0, ttl 62, id 3588, offset 0, flags [DF], proto TCP (6), length 52)
    100.120.0.4.49116 > 100.80.1.2.21: Flags [.], cksum 0xc9f4 (incorrect -> 0xeb38), seq 1, ack 1, win 502, options [nop,nop,TS val 533462787 ecr 1384975601], length 0
09:57:15.345874 IP (tos 0x0, ttl 63, id 15062, offset 0, flags [DF], proto TCP (6), length 101)
    100.80.1.2.21 > 100.120.0.4.49116: Flags [P.], cksum 0xca25 (incorrect -> 0xb9c5), seq 1:50, ack 1, win 510, options [nop,nop,TS val 1384990611 ecr 533462787], length 49: FTP, length: 49
        220 ProFTPD Server (Debian) [::ffff:100.80.1.2]
09:57:15.345965 IP (tos 0x10, ttl 62, id 3589, offset 0, flags [DF], proto TCP (6), length 52)
    100.120.0.4.49116 > 100.80.1.2.21: Flags [.], cksum 0xc9f4 (incorrect -> 0x75c3), seq 1, ack 50, win 502, options [nop,nop,TS val 533477797 ecr 1384990611], length 0

9 packets captured
9 packets received by filter
0 packets dropped by kernel
```

b95d0d4c 0585 42a9 97e4 91ce2a5f5f15

Shorewall

Pour implémenter notre politique avec Shorewall, nous avons:

1. Installé Shorewall : `apt-get install shorewall`
2. Configuré les fichiers de base dans `/etc/shorewall/`:

zones:

```
fw      firewall
net     ipv4
lan     ipv4
dmz     ipv4
admin   ipv4
srv     ipv4
```

interfaces:

```
net     eth0    -
lan     eth1    -
dmz     eth2    -
admin   eth3    -
srv     eth4    -
```

policy:

```
fw      all     ACCEPT
net     all     DROP    INFO
lan     net     ACCEPT
lan     dmz     ACCEPT
lan     srv     ACCEPT
dmz     net     ACCEPT
dmz     srv     ACCEPT  INFO
admin   all     ACCEPT
srv     srv     ACCEPT
all     all     DROP    INFO
```

rules:

```
# (1) SSH vers le firewall depuis admin uniquement
SSH(ACCEPT)   admin      fw

# (2) Accès externes vers la DMZ
HTTP(ACCEPT)   net        dmz
HTTPS(ACCEPT)  net        dmz
DNS(ACCEPT)    net        dmz
SMTP(ACCEPT)   net        dmz
IMAP(ACCEPT)   net        dmz
FTP(ACCEPT)    net        dmz

# (3) DMZ vers Internet
HTTP(ACCEPT)   dmz        net
HTTPS(ACCEPT)  dmz        net
DNS(ACCEPT)    dmz        net

# (4) DMZ vers Services (uniquement LDAP sur 100.80.3.2)
ACCEPT         dmz        srv:100.80.3.2    tcp     389

# (5) LAN vers DMZ
ACCEPT         lan        dmz               tcp     80,443

# (6) LAN vers Services
ACCEPT         lan        srv               tcp     389,445,80,443

# (7) Dev (100.80.0.3) vers Intranet (100.80.3.4) en SSH
ACCEPT         lan:100.80.0.3  srv:100.80.3.4  tcp     22

# (8) Services interne à interne
ACCEPT         srv        srv

# (9) (optionnel) journalisation des rejets
# LOG            all        all
```

Démarrage de Shorewall et vérification du statut:

```
root@mi-target-router:/etc/shorewall# shorewall restart
Compiling using Shorewall 5.2.3.4...
Processing /etc/shorewall/params ...
Processing /etc/shorewall/shorewall.conf...
Compiling /etc/shorewall/zones...
Compiling /etc/shorewall/interfaces...
Determining Hosts in Zones...
Locating Action Files...
Compiling /etc/shorewall/policy...
Compiling TCP Flags filtering...
Compiling Kernel Route Filtering...
Compiling Martian Logging...
Compiling /etc/shorewall/snat...
Compiling MAC Filtration -- Phase 1...
Compiling /etc/shorewall/rules...
Compiling /etc/shorewall/conntrack...
Compiling MAC Filtration -- Phase 2...
Applying Policies...
Generating Rule Matrix...
Optimizing Ruleset...
Creating iptables-restore input...
Shorewall configuration compiled to /var/lib/shorewall/.restart
Stopping Shorewall....
Preparing iptables-restore input...
Running /sbin/iptables-restore --wait 60...
done.
Starting Shorewall....
Initializing...
Setting up Route Filtering...
Setting up Martian Logging...
Preparing iptables-restore input...
Running /sbin/iptables-restore --wait 60...
done.
```

99ea872c 3d44 499d 9492 59c1523c7d14

3. Activé Shorewall : `systemctl enable shorewall && systemctl start shorewall`

Shorewall offre une gestion beaucoup plus simple et lisible des règles de pare-feu, tout en générant les commandes iptables appropriées en arrière-plan.

Fichiers de règles iptables générées par Shorewall:

shorewall-filter.rules :

```
*filter
:INPUT DROP [0:0]
```

```
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [42:2954]
:admin-fw - [0:0]
:admin_frwd - [0:0]
:dmz-admin - [0:0]
:dmz-fw - [0:0]
:dmz-lan - [0:0]
:dmz-srv - [0:0]
:dmz_frwd - [0:0]
:dynamic - [0:0]
:lan-admin - [0:0]
:lan-fw - [0:0]
:lan_frwd - [0:0]
:logdrop - [0:0]
:logflags - [0:0]
:logreject - [0:0]
:net-admin - [0:0]
:net-dmz - [0:0]
:net-fw - [0:0]
:net-lan - [0:0]
:net-srv - [0:0]
:net_frwd - [0:0]
:sha-lh-b1cf911acfddc01e82d1 - [0:0]
:sha-rh-dd25120b6513dabbd699 - [0:0]
:shorewall - [0:0]
:srv-admin - [0:0]
:srv-dmz - [0:0]
:srv-fw - [0:0]
:srv-lan - [0:0]
:srv-net - [0:0]
:srv_frwd - [0:0]
:tcpflags - [0:0]
[35:3898] -A INPUT -i eth0 -j net-fw
[24:1656] -A INPUT -i eth1 -j lan-fw
[59:3677] -A INPUT -i eth2 -j dmz-fw
[4:270] -A INPUT -i eth3 -j admin-fw
[32:2188] -A INPUT -i eth4 -j srv-fw
[0:0] -A INPUT -i lo -j ACCEPT
```

```
[0:0] -A INPUT -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A INPUT -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A INPUT -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A INPUT -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst 10
--hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"INPUT DROP " --log-level 6
[0:0] -A INPUT -j DROP
[79:4464] -A FORWARD -i eth0 -j net_frwd
[0:0] -A FORWARD -i eth1 -j lan_frwd
[82:5738] -A FORWARD -i eth2 -j dmz_frwd
[0:0] -A FORWARD -i eth3 -j admin_frwd
[0:0] -A FORWARD -i eth4 -j srv_frwd
[0:0] -A FORWARD -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A FORWARD -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A FORWARD -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A FORWARD -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"FORWARD DROP " --log-level 6
[0:0] -A FORWARD -j DROP
[4:270] -A admin-fw -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A admin-fw -p tcp -j tcpflags
[4:270] -A admin-fw -j ACCEPT
[0:0] -A admin_frwd -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A admin_frwd -p tcp -j tcpflags
[0:0] -A admin_frwd -o eth0 -j ACCEPT
[0:0] -A admin_frwd -o eth1 -j ACCEPT
[0:0] -A admin_frwd -o eth2 -j ACCEPT
[0:0] -A admin_frwd -o eth4 -j ACCEPT
[0:0] -A dmz-admin -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A dmz-admin -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A dmz-admin -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A dmz-admin -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A dmz-admin -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"dmz-admin DROP " --log-level 6
[0:0] -A dmz-admin -j DROP
```

```
[56:3272] -A dmz-fw -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A dmz-fw -p tcp -j tcpflags
[3:405] -A dmz-fw -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A dmz-fw -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A dmz-fw -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A dmz-fw -m addrtype --dst-type MULTICAST -j DROP
[56:3272] -A dmz-fw -m hashlimit --hashlimit-upto 1/sec --hashlimit-
burst 10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-
prefix "dmz-fw DROP " --log-level 6
[56:3272] -A dmz-fw -j DROP
[0:0] -A dmz-lan -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A dmz-lan -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A dmz-lan -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A dmz-lan -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A dmz-lan -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"dmz-lan DROP " --log-level 6
[0:0] -A dmz-lan -j DROP
[0:0] -A dmz-srv -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A dmz-srv -d 100.80.3.2/32 -p tcp -m tcp --dport 389 -j ACCEPT
[0:0] -A dmz-srv -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"dmz-srv ACCEPT " --log-level 6
[0:0] -A dmz-srv -j ACCEPT
[0:0] -A dmz_frwd -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[82:5738] -A dmz_frwd -p tcp -j tcpflags
[82:5738] -A dmz_frwd -o eth0 -j ACCEPT
[0:0] -A dmz_frwd -o eth1 -j dmz-lan
[0:0] -A dmz_frwd -o eth3 -j dmz-admin
[0:0] -A dmz_frwd -o eth4 -j dmz-srv
[0:0] -A lan-admin -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A lan-admin -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A lan-admin -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A lan-admin -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A lan-admin -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
```

```
"lan-admin DROP " --log-level 6
[0:0] -A lan-admin -j DROP
[24:1656] -A lan-fw -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A lan-fw -p tcp -j tcpflags
[0:0] -A lan-fw -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A lan-fw -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A lan-fw -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A lan-fw -m addrtype --dst-type MULTICAST -j DROP
[24:1656] -A lan-fw -m hashlimit --hashlimit-upto 1/sec --hashlimit-
burst 10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-
prefix "lan-fw DROP " --log-level 6
[24:1656] -A lan-fw -j DROP
[0:0] -A lan_frwd -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A lan_frwd -p tcp -j tcpflags
[0:0] -A lan_frwd -o eth0 -j ACCEPT
[0:0] -A lan_frwd -o eth2 -j ACCEPT
[0:0] -A lan_frwd -o eth3 -j lan-admin
[0:0] -A lan_frwd -o eth4 -j ACCEPT
[0:0] -A logdrop -j DROP
[0:0] -A logflags -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"logflags DROP " --log-level 6 --log-ip-options
[0:0] -A logflags -j DROP
[0:0] -A logreject -m addrtype --src-type BROADCAST -j DROP
[0:0] -A logreject -s 224.0.0.0/4 -j DROP
[0:0] -A logreject -p igmp -j DROP
[0:0] -A logreject -p tcp -j REJECT --reject-with tcp-reset
[0:0] -A logreject -p udp -j REJECT --reject-with icmp-port-unreachable
[0:0] -A logreject -p icmp -j REJECT --reject-with icmp-host-unreachable
[0:0] -A logreject -j REJECT --reject-with icmp-host-prohibited
[0:0] -A net-admin -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A net-admin -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A net-admin -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A net-admin -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A net-admin -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
```

```
"net-admin DROP " --log-level 6
[0:0] -A net-admin -j DROP
[75:4224] -A net-dmz -m conntrack --ctstate RELATED,ESTABLISHED -j
ACCEPT
[2:120] -A net-dmz -p tcp -m multiport --dports 80,443 -m comment --
comment "HTTP, HTTPS" -j ACCEPT
[0:0] -A net-dmz -p udp -m udp --dport 53 -m comment --comment DNS -j
ACCEPT
[2:120] -A net-dmz -p tcp -m multiport --dports 53,25,143,21 -m comment
--comment "DNS, SMTP, IMAP, FTP" -j ACCEPT
[0:0] -A net-dmz -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A net-dmz -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A net-dmz -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A net-dmz -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"net-dmz DROP " --log-level 6
[0:0] -A net-dmz -j DROP
[0:0] -A net-fw -m conntrack --ctstate INVALID,NEW,UNTRACKED -j dynamic
[28:1722] -A net-fw -p tcp -j tcpflags
[35:3898] -A net-fw -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A net-fw -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A net-fw -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A net-fw -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A net-fw -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst 10
--hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"net-fw DROP " --log-level 6
[0:0] -A net-fw -j DROP
[0:0] -A net-lan -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A net-lan -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A net-lan -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A net-lan -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A net-lan -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"net-lan DROP " --log-level 6
[0:0] -A net-lan -j DROP
[0:0] -A net-srv -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A net-srv -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A net-srv -m addrtype --dst-type ANYCAST -j DROP
```

```
[0:0] -A net-srv -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A net-srv -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"net-srv DROP " --log-level 6
[0:0] -A net-srv -j DROP
[4:240] -A net_frwd -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[79:4464] -A net_frwd -p tcp -j tcpflags
[0:0] -A net_frwd -o eth1 -j net-lan
[79:4464] -A net_frwd -o eth2 -j net-dmz
[0:0] -A net_frwd -o eth3 -j net-admin
[0:0] -A net_frwd -o eth4 -j net-srv
[0:0] -A shorewall -m recent --set --name %CURRENTTIME --mask
255.255.255.255 --rsource
[0:0] -A srv-admin -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A srv-admin -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A srv-admin -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A srv-admin -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A srv-admin -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"srv-admin DROP " --log-level 6
[0:0] -A srv-admin -j DROP
[0:0] -A srv-dmz -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A srv-dmz -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A srv-dmz -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A srv-dmz -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A srv-dmz -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"srv-dmz DROP " --log-level 6
[0:0] -A srv-dmz -j DROP
[32:2188] -A srv-fw -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A srv-fw -p tcp -j tcpflags
[0:0] -A srv-fw -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A srv-fw -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A srv-fw -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A srv-fw -m addrtype --dst-type MULTICAST -j DROP
[32:2188] -A srv-fw -m hashlimit --hashlimit-upto 1/sec --hashlimit-
```

```
burst 10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-
prefix "srv-fw DROP " --log-level 6
[32:2188] -A srv-fw -j DROP
[0:0] -A srv-lan -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A srv-lan -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A srv-lan -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A srv-lan -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A srv-lan -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"srv-lan DROP " --log-level 6
[0:0] -A srv-lan -j DROP
[0:0] -A srv-net -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
[0:0] -A srv-net -m addrtype --dst-type BROADCAST -j DROP
[0:0] -A srv-net -m addrtype --dst-type ANYCAST -j DROP
[0:0] -A srv-net -m addrtype --dst-type MULTICAST -j DROP
[0:0] -A srv-net -m hashlimit --hashlimit-upto 1/sec --hashlimit-burst
10 --hashlimit-mode srcip --hashlimit-name lograte -j LOG --log-prefix
"srv-net DROP " --log-level 6
[0:0] -A srv-net -j DROP
[0:0] -A srv_frwd -m conntrack --ctstate INVALID,NEW,UNTRACKED -j
dynamic
[0:0] -A srv_frwd -p tcp -j tcpflags
[0:0] -A srv_frwd -o eth0 -j srv-net
[0:0] -A srv_frwd -o eth1 -j srv-lan
[0:0] -A srv_frwd -o eth2 -j srv-dmz
[0:0] -A srv_frwd -o eth3 -j srv-admin
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG
FIN,PSH,URG -g logflags
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE
-g logflags
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags SYN,RST SYN,RST -g logflags
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags FIN,RST FIN,RST -g logflags
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags FIN,SYN FIN,SYN -g logflags
[0:0] -A tcpflags -p tcp -m tcp --tcp-flags FIN,PSH,ACK FIN,PSH -g
logflags
[0:0] -A tcpflags -p tcp -m tcp --sport 0 --tcp-flags FIN,SYN,RST,ACK
```

```
SYN -g logflags
COMMIT
```

shorewall-nat.rules :

```
*nat
:PREROUTING ACCEPT [120:7626]
:INPUT ACCEPT [4:270]
:OUTPUT ACCEPT [10:712]
:POSTROUTING ACCEPT [14:952]
[0:0] -A POSTROUTING -s 100.80.0.0/16 -o eth0 -j MASQUERADE
COMMIT
```