

TP01 – IMUNES : Passerelles, Routage et Filtrage

Ce TP a deux grands blocs :

- **PARTIE II : Une passerelle (one-gateway.imn)**
- **PARTIE III : Deux passerelles (two-gateways.imn)**

Je te fais **les deux**, propres, complets.

PARTIE I – Préparation (répertoires IMUNES)

À faire une seule fois au début du TP :

1. Sur ton compte personnel :

```
mkdir -p ~/SCR.3.2
```

2. Sur le compte local (celui où tu lances IMUNES) :

Créer :

```
mkdir -p ~/IMUNES/SCR.3.2/TP01
```

IMUNES se lance :

```
sudo imunes &
```

PARTIE II – Une passerelle (one-gateway.imn)

Objectif :

Créer un **réseau S1** (172.16.1.0/24) et un **réseau S2** (172.16.2.0/24), reliés par une **passerelle GW**.

Les règles :

- S1 → S2 : **autorisé**
 - S2 → S1 : **interdit**
-

Étape 1 — Construire la topologie

Dans IMUNES :

1. Créer un switch pour S1
2. Créer un switch pour S2
3. Ajouter **pc1-1** et **pc1-2** sur S1
4. Ajouter **pc2-1** et **pc2-2** sur S2
5. Ajouter une machine type **host** pour la passerelle **GW**
6. Connecter :

```
switch1 -- eth0:GW:eth0 -- switch2
```

Adressage des machines :

Machine	Adresse	Masque	Route ?
pc1-1	172.16.1.1	/24	GW = 172.16.1.254
pc1-2	172.16.1.2	/24	GW = 172.16.1.254

pc2-1	172.16.2.1	/24	GW = 172.16.2.254
pc2-2	172.16.2.2	/24	GW = 172.16.2.254
GW eth0	172.16.1.254	/24	—
	4		
GW eth1	172.16.2.254	/24	—
	4		

Supprimer toutes les adresses IPv6 (elles perturbent les tests).

Étape 2 — Ajouter les routes statiques

Question 1 : "Y a-t-il des routes à ajouter ? Où ?"

OUI : Dans *chaque PC*

Un PC ne peut pas joindre l'autre réseau sans route vers GW.

Sur pc1-1 :

```
ip route add default via 172.16.1.254
```

Sur pc1-2 :

```
ip route add default via 172.16.1.254
```

Sur pc2-1 :

```
ip route add default via 172.16.2.254
```

Sur pc2-2 :

```
ip route add default via 172.16.2.254
```

Étape 3 — Script de démarrage de GW

Dans GW → **Custom configuration** → **Scripts** :

1. Cliquer sur **Fill defaults**
2. Ajouter à la fin :

```
# Activer le routage IP  
sysctl -w net.ipv4.ip_forward=1
```

```
# Autoriser S1 → S2 mais bloquer S2 → S1  
iptables -A FORWARD -s 172.16.1.0/24 -d 172.16.2.0/24 -j ACCEPT  
iptables -A FORWARD -s 172.16.2.0/24 -d 172.16.1.0/24 -j DROP
```

Très important :

- Le routage IP doit être activé (ip_forward=1).
- iptables applique la règle du TP (S2 ne peut pas initier).

Étape 4 — Tests

Depuis S1 → S2 (doit marcher) :

```
sudo himage pc1-1 ping -c 1 172.16.2.2
```

Résultat attendu :

PING OK (comme dans l'énoncé)

tp01.SCR.3.2

Depuis S2 → S1 (doit échouer) :

```
sudo himage pc2-1 ping -c 1 172.16.1.1
```

Résultat attendu :

```
connect: Network is unreachable (comme l'énoncé)
```

tp01.SCR.3.2

FIN DE LA PARTIE II

PARTIE III – Deux passerelles (two-gateways.imn)

Objectif :

Construire **un réseau à 3 segments** avec **2 passerelles**, des accès limités, et un accès Internet (10.0.0.0/8).

Schéma de l'énoncé :

tp01.SCR.3.2

```
S1 -- P1 -- S2 -- P2 -- S3
                        \
                        \---- Internet (10.0.0.0/8)
```

Adresses exactes (selon l'énoncé) :

Interface	Adresse
P1 eth0	172.16.1.25
	4
P1 eth1	172.16.2.25
	3
P2 eth0	172.16.2.25
	2
P2 eth1	172.16.3.25
	1
P2 eth2	10.0.0.250

Étape 1 — Construire la topo two-gateways.imn

Créer :

- **3 switches** : S1, S2, S3
- **2 passerelles** : P1, P2
- **PCs** dans chaque segment

Configurer les adresses comme dans le tableau.

Étape 2 — Ajouter les routes par défaut

S1 → défaut = P1 :

```
ip route add default via 172.16.1.254
```

S2 → défaut = P2 :

```
ip route add default via 172.16.2.252
```

S3 → défaut = P2 :

```
ip route add default via 172.16.3.251
```

Étape 3 — Scripts passerelles (iptables + routage)

Script de démarrage P1

```
sysctl -w net.ipv4.ip_forward=1
```

```
# S1 → S2 autorisé
```

```
iptables -A FORWARD -s 172.16.1.0/24 -d 172.16.2.0/24 -j ACCEPT
```

```
# S2 → S1 interdit
```

```
iptables -A FORWARD -s 172.16.2.0/24 -d 172.16.1.0/24 -j DROP
```

Script de P2

```
sysctl -w net.ipv4.ip_forward=1
```

```
# S2 → S3 autorisé
```

```
iptables -A FORWARD -s 172.16.2.0/24 -d 172.16.3.0/24 -j ACCEPT
```

```
# S3 → S2 interdit
```

```
iptables -A FORWARD -s 172.16.3.0/24 -d 172.16.2.0/24 -j DROP
```

```
# S1 pas visible par S3
```

```
iptables -A FORWARD -s 172.16.3.0/24 -d 172.16.1.0/24 -j DROP
```

```
# S1 et S2 → Internet (10.0.0.0/8) autorisé
```

```
iptables -A FORWARD -s 172.16.1.0/24 -d 10.0.0.0/8 -j ACCEPT
```

```
iptables -A FORWARD -s 172.16.2.0/24 -d 10.0.0.0/8 -j ACCEPT
```

```
# S3 → Internet interdit
```

```
iptables -A FORWARD -s 172.16.3.0/24 -d 10.0.0.0/8 -j DROP
```

Le filtrage correspond **exactement** aux règles de l'énoncé.

Étape 4 — Tests de validation

S1 ↔ S2 :

```
pc1 ping 172.16.2.x → OK
```

```
pc2 ping 172.16.1.x → FAIL
```

S2 ↔ S3 :

```
pc2 ping 172.16.3.x → OK
```

```
pc3 ping 172.16.2.x → FAIL
```

S1 et S2 → Internet :

```
pc1 ping 10.0.0.1 → OK
```

```
pc2 ping 10.0.0.1 → OK
```

S3 → Internet :

```
pc3 ping 10.0.0.1 → FAIL
```

FIN DE LA PARTIE III