Faculté des Sciences & Techniques — Université de Limoges

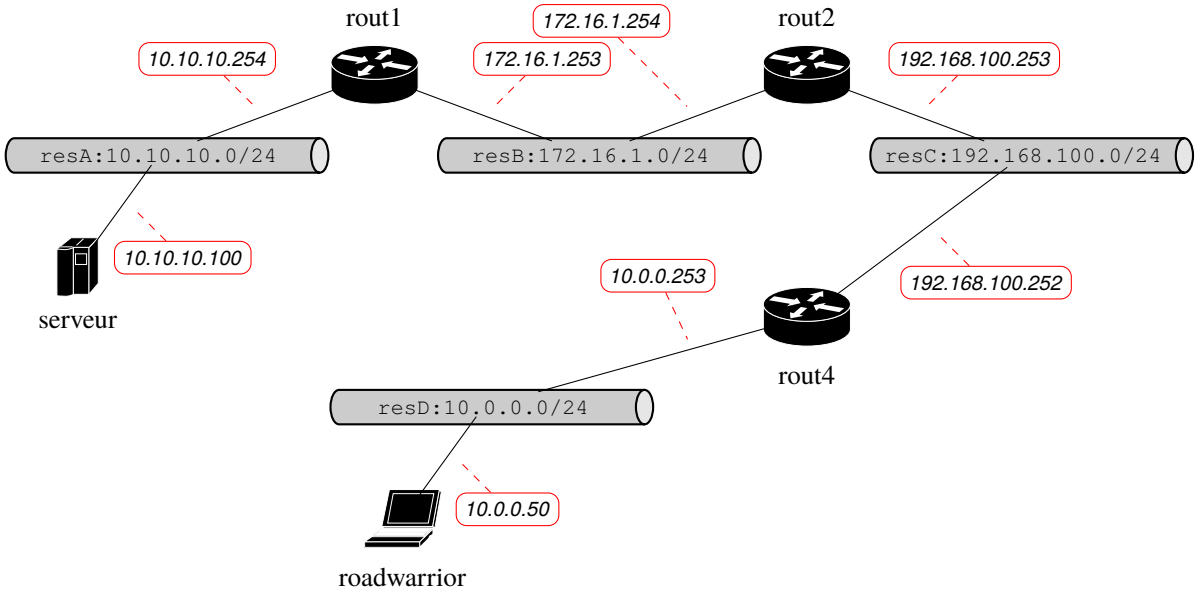
Master 1ère année

Infrastructure Réseaux

TP n°3

Tunnels

Modification du réseau proposé dans le TP n°2



On va se servir du réseau déployé dans le cadre du TP n°2, en supprimant « rout3 » et le VLAN 100.
On utilisera toujours le protocole RIP pour construire les tables de routage des différents routeurs.

But du TP

- adapter le réseau du TP n°2 en modifiant et ajoutant des fichiers de configuration pour les « switches » et les « netns » (on utilisera toujours Quagga pour réaliser le routage RIP) ;
- tester et étudier les VPNs suivants :
  - GRE, « *Generic Routing Encapsulation* », RFC 2784 ;
  - IPsec ;
  - GRE over IPsec ;
  - PPP over SSH ;
  - VPN over SSH.
- ces VPNs seront déployés entre :
  - la machine « *roadwarrior* » et la machine « *serveur* » ;
  - « *rout1* » et « rout4 » ;
- l'analyse sera réalisé à l'aide de l'outil `tcpdump`.

Qu'est-ce qu'un RoadWarrior ?

Un « *Road Warrior* » est une personne qui voyage tout le temps, qui est donc rarement dans son bureau, et qui utilise énormément son ordinateur portable pour travailler : il a besoin d'accéder à distance aux ressources informatiques de son bureau, ce qui nécessite l'établissement de VPN entre sa machine et son bureau (rien à voir avec Mad Max…).

Pour l'installation des outils supplémentaires sur la VM pour PPP et Ipsec :

xterm

```
rezo@ishtar:~$ sudo apt-get install ppp
```

Présentation des différents types de VPN, « Virtual Private Network »

Qu'est-ce qu'un VPN ? « une connexion privée entre deux éléments terminaux », c-à-d une liaison « point-à-point » entre ces deux terminaux dont le contenu n'est accessible qu'à ces deux terminaux.

Plus concrètement ?

- ⋆ il faut pouvoir **encapsuler** des communications quelconques dans cette liaison point-à-point, c-à-d faire passer les datagrammes IP de ces communications dans la liaison ;
- ⋆ cette liaison emprunte InterNet et doit pouvoir être routé : elle utilise des datagrammes IP.

*Ainsi, la liaison est considérée comme un **tunnel** qui peut être emprunter de manière transparente par ces communications.*

Quelles différences avec une liaison physique « point-à-point » ? la liaison physique point-à-point est *physique* (câble de liaison entre deux routeurs par exemple) alors que la liaison VPN est *virtuelle*, elle est simulée au travers d'une communication IP **isolée des autres éléments** du réseau :

- ⋆ seuls deux bouts communiquent au travers de la liaison (par ex. après authentification mutuelle) ;
- ⋆ les données échangées peuvent être rendues « inaccessibles » à l'observation d'un tiers en recourant à la cryptographie (confidentialité assurée par chiffrement), en plus de l'être par routage (ils peuvent passer dans un réseau d'interconnexion contrôlé comme avec MPLS).

Comment mettre en place un VPN ? : il faut transporter des datagrammes IP

- ◦ avec un protocole de niveau 2 comme PPP, « *Point-to-Point Protocol* », RFC 1661, 1547 :
  - ◇ il est considéré, dans le modèle OSI, comme un protocole de niveau 2 car il encapsule dans une **liaison série** des datagrammes ;
  - ◇ il a été crée, à l'origine, pour permettre des communications sur des lignes téléphoniques par modem (avant l'ADSL avec des débits de 56Kbits/seconde) ;
  - ◇ il permet une **authentification mutuelle** avec :
    - ⋆ PAP, « *Password Authentication Protocol* » (transmission de mot de passe en clair) ;
    - ⋆ CHAP, « *Challenge-Handshake Authentication Protocol* » (secret partagé échangé haché) ;
    - ⋆ EAP, « *Extensible Authentication Protocol* » (protocole utilisant différentes méthodes avancés jusqu'à l'utilisation de certificat) ;
  - ◇ il permet de faire de la compression (réduction des transmissions), du contrôle d'erreur, du chiffrement avec ECP, « *Encryption Control Protocol* » ;
  - ◇ il est utilisé au travers d'autres protocoles : PPPoE, « *PPP over Ethernet* », PPPoA, « *PPP over ATM* » pour ses capacités à établir des sessions dans le cadre de l'accès ADSL.
- ◦ avec une encapsulation IP dans IP comme GRE, « *Generic Routing Encapsulation* » , RFC 2784 :
  - ◇ il est intégré dans la couche de niveau 3, où il est désigné par le numéro de protocole 47 ;
  - ◇ il permet de transporté des datagrammes IP :
    - ⋆ pour tous les protocoles encapsulables (ICMP, UDP, TCP *etc.*) ;
    - ⋆ pour des @IP source et destination quelconques (applications à une liaison entre routeurs).
- ◦ avec un protocole d'isolation comme IPsec, « *Encryption Control Protocol* », RFC 1825, 1829 :
  - ◇ il est disponible dans IPv4 et est intégré dans IPv6 ;
  - ◇ il permet la négociation des éléments cryptographiques entre les deux extrémités (choix des algorithmes, des clés, construction de clés de session *etc.*) ;
  - ◇ il travaille au niveau du datagramme IP, c-à-d en mode « connectionless » :
    - ⋆ il permet d'authentifier les deux extrémités de la liaison dans l'en-tête du datagramme IP ;
    - ⋆ il permet de chiffrer le contenu du datagramme IP ;
    - ⋆ il fonctionne en **mode tunnel** (tout le contenu du paquet est protégé y compris les adresses nécessaires au routage) ou **transport** (les adresses nécessaires au routage sont visibles).
- ◦ en combinant du PPP et du GRE et en ajoutant du chiffrement et de la compression : le protocole PPTP, « *Point-to-Point Tunneling Protocol* », RFC 2637 (une connexion TCP vers le port 1723 permet d'établir le tunnel GRE) ;
- ◦ en fractionnant du PPP sur de l'UDP avec L2TP, RFC 2661, 3931 ;
- ◦ en combinant du GRE et de l'IPsec en mode transport ;
- ◦ en utilisant des dérivés d'openSSL :
  - ◇ en combinant du SSH (liaison TCP sécurisée) et du PPP ;
  - ◇ avec SSH et son mode VPN en niveau 3 (interface TUN) ou 2 (interface TAP) ;
  - ◇ avec openVPN en mode UDP ou TCP.

Création de la machine « serveur » & « roadwarrior »

Pour pouvoir utiliser le serveur SSH dans un netns :

- ⋆ vous créerez le sous répertoire `/etc/netns/serveur/ssh` ;
- ⋆ vous copierez le fichier `/etc/ssh/sshd_config` dedans :

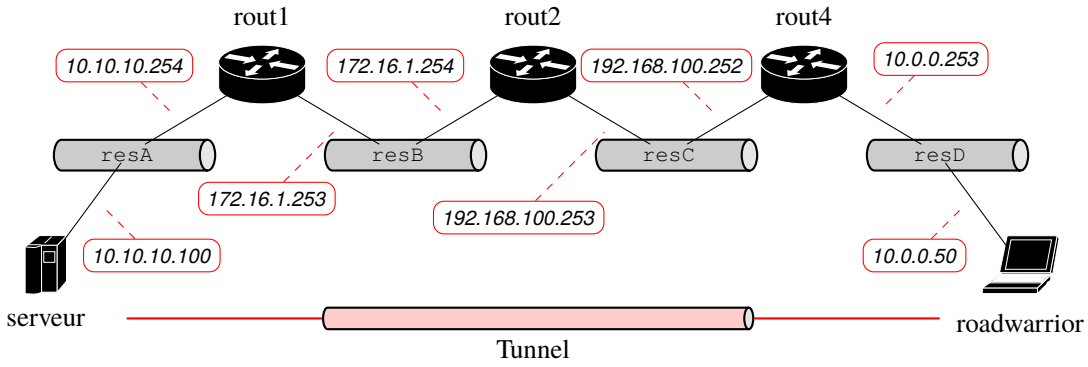
```
rezo@ishtar:~/etc/netns/ssh$ sudo cp /etc/ssh/sshd_config .
```

Dans le fichier `sshd_config` que vous aurez copié, vous rajouterez la ligne :

```
PermitTunnel yes
```

afin d'autoriser l'utilisation des VPNs au travers de SSH.

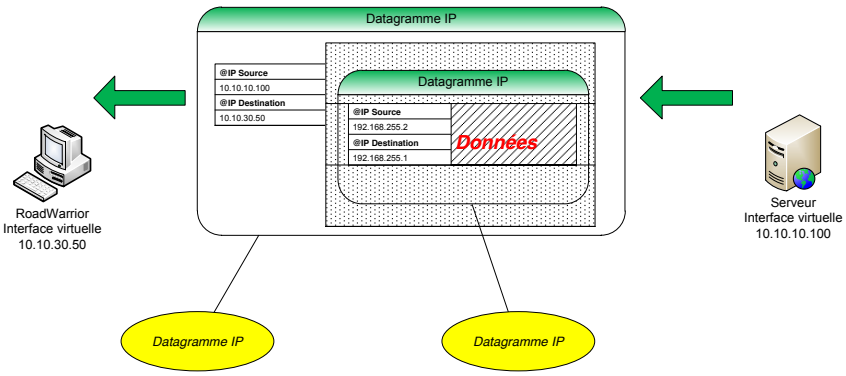
Création de Tunnels entre les machines « Serveur » et « RoadWarrior »



GRE

Avec ce protocole :

- ◦ on va mettre en place un tunnel entre « rout1 » et « rout4 ».
- ◦ le trafic entre la machine RoadWarrior et la machine Serveur empruntera ce tunnel ;
- ◦ le tunnel est « *stateless* » : il n'y a pas de configuration associée au tunnel qui doit être mémorisée sur chaque extrémité, chaque datagramme IP empruntant le tunnel est encapsulé dans un nouveau datagramme IP et envoyé sans contrôle d'erreur ;
- ◦ chaque datagramme empruntant le tunnel (suivant son routage) est traité séparément.



Pour établir le tunnel GRE :

- ⋆ Sur rout1 :

xterm

```
ip tunnel add mon_tunnel mode gre local 172.16.1.253 remote 192.168.100.252
ip link set mon_tunnel up
ip link
31: mon_tunnel: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1476 qdisc noqueue state
UNKNOWN
    link/gre 172.16.1.253 peer 192.168.100.252
ip addr add dev mon_tunnel 10.0.0.1/24
ip route add dev mon_tunnel 10.10.30.0/24
```

⋆ Sur rout4 :

xterm
```
ip tunnel add mon_tunnel mode gre local 192.168.100.252 remote 172.16.1.253
ip link set mon_tunnel up
ip addr add dev mon_tunnel 10.0.0.2/24
ip addr
32: mon_tunnel: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1476 qdisc noqueue state
UNKNOWN
    link/gre 192.168.100.253 peer 172.16.1.253
    inet 10.0.0.2/24 scope global mon_tunnel
ip route add dev mon_tunnel 10.10.10.0/24
```

On remarque que les deux extrémités du réseau appartiennent à un même réseau `10.0.0.0/24` *indépendant de tous les autres (sinon des problèmes de routage peuvent survenir).*

Questions :

a. sur rout2, vous capturerez le trafic réseau à l'aide de tcpdump ;
b. vous étudierez le contenu de ce traffic entre les deux machines « RoadWarrior » et « Serveur » :
   - ◇ en effectuant un « `ping` » entre « Serveur » et « Roadwarrior »
   - ◇ peut-on avoir accès au contenu des échanges ?
   - ◇ capturez du trafic GRE à l'aide de l'option « `-w` » de tcpdump et analysez le à l'aide de Scapy, comment est effectué l'encapsulation GRE ?
   - ◇ vérifiez la MTU du tunnel et comparez avec la MTU standard, que remarquez vous ?

IPsec

▷ IPsec permet la création de VPN, « *Virtual Private Network* », en utilisant la cryptographie.
▷ IPsec est intégré dans IPv6 et peut être utilisé dans IPv4.
▷ IPsec propose :
   - ◇ deux protocoles :
     - ⋆ AH, « *Authentication Header* » pour l'authentification ;
     - ⋆ ESP, « *Encapsulating Security Payload* », pour le chiffrement et l'authentification.

     On peut utiliser séparément l'un ou l'autre et, plus souvent, les deux ensembles.
     Le protocole AH assure l'intégrité et l'authentification de l'origine pour l'ensemble des champs de l'entête du datagramme IP, à l'exception de ceux qui peuvent changer lors du transfert du datagramme, c-à-d les champs « TTL » et « checksum ».

   - ◇ choix entre différents algorithmes cryptographiques : « SHA-1 », « 3DES », « AES », *etc.* : la mise en œuvre d'une connexion IPSec impose de faire des choix, mais chaque connexion ne fait appel qu'à deux, voire trois, algorithmes à la fois.
     - ⋆ L'opération d'authentification calcule un ICV, « *Integrity Check Value* », sur le contenu du paquet, ce qui est réalisé au travers d'une fonction de hachage comme SHA-1. Il incorpore une clé secrète connue des deux interlocuteurs, ce qui permet au destinataire de calculer l'ICV de la même façon. Ainsi, si le destinataire reçoit la même valeur, alors l'émetteur s'est authentifié avec succès (cela repose sur le fait qu'une fonction de hachage ne peut être inversée).
       AH fournit toujours de l'authentification alors qu'ESP peut la fournir en option.
     - ⋆ Le chiffrement utilise un clé secrète pour chiffrer les données avant leur transmission et cela permet de « cacher » le contenu du paquet et de le protéger d'éventuelles écoutes.
       Il est possible de choisir parmi différents algorithmes de chiffrement et en particulier, entre 3DES, Blowfish et AES.

   - ◇ deux **modes de fonctionnement** :
     - ⋆ le mode « **Transport** » :
       - ▷ permet d'établir une liaison sécurisée directement entre deux matériels ;
       - ▷ encapsule le chargement du datagramme IP : les @IP source et destination reste celles de ces matériels ;
     - ⋆ le mode « **Tunnel** »
       - ▷ permet d'établir une liaison sécurisée entre deux routeurs ;
       - ▷ permet d'encapsuler la totalité du datagramme IP passant par ces routeurs ce qui permet d'offrir un « *secure hop* », c-à-d le passage sécurisé entre deux routeurs (les datagrammes IP ne contiennent que les @IP source et destination des routeurs, mais pas celles des machines empruntant ce tunnel ;
       - ▷ permet d'établir des VPNs entre deux sites au travers d'Internet.

Dans le cas du mode Tunnel, on utilise rarement le protocole AH, dans la mesure où il interdit la modification de l'entête du datagramme IP, ce qui rend impossible l'utilisation de NAT ce qui peut être bloquant. On préférera alors l'utilisation d'ESP avec une forme simplifiée d'authentification : elle utilise les mêmes algorithmes que ceux utilisés par AH, mais cette authentification ne porte que sur l'entête et les données du contenu ESP, et pas sur l'entête du datagramme IP qui le contient.
L'utilisation du mode Tunnel est transparente, puisqu'elle s'applique uniquement entre deux routeurs, et que deux interlocuteurs utilisant ces routeurs n'ont rien à faire pour bénéficier de cette protection.

- IKE, « *Internet Key Exchange* » *vs* « Clés fournies manuellement » : les deux extrémités de la communication doivent connaître les valeurs secrètes utilisées pour la fonction de hachage et le chiffrement, ce qui pose le problème de les échanger.
  La « fourniture manuelle » des clés requiert d'entrer manuellement les clés sur les deux extrémités probablement sans se servir du réseau pour le faire.
  IKE est un moyen sophistiqué pour le faire de manière « online ».

- Mode « principal » ou « agressif » : le choix entre ces deux modes représentent un compromis entre efficacité et sécurité pour le protocole IKE d'échange de clés.
  Le mode principal requiert l'échange de 6 paquets dans un sens et dans l'autre, alors que le mode agressif en requiert la moitié, tout en transmettant des informations en clair ce qui diminue la sécurité.

Intégration d'IPsec dans IPV4 :
Le champ « protocole » du datagramme IP indique la nature du contenu :

- 1 : ICMP ;
- 6 : TCP ;
- 17 : UDP ;
- 47 : GRE ;
- 50 : IPsec : ESP ;
- 51 : IPsec : AH ;

Dans le cadre du TP

Nous utiliserons IPsec en mode transport avec AH et ESP configuré manuellement, c-à-d sans négociation, avec des clés choisies de manière aléatoire (on n'utilisera pas le protocole IKE et le démon racoon, l'implémentant sous Linux).

La gestion des éléments cryptographiques :
Il est nécessaire de gérer des secrets sur les deux extrémités de la connexion sécurisée (les secrets permettant l'authentification et le chiffrement).
Lorsqu'un paquet IPsec, AH ou ESP, arrive sur une interface réseau, comment cette interface peut savoir quel ensemble de paramètres (clé, algorithme et « politique de sécurité ») utiliser ?
Chacun de ces ensembles est spécifié au travers d'une SA, « Security Association », c-à-d une collection de paramètres spécifiques à une connexion, et chaque interlocuteur peut en posséder de nombreuses.
Afin de traiter un paquet IP à son arrivée, il faut :

- l'adresse IP de l'interlocuteur qui a envoyé le paquet ;
- la nature du protocole : ESP ou AH ;
- un SPI, « Security Parameters Index ».

Une SA concerne « un seul sens » de communication, c-à-d qu'une communication bidirectionnelle en utilise deux.
Chaque protocole requiert sa propre SA pour chaque direction, ce qui fait que 4 SAs sont nécessaires pour un VPN utilisant AH+ESP.
Chaque interlocuteur dispose d'une SADB, une base de données des SAs qu'il possède.
Dans cette SADB, il y a :

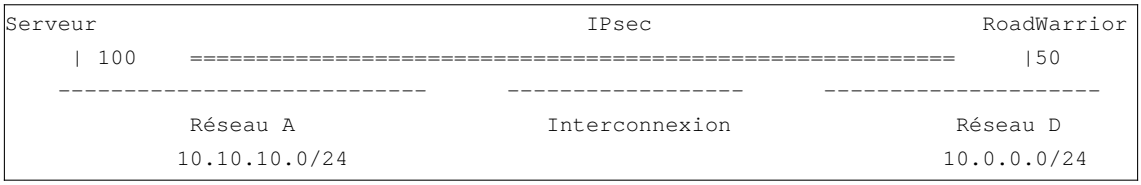
- AH : l'algo. utilisé ;
- AH : le secret d'authentification ;
- ESP : l'algo de chiffrement ;
- ESP : la clé secrète de chiffrement ;
- ESP : la sélection d'une authentification ou pas ;
- des restrictions concernant le routage ;
- des politiques de sélection du contenu IP, « policy » ;
- des paramètres concernant l'échange des clés.

La gestion d'IPsec dans Linux :
On utilisera la commande `ip xfrm`:

- ▷ `ip xfrm state` qui permet de manipuler la SAD, « *Security Association Database* » ;
- ▷ `ip xfrm policy` qui permet de manipuler la SPD, « *Security Policy Database* » ;
- ▷ « openswan » : un démon réalisant IKEv2 pour échanger automatiquement les clés utilisées pour une connexion IPsec.

Ici nous n'utiliserons pas openswan et nous réaliserons manuellement la mise en place du tunnel IPSec.

Exemple de configuration d'un tunnel IPsec :



Le contenu du fichier de configuration d'IPsec en mode transport sur l'extrémité « Serveur » :

```
1  # Flush the SAD and SPD
2  ip xfrm state flush
3  ip xfrm policy flush
4
5  # AH SAs using 256 bit long and ESP SAs using 160 bit long keys
6  ip xfrm state add src 10.10.10.100 dst 10.0.0.50 proto esp spi 0x12345678 reqid 0x12345678 \
7        mode transport \
8        auth sha256 0x323730ed6f1b9ff0cb084af15b197e862b7c18424a7cdfb74cd385ae23bc4f17 \
9        enc "rfc3686(ctr(aes))" 0x27b90b8aec1ee32a8150a664e8faac761e2d305b
10
11 ip xfrm state add src 10.0.0.50 dst 10.10.10.100 proto esp spi 0x12345678 reqid 0x12345678 \
12       mode transport \
13       auth sha256 0x44d65c50b7581fd3c8169cf1fa0ebb24e0d55755b1dc43a98b539bb144f2067f \
14       enc "rfc3686(ctr(aes))" 0x9df7983cb7c7eb2af01d88d36e462b5f01d10bc1
15
16
17 # Security policies
18 ip xfrm policy add src 10.0.0.50 dst 10.10.10.100 dir in tmpl src 10.0.0.50 dst 10.10.10.100 \
19    proto esp reqid 0x12345678 mode transport
20 ip xfrm policy add src 10.10.10.100 dst 10.0.0.50 dir out tmpl src 10.10.10.100 dst 10.0.0.50 \
21    proto esp reqid 0x12345678 mode transport
```

Ce qui donne :

xterm

```
$ sudo ip xfrm policy
src 10.10.10.100/32 dst 10.0.0.50/32
   dir out priority 0
   tmpl src 10.10.10.100 dst 10.0.0.50
      proto esp reqid 305419896 mode transport
src 10.0.0.50/32 dst 10.10.10.100/32
   dir in priority 0
   tmpl src 10.0.0.50 dst 10.10.10.100
      proto esp reqid 305419896 mode transport
```

et :

xterm

```
$ sudo ip xfrm state
src 10.0.0.50 dst 10.10.10.100
   proto esp spi 0x12345678 reqid 305419896 mode transport
   replay-window 0
   auth-trunc hmac(sha256)
0x44d65c50b7581fd3c8169cf1fa0ebb24e0d55755b1dc43a98b539bb144f2067f 96
   enc rfc3686(ctr(aes)) 0x9df7983cb7c7eb2af01d88d36e462b5f01d10bc1
   anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
   sel src 0.0.0.0/0 dst 0.0.0.0/0
src 10.10.10.100 dst 10.0.0.50
   proto esp spi 0x12345678 reqid 305419896 mode transport
   replay-window 0
   auth-trunc hmac(sha256)
0x323730ed6f1b9ff0cb084af15b197e862b7c18424a7cdfb74cd385ae23bc4f17 96
   enc rfc3686(ctr(aes)) 0x27b90b8aec1ee32a8150a664e8faac761e2d305b
   anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
   sel src 0.0.0.0/0 dst 0.0.0.0/0
```

Pour la configuration d'IPsec en mode transport

Le fichier de configuration de l'autre extrémité du tunnel sur « RoadWarrior » est identique au fichier de configuration de « Serveur », **sauf** pour les lignes 18, où il faut mettre « dir out », et 20, où il faut mettre « dir in ».
Ce qui est logique, dans la mesure où les SAs sont partagés et les règles de trafic, SP, sont inversées.

Remarques :

⋆ Pour sortir une clé aléatoire de longueur donnée sous forme hexadécimale :

```
openssl rand -hex 16
```

Ici, la clé est de 16 octets.

Questions :

a. sur « *rout2* » vous capturerez le trafic réseau à l'aide de `tcpdump` ;

b. vous étudierez le contenu de ce trafic entre les deux machines « RoadWarrior » et « Serveur » :
   - ◇ avec un ping ?
   - ◇ peut-on avoir accès au contenu des échanges ?

Utilisation de GRE + IPsec

Questions :

a. En reprenant la configuration de votre tunnel GRE entre « rout1 » et « rout4 », vous ajouterez du chiffrement/authentification à l'aide d'IPsec comme vu dans la section précédente en adaptant la configuration d'IPsec.

b. Vous capturerez du trafic à l'aide de tcpdump sur *rout2* et vous vérifierez que ce trafic est bien chiffré (par exemple lors d'un ping entre « Serveur » et « RoadWarrior ».

Utilisation de PPP over SSH

Sur la machine « serveur » :

◦ vous créerez **une seule fois** les différentes clés associées au serveur SSH sur le netns « *serveur* » :

xterm
```
rezo@ishtar:~# ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
```

◦ vous lancerez le serveur SSH (à effectuer à chaque création du netns) :

xterm
```
rezo@ishtar:~# /usr/sbin/sshd
```

On utilisera le protocole PPP sans authentification, option « noauth », car celle de SSH (surtout à l'aide de clés RSA ou DSA) est plus forte.

Pour lancer le tunnel PPP over SSH et vérifier son fonctionnement :

⋆ Sur le netns « roadwarrior », on établit le tunnel PPP sur SSH :

xterm
```
rezo@ishtar:~# /usr/sbin/pppd noauth updetach pty "ssh root@10.10.10.100
/usr/sbin/pppd noauth notty 192.168.254.254:192.168.254.253"

Using interface ppp0
Connect: ppp0 <--> /dev/pts/6
Deflate (15) compression enabled
local  IP address 192.168.254.253
remote IP address 192.168.254.254
```

une nouvelle interface « `ppp0` » est créée associée à l'@IP « `192.168.254.253` » :

xterm
```
rezo@ishtar:~# ip link
78: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:0a:00:fe:fe:fe brd ff:ff:ff:ff:ff:ff
80: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
90: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN qlen 3
    link/ppp
rezo@ishtar:~# ip addr
90: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN qlen 3
    link/ppp
    inet 192.168.254.253 peer 192.168.254.254/32 scope global ppp0
```

On peut tester le tunnel depuis la machine RoadWarrior :

xterm
```
rezo@ishtar:~# ping 192.168.254.254
PING 192.168.254.254 (192.168.254.254) 56(84) bytes of data.
64 bytes from 192.168.254.254: icmp_req=1 ttl=64 time=0.828 ms
64 bytes from 192.168.254.254: icmp_req=2 ttl=64 time=0.386 ms
64 bytes from 192.168.254.254: icmp_req=3 ttl=64 time=0.400 ms
^C
--- 192.168.254.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.386/0.538/0.828/0.205 ms
```

On peut tester le tunnel avec une connexion TCP en utilisant `socat` :

xterm
```
rezo@ishtar:~# socat - tcp:192.168.254.254:3433
```

⋆ Sur la machine « Serveur », on vérifie l'existence d'une nouvelle interface « ppp0 » :

xterm
```
rezo@ishtar:~# ip addr
82: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:0a:00:ba:db:ad brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.100/24 brd 10.10.10.255 scope global eth0
    inet6 fe80::20a:ff:feba:dbad/64 scope link
       valid_lft forever preferred_lft forever
84: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
85: gre0: <NOARP> mtu 1476 qdisc noop state DOWN
    link/gre 0.0.0.0 brd 0.0.0.0
91: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UNKNOWN qlen 3
    link/ppp
    inet 192.168.254.254 peer 192.168.254.253/32 scope global ppp0
```

On teste le tunnel en attendant une connexion TCP à l'aide de `socat` en provenance de « RoadWarrior » :

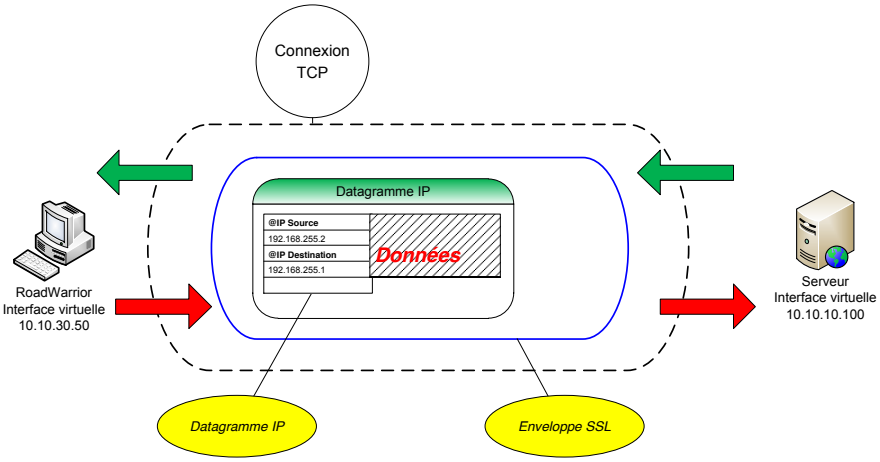
xterm
```
rezo@ishtar:~# socat - tcp-listen:3433
```

Questions :

a. sur « rout2 », vous capturerez le trafic réseau ;

b. vous étudierez le contenu de ce trafic entre les deux machines « RoadWarrior » et « Serveur » :
   - ◇ comment évolue-t-il lors du ping ?
   - ◇ peut-on avoir accès au contenu des échanges ?

VPN over SSH



- Vous n'oublierez pas de lancer le serveur SSH sur le netns « Serveur » :

```
xterm
rezo@ishtar:~# /usr/sbin/sshd
rezo@ishtar:~# ss -l
Recv-Q Send-Q              Local Address:Port                  Peer
Address:Port
0      0                         *:ssh                             *:*
0      0                        :::ssh                            :::*
```

- sur la machine « RoadWarrior » :

```
xterm
rezo@ishtar:~# ssh -NTCf  -w 0:0 root@10.10.10.100
```

Vous vérifierez qu'une nouvelle interface « tun0 » a été créée :

```
xterm
rezo@ishtar:~# ip link
24: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 500
    link/none
```

Ainsi que sur « Serveur » :

```
xterm
rezo@ishtar:~# ip link
25: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN qlen 500
    link/none
```

Vous paramètrerez cette nouvelle interface :

```
xterm
rezo@ishtar:~# ip link set tun0 up
rezo@ishtar:~# ip addr
rezo@ishtar:~# ip addr add 10.87.0.100/32 peer 10.87.0.200 dev tun0
```

- sur « Serveur » :

```
xterm
rezo@ishtar:~# ip link
rezo@ishtar:~# ip link tun0 up
rezo@ishtar:~# ip link set tun0 up
```

- Vous capturerez le trafic sur « rout2 » ;
- Vous testerez le tunnel depuis le « RoadWarrior » :

```
xterm
rezo@ishtar:~# ping 10.87.0.200
```

Questions :

a. Que se passe-t-il ? Quel trafic capturez vous et à quoi correspond-il ?

b. Pourquoi le ping ne fonctionne pas ?

c. Sur le « Serveur », tapez les commandes suivantes :

```
xterm
rezo@ishtar:~# ip addr add 10.87.0.200 dev tun0
rezo@ishtar:~# ip route add 10.87.0.100 via 10.87.0.200
```

Que se passe-t-il ?

d. Entre un VPN IPsec et un VPN SSH de sécurité équivalente lequel privilégier ?

Wireguard

C'est le nouveau protocole en vogue disponible sur différentes plateformes..

Il utilise les algorithmes ChaCha20 et Poly1305 pour l'intégrité et le chiffrement des données.

Vous trouverez la description de son installation à la page :
`https://www.wireguard.com/install/`

Vous trouverez une démo de mise en œuvre à la page :
`https://www.wireguard.com/quickstart/`

Un étude de performance à `https://www.wireguard.com/performance/`

Vous essaierez de mettre en œuvre un tunnel de cette nature.

Questions :

a. En utilisant l'outil de mesure de performance `iperf` ou `iperf3`, vous vérifierez s'il est plus rapide que les autres types de tunnel.

b. Quels sont les ports utilisés pour la mise en place du tunnel ?
Vous pourrez utiliser l'outil `sudo ss -lnp`

c. Vous sniffferez le trafic avec `tcpdump` pour étudier comment se fait l'encapsulation des échanges.