M2-RADIS Rezo
TP12 : IPSEC

Jean Saquet
Jean.Saquet@unicaen.fr

12/12/12

1 Introduction

L'objectif de ce TP est d'établir des associations de sécurité IPSEC entre deux réseaux virtuels configurés sur deux machines différentes. Nous étudierons le mode tunnel comme le mode transport.
Le mécanisme IPSEC fonctionne ainsi :
– à chaque datagramme, le système regarde dans la base de données SPD si la politique de sécurité impose d'utiliser IPSEC ;
– si oui, on recherche les paramètres de la SA et la clé dans la base SAD
– si on ne la trouve pas, le protocole IKE permet les échanges de clés et la base SAD est alors renseignée et utilisée pour traiter le datagramme.

2 Architecture utilisée

On utilisera un réseau semblable à celui de la figure 1, avec les particularités suivantes :
– nous configurerons des adresses IPv4 et des adresses IPv6 mais travaillerons essentiellement en IPv6 ;
– le TP se fera par binôme, l'objectif étant de communiquer entre les réseaux des deux éléments du binôme ;
– nous utiliserons une machine au lieu d'un routeur car l'image de ces derniers ne contient pas le logiciel nécessaire ;
Configuration :
Comme dans beaucoup de TPs, un nombre `n` sera attribué à chacun, entre 11 et 1F. Il sera utilisé pour configurer les adresses :
– réseau interne `192.168.n.0/24` en v4, `2001:660:7101:n::/64` en v6 ;
– adresse v4 `192.168.128.n` sur l'interface "externe" du routeur ;
– `2001:660:7101:ffff:10::n` en v6 sur la même interface.
N'oubliez pas de :
– activer le forwarding v4 et v6 sur le routeur ;
– mettre une route par défaut sur la machine interne ;
– activer le routage v6 avec quagga
– Configurer un firewall v6 pour interdire les liaisons directes (on peut laisser passer dans la phase de test). Il faut ajouter une règle pour laisser passer udp sur le port 500 (indispensable pour IKE) ;

3 principe du TP

On communique entre les deux réseaux virtuels des deux éléments du binôme, ou entre deux machines situées chacune dans un de ces réseaux.

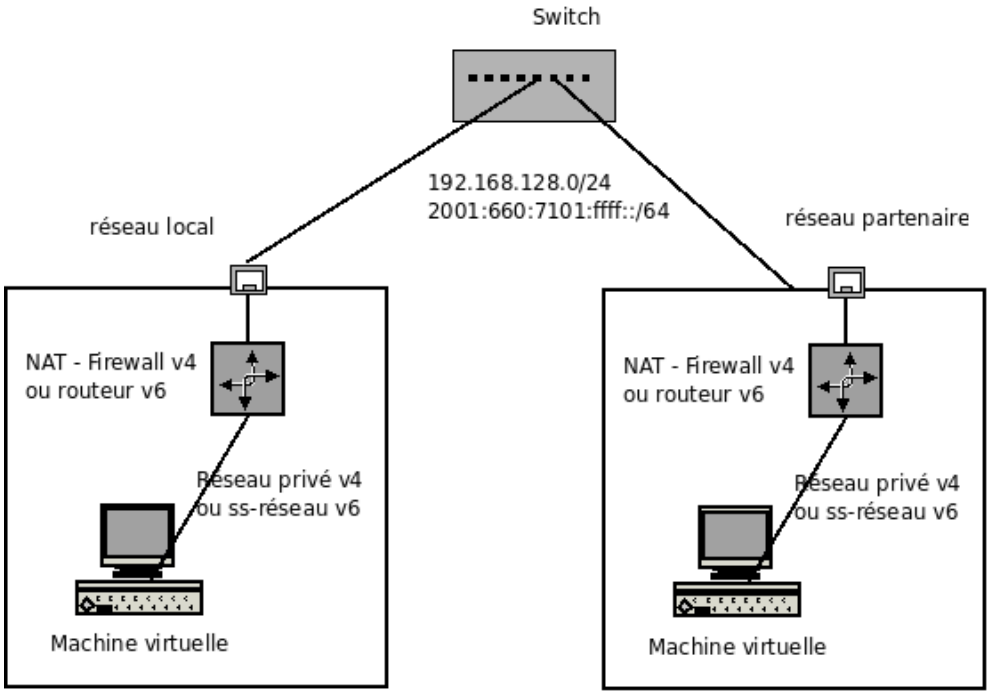

FIGURE 1 – Notre architecture
Attention, l'adresse v6 du réseau de TP peut être un peu différente

Tester d'abord la communication sans IPSEC, puis activer IPSEC avec un chiffrement et tester à nouveau. La différence se verra en examinant les trames avec `tcpdump` aux endroits stratégiques ...
On essaiera un mode tunnel entre les deux routeurs, et un mode transport entre deux machines internes situées dans des réseaux virtuels différents.

4 Configuration

Il y a dans le répertoire `/etc/racoon` des exemples de fichiers. ce sont :
– le fichier `psk.txt` qui contient la ou les clés secrètes partagées. Nous utiliserons cette méthode.
– un exemple de fichier de configuration `racoon.conf`
Le fichier de configuration de racoon peut se présenter ainsi :

```
remote <passerelle ou machine distante> {
        exchange_mode main,aggressive;
        lifetime time 2 min ;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm sha1;
                authentication_method pre_shared_key;
                dh_group modp1024;
        }
        generate_policy off;
}
```

```
sainfo address <reseau local>[any] any address <reseau distant>[any] any {
       pfs_group modp768;
       encryption_algorithm 3des;
       authentication_algorithm hmac_md5;
       compression_algorithm deflate;
       lifetime time 1 min ;
}
```

Il faut tout d'abord définir la politique de sécurité à appliquer. Le mieux est de créer un fichier de configuration `setkey.conf` pour ne pas tout retaper à chaque essai. Pour un mode tunnel, ce fichier se présente sous la forme :

```
#!/usr/sbin/setkey -f
flush;
spdflush;
spdadd <reseau_local> <reseau_partenaire> any -P out ipsec
         esp/tunnel/<passerelle_locale>-<passerelle_partenaire>/require ;
spdadd <reseau_partenaire> <reseau_local> any -P in ipsec
         esp/tunnel/<passerelle_partenaire>-<passerelle_locale>/require ;
```

Remarque : les réseaux, local ou partenaire, peuvent aussi être une simple machine (par exemple la passerelle). Ceci est en particulier le cas en v4 où le réseau partenaire est inaccessible directement, pour cause d'adresses privées.
Configurer IPSEC pour v4 est un peu sportif. Il est conseillé de commencer par v6 et on essaiera v4 si on a le temps.
NB : il peut aussi être parfois utile de remplacer un des réseaux par `0.0.0.0/0`, c'est-à-dire qu'on accepte de dialoguer avec n'importe qui, mais dans ce cas on utilisera sans doute une autre technique que le clé secrète partagée.
On lancera ce fichier par la commande :
`setkey -f <nom du fichier>`
La commande setkey permet également de voir ou vider les contenus des bases SPD et SAD :

```
setkey -DP:  Dumps all SPD entries
setkey -FP: Clear all SPD entries
setkey -D: Dump the SAD entries
setkey -F: Flush the SAD entries
```

Le fichier setkey.conf ayant été interprété, on doit déjà voir la politique avec setkey -DP. Les SA se lanceront à la demande.
Puis lancer `racoon`, de préférence avec l'option `-l <fichier de log>` pour la phase de mise au point ...
Il reste à essayer de communiquer à nouveau, et de vérifier que nos messages sont chiffrés, par tcpdump sur une passerelle, voir la différence entre un dump sur eth0 et eth1 par exemple lorsqu'on utilise le mode tunnel entre les passerelles.
On peut voir aussi dans le dump, si on est un peu patient, les échanges de clés ISAKMP lors des renouvellements périodiques. Remarquer le changement des SPI à cette occasion.
Vérifiez également la présence des SA avec setkey -D.

On pourra en outre, si on est téméraire, essayer de bloquer les pings (voire tout ICMP) entre les passerelles et vérifier que, encapsulés dans ESP, ils passent.
Si on est vraiment courageux, on peut essayer de mettre une politique par défaut de tout bloquer dans le firewall, et ne laisser passer que ce qu'il faut ... Il est conseillé d'observer ce qui doit passer avant.

Pour le mode transport, voici un exemple de fichier, où n et m sont les nombres distribués aux deux partenaires, a1 et b1 des adresses machines situées dans leurs réseaux respectifs :

```
#!/usr/sbin/setkey -f
flush;
spdflush;

spdadd 2001:660:7101:n::a1 2001:660:7101:m::b1 any -P out ipsec
       ipcomp/transport//use
       esp/transport/2001:660:7101:n::a1-2001:660:7101:m::b1/require;

spdadd 2001:660:7101:m::b1 2001:660:7101:n::a1 any -P in  ipsec
       ipcomp/transport//use
       esp/transport/2001:660:7101:m::b1-2001:660:7101:n::a1/require;
```

Une version simplifiée (vue sur kame.net mais non testée ...) :

```
#!/usr/sbin/setkey -f
flush;
spdflush;

spdadd 2001:660:7101:n::a1 2001:660:7101:m::b1 any -P out ipsec
       esp/transport//require;

spdadd 2001:660:7101:m::b1 2001:660:7101:n::a1 any -P in  ipsec
       esp/transport//require;
```

Il semble en effet logique, en mode transport, de n'avoir pas à spécifier les deux extrémités entre “transport” et “require” puisque ce sont la machine locale et la machine remote définies au dessus.