IPSec - Partie 1

Version du 30 septembre 2021, 15:47

Objectifs

L'objectif de ce TP est de vous familiariser avec le protocole IPSec et plus précisément avec la mise en place de VPN IPSec. DAns la première partie nous nous intéresserons à la configuration statique.

1 IPSec, Rappel

Selon la RFC de IETF : IPSec est un protocole de sécurité au sein de la couche réseau, qui a été développé pour fournir un service de sécurité à base de cryptographie, permettant de garantir l'authentification, l'intégrité, le contrôle d'accès et la confidentialité des données.

D'une manière plus commune : IPSec = formatage de trame permettant le chiffrement des données au niveau IP.

Il existe deux types de transformations de données pour IPSec :

1. Authentication Header (AH)
2. Encapsulating Security Payload (ESP)

1.1 Authentication Header (AH)

Paquet origine : IP Header + le reste du paquet (autres headers + payloads) = A

A l'aide de l'algorithme MD5, la clef secrète (connue des deux parties) + A on obtient :

IP Header + AH + reste du paquet (autres headers + payloads)

Avec AH : pas de confidentialité = pas de chiffrement

1.2 Encapsulating Security Payload (ESP) :

Paquet origine : IP Header + le reste du paquet (autres headers + payloads) = A

1. Si on est en mode transport (mode de bout en bout, entre hôtes) : On chiffre le "reste du paquet (autres headers + payloads)" avec la clef secrète, et on ajoute au paquet un header ESP.
   Le paquet devient donc :

IP Header + Header ESP + reste du paquet (autres headers + payloads) chiffré

2. Si on est en mode tunnel (avec des routeurs + des firewalls entre hôtes : Internet) : De nouveaux IP headers sont rajoutés lors du routage des paquets, mais le paquet origine est chiffré avec la clef secrète. Le IP header d'origine est conservé intacte dans la partie chiffrée.
   Le paquet devient donc :
   New Ip Header + Header ESP + Chiffré [Ip Header origine + reste du paquet (autres headers + payloads)]

Avec ESP : confidentialité car on a chiffrement des données.

1.3 Les associations de sécurité (SA) :

Une SA est une relation à sens unique (unilatérale) entre un émetteur et un destinataire. Elle définit l'ensemble des opérations IPSec devant être appliquées aux paquets. Une SA doit être définie dans chaque direction.

2 Revenons au TP

Dans un premier temps nous allons mettre en place un tunnel IPSec s'appuyant sur des clés statiques et reposant sur L2TP. Pour cela nous allons utiliser la commande `ip` du packetage `iproute2`. Dans cette première partie nous n'allons pas utiliser le protocole IKE et donc simplement mettre en place un tunnel reposant toujours sur les mêmes clés. Ce n'est donc pas la solution recommandée pour mettre en place des VPNs persistants mais plutot pour dépanner sur une courte période de temps.

Le but va être d'établir un tunnel IPSec en mode transport entre deux machines situées chacune derrière une passerelle. Pour cela vous pouvez soit utiliser une architecture virtualisée comme vu l'année dernière avec M. Bonnefoi, soit travailler avec des machines virtuelles Virtualbox. Il vous faudra au minimum trois machines : 2 pour les machines d'extrémité qui vont mettre en place le tunnel et 1 pour la passerelle. Il est néanmoins recommandé de mettre en place 2 passerelles.

Pour la gestion d'IPSec on utilisera la commande `ip xfrm` et plus précisément `ip xfrm state` qui va permettre de définir les SA et `ip xfrm policy` qui permet de définir les SP.

2.1 Préparation

Les adresses réseaux à utiliser seront le réseau 10.10.10.0/24 pour une des extrémités et 10.0.0.0/24 pour l'autre extrémité.

Pour la mise en place du tunnel, vous aurez besoin de 4 clés : 2 pour les SA et 2 pour les SPD (une paire pour chaque extrémité) que nous appellerons dans la suite `@K1, @K2, @K3, @K4`.

Pour cela, vous pouvez utiliser la commande `xxd` pour générer ces clés. Nous allons utiliser un chiffrement AES-256 pour notre tunnel donc nous aurons besoin de 4 clés de 32 bits ce qui peut s'obtenir grâce à la commande `xxd -p -l 32 -c 32 /dev/urandom`. La commande vous génère une sortie aléatoire sur 32bits. En la préfixant par $0x$ vous obtenez la clé à utiliser. Vous pouvez aussi passer par la commande `openssl rand -hex 32` si vous préférez.

Vous aurez également besoin de définir 2 identifiants appelés `reqid` qui permettent à iproute de faire le lien entre une SA et un SPD (et que nous appellerons dans la suite @Rid1 et @Rid2). Ces identifiants sont sur 4 octets (ex : 0x12345678) et peuvent être générés aléatoirement de la même manière que les clés (en remplaçant bien sur 32 par 4).

Sur le même principe vous aurez besoin de deux autres identifiants également sur 4 octets pour les SPI que nous appellerons @SPI1, @SPI2 dans la suite.

Une fois que vous avez vos clés et vos identifiants commencez par vider les bases SAD et SPD à l'aide des deux commandes suivantes :

```
ip xfrm state flush;
ip xfrm policy flush;
```

2.2 Mise en place du tunnel

Dans cette partie nous allons configurer les SA puis mettre en place la partie SPD pour établir le tunnel. **Il est fortement recommandé d'écrire des scripts pour exécuter les différentes commandes.**

Pour les SA saisissez sur chacunes des machines d'extrémité du tunnel les commandes suivantes :

```
ip xfrm state add src 10.10.10.1 dst 10.0.0.1 proto esp spi "@SPI1" reqid\
 "@Rid1" mode tunnel auth sha256 "@K1" enc aes "@K2"
ip xfrm state add src 10.0.0.1 dst 10.10.10.1 proto esp spi "@SPI2" reqid\
 "@Rid2" mode tunnel auth sha256 "@K3" enc aes "@K4"
```

Vous noterez l'utilisation d'AES-256 pour le chiffrement et de sha256 pour le controle d'intégrité.

Une fois les SA configurées, il faut ensuite passer aux SPD. Plus précisément, le rôle des SPD va être de définir quel traffic doit passer dans le tunnel. Dans notre cas c'est tout le traffic entre les deux machines d'extrémités. Notez qu'en général ces machines d'extrémités sont plutot des routeurs et que l'on définit plutot au niveau du SPD le routage entre les réseaux auxquels elles relient.

Sur la machine 10.0.0.1 :

```
ip xfrm policy add src 10.10.10.1 dst 10.0.0.1 dir in tmpl src 10.10.10.1\
```

```
 dst 10.0.0.1 proto esp reqid "@Rid1" mode tunnel
ip xfrm policy add src 10.0.0.1 dst 10.10.10.1 dir out tmpl src 10.0.0.1\
 dst 10.10.10.1 proto esp reqid "@Rid2" mode tunnel
```

Lorsque l'on interconnecte des réseaux il est souvent nécessaire d'ajouter une règle de forwarding (idem à la deuxième commande en remplacant out par fwd).

Sur l'autre machine d'extrémité, vous avez devez exécuter les mêmes commandes en inversant bien sur les entrées et les sorties.

A partir de là votre tunnel est configuré, il ne vous reste plus qu'à définir les règles de routage pour que le traffic passe par ce tunnel.

2.3 Vérification du fonctionnement

Mettez en place un échange de données entre les deux machines d'extrémité (ping, socat...) et capturez les paquets qui vont circuler au niveau du routeur intermédiaire à l'aide de wireshark (ou tcpdump au choix). Que voyez vous ? Conservez votre capture pour comparaison dans l'exercice précédent.

2.4 IPSec en mode transport

Après avoir remis à zéro vos configurations, reprennez les étapes précédentes en remplacant `tunnel` par `transport` dans les commandes précédentes.

Refaites ensuite une capture Wireshark. Quelles différences observez vous avec la précédente capture.