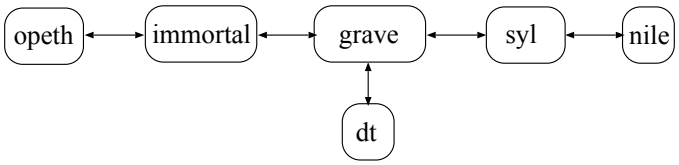
Université de Bordeaux — année 2019-2020

Master CSI/Master Info. — Sécurité des réseaux

TD - IPSec & `libreswan`

Le but de ce TP est de mettre en place un tunnel IPSEC entre deux machines distantes afin de créer un VPN (réseau privé virtuel entre ces deux machines). Ce mécanisme sera mis en place entre les machines immortal et syl de la plate-forme décrite ci-dessous :



La topologie réseau correspondante peut être obtenue en lançant le script de démarrage `/net/stockage/aguermou/SR/TP/8/qemunet.sh` en lui fournissant la description de la topologie réseau à l'aide de l'option `-t` ainsi que l'archive contenant la configuration initialte des machines à l'aide de l'option `-a`. Ceci revient à lancer les commandes suivantes :

```
cd /net/stockage/aguermou/SR/TP/8/; ./qemunet.sh -x -t topology -a archive_tp8.tgz
```

1. Nous allons tout d'abord générer trois certificats X509 pour dt, immortal et syl. Attention, il est important de donner un numéro de série différent pour chacun des certificats.

   (a) Nous allons commencer par créer la clé privée de la CA :

   ```
   certtool --generate-privkey --outfile ca-key.pem
   ```

   (b) Puis nous allons générer le certificat de cette dernière (Attention le champ Common Name doit absolument être saisi).

   ```
   certtool --generate-self-signed --load-privkey ca-key.pem --outfile \
      ca-cert.pem
   ```

   (c) Ensuite, nous allons générer une clé privée pour l'une de nos machines.

   ```
   certtool --generate-privkey --outfile key.pem
   ```

   (d) Enfin, nous allons générer et signer le certificat :

   ```
   certtool --generate-certificate --load-privkey key.pem --outfile \
      cert.pem --load-ca-certificate ca-cert.pem  --load-ca-privkey \
      ca-key.pem
   ```

   Le contenu du certificat peut être consulté soit en éditant le fichier, ou en utilisant la commande :

   ```
   certtool --certificate-info --infile cert.pem
   ```

   Attention : Pour que l'étape suivante fonctionne correctement, vous devez lors de la génération des certificats :
   — Activer les extension TLS web server et TLS web Client pour que les programmes de test `gnutls-cli` et `gnutls-serv` soient fonctionnels.
   — Saisir l'adresse IP du propriétaire lorsque `certtool` le demande.

2. Nous allons maitenant vérifier que nos certificats ont bien été générés. Pour se faire nous allons utiliser les outils `gnutls-cli` et `gnutls-serv`. L'idée et de lancer un serveur *https* léger sur une de nos machines (avec `gnutls-serv` sur immortal par exemple) et de faire communiquer nos autres machines (avec `gnutls-cli` sur syl et dt) avec ce serveur. Bien entendu, il faut demander au serveur (resp. client) d'utiliser le couple certificat/clé privée de la machine associée. Il est important qu'à l'issue de cette vous n'observiez aucune erreur.
3. Le daemon utilisé par `libreswan` requiert que les certificats et clés connus par une machine soit stockés dans un base NSS de type SQL. Cette base de donnée est stockée dans notre cas dans le chemin suivant : `/var/lib/ipsec/nss`. Pour ce faire, nous allons tout d'abord convertir les fichiers associés à chaque machine pour les mettre au formt PKCS#12. Un exemple pour `syl` est fourni ci-dessous.

   ```
   certtool --load-certificate syl-cert.pem --load-privkey syl-key.pem\
     --load-ca-certificate ca-cert.pem --to-p12 --outder --outfile syl.p12
   ```

   Bien entendu, si sur une machine donnée vous n'avez accès à la clé privée d'une autre machine, il suffit de supprimer l'option `--load-privkey`.
   Il faut tout d'abord initialiser la base :

   ```
   rm -f /var/lib/ipsec/nss/*; ipsec initnss
   ```

   Enuite, il suffit d'ajouter le couple de clé au format PCKS#12 à la base IPSec. Ceci peut être fait de la manière suivante :

   ```
   ipsec import syl.p12
   ```

   Il est à noter qu'on peut consulter la liste des certificats contenus dans la base de donnée de la manière suivante :

   ```
   certutil -L -d sql:/var/lib/ipsec/nss
   ```

   Pensez donc à vérifier les *trust attributes* de chaque entrée. Ils doivent être de la forme :
   — `'u,u,u'` pour l'entrée correspondant à la machine courante.
   — `'CT,,'` pour l'autorité de certification.
   — `'P,,'` pour le certificat de la machine avec laquelle vous souhaitez interagir.
   Il est possible de modifier les flags d'une entrée en utilisant la commande suivante :

   ```
   certutil -M -d sql:/var/lib/ipsec/nss -n <id> -t <flags>
   ```

   De même, lister la liste des clé privées connues peut être obtenue de la manière suivante ;

   ```
   certutil -K -d sql:/var/lib/ipsec/nss
   ```

   Enfin pour supprimer une clé, il suffit d'exécuter la commande suivante

   ```
   certutil -D <nom> -d sql:/var/lib/ipsec/nss
   ```

4. Il faut ensuite spécifier au moteur IPSec quelles sont les clés privées connues en fournissant la passphrase asscociée. Ceci se fera en modifiant le fichier `/etc/ipsec.secrets` en y ajoutant une ligne ayant le format suivant :

   ```
   : RSA <id>
   ```

   Où id représente le nom que vous avez donné à l'objet cryptographique lors de la création du fichier PKCS#12.
5. La configuration de `libreswan` se fait par le biais du fichier `/etc/ipsec.conf`. Configurer un premier tunnel en utilisant la configuration suivante (vu par immortal, il faut faire le symétrique sur syl) :

```
version 2.0

config setup
        plutodebug = all
        plutostderrlog=/var/log/pluto.log
        protostack=netkey
        nat_traversal=no
        nhelpers=0

conn tunnelipsec
        type=           tunnel
        left=           @IP externe immortal
        leftsubnet=     @IP reseau interne immortal
        leftcert=       <id_immortal>
        right=          @IP externe syl
        rightsubnet=    @IP reseau interne syl
        rightcert=      <id_syl>
        phase2alg=            alg. chiffrement-alg. hachage #(par exemple aes-sha1)
```

(attention l'indentation est importante).

Une fois la configuration terminée, il faut :

(a) redémarrer le démon `libreswan` : `service ipsec restart`

(b) ajouter la connexion sécurisée : `ipsec auto --add tunnelipsec`

(c) activer la connexion sécurisée : `ipsec auto --up tunnelipsec`

(d) on peut maintenant communiquer en sécurisé (on peut regarder l'état de la connexion avec ipsec auto)

6. Un exemple de configuration pour ce fichier est disponible dans le dossier `/net/stockage/-aguermou/SR/ipsec/roadwarrior`. La configuration correspond au cas d'un client isolé (dt) qui veut se rattacher à un réseau existant par le biais d'un tunnel IPSec. Testez ces deux configurations. Regarder ce qui se passe sur le réseau lorsque nile ou opeth communique avec dt.