MISE EN PLACE DE IPSEC

Crééz un fichier texte (.txt) à votre nom. Vous y placerez toutes les commandes que vous aurez saisies pour effectuer les différents exercices. Vous y placerez aussi la réponse aux questions posées dans certains exercices. N'oubliez pas de préciser le numéro de l'exercice

Le but de ce travail est de mettre en place le protocole IPSEC. Avant toute chose, vérifiez que le paquetage `ipsec-tools` est bien installé sur votre machine :
`% rpm -qa | grep -i ipsec-tools`

L'ensemble du travail devra être effectué avec les droits de l'administrateur système.

1. Une première approche

La configuration d'une SA et d'une SP se fait via un script dont la première ligne est : `#!/sbin/setkey -f`. Le script commencera ensuite par les deux lignes suivantes :
`flush;`
`spdflush;`
qui permettent de détruire toute SA ou SP présente sur la machine. Consultez le manuel de la commande `setkey` afin de connaître la syntaxe de la commande `add` qui permet de rajouter une SA pour la communication d'un hôte source vers un hôte destination. La commande `add` permet de rajouter une SA de type AH ou ESP. Chaque SA est identifiée par un numéro unique (SPI) que vous choisirez, ce numéro doit être identique sur l'hôte source et l'hôte destination. Ne vous préoccupez pas du champ extension.

Exercice 1. Définissez sur votre poste une SA comportant 2 règles permettant de communiquer avec un autre poste en utilisant AH avec hmac-sha1 (clé de 160 bits) et d'utiliser ESP avec rijndael-cbc (clé de 128 bits). Pour engendrer une clé aléatoire utilisez la commande dd avec pour périphérique d'entrée /dev/random et combinez le tout avec la commande od pour obtenir un entier sous forme hexadécimale.

Consultez dans le manuel de `setkey` la syntaxe de `spdadd` qui permet de configurer une politique de sécurité.

Exercice 2. Rajoutez dans votre script une politique de sécurité spécifiant que pour tout protocole, tous les paquets sortant de votre machine à destination de l'hôte distant doivent subir un traitement ipsec combinant ah et esp en mode transport. Ne tenez pas compte de la notion de priorité qui est optionelle. Lancez votre script. Vérifiez avec la commande setkey (options -D et -DP) que votre SA et votre SP sont opérationelles. Lancez ethereal et effectuez un ping vers l'hote distant. Observez les trames pour constater qu'elles sont encapsulées par IPsec.
L'hôte distant ne répond pas. Pourquoi ?
L'ordre dans lequel vous avez défini les régles pour votre SP est-il important ? (échangez les régles et observez les trames.)

Afin que l'hôte distant puisse répondre, il faut qu'il possède une SA identique à celle que vous avez mis en place sur votre poste.

Exercice 3. Créez sur le poste distant une SA permettant à ce dernier de vérifier et déchiffrer tous les paquets provenants de votre poste. Relancez le ping et observez ce qui se passe. La communication est-elle chiffrée dans les 2 sens ?

Dans la configuration que vous utilisez actuellement, le poste distant sait comment contrôler et déchiffrer les

Date: Novembre 2009.

trames provenant de votre poste. Cependant, il acceptera aussi de la part de votre poste tout traffic non chiffré et non authentifié. Quelqu'un usurpant votre adresse IP pourra donc correspondre avec le poste distant. Pour vérifier cela désactivez avec la commande `setkey` votre SA et votre SP (options -F et -FP) sur votre poste et vérifiez que vous pouvez envoyer un ping vers le poste distant.

Exercice 4. Rajoutez sur votre poste client une politique de sécurité afin que seuls les paquets authentifiés et chiffrés provenants de votre poste soient acceptés. Activez IPsec uniquement sur le poste distant et vérifiez qu'en lançant un ping de votre poste vers le poste distant, la communication ne s'effectue pas. Activez alors IPsec sur votre poste et réessayez. Observez les trames échangées. Sont-elles chiffrées dans les 2 sens ?

Pour que le traffic soit chiffré du poste distant vers votre poste il va falloir définir de nouvelles SA et SP sur les deux machines.

Exercice 5. Rajoutez sur le poste distant une SA indiquant les méthodes et clés à utiliser pour AH et ESP lors de l'envoi de paquet vers votre poste. Cette SA doit aussi figurer sur votre poste afin que ce dernier puisse vérifer et déchiffrer les paquets qu'il recevra. Il est recommandé d'utiliser des clés différentes de celles utilisées dans le sens votre poste → poste distant. Rajoutez ensuite sur le poste distant une politique de sécurité indiquant que tous les paquets sortant à destination de votre poste doivent être authentifiés et chiffrés. Placez alors sur votre poste une politique de sécurité stipulant que seuls les paquets entrants chiffrés et authentifiés provenants du poste distant seront acceptés. Vérifiez alors que tous les échanges sont à présent chiffrés.

2. Automatisation de la gestion des clés

Le contexte dans lequel vous avez travaillé oblige l'administrateur système de votre poste et celui du poste distant à se mettre d'accord sur un ensemble de clés et d'algorithmes à utiliser et à placer eux-mêmes ces informations dans les fichiers de configuration. Nous allons voir à présent comment automatiser cette phase. C'est le protocole IKE qui se charge de négocier entre les deux machines les paramètres de sécurité, chacune d'entre elles pouvant émettre des voeux quant aux différents algorithmes à utiliser. Le démon correspondant à ce protocole sous Linux s'appelle `racoon`. Par rapport au travail réalisé précédemment, vos fichiers de configuration ne devront contenir que la politique de sécurité, les différentes SA seront négociées entre chaque machine via le protocole IKE. Avant d'entamer la phase de négociation des paramètres de sécurité, les deux machines doivent s'authentifier mutuellement. Pour l'instant nous gérerons cette authentification en utilisant un secret partagé (un mot de passe) entre les deux machines.

Exercice 6. Modifiez les fichiers de configuration sur votre poste et l'hôte distant de façon à ce qu'il ne reste que les politiques de sécurité permettants des échanges authentifiés et chiffrés entre les deux machines. Dans le répertoire /etc/racoon de chaque machine, éditez le fichier psk.txt (pre-shared key) et placez-y l'adresse IP de la machine avec qui vous voulez communiquer suivi d'un mot de passe commun.
Le démon racoon s'appuie sur le fichier de configuration /etc/racoon/racoon.conf. Editez ce dernier, il doit comporter 2 sections :

- remote : section permettant de fixer les paramètres de la phase 1 de IKE;
- sainfo : section permettant de fixer les paramètres de la phase 2 de IKE.

Consultez le manuel de racoon pour remplir ces sections.

Concernant la section remote vous ne vous intéresserez qu'aux options : exchange_mode, my_identifier, peers_identifier, verify_identifier et proposal. Configurez ces dernières de façon à ce que chacun des postes envoie comme information à l'autre son adresse IP et que chaque poste vérifie cette information. La phase 1 devra utiliser le 3des et sha1. L'authentification mutuelle se fera par secret partagé et l'échange de clés Diffie-Helmann devra manipuler des entiers de 2048 bits. La phase 1 devra être renégociée toutes les 4 minutes.

Concernant la section sainfo, un exemple est donné dans le fichier /etc/racoon/racoon.conf. Assurez-vous que le rijndael et hmac-sha1 sont proposés et que la méthode Diffie-Helmann manipule des entiers de 2048 bits. La phase 2

devra être renégociée toutes les 2 minutes.

Exécutez sur chaque machine votre politique de sécurité, puis lancez sur les 2 machines la commande racoon (avec la commande -F pour qu'elle tourne en avant-plan afin d'observer les échanges). Lancez alors un ping et vérifiez le bon déroulement des communications.

3. Authentification à base de certificats

La méthode étudiée précédemment repose sur l'existence d'un secret partagé entre les deux postes. Ce dernier permet entre autre aux deux machines communicantes de s'authentifier mutuellement. La mise au point d'un secret partagé et sa distribution (sécurisée !) entre plusieurs machines géographiquement éloignées ne permet pas d'envisager d'utiliser une telle solution pour un ensemble de machines. IPsec permet de réaliser la phase d'authentification mutuelle à l'aide de certificats X.509.

Exercice 7. Utilisez la commande openssl pour créer pour chaque machine :

. une clé secrète RSA de 1024 bits non protégée et la clé publique correspondante,
. une requête de certificat pour la clé publique créée utilisant SHA1 pour fonction de hachage.

Remarque: Ces deux étapes se font simultanément en utilisant la commande req de openssl.

Récupérez sur http://veron.univ-tln.fr/ENSEIGNEMENT/M2/ le certificat et la clé privée de l'autorité de certification afin de pouvoir signer vos requêtes et obtenir ainsi un certificat pour chaque clé publique engendrée. Le mot de passe protégeant la clé privée de la CA est *bonjour*.

Placez sur chaque poste dans /etc/racoon/certs :

. la clé privée du poste,
. le certificat de la clé publique correspondante,
. le certificat de l'autorité.
. le certificat de votre correspondant.

Editez ensuite sur chacun des postes le fichier de configuration de racoon. Vous devrez :

. modifier les champs my_identifier et peers_identifier afin de spécifier que l'identité est au format ASN.1 (codage utilisé pour représenter un nom dans un certificat).
. ajouter trois lignes stipulant les noms des fichiers contenant votre certificat et votre clé privée,le nom du fichier contenant le certificat de l'autorité et le nom du fichier contenant le certificat de votre correspondant.
. préciser que l'authentification utilise maintenant une méthode à base de signature RSA.

Relancez racoon et testez votre configuration.