Principes et pratiques de l'administration des systèmes

Module 3I015

Semaine 7

1. Sécurisation par iptables

Étape 1 – Quotas

Pour pimenter le TME, installez sur vos 3 machines virtuelles via `apt` le package `quotatool`, qui propose des fonctionnalités pour gérer les quotas des utilisateurs sur disque. Sur vos machines msLDAP et ssLDAP, vous allez mettre en place des quotas sur les *partitions locales* (celles en `/dev/sdaX`, il n'y a en effet pas besoin de quotas sur les partitions montées par NFS puisque celles-ci ont déjà des quotas en tant que partitions locales sur leurs serveurs NFS). Pour cela, rajoutez dans les `/etc/fstab` des deux machines l'option `usrquota` après les options de montage courantes des partitions locales `/users`, `/nfs_bin` et `/nfs_tmp` (probablement, ces options se limitent actuellement à `defaults`). Ne rajoutez pas l'option `usrquota` à `/backup` car elle n'est pas compatible avec `btrfs`. Pour que ces nouvelles options soient prises en compte sans avoir à faire des montages/démontages manuels, rebootez vos 2 machines.

Sur les deux machines, il faut maintenant créer un fichier de quota. Pour cela, arrêtez leurs serveurs NFS et utilisez la commande `quotacheck -um point_de_montage` où `point_de_montage` correspond au point de montage de votre partition locale. Par exemple, sur msLDAP, cela reviendra à taper `quotacheck -um /users`. Effectuez cette opération sur toutes les partitions locales qui sont montées avec l'option `usrquota`.

Enfin, il ne reste plus qu'à activer sur chaque point de montage local les quotas via la commande `quotaon -uv point_de_montage` (par exemple, `quotaon -uv /users`). Voila, vos quotas sont mis en place. Pour l'instant, vous n'avez pas encore limité la place disponible pour les utilisateurs et ce n'est pas l'objet de ce TME. Si vous souhaitez le faire par la suite, vous pourrez tester la commande :

`edquota -u nom_de_user`.

Par exemple, `edquota` pourrait produire un affichage similaire à :

```
Quotas disque pour user student3 (uid 2000) :
 Système de fichiers           blocs       souple     stricte   inodes    souple   stricte
  /dev/sda3                      1716          0        1800       88        92        95
```

Cela indique qu'actuellement `student3` a créé 88 fichiers/inodes, totalisant 1716 kilo octets. `student3` ne pourra jamais écrire plus de 1800 kilo octets sur `/users`. Il ne pourra jamais utiliser plus de 95 inodes. En principe, il est censé utiliser au plus 92 inodes mais il est autorisé pendant un certain temps à dépasser 92 inodes (en restant dans la limite des 95 inodes) et, passé ce délai, il ne sera plus autorisé à dépasser les 92 inodes.

Afin que les quotas soient accessibles via les machines distantes, démarrez via `systemctl` le service `quotarpc.service` et pérennisez le démarrage de ce service.

Étape 2 – Sécurisation : arrêt des serveurs

Afin de sécuriser vos machines, vous allez commencer par stopper toutes leurs communications. Les clients (NFS, LDAP) que vous avez mis en place les semaines précédentes vont moyennement apprécier cela. Donc, dans un premier temps, vous allez démonter via la commande `umount` les répertoires distants que vous avez montés par NFS sur msLDAP, ssLDAP et c1LDAP. De même, éditez les fichiers `/etc/nsswitch.conf` des trois machines et supprimez les références à « `ldap` » puis arrêtez les services `slapd` de msLDAP et ssLDAP. Maintenant, vos machines n'ont plus aucune raison de communiquer les unes avec les autres.

Étape 3 – Paranoïa absolue sur msLDAP

Sur msLDAP, vous allez placer vos règles d'iptables dans un fichier `/etc/iptables`. En principe, dans les distributions Debian, ce fichier n'existe pas et il faut le créer. Ici, il l'est déjà : l'équipe enseignante a en effet créé des règles pour la table « `nat` », qui permet à votre passerelle ssLDAP de faire ce que l'on appelle du « masquerading ». Cela sert à faire en sorte que les machines de votre réseau local, qui possèdent des adresses IP non routables sur internet (et qui ne peuvent donc absolument pas communiquer avec d'autres machines sur internet), puissent tout de même communiquer avec internet. L'idée est la suivante : quand msLDAP *pinge* `google.fr`, msLDAP, d'adresse IP 192.168.X.1, envoie le paquet correspondant à ssLDAP, sa passerelle. ssLDAP devrait *forwarder* ce paquet à `google.fr` mais si elle faisait cela directement, `google.fr` devrait renvoyer la réponse vers la machine 192.168.X.1. Or, il lui est impossible de faire cette opération car les adresses débutant par 192.168 ne sont pas routables sur internet. Au lieu de cela, ssLDAP exploite l'IP *masquerading* et forwarde le paquet en indiquant que le paquet provient de sa 2ème carte réseau. Vous avez configuré celle-ci avec un mode d'accès NAT, ce qui signifie que votre 2ème carte réseau va transmettre les paquets à un « routeur » virtuel de VirtualBox qui, lui, va transmettre les paquets sur internet via la carte réseau de votre machine hôte en utilisant également de l'IP *masquerading*. Autrement dit, quand le paquet sort de votre machine hôte et transite sur internet, il est indiqué qu'il provient de l'adresse IP de votre machine hôte qui, elle, possède une adresse routable sur internet. Étant donné que cette adresse est routable, `google.fr` peut renvoyer la réponse à votre machine hôte qui la rentransmet au routeur, qui la retransmet à son tour à ssLDAP et, enfin, celle-ci retransmet alors cette réponse à msLDAP. Vous le comprenez, toute l'idée est de faire en sorte que, quand les paquets sortent de votre machine hôte, ils sont indiqués comme étant issus d'une machine d'adresse IP routable sur internet.

Le fichier `/etc/iptables` étant déjà créé, vous n'aurez par la suite qu'à mettre à jour le contenu de sa table « `filter` », qui a été initialisé à `ACCEPT` pour tous les paquets.

Faites maintenant en sorte que les politiques par défaut des chaînes `INPUT`, `OUTPUT` et `FORWARD` soient égales à « `DROP` » (cf. le transparent « Exemple d'utilisation (2/6) »). Mettez à jour les iptables utilisées par votre système en utilisant la commande :

```
iptables-restore < /etc/iptables
```

Vérifiez que tout s'est bien passé en listant ses règles via la commande `iptables -L`. Vous devriez observer un affichage similaire à :

```
[root@msLDAP ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
```

Testez le bon fonctionnement en `ping`ant votre loopback (127.0.0.1). Le `ping` devrait indiquer que votre machine n'arrive plus à communiquer avec son loopback :

```
[root@msLDAP ~]# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

Étape 4 – Ce sont les autres, les méchants

Rééditez le fichier `/etc/iptables` de msLDAP et ajoutez maintenant des règles afin d'accepter tous les messages envoyés au loopback (interface `lo`) ainsi que ceux que lui-même envoie. **Attention :** si votre machine communique avec son loopback, elle lui envoie des messages et le loopback lui répond, ce qui implique que vous devez ajouter une règle pour la chaîne `INPUT` pour permettre les envois de paquets vers le loopback ainsi qu'une règle pour la chaîne `OUTPUT` pour permettre au loopback de vous envoyer des paquets.

Évidemment, n'oubliez pas de refaire un `iptables-restore < /etc/iptables` afin que vos modifications soient prises en compte. Maintenant, msLDAP ne peut toujours pas `ping`er ssLDAP, mais il peut `ping`er 127.0.0.1.

Étape 5 – Mode paranoïa sur c1LDAP et ssLDAP

Sur c1LDAP et ssLDAP, faites comme pour msLDAP : ne conservez que des politiques par défaut égales à `DROP` et n'autorisez que les communications sur les loopbacks.

Étape 6 – Ping pour tout le monde

La commande `ping` utilise le protocole `icmp` (Internet Control Message Protocol). Autorisez sur toutes les machines les `ping` provenant de n'importe quelle machine en rajoutant des règles indiquant que tous les paquets de type `ICMP` doivent être acceptés. **Attention :** n'oubliez pas que, quand vous faites un `ping`, vous envoyez un paquet vers une machine (donc vous utilisez votre chaîne `OUTPUT`) et vous récupérez sa réponse (pour cela, vous utilisez votre chaîne `INPUT`).

Vérifiez que vos règles ont bien été prises en compte en exécutant des `ping` vers les autres machines de votre réseau.

Étape 7 – Rétablissement de nfs : `/users`

Le fichier `/etc/services` contient les numéros de port ainsi que les protocoles (TCP, UDP, ICMP) utilisés par vos services. Recherchez dans ce fichier quels sont les ports TCP et/ou UDP utilisés par votre serveur NFS. On souhaite ajouter sur msLDAP les règles qui vont permettre à votre serveur d'interagir avec ses clients. *Rappel de cours :* quand deux machines communiquent, chacune ouvre un port pour cette communication. Par exemple, si vous accédez via votre browser habituel à google.fr, celui-ci va contacter le port 80 de la machine google.fr (cf. le fichier `/etc/services`) et pour que google puisse lui répondre, votre machine va également affecter à cette communication un port (≥ 1024). En principe, on ne sait pas quel port votre browser va utiliser (cela change en fonction des requêtes faites aux différents services, de l'ordre de celles-ci, *etc.*). Par conséquent, dans les règles iptables, on ne spécifie que le port du serveur, pas celui utilisé par le client. Pour permettre à toutes les machines de votre réseau d'accéder à votre serveur NFS sur msLDAP, vous devez donc rajouter :

- sur la chaîne `INPUT` de msLDAP, une règle spécifiant que vous acceptez les paquets provenant des machines de votre réseau ($192.168.X.0/24$) et à destination de votre serveur NFS (spécifié par le protocole qu'il utilise ainsi que par son numéro de port) ;
- sur la chaîne `OUTPUT` de msLDAP, une règle spécifiant que vous acceptez les paquets à destination des machines de votre réseau ($192.168.X.0/24$) et provenant de votre serveur NFS (là

encore, spécifié par le protocole qu'il utilise ainsi que par son numéro de port).
— sur les chaînes `INPUT` de ssLDAP et c1LDAP, une règle spécifiant que vous acceptez les paquets provenant du serveur NFS de msLDAP (spécifié par l'adresse IP de msLDAP, le protocole utilisé par le serveur ainsi que par son numéro de port) ;
— sur les chaînes `OUTPUT` de ssLDAP et c1LDAP, une règle spécifiant que vous acceptez les paquets à destination du serveur NFS de msLDAP (spécifié par l'adresse IP de msLDAP, le protocole utilisé par le serveur ainsi que par son numéro de port).

Vous pouvez maintenant tester si cela a bien fonctionné en effectuant un `mount /users` sur ssLDAP et en testant si vous pouvez lire des fichiers dans `/users` à partir de ssLDAP (n'oubliez pas de redémarrer vos serveur NFS s'ils sont éteints). Lorsque cela fonctionne, vous pouvez reproduire la même opération sur c1LDAP.

Moralité de cette étape : en utilisant le contenu du fichier `/etc/services`, on peut aisément déduire les règles à ajouter dans les iptables afin de permettre l'accès aux différents services que l'on propose ou utilise.

Étape 8 – Rétablissement du serveur NFS de ssLDAP

Ajoutez les règles permettant les accès NFS à `/backup`, `/nfs_tmp` et `/nfs_bin` que vous avez exportés/importés les semaines précédentes.

Testez si vous pouvez bien monter ces répertoires par NFS. Si vous n'arrivez pas à les monter, effectuez l'étape 9.

Étape 9 – Rétablissement du serveur NFS de ssLDAP

À première vue, il peut être étrange d'arriver à réaliser les montages NFS de `/users` et pas ceux de `/backup`. La raison en est la suivante : sur msLDAP, votre serveur NFS n'exporte qu'un seul répertoire. À ce titre, il peut fonctionner en NFSv4 et c'est effectivement en version 4 qu'il fonctionne. En revanche, sur ssLDAP, plusieurs répertoires sont exportés. Or, pour qu'ils le soient en NFSv4, il faut satisfaire certaines contraintes, notamment le fait que ces répertoires doivent être issus d'une même hiérarchie de répertoires, dont la racine est elle-même exportée avec une option « `fsid=0` ». Comme ce n'est pas le cas sur ssLDAP, le serveur NFS de ssLDAP fonctionne en NFSv3. Une des différences entre ces deux versions est que l'on peut contacter directement un serveur en v4 en TCP sur son port 2049, ce qui n'est pas le cas pour la v3 et c'est pour cela que vos montages NFS de `/backup`, `/nfs_bin` et `/nfs_tmp` échouent.

Mais alors, comment faire pour trouver les ports qui sont utilisés par NFSv3 avant que le port 2049 soit effectivement exploité ? Ici, plusieurs solutions s'offrent à vous : la première consiste à se renseigner sur internet et vous trouverez rapidement les règles à rajouter. Une autre solution, une astuce qui fonctionnera dans tous les cas, consiste à rajouter des règles de log : dans le fichier `/etc/iptables` de ssLDAP, **à la fin** des règles de `filter` juste avant son `COMMIT`, rajoutez une règle pour la chaîne `INPUT` ainsi qu'une règle pour la chaîne `OUTPUT` indiquant que tout paquet envoyé vers ou provenant de msLDAP doit être loggué (action = `LOG`, comme vu en cours). Il est important de placer cette règle juste avant le `COMMIT` car elle ne s'activera que si toutes les règles précédentes échouent, autrement dit que lorsque vous essayerez de vous connecter à un port que vous n'avez pas encore autorisé (celui-là même qui bloque vos montages NFS). Faites les mêmes opérations sur msLDAP. Votre démon `syslogd` placera alors les logs que vous aurez générés dans le fichier `/var/log/kern.log` qui correspond aux messages de votre noyau (kernel) linux. Pour voir ces logs au fur et à mesure qu'ils sont écrits dans le fichier, nous vous suggérons d'ouvrir une nouvelle console, de passer `root` et de taper la commande

`tail -f /var/log/kern.log`. Sur msLDAP, exécutez la commande `mount /backup` et observez les logs. Vous devriez voir l'apparition de nouvelles lignes similaires à :

```
Aug 12 11:26:59 msLDAP kernel: [ 3785.448854]
IN= OUT=enp0s3 SRC=192.168.X.1 DST=192.168.X.2 LEN=84 TOS=0x00 PREC=0x00 TTL=64
ID=42239 DF PROTO=UDP SPT=50666 DPT=111 LEN=64
Aug 12 11:26:59 msLDAP kernel: [ 3785.449208]
IN= OUT=enp0s3 SRC=192.168.X.1 DST=192.168.X.2 LEN=84 TOS=0x00 PREC=0x00 TTL=64
ID=42240 DF PROTO=UDP SPT=58903 DPT=111 LEN=64
```

Les champs `IN` et `OUT` indiquent s'il s'agit de log sur la chaîne `INPUT` ou bien `OUTPUT`. Étant donné que vous envoyez une requête `mount` vers ssLDAP, on peut observer que seul `OUT` est renseigné. Les champs `SRC` et `DST` sont les adresses IP des machines source et destination du paquet, et `SPT` et `DPT` les ports correspondants. Ici, comme votre paquet est envoyé vers ssLDAP, seul les champs `DST` et `DPT` s'avéreront utiles pour vos iptables. Notez que les ports élevés correspondent aux ports locaux (ici le champ `SPT`) et ne doivent pas être utilisés dans vos règles iptables car ils changent à chaque connexion. Manifestement, ici, msLDAP essaye d'envoyer un paquet vers le port 111 de ssLDAP avec le protocole `UDP` (champ `PROTO`). Rajoutez les règles dans vos iptables afin de pouvoir accéder au service du port 111 (comme d'habitude, il doit y avoir des règles à la fois sur msLDAP et ssLDAP afin que les deux machines communiquent entre elles).

Réessayez la commande `mount /backup` sur msLDAP. Elle ne fonctionne toujours pas. On pourrait réutiliser les logs afin de rajouter les règles nécessaires pour que `mount` fonctionne (il ne reste qu'un seul autre port à ouvrir). Malheureusement, cela ne fonctionnera probablement que jusqu'à extinction de vos machines : au prochain démarrage, `mount` produirait probablement des erreurs. La cause provient de ce que représente le port 111 : il s'agit de ce que l'on appelle le `portmapper`, le service RPC (*Remote Procedure Call*). Celui-ci a pour tâche, entre autres, d'affecter dynamiquement des ports aux services qui dépendent de lui et dont fait partie le service NFSv3. Par conséquent, au prochain redémarrage, il est fort probable que le `portmapper` affecte à `mountd`, une des parties de votre serveur NFS, un nouveau numéro de port. L'étape suivante vous permettra de résoudre ce problème.

Moralité de cette étape : en vous appuyant sur les logs générés par les iptables, vous pouvez déduire les services "auxiliaires" dont il est nécessaire de permettre l'accès pour que les services qui vous intéressent soient également accessibles.

Étape 10 – Les services RPC

Sur ssLDAP, faites un `man mountd` pour comprendre comment fonctionne ce démon. Observez que ce démon est bien décrit comme un service RPC. Tapez la commande `rpcinfo -p` pour voir la liste des services s'appuyant sur RPC qui sont démarrés sur votre machine. Vous devez observer un affichage similaire à :

```
[root@msLDAP /]# rpcinfo -p
program  vers  proto   port  service
 100000     4    tcp    111  portmapper
 100000     3    tcp    111  portmapper
 100000     4    udp    111  portmapper
 100000     3    udp    111  portmapper
 100024     1    udp  48873  status
 100024     1    tcp  60222  status
 100011     1    udp    799  rquotad
 100011     1    tcp    799  rquotad
 100021     4    udp  32769  nlockmgr
 100021     4    tcp  32803  nlockmgr
 100003     4    tcp   2049  nfs
 100003     4    udp   2049  nfs
 100005     3    udp  47234  mountd
 100005     3    tcp  47635  mountd
```

On peut y voir, notamment, les ports utilisés par le `portmapper` (le 111 que vous aviez dû utiliser) ainsi que celui actuellement utilisé par `mountd`. Le fait que le numéro de port de `mountd` change dynamiquement est un peu gênant vis à vis des iptables. Mais, fort heureusement, on peut demander à `mountd` d'utiliser un port fixé. Pour cela, éditez le fichier `/etc/default/nfs-kernel-server` et modifiez la ligne `RPCMOUNTDOPTS` en :

```
RPCMOUNTDOPTS="-manage-gids -p 3000"
```

Cela passera l'option `-p` à `mountd` qui lui indiquera d'écouter le numéro de port 3000 (cf. `man mountd`). Redémarrez via `systemctl` le service `nfs-kernel-service.service`. Si vous effectuez un `rpcinfo -p`, vous pourrez observer que `mountd` utilise maintenant le port 3000. Maintenant que vous avez la certitude que c'est le port 3000 qui est utilisé par le serveur `mountd` de ssLDAP, rajoutez les règles iptables qui vont permettre à la commande `mount` de fonctionner sur vos trois machines virtuelles. Lorsque ce sera la cas, vos montages NFS réussiront.

Moralités de cette étape : la commande `rpcinfo -p` vous permet d'accéder à des informations complémentaires du fichier `/etc/services` pour établir les règles de vos iptables concernant les services s'appuyant sur RPC.

De plus, les fichiers de configuration des services dépendant de RPC permettent souvent d'affecter « en dur » des numéros de port à écouter. Pour cela, il faut en général éditer leur fichier de configuration qui se trouve dans le répertoire `/etc/default`.

Étape 11 – Rétablissement de LDAP

Comme pour nfs, recherchez les ports utilisés par LDAP dans le fichier `/etc/services` et ajoutez les règles qui vous permettront de faire fonctionner vos services LDAP. ssLDAP étant un client de msLDAP et en même temps un serveur (esclave), il faudra configurer les règles de cette machine à la fois en tant que serveur LDAP et en tant que client.

Lorsque vous aurez effectué un `iptables-restore` sur vos 3 machines, démarrez les services `slapd` sur msLDAP et ssLDAP puis rétablissez les règles ldap dans les fichiers `/etc/nsswitch.conf` des 3 machines.

Testez que votre installation fonctionne bien en vous logguant sur c1LDAP en tant que `student3`.

Étape 12 – Et les quotas dans tout ça ?

Placez-vous maintenant sur ssLDAP et tapez la commande `quota` afin d'afficher les limites d'espace disque imposées aux utilisateurs. Vous pourrez constater que `quota` émet un message d'erreur du type :

```
quota: error while getting quota from msLDAP:/users for root (id 0): Connexion refusée
```

car il n'est pas encore autorisé à converser avec msLDAP. On souhaite régler ce problème. Là encore, regardez dans `/etc/services` quels sont les ports TCP et/ou UDP utilisés par le service de `quota`.

Vous ne trouvez pas de référence à quota dans `/etc/services` ? Qu'à cela ne tienne, regardez les logs obtenus dans `/var/log/kern.log`. Vous pouvez observer que, là encore, un démon `portmapper` est contacté mais, cette fois-ci, c'est sur msLDAP qu'il se situe. Rajoutez donc les règles iptables sur msLDAP et ssLDAP afin que ssLDAP soit client et msLDAP soit serveur `portmapper`. Puis réesayez la commande `quota`.

Encore une fois, cela ne fonctionne pas. Les logs de `/var/log/kern.log` de ssLDAP sont similaires à :

```
Aug 12 11:26:59 ssLDAP kernel: [ 3785.448854]
IN= OUT=enp0s3 SRC=192.168.X.2 DST=192.168.X.1 LEN=120 TOS=0x00 PREC=0x00 TTL=64
ID=51925 DF PROTO=UDP SPT=769 DPT=755 LEN=64
Aug 12 11:26:59 ssLDAP kernel: [ 3785.449208]
IN= OUT=enp0s3 SRC=192.168.X.2 DST=192.168.X.1 LEN=116 TOS=0x00 PREC=0x00 TTL=64
ID=51927 DF PROTO=UDP SPT=769 DPT=755 LEN=64
```

Notez que, sur cet exemple, le port de destination est 755 sur la machine `192.168.X.1`, autrement dit, msLDAP. Si l'on procède à un `rpcinfo -p` sur msLDAP, on peut observer que cela correspond au démon `rquotad`, qui est géré par le `portmapper`, comme le souligne `rpcinfo -p`.

Donc, comme dans l'étape 10, on doit faire en sorte que le port affecté par le `portmapper` à `rquotad` soit toujours le même. Pour cela, comme dans l'étape 10, il faut éditer le fichier de configuration du serveur `rquotad` qui se trouve dans le répertoire `/etc/default`. Ici, il s'agit du fichier `/etc/default/quota`. Indiquez dans celui-ci que :

```
RPCRQUOTADOPTS="-p 800"
```

Cela passera l'option `-p` à `rquotad` qui lui indiquera d'écouter le numéro de port 800 (cf. `man rquotad`). Redémarrez via `systemctl` le service `quotarpc.service`. Si vous effectuez un `rpcinfo -p`, vous pourrez observer que `rquotad` utilise maintenant le port 800. Faites ces opérations à la fois sur msLDAP et ssLDAP. Maintenant que vous avez la certitude que c'est le port 800 qui est utilisé par `rquotad`, rajoutez les règles iptables qui vont permettre à la commande `quota` de fonctionner sur vos trois machines virtuelles. Lorsque ce sera la cas, cette commande vous rendra la main après vous avoir affiché un message similaire à :

```
Disk quotas for user root (uid 0): aucun
```

2. Sécurisations optionnelles

Les étapes ci-dessous ne seront pas comptabilisées dans la deuxième note de TME Linux. Elles ont pour but de faire en sorte que vous puissiez à nouveau déployer des machines virtuelles avec FAI, même lorsque votre parc informatique est protégé par des iptables. Réalisez ces étapes s'il vous reste du temps et que vous êtes intéressés par un approfondissement de la sécurisation par iptables.

Étape 13 – Autorisation d'accès à internet sur ssLDAP

Rajoutez les règles sur les chaînes `INPUT` et `OUTPUT` permettant à ssLDAP de se connecter à internet (quel que soit le protocole et le serveur que l'on souhaite joindre). N'oubliez pas que cette communication se fera sur votre 2ème carte réseau. Seules les communications initiées par ssLDAP doivent être autorisées, les autres doivent être refusées par votre firewall.

Étape 14 – Forward des accès à internet sur ssLDAP

Sur ssLDAP, rajouter les règles sur la chaîne `FORWARD` autorisant ssLDAP à forwarder les communications des autres machines virtuelles vers internet.

Sur msLDAP et c1LDAP, rajoutez les règles permettant de communiquer avec des serveurs web sur internet (on rappelle que les serveurs `http` communiquent via le protocole `tcp` sur le port 80).

Testez bien que msLDAP et c1LDAP peuvent lire les pages web de `http://ftp.lip6.fr`

Étape 15 – DHCP accessible sur ssLDAP

Lorsque vos nouvelles machines virtuelles se déploient, elles récupèrent leur adresse IP via le serveur DHCP installé sur ssLDAP. Vous devez donc autoriser les connexions sur ce serveur. Les échanges de paquets entre le serveur et ses clients se déroulent des deux côtés sur les ports 67 et 68 en UDP. Vous devez donc permettre ces échanges. Notez que, dans les règles concernant le DHCP, on ne peut pas spécifier l'adresse des machines distantes puisque, quand celles-ci contactent le serveur, elles n'ont pas encore d'adresse IP. Écrivez donc les règles iptables correspondantes sur ssLDAP.

Le plus simple pour tester si celles-ci fonctionnent est de démarrer votre machine `c2LDAP` sur laquelle vous n'avez pas encore installé d'iptables. Si cette machine parvient à obtenir son adresse IP, c'est que vos règles sont bonnes.

Étape 16 – Commandes TFTP accessibles sur ssLDAP

La deuxième opération effectuée lorsque vos machines virtuelles se déploient consiste à télécharger le noyau linux via tftp. Ce serveur attend les connexions en UDP sur son port 69. Écrivez donc sur ssLDAP les règles iptables qui permettent au serveur de converser avec ses clients.

Afin de tester si vos règles sont correctes, sur c1LDAP, arrêtez les iptables puis exécutez la commande `tftp ssLDAP`. Si vous voyez l'invite de commandes de tftp, vos règles sur ssLDAP sont correctes. Vous pourrez taper la commande `status` dans tftp puis la commande `quit` pour sortir.

Étape 17 – Transferts de fichiers TFTP accessibles sur ssLDAP

En fait, comme dans le protocole ftp, le port 69 de tftp ne sert qu'à émettre des commandes. Lors des téléchargements, les fichiers sont transmis via un autre port créé dynamiquement. Afin de tenir compte de ces ports, il ne suffit pas d'utiliser les options `ESTABLISHED,RELATED` que nous avons vu en cours car le protocole UDP est *stateless* (sans état). Ici, sur ssLDAP, il faut utiliser un module du noyau linux appelé `nf_conntrack_tftp`, qui analysera les paquets transmis afin de déterminer les nouveaux ports créés dynamiquement utilisés pour les transferts. Pour charger ce module, il faut taper la commande :

```
modprobe nf_conntrack_tftp
```

et, pour pérenniser ce chargement, rajouter une ligne :

```
nf_conntrack_tftp
```

dans le fichier `/etc/modules`
Enfin, il faut indiquer que les connexions « traquées » par `nf_conntrack_tftp` sont acceptées :

```
-A INPUT  -i enp0s3 -p udp -s 192.168.X/24 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -o enp0s3 -p udp -d 192.168.X/24 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

Voilà, après avoir restauré vos iptables, votre serveur tftp devrait pouvoir vous transmettre des fichiers. Pour tester cela, sur c1LDAP, réexécutez un `tftp ssLDAP` et tapez la commande `get fai/pxelinux.0`. Le transfert devrait alors se faire correctement.

Étape 18 – Nouveau déploiement

Vous pouvez maintenant tester le déploiement d'une nouvelle machine virtuelle. Celle-ci a toutes les autorisations du firewall de ssLDAP pour s'installer.

Étape 19 – Pérennisation

Pour pérenniser l'exploitation du fichier `/etc/iptables`, il suffit de créez un fichier :
`/etc/network/if-pre-up.d/iptables` et d'indiquer dedans :

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables
```

puis de rajouter pour tout le monde les droits en exécution sur ce fichier. Les exécutables du répertoire `/etc/network/if-pre-up.d` sont exécutés lorsque l'on active (up) les interfaces réseau. Au prochain démarrage du service `networking`, cela obligera votre noyau Linux à prendre en compte vos iptables. Ici, ce fichier existe déjà, il a été créé par l'équipe enseignante pour que ssLDAP puisse faire de l'IP `masquerading`.

Étape 20 – Quelques lectures à faire chez vous

Vous avez finalisé la configuration de vos iptables. Félicitations ! Les étapes précédentes vous ont montré plusieurs techniques assez générales afin de mettre en place des iptables.

Si vous souhaitez compléter vos connaissances sur les RPC et les différents services que vous utilisez, voici quelques lectures intéressantes :

- Description de ce que sont les RPC :
  `http://fr.wikipedia.org/wiki/Remote_procedure_call`
- Une autre description plus approfondie des RPC :
  `http://okki666.free.fr/docmaster/articles/linux116.htm`
- Encore une autre description des RPC :
  `http://www.cs.cf.ac.uk/Dave/C/node33.html`
- Éléments d'information sur la sécurisation des services RPC par iptables :
  `http://www.cyberciti.biz/faq/linux-secure-portmap-with-iptables-tcp-wrappers/`
- Sécurisation de NFS par iptables :
  `http://www.cyberciti.biz/faq/centos-fedora-rhel-iptables-open-nfs-server-ports/`
- IP masquerading :
  `http://fr.tldp.org/HOWTO/lecture/IP-Masquerade-HOWTO.html`