Sécurité - TP
Enoncé du projet `iptables`

Virginie Van den Schrieck

10 novembre 2015

Le but de ce projet est d'implémenter une politique de sécurité au travers de la configuration d'un firewall, et de valider cette politique de deux manières : D'une part en garantissant le bon fonctionnement des services, et d'autre part en garantissant l'absence de possibilité d'intrusions.

Le projet s'effectuera par groupe de quatre étudiants, et se déroulera en deux étapes. La première, qui s'étendra sur quatre semaines, visera à développer en parallèle la configuration firewall d'un réseau, ainsi qu'un ensemble de scripts de validation. La configuration iptables devra être validée sur base de ces scripts.

La seconde étape consiste à analyser la configuration d'un autre groupe, et d'identifier d'éventuelles vulnérabilités. Si chaque groupe a correctement concu ses scripts de validation, il suffira de les appliquer sur ce réseau et d'analyser le résultat.

Deux rapports sont demandés : Un premier rapport justifiant la configuration choisie et les scripts proposés, et un second analysant le travail d'un autre groupe.

A l'exception des modalités pratiques, ce projet est intégralement repris du cours INFO0045 du Pr. Donnet de l'ULg (`http://www.montefiore.ulg.ac.be/~bdonnet/info0045/index.html`).

1 Description du réseau

Vous êtes en charge de la sécurité réseau de Paranoyak ^{TM}, une nouvelle compagnie spécialisée dans la photographie développant des technologies innovantes. Cette compagnie remporte de plus en plus de succès, et ils ont décidé de construire leur premier quartier général à Louvain-la-Neuve. Un réseau interne a donc été déployé dans le nouveau batiment. Ils souhaitent que vous protégiez ce réseau afin de garantir le secret de leurs nouvelles techniques en implémentant différentes règles dans leurs firewalls.

Le labo virtuel implémentant ce réseau est disponible sur le Campus Virtuel (fichier FW.zip). Si vous rencontrez des problèmes pour l'exécution de ce

labo Netkit, il vous faudra modifier le fichier `/home/user/netkit/netkit.conf`, et indiquer la valeur `no` pour la variable `USE_SUDO`.

Dans ce réseau, deux machines R1 et R2 sont utilisées à des fins de recherche. Elles sont localisées sur le même LAN, qui est protégé par le firewall stateful `FW1`. Ce firewall est connecté à l'Internet. Combiné avec un second firewall `FW2`, il délimite une DMZ en sandwich contenant un serveur DNS local (`LDNS`) utilisé par les appareils à l'intérieur du réseau de la compagnie, deux proxies `HTTP` et `HTTPS` (non utilisés par les machines de recherche) et un relai `SMTP` implémentant également IMAP. Les utilisateurs peuvent soliciter le serveur mail pour recevoir leurs emails.

Les machines disponibles pour l'administration, `U1` et `U2` sont sur le même LAN et sont isolées de l'Internet.

Différents serveurs sont déployés dans le réseau. Le serveur `Web` héberge le site web de la compagnie, et un serveur DNS public (`PDNS`) peut recevoir les requêtes de l'Internet. L'administrateur réseau a configuré deux serveurs pour sauver et partager des documents importants. Un serveur `RSYNC` est utilisé pour le backup des données. Les employés peuvent télétravailler, et donc, ce serveur doit également être accessible de l'extérieur. Un autre serveur `NFS` est utilisé pour le partage de documents entre les machines du département recherche.

La compagnie étant spécialisée en photographie, le département recherche doit être capable de réaliser des opérations couteuses en CPU/mémoire sur des images. Un ordinateur de calcul dédié, appelé `processor`, a donc été mis à disponibilité des ingénieurs. Les algorithmes de traitement d'images utilisés font partie du business de la compagnie, et doivent donc être tenus secrets. `processor` doit donc être soigneusement sécurisé. Tout d'abord, cet appareil est situé derrière un troisième firewall stateful, `FW3`. Ensuite, seule la machine `R1` doit pouvoir accéder directement à cette machine en SSH. Les autres machines doivent utiliser un relai SSH (machine `SSH`) pour se connecter à `processor`. Les images qui doivent être traitées par cette machine ne peuvent pas être transférées directement, elles doivent d'abord être chargées sur le serveur `FTP`, d'où elles seront téléchargées par `processor`.

Le protocole FTP n'étant pas sécurisé, ce serveur ne peut être accédé que depuis le réseau de la compagnie. Si quelqu'un a besoin de traiter des données, il doit procéder comme suit :

1. Transférer les données sur le serveur `FTP`
2. Créer une connexion SSH depuis le relai `SSH`
3. Etablir une nouvelle connexion SSH du relai jusqu'à `processor`
4. Utiliser `processor` pour charger les données depuis `FTP`
5. Exécuter le traitement des données sur `processor`
6. Envoyer les résultats sur `FTP`, et les récupérer depuis une autre machine.

Seul `R1` peut accéder directement en SSH à `processor`, mais il doit néanmoins toujours passer par le serveur FTP pour le transfert de données.

2 Comptes utilisateurs

La compagnie comprend deux ingénieurs, Bill et Steve, qui ont un compte sur la plupart des machines du réseau. Vous pouvez utiliser ces comptes pour tester le déploiement de votre système de sécurité. Les logins sont `bill` et `steve`, et, par simplicité, les mots de passe sont identiques aux logins. Bill et Steve ont également chacun une adresse email au sein de la compagnie :

`{steve|bill}@paranoyak.omg`

3 Informations supplémentaires sur les services

3.1 Adressage IP

Netkit est exécuté derrière un NAT, ce qui fait que toutes les adresses du réseau émulé sont privées. Dans le cadre de ce projet, nous considérerons que le sous-réseau 172.16.0.0/12 appartient à l'Internet et est donc routable. Pour réaliser vos tests, vous avez accès à l'ordinateur personnel de Steve, `T1`. Cette machine est configurée pour utiliser `SMTP` comme relai pour les emails, et `PDNS` comme serveur DNS.

3.2 Le DNS

Deux serveurs DNS sont disponibles dans le réseau virtuel. Le premier, `LDNS`, est utilisé par les machines du réseau local (`R1`, `R2`, `U1` et `U2`). Le second, `PDNS`, est un serveur DNS public. Il n'accepte de requêtes que depuis l'Internet. Ces deux serveurs sont configurés avec les entrées suivantes :

- `www.paranoyak.omg` : adresse du serveur web
- `mail.paranoyak.omg` : adresse du serveur SMTP/IMAP
- `ssh.paranoyak.omg` : adresse du relai SSH
- `rsync.paranoyak.omg` : adresse du serveur RSYNC

Le serveur DNS local contient également plusieurs entrées mappant des noms de machines à leurs adresses IP dans le réseau local. Seules les machines à interface unique sont considérées.

3.3 Service mail

Bill et Steve peuvent envoyer et recevoir des emails avec leurs adresses `@paranoyiak.omg`. Les machines de l'environnement virtuel (`R1, R2, U1, U2 et T1`) disposent d'un client mail, appelé `mutt`. Ce client est configuré pour utiliser la machine `SMTP` de la compagnie comme relai SMTP et serveur

IMAP. Pour utiliser `mutt` depuis une de ces machines, tapez simplement `mutt` dans le terminal de la machine virtuelle. Le client est configuré pour utiliser l'adresse email correspondant au compte utilisé. Dans le cadre de ce projet, la configuration du labo ne vous permet pas d'envoyer des mails vers d'autres domaines, mais vous devez néanmoins prévoir cette possibilité.

3.4 Navigateur web

Le client web `lynx` est disponible sur chaque machine. Pour demander une page web, tapez simplement en ligne de commande :

```
lynx http://website.domain
```

Pour un transfert sécurisé, remplacez http par https. Dans le réseau virtuel, lynx est configuré pour utiliser automatiquement les proxies lorsque c'est nécessaire.

3.5 FTP

Un client FTP est disponible sur les machines et permet de se connecter au serveur FTP comme suit :

```
ftp ip_server
```

avec `ip_server` l'adresse IP du serveur FTP. Les commandes suivantes sont disponibles :

- `?` pour obtenir la liste des commandes disponibles
- `ls` pour afficher le contenu du répertoire courant sur le serveur
- `!ls` pour afficher le contenu du répertoire courant local
- `cd` pour changer le répertoire courant sur le serveur
- `lcd` pour changer le répertoire courant local
- `mkdir` pour créer un répertoire sur le serveur
- `put` pour envoyer un fichier au serveur
- `get` pour récupérer un fichier depuis le serveur

3.6 RSYNC

`rsync` est un logiciel utilisé pour la synchronisation de fichiers. Il est ici utilisé pour implémenter un système de backup distant. Afin de synchroniser un fichier avec le serveur, utilisez la commande suivante :

```
rsync -v file user@server::module
```

avec :

- `file` le nom du fichier à synchroniser
- `user` le nom d'utilisateur (root n'est pas autorisé)
- `server` l'adresse du serveur `RSYNC`

– `module` le nom d'un module, c'est-à-dire un ensemble d'information permettant à `RSYNC` de savoir où et comment stocker les fichiers synchronisés sur le serveur, les autorisations, etc. Deux modules sont disponibles sur le serveur : `backup_bill` et `backup_steve`. Les fichiers sont synchronisés dans le répertoire home de chaque utilisateur sur le serveur.

Pendant le transfert, les données ne sont pas chiffrées. Cela signifie que n'importe qui peut lire les documents transférés. Pour le transfer d'informations sensibles, il vaut donc mieux utiliser `RSYNC` en combinaison avec `SSH`. Pour cela, la commande à taper devient :

```
rsync -v file user@server:destination_directory
```

avec `destination_directory` le répertoire sur le serveur où le fichier doit être synchronisé. Le chemin de ce répertoire peut être absolu ou relatif au répertoire home de l'utilisateur.

3.7 NFS

NFS est un protocole utilisé pour le partage de données à travers un réseau. L'intérêt de ce système est qu'il est possible d'utiliser un répertoire (voire même l'entièreté du système de fichier) d'une machine distante comme s'il s'agissait d'un disque dur directement connecté à la machine locale.

Dans le réseau PARANOYAK, le répertoire `/home/sharing` sur le serveur `NFS` est partagé avec `R1` et `R2`. Chaque fois que ces deux machines sont démarrées, elles montent automatiquement ce répertoire partagé sur leur répertoire local `/home/sharing`. Dès que `R1` ou `R2` modifient des données dans le répertoire, ces modifications sont envoyées au serveur. Ce dernier avertit alors les autres clients de la modification effectuée.

4 Méthodologie

La première étape, effectuée en classe, va consister à représenter le réseau décrit plus haut sur un schéma reprenant le nom des machines, leurs interconnexions et les différents subnets et adresses utilisés. Ce travail vous permettra de vérifier votre compréhension du réseau et les enjeux de la politique de sécurité à implémenter.

En deuxième lieu, vous aurez soin de tester ce réseau via l'implémentation Netkit fournie, pour vérifier que vous maitrisez bien les services réseaux qui sont en jeu.

Ensuite, vous prendrez le temps de rédiger clairement la politique de sécurité à implémenter au niveau des accès réseaux dans les firewalls.

Une fois identifiée la politique de sécurité à implémenter, le travail demandé nécessitent deux tâches en parallèle : La conception et la configuration

des règles de firewall, et la conception de scripts de validation. Pour chaque élément de la politique (ex : pour chaque service), il pourrait être intéressant de diviser le travail en deux sous-groupes : Le premier s'occupe des règles iptables, et le second s'occupe des scripts de validation. Afin d'uniformiser les apprentissages au sein du groupe, il est recommandé d'intervertir les rôles à chaque élément.

Les étapes conseillées pour mener à bien ce projet sont donc :

1. (en classe) Faire le schéma du réseau
2. Lister les services dont il faut tenir compte
3. Tester ces services dans le réseau Netkit sans firewall
4. Décrire les règles de haut niveau nécessaires pour le firewall, en français (ex : seul le trafic http est autorisé entre la machine X1 et la machine X2). Ces règles devront figurer dans le rapport final
5. Décider de la manière dont seront construits les scripts de validation (langage de scripting choisi, liste des étapes à effectuer, documentation des scripts, ...)
6. Créer le squelette du rapport afin qu'il puisse être complété au fur et à mesure
7. Pour chaque service :
   (a) Vérifier à nouveau que le service fonctionne tel que prévu en l'absence de règles
   (b) Désigner deux étudiants pour la partie config et deux autres pour la partie validation.
   (c) Réaliser la config des règles iptables concernant ce service ainsi que sa validation
   (d) Documenter la config et la validation (rédaction du rapport au fur et à mesure)
   (e) Réunir le travail des deux sous-groupes et valider la config du service
   (f) Intégrer la config spécifique à ce service à l'ensemble de la configuration et vérifier que les autres règles fonctionnent toujours (tests de régression)
8. Finaliser le rapport

A titre indicatif, voici une proposition d'agenda pour l'ensemble de ces étapes :

- **Semaine du 9 novembre** : Formation des groupes, lecture de l'énoncé, prise de connaissance du réseau, listing des services et des règles de haut niveau. Planification précise des tâches et préparation des documents de travail.

- **Semaine du 16 novembre** : Implémentation et validation des deux premiers services (par ex : Web et DNS)
- **Semaine du 23 novembre** : Suite de la configuration
- **Semaine du 30 novembre** : Fin de la configuration, finalisation du rapport et dernier ajustement avant remise.
- **Semaine du 7 décembre** : Envoi par email du rapport au groupe correcteur et via le Campus Virtuel au titulaire du cours pour le **8 décembre à 18h**
- **Semaine du 14 décembre** : Application des scripts de validation sur la configuration reçue, et rédaction d'un petit rapport d'analyse à rendre le **18 décembre avant 18h**.

5 Deadlines et déliverables

Vous avez deux échéances pour ce projet :

5.1 Le projet lui-même

Le projet, à remettre pour le **8 décembre**, contient le rapport, la configuration et les scripts de validation. Il doit être envoyé par mail au groupe de l'autre classe ayant le même numéro (ex : 3TL1.1 envoie à 3TL2.1 et inversément). Les numéros de groupe vous seront communiqués lors de la première séance du projet.

Le rapport doit contenir un fichier pdf expliquant, d'une part, les règles de haut niveau implémentées, et d'autre part, la stratégie de validation adoptée et les résultats de cette dernière sur votre réseau.

Il contiendra également la configuration iptables des trois firewalls, sous forme de trois fichiers appelés `config_FWx.sh`, avec `x` le numéro du firewall concerné.

Enfin, il contiendra également les scripts ou la méthodologie de validation suivie.

L'ensemble des fichiers sera remis à l'intérieur d'un répertoire appelé FW_gr3TLY.X, avec Y le numéro de votre classe et X le numéro de groupe qui vous aura été attribué. Ce répertoire sera zippé avant la remise. La structure de ce répertoire est la suivante :

- Fichier `FW_gr3TLY.X_rapport.pdf`
- Répertoire `FW_gr3TLY.X_config` contenant :
  - config_FW1.sh
  - config_FW2.sh
  - config_FW3.sh
- Répertoire `FW_gr3TLY.X_validation` contenant les scripts de validation ou tout autre document utile à cette fin.

5.2 Rapport d'analyse

Le rapport d'analyse, à remettre sur le Campus Virtuel pour le **18 décembre à 18h**, consiste en un document pdf de maximum 3 pages expliquant les résultats de l'analyse effectué sur le réseau du groupe « adversaire ». Ce document sera muni d'une conclusion résumant les observations effectuées et statuant sur la sécurité de la configuration firewall proposée par l'autre groupe.