TP 4 - Services SSH / Apache / MySQL / NFS / Tomcat

Pour ce TP, on considère la mise en place d'un serveur hébergeant les services utiles à notre entreprise *istycorp.fr*. Pour cela nous allons considérer la mise en place de :

- ssh pour faciliter la gestion distante du serveur.
- Les services habituels Apache / PHP / MySql pour fournir l'hébergement d'un Wordpress.
- Un serveur NFS pour partager les dossiers home en réseaux.
- Un serveur tomcat pour héberger Jenkins.

Intégration dans l'infrastructure

Nous repartirons du réseau mis en place lors du TP 3 à savoir un routeur et un client. Nous ajouterons donc une machine *server.istycorp.fr*. De même, cette dernière sera connectée à internet au travers du routeur.

Dans la suite nous allons mettre en place divers services qui devront être exposés à l'extérieur par le routeur. Si rien n'est fait, les connexions entrantes arriveront sur le routeur et ne trouveront aucune réponse à leur requête. Nous devons donc faire une redirection de port sur ce dernier pour renvoyer les paquets entrant vers le serveur approprié en fonction du port cible.

Pour les services à mettre en place dans la suite vous devrez ajouter des règles au fichier */etc/iptables/rules.v4* du routeur dans la partie NAT :

```
-I PREROUTING -p tcp --dport {DEST_PORT} -i ${WAN} -j DNAT \
--to {REDIRECT_TO_IP}:{REDIRECT_TO_PORT}
```

Nous sommes dans des machines virtuelles, les ports doivent donc également être exposés à votre machine locale en reconfigurant les cartes réseau de VirtualBox sur le routeur.

Service SSH

Exercice 4.1

Installez le serveur SSH (*openssh-server* sous Debian) sur le serveur et vérifiez son fonctionnement. Pensez à activer le routage du port 22 (iptables du routeur et VirtualBox).

Exercice 4.2

Mettez en place vos clés SSH pour vous connecter avec ces dernières plutôt qu'avec un mot de passe pour le compte 'isty'. Pour ce faire, il faut copier votre clé publique dans le fichier *.ssh/authorized_keys*. Sous Linux cette tâche peut être automatisée avec la commande :

```
ssh-copy-id {user}@{hostname}
```

Exercice 4.3

Afin de sécuriser l'accès à notre serveur nous n'autoriserons que l'accès par clé SSH et interdirons les connexions directes au compte root. Modifiez la configuration du serveur en conséquence (*/etc/ssh/sshd_config*).

Exercice 4.4

Mettez en place l'outil *fail2ban* afin de bloquer les tentatives d'intrusions par forte brute sur le ssh. Vérifiez que vous vous faites bien bannir en échouant plusieurs fois à vous connecter.

Exercice 4.5

Après vous êtes fait bannir, regardez la sortie de la commande *iptables-save* et expliquez.

Exercice 4.6

Observez vos traces dans les fichiers */var/log/auth.log* et */var/log/fail2ban.log*.

Exercice 4.7

Retrouvez votre accès en relançant le service après avoir supprimé les lignes dans *auth.log*.

Apache / PHP / Mysql / Wordpress

Nous allons considérer ici que notre entreprise a besoin du service WordPress pour diffuser des informations sur l'entreprise. Pour cela nous aurons besoin d'un serveur supportant PHP et d'une base de données MySQL.

Exercice 4.8

Installez Apache et PHP, vérifiez le fonctionnement. Comme pour SSH il faudra activer le transfert de port sur le routeur et dans VirtualBox.

Exercice 4.9

Installez votre serveur MySQL. Choisissez le mot de passe de son compte root comme demandé par le gestionnaire de paquet.

Exercice 4.10

Sur les distributions de type debian on aime en générale utiliser le script *mysql_secure_installation* pour désactiver le compte anonyme et limiter l'accès root à un usage local.

Exercice 4.11

Afin de nous faciliter la gestion de notre base de données, nous installerons l'application *phpmyadmin*.

Exercice 4.12

Installez WordPress et vérifiez son fonctionnement.

Exercice 4.13

Validez le fonctionnement en activant SSL avec un certificat auto signé. Debian fournit déjà la configuration, vous n'avez plus qu'à l'activer avec :

```
a2ensite default-ssl
a2enmod ssl
```

Exercice 4.14

Quelle remarque pouvez-vous faire à propos de la sécurité des certificats auto signés ?

Serveur NFS

Nous décidons de partager les dossiers des utilisateurs sur le réseau interne à l'entreprise afin qu'ils puissent y accéder depuis n'importe quelle station de travail.

Exercice 4.15

Mettez en place et testez le sur une machine cliente.

Exercice 4.16

Discutez la sécurité des droits utilisateurs dans cette configuration, les restrictions d'accès aux fichiers seront-elles nécessairement toujours vérifiées ?

Jenkins

Les développeurs de notre entreprise ont besoin d'une plateforme d'intégration continue pour leur travail, nous leur proposons Jenkins. Ce service est implémenté en J2EE et nécessite donc la présence d'un serveur applicatif java tel que *tomcat*.

Exercice 4.17

Installez tomcat et déployez y *Jenkins*. Pour ce faire, installez l'archive *war* sans l'extraire dans le dossier */var/lib/tomcat/webapps*.

Exercice 4.18

Avant de lancer tomcat, créez le dossier /var/jenkins, changez ses droits pour qu'ils appartiennent à l'utilisateur et au groupe tomcat7. Ajoutez la ligne suivante au fichier

Exercice 4.19

On préfère habituellement ne pas exposer directement les serveurs applicatifs, on les cache donc derrière des serveurs frontaux de type apache, réputés plus robustes. Apache joue alors le rôle de *reverse proxy*. Pour activer ce fonctionnement :

1. Activer le connecteur AJP sur le port 8009 dans */var/lib/tomcat7/conf/server.xml*.
2. Activer le module *proxy* et *proxy_ajp* d'apache avec la commande *a2enmod*.
3. Créez un fichier de configuration *proxy-jenkins.conf* dans */etc/apache/mods-available/* avec le contenu suivant :
   ```
   <Proxy /jenkins>
   ProxyPass ajp://localhost:8009/jenkins
   ProxyPassReverse ajp://localhost:8009/jenkins
   </Proxy>
   ```
4. Créez le fichier *proxy-jenkins.load* :
   ```
   # Depends: proxy_ajp
   ```
5. Activez ce module avec a2enmod

Relancez le serveur et vérifiez l'accès au travers d'apache.

Bonus : webmail roundcube

Installez le webmail Roundcube pour accéder à votre mail étudiant.

Bonus : redmine

Installez et configurez le service redmine.