Université des Sciences et de la Technologie
Houari Boumediène

FACULTÉ D'ÉLECTRONIQUE ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

DEUXIÈME RAPPORT DU PROJET : SÉCURISER UN SERVEUR WEB

MODULE : SÉCURITÉ DES SYSTÈMES D'EXPLOITATION

PRÉSENTÉ PAR :

— Bennabi Abdelhakim.
— Bensaid Nabil.
— Bentorcha Camelia.
— Medouni Lotfi.
— Namane Madjid.
— Toumi Nassima.

Master I SSI - 2015/2016

Table des matières

Chapitre 1

Introduction

Ce rapport a pour but de détailler toutes les mesures de sécurité appliquées afin de sécuriser le serveur web, certaines mesures n'ont pas été citées dans le rapport précédent.

Remarques importantes :

- L'adresse du serveur doit rester statique : **192.168.1.111** en raison du certificat SSL.
- Malgré la police SELinux chargée, l'utilisateur *webdev1* n'est pas autorisé à modifier le contenu du dossier /protect/www, il faut donc employer le **sudo** avec l'utilisateur *sysadmin* pour ajouter des fichiers au serveur.
- A chaque démarrage du serveur, le serveur Apache, NTP et IPTables doivent également être démarrés, la phrase de passe pour la clé SSL est : **test**.

```
1 sudo service iptables start
2 sudo service ntpd start
3 sudo service httpd start
```

Chapitre 2

Comptes et différents mots de passe

2.1 Comptes utilisateurs :

- **Root :** L'ouverture de session a été désactivée sur ce compte.
- **sysadmin :**
  - **Mot de passe :** Masterssi2016
  - **Groupe :** wheel
  - **Utilisateur SELinux :**unconfined_u
- **webdev1 :**
  - **Mot de passe :** Apache2016
  - **Groupe :** webdevs
  - **Utilisateur SELinux :**user_u
- **netadmin1 :**
  - **Mot de passe :** Reseau2016
  - **Groupe :** netadmins
  - **Utilisateur SELinux :**staff_u
- **Autres comptes pouvant être créés :**
  - **Utilisateur SELinux :**guest_u

2.2 Mot de passe de chiffrement des partitions :

Les partitions /tmp, /protect/log, /protect/www et /home ont été chiffrées lors de l'installation avec le mots de passe : **masterssi2016**

2.3 Compte d'administration GRUB :

Login : grub **Mot de passe :** grub

A été ajouté en modifiant le fichier : */etc/grub.d/10_linux*

```
1 cat << EOF
2 set superusers="grub" password grub grub
3 EOF
```

Puis en exécutant la commande :

```
1 grub2-mkconfig --output=/boot/grub2/grub.cfg
```

2.4 Mot de passe du BIOS :

Le BIOS étant indisponible pour les machines virtuelles Virtual Box, il nous a été impossible de le sécuriser par mot de passe, toutefois nous avons désactivé le démarrage à partir de périphériques amovibles dans les options de la VM.
Il est possible de configurer le mot de passe du BIOS pour les machines virtuelles VMWare en appuyant sur la touche F2 au démarrage.

Chapitre 3

Système d'exploitation du serveur

3.1 Table des partitions :

Table	Taille	Remarque
/	5 Go	
swap	2Go	
/boot	200Mo	
/tmp	2Go	nosuid, noexec +t et g+s
/home	6Go	
/protect/www	2Go	
/protect/log	2Go	+t

3.2 Patch vulnérabilités :

Après avoir effectué une installation du système offline, nous avons téléchargé les différents patchs sur une machine différente sans les installer avec la commande **yum update –downloadonly –downloaddir /tmp/rpm_updates**, nous les avons ensuite copié sur un support amovible et installé sur notre machine avec la commande **rpm -Uvh /tmp/rpm_updates/***.

3.3 Politique de gestion des utilisateurs/mots de passe :

Mots de passe :

Nous avons défini la politique des mots de passe suivante à l'aide de PAM en modifiant le fichier : **/etc/security/pwquality.conf**

```
1 difok=3 (on change au moins 3 caracteres par rapport a l'ancien mot de passe)
2 minlen=8 (longueur minimale de 8 caracteres)
3 ucredit=-1 (au moins un caractere uppercase)
4 lcredit=-1 (au moins un caractere lowercase)
5 dcredit=-1 (au moins un chiffre)
```

Commande sudo :

— Les membres du groupe *wheel* pourront effectuer n'importe quelle commande.
— Les membres du groupe *webdevs* pourront effectuer la lecture, écriture, modification sur le dossier /protect/www.
— Les membres du groupe *netadmins* pourront exécuter les commandes de l'alias NETWORKING ainsi que la commande **nmap** afin de lister les ports ouverts.

3.4 Services/protocoles :

Serveur NTP :

Nous avons synchronisé l'horloge du système avec des serveurs NTP en modifiant le fichier **/etc/ntp.conf** :

```
1 server ntp1.jst.mfeed.ad.jp iburst
2 server ntp2.jst.mfeed.ad.jp iburst
3 server ntp3.jst.mfeed.ad.jp iburst
```

L'horloge est mise à jour lorsque le service est redémarré : **sudo service ntpd restart**.

SSH :

L'accès à distance se fait à l'aide du protocole SSH, nous l'avons configuré en modifiant le fichier :**/etc/ssh/sshd_config**

```
1 PermitRootLogin no
2 AllowGroups wheel netadmins webdevs
```

L'utilisateur *sysadmin* par exemple pourra se connecter à distance avec la commande : ssh 192.168.1.1111 -l sysadmin

Upload de fichiers :

Le upload de fichiers se fait encore une fois à l'aide de SSH, avec la commande **scp**, par exemple pour copier le fichier *filetest* sur le serveur avec le compte *sysadmin* : **scp filetest sysadmin@192.168.1.111 :/home/sysadmin** Pour le téléchargement les deux arguments sont inversés.
Dans le cas où l'utilisateur est connecté sur une machine d'OS différent tel que Windows, il pourra passer par le site **transfer.sh** qui gère le transfert de fichiers via **https**, l'upload sur leur serveur se fait avec la commande : **curl –upload-file ./hello.txt https ://transfer.sh** Qui nous donnera en sortie un lien de téléchargement, nous pourront récupérer le fichier sur notre serveur en passant ce lien en argument de la commande **wget**.

Nous n'aurons donc pas besoin de FTP pour le transfert de fichiers.

3.4.1 Contrôle d'accès :

Comme expliqué dans le précédent rapport.

3.5 Maintenance :

3.5.1 Scan vulnérabilités et pen-testing :

Nous avons installé le scanner de vulnérabilités et anti-malware **Nessus**, son interface de configuration est accessible sur l'adresse : **https ://192.168.1.111 :8834** Le compte d'administration est le suivant :
Login : sysadmin **Mot de passe :** Masterssi2016

3.5.2 Logging :

Nous avons choisi de changer l'emplacement des fichiers log au répertoire /protect/log, nous avons donc dû changer son contexte SELinux afin qu'il soit accessible au démon rsyslog :

```
1 semanage fcontext -a -t var_log_t /protect/log
2
3 restorecon -v /protect/log
```

Ensuite nous avons configuré le logging de sorte que chaque processus ait son propre fichier en modifiant **/etc/rsyslog.conf** :

```
1 $template CUSTOM_LOGS,"/var/log/%programname%.log"
2 $template sample,"Message:%msg%, Priorite : %pritext%, %timegenerated%, %
3 HOSTNAME%, %syslogtag%\n"
4 *.* ?CUSTOM_LOGS;sample
```

La rotation s'effectue chaque semaine, les fichiers sont compressés et la limite d'archives est de 24 fichiers, c'est à dire que les logs sont sauvegardés pendant 24 semaines.

3.5.3 Audit :

Pour l'audit également nous avons dû changer le contexte du fichier /protect/log/audit/audit.log.

Chapitre 4

Réseau :

4.1 Configuration réseau :

Pour les besoins du serveur web, notamment pour installer un certificat SSL nous avons dû assigner une adresse IP statique au serveur : **192.168.1.111/24** avec comme passerelle : **192.168.1.1**
Les serveurs DNS sont ceux de Google 8.8.8.8 et 8.8.4.4

4.2 xinetd et TCPWrappers :

Parmi les services proposés, seul le SSH passe par TCPWrappers.
Le fichier **/etc/hosts.deny** contient la ligne : ALL :ALL Le fichier **/etc/hosts.allow** contient la ligne : sshd : ALL

4.3 Pare-feu :

Afin de garantir les différents accès aux services cités précédemment (SSH, HTPP, HTTPS...) nous avons configuré le firewall IPTables avec les commandes suivantes :

```
1  iptabes -F
2
3  iptables -P INPUT DROP
4  iptables -P FORWARD DROP
5
6  iptables -A INPUT -p udp --sport 53 -j ACCEPT
7
8  iptables -A INTPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j
       ACCEPT
9
10 iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j
       ACCEPT
11
12 iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

```
13
14 iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst
       100 -j ACCEPT (pour limiter DDos)
15
16 iptables -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j
       ACCEPT
17
18 iptables -A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j
       ACCEPT
19
20 iptables -A INPUT -p udp --sport 123 -j ACCEPT (pour synchronisation avec NTP
       )
21
22 iptables -A INPUT -p tcp --dport 8834 -j ACCEPT
23
24 iptables -A INPUT -i lo -j ACCEPT
25
26 service iptables save
```

Les connexions entrantes sur les ports de NTP, HTTP,HTTPS, DNS et SSH sont donc autorisées, les autres sont refusées. Il n'y a pas de restriction sur les connexions sortantes.

4.4 IDS et IPS :

Nous avons installé le logiciel SNORT et mis à jour les différentes règles permettant de protéger la machine contre la plupart des attaques. SNORT peut être lancé avec la commande : **sudo snort -i <interface> -u snort -g snort -c /etc/snort/snort.conf**

Chapitre 5

Serveur web :

5.1 Configuration :

Nous avons désactivé les différents modules tels que mod_userdir, mod_status, mod_info pouvant révéler des informations sur la version d'Apache ou de l'OS. Nous avons également défini le champ Options à None et supprimé le fichier de documentation et le fichier welcome affiché lorsque index.html n'existe pas car révèle la distribution de l'OS.
Pour une sécurité accrue, la racine des fichiers du serveur est **/protect/www/html** et non /var/www/html.

5.2 Chroot Jail :

Plusieurs articles sur le web affirment que SELinux apporte un meilleur niveau de sécurité que le chroot jail, néamoins, nous avons essayé mais sans succès de démarrer le serveur dans un chroot jail. Après avoir suivi toutes les étapes du tutoriel ci-dessous, la racine des processus de httpd reste celle du système et non du jail.De plus, le module **mod_chroot** d'Apache n'est pas disponible pour CentOS.
Lien : http ://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap29sec254.html

5.3 Utilisation de SSL/TLS :

Nous avons généré une clé RSA 2048 bits et un certificat X509 pour notre adresse 192.168.1.111 et l'avons intégré à notre serveur Apache en modifiant le fichier **/etc/httpd/conf/httpd.conf** :

```
1 <VirtualHost 192.168.1.111:80>
2     Redirect / https://192.168.1.111
3 </VirtualHost>
4
5 <VirtualHost 192.168.1.111:443>
6     SSLEngine on
7     SSLCertificateFile /etc/httpd/server.crt
8     SSLCertificateKeyFile /etc/httpd/server.key
9 </VirtualHost>
```

De ce fait toute requête HTTP sera directement redirigée vers HTTPS.

Remarque : Lorsque le serveur est redémarré, la phrase de passe de la clé est : **test**.

5.4 Protection contre les attaques :

Par exemple pour contrer les attaques DDos et BufferOverflow nous avons ajouté les lignes suivantes au fichier **/etc/httpd/conf/httpd.conf** :

```
Timeout 10
KeepAlive On
MaxClients 256
MaxKeepAliveRequests 100
KeepAliveTimeout 15
LimitRequestLine 512
LimitRequestFields 100
LimitRequestFieldSize 6000
LimitRequestBody 512000
```

5.5 Apache et SELinux :

Nous avons changé le contexte du dossier /protect/www et de son contenu à **httpd_sys_content_t** . Nous avons également ajouté une police personnalisée pour autoriser les utilisateurs du groupe webdevs à effectuer des opérations sur les fichiers de ce répertoire :

```
policy_module(webdevs, 1.0)
gen_require(‘
type httpd_sys_content_t;
type user_t;
’)
allow user_t httpd_sys_content_t :file{read};
allow user_t httpd_sys_content_t :file{create};
allow user_t httpd_sys_content_t :file{write};
allow user_t httpd_sys_content_t :file{rename};
allow user_t httpd_sys_content_t :dir{write};
allow user_t httpd_sys_content_t :dir{read};
```