ATELIER Bonnes pratiques– Sécurisation du web Apache

Préambule et préparation.

Nous allons dans cet exercice travailler avec docker. Dans un terminal, lancez toujours la commande docker avec les privilèges sudo.

Téléchargez le fichier securel3.tar : il s'agit d'une image docker préalablement configurée. La distribution est Fedora 24. Une fois téléchargée, vous pouvez, pour extraire l'image, lancer la commande :

docker load < securitel3.tar

Puis pour créer un conteneur :
docker run -it --name secu -p8080:80 -p443:443 -v $PWD:/tmp michel17:oko /bin/bash

Vous être à présent dans un terminal du conteneur : [root@XXXXXXXXXX /]
Dans un autre terminal, effectuez un sudo *docker images* et un sudo *docker ps --all* pour lister les images et les conteneurs lancés.

Dans le terminal du conteneur, lancez httpd et /usr/sbin/sshd pour lancer le serveur apache et le serveur sshd. Des warnings apparaissent. Vérifiez cependant leur lancement en utilisant :

- **netstat -an | grep ':80' (httpd lancé)**
- **netstat –an | grep ':22' (sshd lancé)**

Modifiez le fichier /etc/resolv.conf : il doit contenir sur la première ligne nameserver 10.2.40.230

Pour la suite de l'exercice, vous travaillerez soit à partir du terminal du conteneur, soit en vous connectant en ssh sur ce conteneur à l'adresse ip de celui-ci (en tant que root, mot de passe: jetson).

REMARQUE IMPORTANTE

Docker sous ubuntu ne peut actuellement lancer le service httpd, ni le service ssh. Pour la suite, le plus simple est de remplacer les commandes de lancement ou d'arrêt des services (systemctl start httpd.service, systemctl stop httpd.service, …) par httpd (lancement) et pkill httpd (arrêt) ou encore /usr/sbin/sshd (lancement) et pkill sshd (arrêt).

1. **Lancement de httpd.**
   httpd
   Puis : netstat -an | grep ':80'
   Et lancez ensuite un navigateur sur la machine hôte : http://localhost:8080: vous devrez avoir la page de Fedora.

2. **Tester l'accès au serveur via lynx ou wget (sur le conteneur)**

Via lynx (s'il est présent) : lynx -dump 'http://localhost' | head -2
ou wget --no-proxy http://localhost/testfile (avec testfile créé sous l'utilisateur root par touch /var/www/html/testfile).
Via un navigateur (à partir de l'hôte : n'oubliez pas de passer par le port 8080 pour être transféré sur le port 80).

3. **Visualiser les journaux (l'emplacement du fichier peut dépendre de votre configuration)**
   tail /var/log/httpd/error_log
   tail /var/log/httpd/access_log

4. **Modifier la configuration. Relancer le serveur**
   Mettez en place une authentification pour les pages web présentes dans le répertoire club_prive. Les accès sont interdits sauf à partir du poste local.
   vi /etc/httpd/conf.modules.d/clubprive.conf

```
<Directory "/var/www/html/club_prive">

        AuthType Basic
        AuthName "Club prive"
        AuthUserFile /etc/httpd/conf/users
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1 localhost et  l'adresse IP
 de  votre machine liée à docker0
        Require valid-user
 </Directory>
```

   réactivez le service avec la commande pkill httpd, puis httpd

5. **Créer les comptes des utilisateurs autorisés (binôme)**
   Utilisez vos logins et mots de passe :
   htpasswd -c /etc/httpd/conf/users mmenard (l'option -c crée le fichier, pour le second omettez-le.
   visualisez vos autorisations :
   cat /etc/httpd/conf/users

6. **Créer le site Web**
   mkdir /var/www/html/club_prive
   echo ''<h1>Club prive</h1>'' > /var/www/html/club_prive/index.html

7. **Accéder au site**
   Si l'on accède au site à partir du poste du binôme ou du poste local, le navigateur demande à l'utilisateur de s'authentifier. Le couple login, mot de passe permet d'accéder à la page Web. A partir d'un autre poste, l'accès est interdit (si ce n'est déjà fait, ouvrez le port 80 (http) du firewall local) .
   Si lynx est installé, on peut l'utiliser de deux manières :
   a. De manière interactive

lynx "http://localhost/club_prive"
Et authentifiez-vous.

b. En mode scriptable

lynx -dump 'http://localhost/club_prive' | head -2
lynx -dump -auth=login:mot_de_passe 'http://localhost/club_prive' | head -2

Vous pouvez également utiliser (sur le poste du binôme ou en local): wget --http-user=Nom --http-password=mot de passe –no-proxy 'http://localhost/club_prive'

Avec le navigateur de votre choix.
Accédez au site à partir d'un poste non autorisé

Objectif 2

Sécuriser Apache - Utilisation de SSL

1. **Créer un certificat autosigné. Le visualiser**

   openssl genrsa -out server.key 1024
   chmod 440 server.key
   openssl req -new -key server.key -out server.req
   Entrer l'adresse IP du serveur sur lequel vous avez lancé httpd
   openssl x509 -req -days 365 -in server.req -signkey server.key -out server.crt
   openssl x509 -in server.crt -text -noout | more
2. **Vérifier la présence du module SSL (installez le si nécessaire par yum install mod_ssl)**

   rpm -q mod_ssl
3. **Afficher la configuration par défaut du module SSL**

   grep "^[a-zA-Z]" /etc/httpd/conf.d/ssl.conf

   (cherchez localhost.crt et localhost.key ; Il s'agit des données précédentes normalement.
   Sauvegardez les fichiers localhost.crt et localhost.key.
4. **Installer le certificat et la clé privée associée au bon emplacement**

   cp server.crt /etc/pki/tls/certs/localhost.crt
   cp server.key /etc/pki/tls/private/localhost.key
5. **Relancer le serveur et vérifier l'ouverture du port SSL**

   pkill httpd puis httpd
   netstat -an | grep ':443'

6. **A partir du poste local, tester l'accès à la page d'accueil avec lynx et avec un navigateur**

   Testez d'abord en mode normal (http) puis en mode SSL (https)
   lynx -dump -auth=login:password 'http://num_ip_serveur/club_prive' | head -2
   (ou bien-sûr localhost) : OK
   lynx -dump -auth=login:password 'https://num_ip_serveur/club_prive' | head -2
   (ou bien-sûr localhost) : NO

lynx -auth=login:password 'http://num_ip_serveur/club_prive': message d'avertissement mais accès OK
lynx -auth=login:password 'https://num_ip_serveur/club_prive' : NO

Avec un navigateur : http : OK
Avec un navigateur : https : présentation d'un avertissement, certificat auto-signé. Téléchargez le certificat et consultez le.

7. **Modifier la configuration dans clubprive.conf pour que l'accès puisse se faire également à partir d'un autre PC (celui à coté de vous).**
   Relancez les commandes de la question 6, après avoir relancé le serveur.
   Avant l'installation
   a. (Si lynx installé, sinon utiliser un navigateur) Téléchargez en mode scriptable la page SSL. L'opération doit échouer.
      lynx -dump 'https://nom_serveur/club_prive'
   Installation
   b. Récupérez le certificat du CA depuis le poste distant. Paramétrer lynx pour l'utiliser.
      scp num_ip_serveur:/root/serveur.crt /root
      export SSL_CERT_FILE=/root/server.crt
      (Vérifiez bien les noms des dossiers. Vous pouvez utilisez /tmp).

   Essayez de nouveau d'attendre la page en SSL (avec lynx ou un navigateur)

8. **Modifier la configuration du serveur où a été lancé Apache. Le forcer à l'utilisation de SSL**
   vi /etc/httpd/conf.d/clubprive.conf

```
<Directory "/var/www/html/club_prive">

        AuthType Basic
        AuthName "Club prive"
        AuthUserFile /etc/httpd/conf/users
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1 localhost et  l'adresse IP de  votre machine distante
à partir de laquelle vous  cherchez à vous connecter
        SSLRequireSSL
        Require valid-user
</Directory>
```

pkill httpd puis httpd

Tester à partir du poste distant (avec un navigateur ou wget)
wget --no-check-certificate --http-user=login --http-password=password --no-proxy 'https://num_ip_serveur/club_prive'
L'accès doit réussir

wget --no-check-certificate --http-user=login --http-password=password --no-proxy 'http://num_ip_serveur/club_prive'
L'accès doit échouer

9. **Idem, mais on veut un chiffrement fort.**

vi /etc/httpd/conf.d/clubprive.conf

```
<Directory "/var/www/html/club_prive">

        AuthType Basic
        AuthName ''Club prive''
        AuthUserFile /etc/httpd/conf/users
        Order deny,allow
        Deny from all
        Allow from 127.0.0.1 localhost et  l'adresse IP de  votre machine
distance à partir de laquelle vous   cherchez à vous connecter
        SSLRequireSSL
        SSLCipherSuite ALL :!ADH :!LOW :!EXP :+HIGH :+MEDIUM
        Require valid-user
</Directory>
```

Vérifiez la configuration et relancer le serveur

Les chiffrements LOW (DES simple), ADH (Anonymous Diffie Hellman) ou EXP (40 bits) sont interdits. Sont autorisés tous les chiffrements (ALL), y compris HIGH (triple DES) et MEDIUM (chiffrement 128 bits).

Testez. L'accès lynx doit échouer. Les chiffrements proposés par Lynx ne conviennent pas à Apache. Par contre l'accès par Firefox fonctionne.

10. **Visualiser (sous le poste serveur) les journaux d'Apache concernant SSL**

tail -1 /var/log/httpd/ssl_error_log
tail -1 /var/log/httpd/ssl_request_log
tail -1 /var/log/httpd/ssl_access_log

supprimez dans le fichier de configuration la ligne :
SSLCipherSuite ALL :!ADH :!LOW :!EXP :+HIGH :+MEDIUM

Objectif 3

Sécuriser un serveur : renforcer la sécurisation de l'application

1. Supprimer les bannières
    a. Affichez la bannière d'Apache à partir du poste distant
    echo -e "GET /toto.html http/1.0\r\n\r\n" | nc num_ip_pc2 80
    b. Modifiez la configuration d'Apache

    cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.000

    vi /etc/httpd/conf/httpd.conf

    ```
    #ServerTokens OS
    ServerTokens ProductOnly
    …
    #ServerSignature On
    ```

ServerSignature Off

service httpd restart

c. Affichez de nouveau la bannière à partir du poste distant. Elle n'apparait plus.
echo -e ''GET /toto.html HTTP/1 .0\r\n\r\n'' | nc num_ip_pc2 80

d. Affichez la bannière PHP à partir du poste distant
(remarque : si php n'est pas installé, faire yum install php php-common)
echo -e ''GET /index.php HTTP/1 .0\r\n\r\n'' | nc num_ip_pc2 80

e. Supprimez la bannière PHP
cp /etc/php.ini /etc/php.ini.000
vi /etc/php.ini
…
; expose_php=On
expose_php=Off

service httpd restart

f. Accédez de nouveau à la page PHP, la bannière n'apparaît plus.
echo -e ''GET /index.php HTTP/1 .0\r\n\r\n'' | nc num_ip_pc2 80

Objectif 4

Supprimer les messages d''avertissement dû à un certificat auto-signé.

Lorsque vous vous connectez à partir d'un poste distant sur le serveur web, vous rencontrez un message d'avertissement précisant que le certificat du serveur est auto-signé, et qu'il n'est pas conseillé de vous y connecter. Proposez une solution pour supprimer cet avertissement. Réalisez-là. Dans quelle mesure cette solution ne pose pas de problème de sécurité ?

Objectif 5

Création d'une Autorité de Certification et signature du certificat de l'AC

En vous aidant du TP 3, créez une Autorité de certification. Signez le certificat du serveur avec cette Autorité. Installez le certificat de l'Autorité dans le magasin des certificats de votre navigateur.

Testez. Vous ne devriez plus avoir de message d'avertissement.

ATELIER Bonnes pratiques – Vérifiez la signature d'un logiciel avant installation

Attention : pour wget, il faut configurer la commande pour qu'elle passe le proxy. Vous devez écrire dans le fichier /home/tpuser/.wgetrc
http_proxy=http://nom :password@10.1.30.18 :3128/
https_proxy=http://nom :password@10.1.30.18 :3128/
ftp_proxy=http://nom :password@10.1.30.18 :3128/

1. **Télécharger Apache, son empreinte MD5 et la signature. Vous pouvez également le faire à partir d'un navigateur.**
   wget "http://mirrors.ircam.fr/pub/apache/httpd/httpd-2.4.34.tar.bz2"
   wget "http://www.apache.org/dist/httpd/httpd-2.4.34.tar.bz2.md5" (signature md5)
   wget "http://www.apache.org/dist/httpd/httpd-2.4.34.tar.bz2.asc" (signature)
   Afficher les fichiers téléchargés.

2. **Télécharger les clés GPG des développeurs d'Apache et les incorporer à votre trousseau**
   wget "http://www.apache.org/dist/httpd/KEYS" (clés publiques des développeurs Apache)
   (Vous pouvez également utiliser un navigateur et copier l'intégralité de la page dans un fichier texte ; et ensuite importer les clés).
   Importez les clés : gpg --import KEYS

3. **Vérifier la somme MD5 du fichier httpd-2.4.34.tar.bz2**

4. **La comparer avec la signature dans le fichier htppd-2.4.34.tar.bz2.md5** (commande more) ou à l'aide du navigateur**.**

5. **Vérifier la signature avec l'option --verify (commande gpg) (utilisation des clés publiques importées)** :

   gpg --verify httpd-2.4.34.tar.bz2.asc

   Vous devez vous trouver dans le dossier où se trouvent les fichiers httpd-2.4.34.tar.bz2 et httpd-2.4.34.tar.gz.asc
   Le fichier est signé par Jim Jagielski, son identifiant de clé publique est ID 791485A8 (vous pouvez avoir une signature différente en fonction du numéro de version téléchargée). Que faudrait-il faire pour être sûr de la signature ?