Secure Socket Layer, apache 2, ettercap et ufw

Buts : créer une page d'authentification web, non-securisée et sécurisée, filtrer les accès par firewall. Réaliser des attaques Mitm.

1 Préparation de l'environnement

On installe lubuntu (avec 12Go de disque) et la VM kali (qui fera 40Go de disque à la fin) [1]. La mise en réseau des 2 VM sera en NAT. Installez `open-vm-tools` sur la lubuntu si vous utilisez VMware comme gestionnaire de VM, mettez à jour lubuntu, voire la kali (pour laquelle ça prend du temps !). Notez bien les indications sur PostgreSQL qui vous seront utiles pour le dernier TP. Comptez plus d'1h d'installation.

2 Installation d'`apache 2` sur la lubuntu

On installe le paquet `apache2` qui contient déjà les modules et les fichiers de configuration principaux d'un serveur `http` ou `https`. Une fois le paquet installé, vérifiez que vous avez un serveur actif en vous connectant sur le loopback. Vous devez voir la page d'accueil (en local seulement pour l'instant). Ensuite

- activez les modules nécessaires par `a2enmod` : `ssl, headers, cgid` (un par un) ;
- ultérieurement, à l'ajout des fichiers de configuration spécifiques utilisez l'utilitaire `a2enconf` ;
- et plus tard, à l'ajout d'un hôte virtuel, l'utilitaire `a2ensite`.

Pensez à redémarrer votre serveur après les changements par : `systemctl restart apache2`.

2.1 Génération d'un certificat auto-signé et configuration `https`

Il faut créer la biclé et un certificat auto-délivré avant de modifier les fichiers de configuration. Il faut utiliser successivement (et avec les bons paramètres) les commandes :

```
openssl genrsa # au moins 2048 bits
openssl req
openssl x509
```

Quelques indications sur les pages de `man` ou bien ici pour créer clés et certificat. Il est aussi possible de le faire en une commande `openssl`. Rangez le certificat et la clé dans le répertoire `/etc/ssl/private` et `/etc/ssl/certs` avec les bons droits.
Faire ensuite un `a2ensite default-ssl.conf` qui recopie le fichier de configuration de `ssl` du répertoire `sites-available` dans `sites-enabled` puis modifier le fichier `default-ssl.conf` pour mettre à jour le chemin vers le certificat généré et la clé.

2.2 Personnalisation

Créez une page web d'authentification en remplaçant `/var/www/html/index.html` par :

```
<html>
<head><title> Petit formulaire</title></head>
<body>
<h1> Formulaire </h1><hr>
<form action="/cgi-bin/scriptpass.pl" method="get">
login:<input type="text" name="login" size=40><p>
pass:<input type="text" name="pass" size=40><p>
<input type="submit" value="soumettre">
<input type="reset" value="RAZ">
</form><hr>
</body>
</html>
```

1. saisir la commande `setxkbmap fr` pour avoir un clavier français ou dpkg-reconfigure keyboard-configuration pour le rendre persistant.

Ajoutez, avec les droits (`-rwxr-xr-x`), le script `cgi` dans `/usr/lib/cgi-bin/scriptpass.pl`

```
#! /usr/bin/perl
print "Content-type: text/html\r\n\r\n";
$query_string =$ENV{'QUERY_STRING'};
($champ1, $champ2) = split (/&/, $query_string);
print $champ1,"<br>";
print $champ2;
exit(0);
```

Le module `cgid` a normalement déjà été chargé par la commande `a2enmod` et un redémarrage du serveur devrait permettre d'afficher la page d'authentification. Vérifiez le bon fonctionnement par une connexion locale en `http` et en `https`. Regardez au passage le certificat que vous avez généré. Attention, certains navigateurs récents refusent la connexion à un site sécurisé par un certificat auto-délivré.

3 Firewall `ufw`

Les distributions postérieures à 2016 basées sur `Debian` et `Ubuntu` intègrent *Uncomplicated Firewall*, une surcouche à `iptables` ou `nftables`, gérée par la commande `ufw` inactive par défaut sur la lubuntu. Il faut l'activer par la commande `ufw enable` et définir une politique par défaut pour autoriser le trafic sortant et interdire tout trafic entrant (sans bit d'acquitement) :

```
# ufw default deny incoming
# ufw default allow outgoing
```

Pour autoriser l'accès à un service hébergé sur la machine (comme `http` qui écoute sur le port 80), on utilise la directive `allow` de la commande `ufw` en précisant soit le service soit le port.

```
# ufw allow http
```

En utilisant la directive `deny`, on refuserait la connexion à un service avec la même syntaxe.
Il est utile de voir les règles actives pour la mise au point. On accède à la liste numérotée des règles par :

```
# ufw status numbered
```

qui affiche

```
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80                         ALLOW IN    Anywhere
```

Il est possible de changer la verbosité par :

```
# ufw status verbose
```

Une règle inutile ou obsolète pourra être supprimée par la directive `delete` en précisant le numéro :

```
# ufw delete 1
```

Voici une documentation plus riche qui permettra d'autoriser les connexions entrantes pour votre serveur web, voire d'activer la journalisation.

Vérifiez que vous pouvez accéder à votre serveur web depuis une autre machine (la machine physique qui héberge le gestionnaire de machines virtuelles).

4 MIM par ettercap

Le but est d'intercepter la transmission `login/pass` sur la page d'authentification du serveur web. **Indication** : Il faudra peut-être modifier le fichier de configuration `etter.conf` (de `/etc/ettercap`) pour réaliser l'attaque MITM depuis la `kali` (`redir_command*`).
Après avoir lancé `ettercap`, vérifiez que vous êtes en `unified sniffing`, sélectionnez l'interface réseau à utiliser, scannez les hôtes du réseau, sélectionnez les cibles, lancez le type d'attaque (ici l'arp poisonning) puis le "sniffing".

4.1 Attaque du serveur classique

Le formulaire ajouté est publié par le serveur `http` (i.e. sans être sécurisé par `openssl`). Votre première mission est d'intercepter la connexion qui transmet le couple login/password non sécurisés.

Comment pouvez-vous empêcher le serveur apache de publier la page de connexion sans qu'il soit sécurisé par `openssl` ?

4.2 Attaque MITM sur le serveur sécurisé

Interceptez maintenant la connexion qui transmet le couple login/password sécurisés, comme vous l'avez déjà fait pour la session `http`.
Vérifiez par la commande `arp` sur chacune des victimes que le contenu des tables `arp` a bel et bien été modifié ! L'attaque est maintenant active et tous les paquets circulant entre la source et la cible sont redirigés vers la machine qui héberge `ettercap`. Stoppez `ettercap` puis connectez-vous par un navigateur et affichez le certificat. Notez ce qui se passe et comparez avec les informations d'origine.
Puis relancez `ettercap` et constatez les différences.

> Je rappelle que l'usage de cet outil est TOTALEMENT INTERDIT sauf sur votre propre réseau local –physique ou virtuel–. Vous vous êtes engagés dans la charte informatique que vous avez signée à ne pas employer de tels outils. Les sanctions, dans la jurisprudence (il y en a) vont du simple conseil de discipline (avec interdiction de passer des examens pendant 5 ans) à des peines de prison (sans sursis) de plusieurs mois.