Maintien d'accès sur un site web

1 Contexte :

Il s'agit d'une intervention très technique, qui permet de déterminer le potentiel réel d'intrusion et de destruction d'un pirate sur l'infrastructure à auditer, et de valider l'efficacité réelle de la sécurité appliquée aux systèmes, au réseau et à la confidentialité des informations.

Cette véritable procédure d'audit se réalise en 5 étapes :

- ✓ Collecte d'information.
- ✓ Scan.
- ✓ Acquisition d'accès.
- ✓ Maintien d'accès.
- ✓ Effacement de traces.

Notre atelier porte sur la 4ème phase à savoir le maintien d'accès qui a pour objectif de garder et utiliser les accès obtenus pour poursuivre l'audit du réseau ainsi que Maintenir le contrôle de la machine une fois pénétrée.

2 Réalisation :

Cet atelier comprend deux principales parties, la première concerne les attaques pouvant être réalisées une fois qu'on a l'accès sur un site web tout en garantissant un accès ultérieur, la deuxième met l'accent sur la politique de sécurité qu'il faut adopter pour rendre le site inaccessible et empêcher toute intrusion possible.

2.1 Outils de maintien d'accès :

- ❖ Pour assurer un premier accès sur le site web, on a utilisé l'outil c100.txt qui doit être ajouté aux fichiers du site web.
- ❖ En se connectant sur le site, on accède à ce fichier qui offre plusieurs méthodes intégrées permettant d'auditer le serveur apache 2 du site et de déterminer les failles existantes.
- ❖ Failles découvertes :

✓ Installation d'un backdoor sur la machine cible et possibilité d'accéder sans utiliser le site une autre fois.

```
nc adresse_host port
```

✓ Exécution de commandes critiques et même avoir le contrôle total du la machine hébergeant le site.

Exemples:

- Suppression de fichier ou répertoire :

```
rm  /opt/lampp/htdocs/test_securinets/test
```

- Détection des ports ouverts

```
Show opned ports
```

✓ Accès et modification de la base de données du site web.

Exemples:

- Création d'une nouvelle BD : Create new DB
- Suppression : Delete
- Exécution de requête SQL : SQL-Query

2.2 Politique de sécurité :

Pour assurer la sécurisation de notre site on propose deux approches, l'une basée sur l'utilisation d'un gestionnaire d'intégrité pour nous signaler toute éventuelle modification du site, et l'autre se base sur la limitation des droits d'accès de notre serveur.

a) Utilisation du gestionnaire d'intégrité Tripwire :

Etapes élaborées :

✓ Install tripwire

✓ Configuration de tripwire : En spécifiant la liste des dossiers à superviser ainsi que la stratégie de contrôle dans le fichier */etc/tripwire/twpol.txt*

- ✓ Initialisation de Tripwire à l'aide de la commande

```
/usr/sbin/tripwire --init
```

- ✓ Vérification manuelle (si un fichier est changé ou pas) à l'aide de la commande :

```
/usr/sbin/tripwire --check
```

- ✓ Génération du rapport après la vérification à l'aide de la commande :

```
/usr/sbin/twprint -m r --twrfile /var/lib/tripwire/report/<name>.twr

<name> : est le fichier le plus récent sous /var/lib/tripwire/report, son nom est composé de la date et l'heure de son création c.à.d. le moment de réalisation de CHECK.
```

- ✓ Après changement de la stratégie (fichier /etc/tripwire/twpol.txt) il faut générer un nouveau fichier signé /etc/tripwire/tw.pol pour appliquer la modification de la stratégie qui se trouve dans ce fichier à l'aide des commandes:

```
/usr/sbin/twadmin --create-polfile -S site.key /etc/tripwire/twpol.txt rm

/var/lib/tripwire/nom.domain.com.twd

 /usr/sbin/tripwire --init
```

b) Modification des permissions du serveur:

Dans cette étape de sécurisation, on va limiter les privilèges avec lesquels le serveur Apache s'exécute, pour cela on doit modifier le fichier `/opt/lampp/etc/httpd.conf`

- ✓ Changer le groupe d'apache a nobody :

```
user nobody

group nobody
```