TP milxc et sécurité des échanges par mail

Introduction

Nous allons voir dans ce TP les différentes failles présentes dans les échanges par mail, ainsi que différentes mesures pour réduire les risques. Pour ce faire, nous allons utiliser **milxc** qui est un système basé sur des conteneurs **lxc** simulant un mini-internet. Vous pouvez entrer la commande `./mi-lxc.py print` pour afficher la représentation du réseau étudié si besoin.

Nous allons principalement nous intéresser aux sous-réseaux **target**, **gcorp** et **isp-a**. Dans ceux-ci, les machines que nous allons manipuler seront principalement **admin** et **dmz**, **user** et **infra** ainsi que **hacker** et **infra**.

L'utilisateur **hacker** va chercher à exploiter les failles de l'infrastructure, et votre travail va être de sécuriser l'infrastructure de mail des compagnies **gcorp** et **target**. Les serveurs de mail se trouvent sur les machines **infra** et **dmz**, les clients sont les machines **user**, **admin** et **hacker**. Les machines des clients possèdent un client de messagerie **claws mail** vous permettant d'envoyer vos messages pour tester. Pour les plus curieux, vous pouvez utiliser la commande `nc`, une annexe avec des exemple est fournie à la fin de ce document. Enfin, des mails automatique peuvent être écris dans le fichier **mailscript/scenarii.json**. Ces tests seront éxécutés avec la commande `./mi-lxc.py mail`.

Première approche

Question 1 : Avec l'architecture actuelle, donnez 2 exemples d'échange de mail frauduleux, essayez de les mettre en place.

Il est bon de savoir que les serveurs de mail on pris l'habitude de se diviser les taches en deux. Deux ports sont ouverts sur le serveur ; les ports **587** et **25**. Le port 25 était le port par défaut par lequel tous les échanges passaient. Maintenant, le port 587 est ouvert pour permettre aux clients locaux de parler avec leur serveur directement, et le port 25 sert à communiquer uniquement entre serveurs mail. Cette division a pour but de réduire les spams en forçant à communiquer via un serveur de mail. Si ce dernier obtient une mauvaise réputation il pourra être black-listé et les serveurs ne l'écouteront plus.

Question 2 : Dans la configuration actuelle, cette division des tâches est-elle mise en place sur tous les serveurs mails ? Mettez en place cette configuration dans le cas où elle n'est pas encore appliquée.

Par la suite, on se concentrera sur le serveur mail **smtp.target.milxc**.

Comme expliqué ci-dessus, le port **587** autorise seulement les clients locaux du serveur à communiquer avec ce dernier.

Question 3 : Donnez les restrictions pouvant satisfaire cette condition. Ces restrictions sont-elles actuellement appliquées sur le serveur mail de target ?

Question 4 : En jouant avec la configuration postfix des serveurs, essayez d'obtenir un message de refus en voulant communiquer depuis l'extérieur du sous-réseau du serveur mail de target sur le port 587.

Autoriser seulement son sous-réseau n'est pas suffisant pour pour protéger le port 587 de son serveur mail. On veut éviter que les utilisateurs locaux effectuent des actions malveillantes.

Question 5 : Avec la configuration établie aux questions précédentes, que peut faire un utilisateur local de malveillant en communiquant sur le port 587 ?

Question 6 : Proposer et implémenter une solution pour empêcher cette action.

Maintenant que l'on a empêché nos utilisateurs d'effectuer des actions malveillantes, rien ne nous garantit que tous les serveurs de mails sont configurés correctement. De plus, nous n'avons pas protégé le port 25 du serveur. Il faut ainsi se protéger des usurpateurs extérieurs et protéger son port 25 pour n'autoriser que les serveurs de mail à communiquer dessus. Pour cela, vous allez implémenter une politique **SPF**.

SPF

Question 7 : Montrez qu'un utilisateur extérieur peut toujours usurper l'adresse mail d'**admin**.

Pour implémenter la politique **SPF**, nous allons utiliser **policyd-spf**. Il faut installer le package avec la commande `sudo apt-get install postfix-policyd-spf-python`. En vous aidant de `man` (ou autres sources) effectuez les manipulations pour activer **SPF** avec **postfix**.

Question 8 : Avez vous pensé à rajouter la politique SPF dans la configuration DNS ? Si non, vous pouvez trouver le fichier à modifier dans `/etc/nsd/*.zone` dans les serveurs de mail.

Question 9 : Pouvez vous toujours usurper l'adresse d'**admin** ou de **user** avec **hacker** ?

Question 10 : Proposez divers scénarios à tester pour montrer que votre implémentation est bonne. Aidez-vous du fichier **scénarii.txt** et de la commande `./mi-lxc.py mail` pour vérifier votre configuration.

Annexes et remarques

Lien intéressant pour manipuler les différents paramètres de postfix : https://man.archlinux.org/man/postconf.5

Pour les questions 1 à 6, regardez principalement :

`smtpd_client_restrictions` et `smtpd_sender_restrictions`.

Dans la doc, on trouve des références à cette documentation : **virtual(5) alias** ou **canonical(5) mapping**. Cela veut simplement dire que vous devez créer des fichiers `/etc/postfix/file` et que vous devez ensuite exécuter : `postmap /etc/postfix/file`. Les fichiers `file` sont de la forme :

```
pattern action
```

Par exemple, dans le fichier `/etc/postfix/file` nous avons :

```
127.0.0.1 OK
nomde.domaine OK
```

Plus de détails : http://www.postfix.org/access.5.html