

TP milxc et sécurité des échanges par mail

Introduction

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

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

En ce qui concerne le contexte, **hacker** va chercher à exploiter les failles et votre travail va être de sécuriser l'infrastructure de mail des compagnies **gcorp** et **target**. Pour vous aider, les serveurs de mail se trouvent sur les machines **infra** et **dmz**, les clients sont donc les machines **user** et **admin**. Les machines des clients possèdent un client de messagerie **claw mail** vous permettant d'envoyer vos messages pour tester. Pour les plus curieux vous pouvez utiliser la commande `nc`, une annexe avec des exemples est fournie à la fin de ce document. Enfin, des mails automatiques peuvent être écrits dans le fichier **scenarios.json**. Ces tests seront exécutés 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 ont pris l'habitude de se diviser les tâches en 2. 2 ports sont ouverts sur le serveur. Ce sont 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. Le port 25 sert maintenant à communiquer uniquement entre serveur 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 n'écouteront plus ce dernier.

Question 2 Dans la configuration actuelle, cette protection est-elle mise en place ? Si oui expliquez comment. Si non, montrez le.

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

Question 3 Donnez les restrictions qui peuvent s'appliquer pour s'assurer de cette condition. Ces restrictions sont-elles appliquées ?

Question 4 En jouant avec la configuration postfix des serveurs et les clients essayer d'obtenir les messages de refus en voulant communiquer depuis extérieur sur le port 587.

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

Question 5 Avec la bonne configuration précédente, en communiquant sur le port 587 que peut faire un utilisateurs local de malveillant ?

Question 6 Proposer un solution pour empêcher cette action.

Maintenant que l'on a empêché nos utilisateurs d'effectuer des actions malveillantes, rien ne nous garantis que tous les serveurs de mails sont configurés correctement. De plus, nous n'avons pas protégé le port 25 des serveurs ce qui réduis à néant ce que nous venons de faire. Il faut alors se protéger des usurpateurs extérieur 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 `sudo apt-get install postfix-policyd-spf-python`. En vous aidant du 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 tous les tests intéressants et différents pour montrer que votre implémentation est bonne. Pensez à rajouter des commentaires pour expliquer vos tests. Aidez vous du fichier **scénarios.txt** et de la commande `./mi-lxc.py mail` pour vérifier votre bonne 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** or a **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>