Lutte Informatique Défensive
SOC, CERT et CTI

TP3 - Analyse de malware

Objectifs de réalisation

En phase de réponse à incident : savoir analyser des échanges réseau, extraire des fichiers capturés dans un PCAP, analyser ces fichiers pour identifier le niveau de nocivité et produire des règles de détection.

Format du TP

Travail en groupe (2 ou 3 personnes par groupe) à réaliser sur 4h.

Pré-requis:

- VirtualBox
- Wireshark

Livrable

Le rapport devra être envoyé par email à l'adresse *frederic.guihery@amossys.fr* **à la fin de la séance de TP.**

Consignes

Chapitre 1 - Installation de votre environnement

1.1 Création d'une machine virtuelle Debian avec l'outil Vagrant

Vagrant est un logiciel libre et open-source pour la création et la configuration des environnements de développement virtuel. Il peut être considéré comme un *wrapper* autour de logiciels de virtualisation comme VirtualBox.

Dans le cadre de ce TP, nous allons utiliser Vagrant pour obtenir une machine virtuelle Debian installée et prête à l'emploi.

1.1.1 Installation de Vagrant

Sur une Ubuntu ou une Debian à jour, l'installation de vagrant se résume à l'exécution de la commande suivante :

```
$ sudo apt-get install vagrant
```

Si vous utilisez une autre distribution, utilisez votre package manager pour installer le package officiel Vagrant disponible sur: https://www.vagrantup.com/downloads.html.

Une fois installé, vérifiez le bon fonctionnement de Vagrant en affichant sa version tel qu'illustré ci-dessous.

```
$ vagrant -v
Vagrant 2.1.5
```

1.1.2 Création d'une machine virtuelle Debian en mode Desktop

Une fois Vagrant installé, créez un répertoire dédié à votre TP.

```
$ mkdir tp3

$ cd tp3

$ vagrant init 4linux/debian9-desktop

$ vagrant up
```

Vous pouvez vérifier que la machine a bien été créé, en ouvrant VirtualBox.

Maintenant, lancer la machine virtuelle nouvellement créée, avec VirtualBox.

La suite de ce TP sera réalisé dans cette nouvelle VM et non plus sur votre Host.

- Login : **vagrant**
- Mot de passe : **vagrant**

Attention : le clavier est en qwerty par défaut.

Pour changer la disposition du clavier, une fois connecté, ouvrez un terminal et tapez :

```
$ sudo setxkbmap fr
```

Pour permettre le partage de fichiers entre votre Host et cette machine virtuelle, tapez :

```
$ mount -t vboxsf vagrant /media/cdrom/
```

Chapitre 2 - Phase de réponse à incident

2.1 Récupération du fichier PCAP à analyser

Lien de téléchargement : http://dl.free.fr/gqq5EBdid

Login : istic
Mot de passe : istic

2.1 Identification des conversations dans le fichier PCAP

Analysez le fichier PCAP et répondez aux questions suivantes :

- Combien il y a t-il d'échanges sur cette capture ?
- Quelles conversations identifiez-vous (ip/port) ?
- Quels sont les échanges applicatifs ?
- Quel peut être le rôle de chaque échange ?

Note : vous pouvez colorier les échanges avec Wireshark pour mieux vous y retrouver.

2.2 Extraction de fichier

Avec Wireshark, dumpez le fichier transmis dans un des échanges.

Quelles sont les caractéristiques de ce fichier ? (métadonnées)

2.3 Analyse statique de binaire

Outil : IDA Pro free

Installez la version Freeware pour Linux :

https://www.hex-rays.com/products/ida/support/download_freeware.shtml

Au travers d'une analyse statique avec IDA Pro, qu'arrivez vous à identifier en ce qui concerne :

- le fonctionnement du binaire ?

- d'éventuels éléments permettant de le caractériser ? (IoC - Indicateurs de compromission)

2.4 Analyse réseau

Outil : Wireshark

Analysez le canal de commande et contrôle (C&C), et répondez aux questions suivantes :

- d'après vous, quelle est la syntaxe du protocole ?
- identifiez vous une séquence de messages ? Si oui, quelle est cette séquence ?
- pouvez vous en déduire des caractéristiques permettant de tracer des échanges de ce protocole sur le réseau ? (dans l'optique de produire des règles NIDS)

2.5 Production de règles NIDS

A partir de l'analyse du canal de communication du protocole, produisez une règle de détection pour Snort ou Suricata, qui sera à inclure dans le rapport.

Note : la qualité de la règle dépend notamment de deux aspects : sa généralisation au protocole étudié (pour limiter les faux négatifs), et également le fait qu'elle ne doit pas englober d'autres protocoles légitimes (afin d'éviter d'éventuels faux positifs).

2.6 Analyse dynamique de binaire

Etant donné que le binaire extrait du fichier PCAP est compilé pour une architecture i386, si votre environnement est en 64 bits, vous devez ajouter la comptabilité i386 :

```
$ sudo dpkg --add-architecture i386
$ sudo apt-get update
$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
```

A l'aide des outils *netstats* et *auditd*, identifiez l'impact de l'exécution du binaire sur l'OS. Dans le rapport, décrivez à la fois votre démarche d'analyse avec ces deux outils ainsi que les résultats (en commentant les traces obtenues).

Indice : il peut être nécessaire de stimuler le binaire exécuté.

Note : l'installation d'*auditd* se fait avec la commande suivante :

```
$ sudo apt-get install auditd
```

Avec l'outil *auditd*, il est conseillé de seulement tracer les accès en écriture sur le système de fichiers.

Il est recommandé de s'appuyer sur logstash pour filtrer les événements et les métadonnées produits par auditd.

Logstash nécessite java :

```
$ sudo apt-get install openjdk-8-jre
```

L'installation de logstash est décrite ici :

https://www.elastic.co/guide/en/logstash/current/installing-logstash.html

Pour manipuler logstash, il est conseillé de travailler en ligne de commande. Il faut donc au préalable stopper le service :

```
$ sudo service logstash stop
```

Ensuite, il faut créer un fichier de configuration, tel que /etc/logstash/conf.d/my_file.conf

```
input {
  file { path => ["/var/log/audit/audit.log"] }
}

output {
        stdout {}
        file { path => "/tmp/audit_logstash.log" }
}
```

2.7 Production des indicateurs de compromission (IoC)

Outil : https://www.iocbucket.com/openioceditor#

Décrivez dans cette interface web l'ensemble des indicateurs identifiés.

Exporter les indicateurs au format OpenIOC (XML) et inclure le fichier XML dans le rapport.