ESIEA - IDS/IPS - TD2

Éric Leblond

10 décembre 2018

Introduction

Le but de ce TD est d'expérimenter l'écriture de règles. Pour chaque signature, veuillez à bien fournir l'explication de l'ensemble des arguments utilisés.

Note : Les fichiers pcap communiqués ne doivent pas sortir de la salle.

1 Extraction de fichiers

Question 1 *Ecrire une règle réalisant sur l'extraction des fichiers PDF.*

Question 2 *Récupérer les slides depuis le fichier slides.pcap* `http://home.regit.org/~regit/slides.pcap`.

2 Cas 1

La fichier `case1.pcapng` contient les traces d'un malware de type Havex. Le passage de commande se fait par le biais de commentaires dans le retour envoyé par le serveur :

```
<\!--havex COMMANDE havex-->
```

Question 3 *Écrire une règle cherchant le motif* `havex` *dans les réponses HTTP.*

Question 4 *Analyser le flux concerné avec Wireshark. Que remarque-t-on ?*

Question 5 *Procéder à l'analyse du moteur pour cette règle. Fixer les warnings.*

Question 6 *Raffiner la règle avec une expression régulière pour trouver les commentaires recherchés.*

3 Cas 2

La fichier `case2.pcapng` contient les traces d'un malware de type Havex réalisant un passage de commande par stéganographie. Des images PNG modifiées sont récupérées pour y trouver des commandes.

Le répertoire tools contient un outil de stéganographie.

Question 7 *Extraire les fichiers png des échanges.*

Question 8 *Les passer à show.py.*

4 Cas 3

19 Dans ce cas le malware réalise une exfiltration de données par requêtes DNS. Voir `http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DNS_Exfiltration_2011-01-01_v1.1.pdf` pour plus d'information.

Question 9 *Écrire une alerte prévenant d'une résolution sur le domaine* `supervilain.ru`.

L'algorithme de camouflage des données est inversé par XOR avec `0xF2`.

Question 10 *Modifier l'alerte pour qu'elle appele un script lua affichant la chaine décodée sur la sortie standard.*

5 Réaction

La bibliothèque paramiko permet d'écrire facilement un client SSH en python. Voir `http://jessenoller.com/blog/2009/02/05/ssh-programming-with-paramiko-completely-different` par exemple.

Question 11 *Utiliser paramiko pour effectuer une connexion sur la machine SELKS.*

Question 12 *Analyser les traces générées dans Suricata.*

L'outil DOM `https://github.com/regit/DOM` recherche dans un fichier EVE des connexions SSH suspectes et ajoute les IPs à l'origine de ces connexions dans un ensemble ipset.

Question 13 *Créer un ensemble ipset où les IPs contenues résident pour 5 min. Ajouter une règle pour cet ensemble interdisant les connexions SSH sur le machine.*

Question 14 *Installer DOM sur la machine et le paramétrer pour qu'il ajoute les connexions utilisant paramiko à l'ensemble précédemment défini.*

Question 15 *Valider le bloquage et le retrait de l'IP de la liste de bloquage au bout de 5 minutes.*

6 Mode IPS

Question 16 *Ajouter un fichier de règle au YAML contenant une règle de bloquage des connexions SSH avec paramiko.*

Question 17 *Arréter suricata et le relancer en mode IPS sur la queue 1.*

Question 18 *Ajouter une règle de filtrage pour que les flux SSH passent par Suricata*

7 Kill the player

Question 19 *Modifier le script lua de pour stocker le résultat du déchiffrage dans un fichier.*

Question 20 *Modifier DOM pour qu'il ajoute à l'ensemble ipset en fonction du user agent HTTP.*