

Rapport d'Analyse de Malware

Projet : Constructing Defense

Sujet : Déploiement d'un laboratoire d'analyse et étude d'un malware

Étudiant : Mohamed Rayen AKAICHI

Infrastructure : DetectionLab (Windows 10) & Kali Linux

25 février 2026

Table des matières

1	Résumé Exécutif	3
2	Mise en Place de l'Environnement (Lab Setup)	3
2.1	Déploiement de DetectionLab	3
2.2	Transfert du Malware (Kali vers Windows)	4
3	Identification de l'Échantillon	4
4	Analyse Statique (Kali Linux)	4
5	Analyse Dynamique (Windows 10)	5
5.1	Chaîne d'Exécution (Process Monitor)	5
5.2	Activité Botnet (Relais Proxy)	5
6	Conclusion	6

1 Résumé Exécutif

Ce rapport détaille la mise en place d'une infrastructure d'analyse de malwares (DetectionLab) et son utilisation pour étudier un échantillon de la famille **Socks5Systemz**.

L'analyse démontre comment ce malware utilise l'ingénierie sociale (faux installateur de convertisseur audio) et des techniques de "packing" (Inno Setup) pour infiltrer un système. Une fois exécuté, il enrôle la machine dans un botnet en communiquant avec un serveur de commande (C2) via HTTP. Le rapport fournit les indicateurs de compromission (IOCs) et une règle YARA pour la détection.

2 Mise en Place de l'Environnement (Lab Setup)

L'analyse dynamique nécessite un environnement isolé et instrumenté. j'ai choisi de déployer **DetectionLab** sur une machine hôte Windows.

2.1 Déploiement de DetectionLab

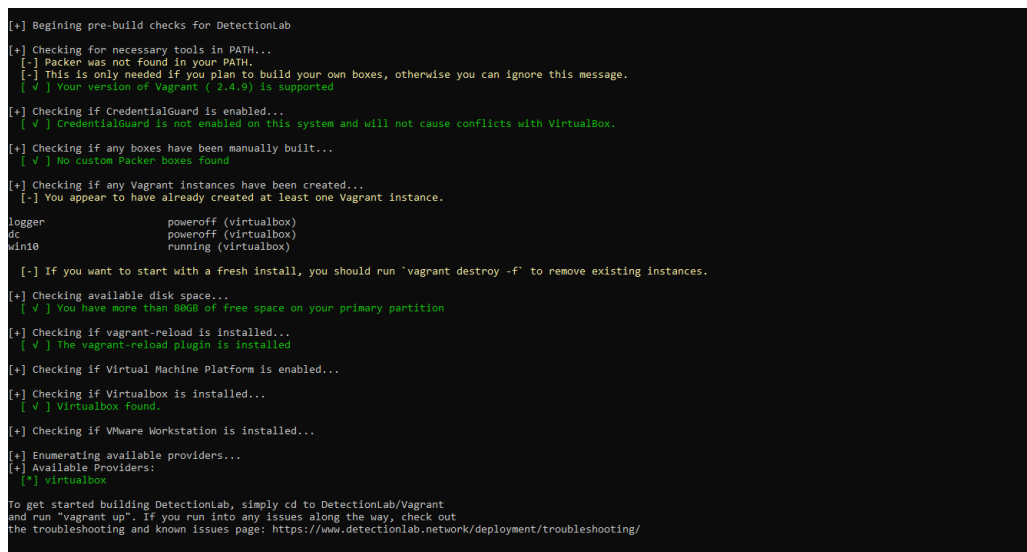
DetectionLab automatise la création d'un domaine Windows surveillé (Active Directory, Splunk, Sysmon).

Pré-requis et Installation :

1. Installation de **VirtualBox** et **Vagrant** sur l'hôte Windows.
2. Désactivation des conflits de virtualisation (Hyper-V, Isolation du noyau) pour assurer la stabilité des VMs.
3. Déploiement via PowerShell :

```
1 git clone https://github.com/clong/DetectionLab.git
2 cd DetectionLab/Vagrant
3 ./prepare.ps1
4 vagrant up
```

Listing 1 – Commandes de déploiement



```
[*] Beginning pre-build checks for DetectionLab
[*] Checking for necessary tools in PATH...
[-] Packer was not found in your PATH.
    [-] This is only needed if you plan to build your own boxes, otherwise you can ignore this message.
    [✓] Your version of Vagrant ( 2.4.9 ) is supported
[*] Checking if CredentialGuard is enabled...
    [✓] CredentialGuard is not enabled on this system and will not cause conflicts with VirtualBox.
[*] Checking if any boxes have been manually built...
    [✓] No custom Packer boxes found
[*] Checking if any Vagrant instances have been created...
    [-] You appear to have already created at least one Vagrant instance.

logger      poweroff (virtualbox)
dc           poweroff (virtualbox)
win10       running (virtualbox)

[-] If you want to start with a fresh install, you should run 'vagrant destroy -f' to remove existing instances.

[*] Checking available disk space...
    [✓] You have more than 800G of free space on your primary partition
[*] Checking if vagrant-reload is installed...
    [✓] The vagrant-reload plugin is installed
[*] Checking if Virtual Machine Platform is enabled...
[*] Checking if Virtualbox is installed...
    [✓] Virtualbox found.
[*] Checking if VMware Workstation is installed...
[*] Enumerating available providers...
[*] Available Providers:
    [*] virtualbox

To get started building DetectionLab, simply cd to DetectionLab/Vagrant
and run 'vagrant up'. If you run into any issues along the way, check out
the troubleshooting and known issues page: https://www.detectionlab.network/deployment/troubleshooting/
```

FIGURE 1 – Déploiement automatisé des VMs via Vagrant

2.2 Transfert du Malware (Kali vers Windows)

Pour transférer l'échantillon de la machine d'attaque (Kali Linux) vers la victime (Win10) sans exposer l'hôte, j'ai mis en place un serveur HTTP temporaire sur un réseau Host-Only isolé.

Configuration Réseau :

- Kali Linux IP (Host-Only) : 192.168.56.108
- Win10 Victime IP : 192.168.56.104

Exécution du transfert : Sur Kali, un serveur Python a été lancé dans le répertoire du malware :

```
1 sudo python3 -m http.server 80
```

Listing 2 – Lancement du serveur Python sur Kali

Sur la machine victime Windows 10, le fichier a été récupéré via le navigateur à l'adresse `http://192.168.56.108/malware.exe`.

Directory listing for /



FIGURE 2 – Méthode de transfert du malware via HTTP

3 Identification de l'Échantillon

Nom du fichier	f027ab54...2b.exe (Renommé SystemUpdate.exe)
Type de fichier	PE32 Executable (GUI) Intel 80386
Taille	546 KB
Hash SHA256	f027ab542dc3fa47097d26472933cdb50c9960f00aab2e7a32443db5ae89ed2b
Classification	Dropper / Botnet Client (Socks5Systemz)

TABLE 1 – Carte d'identité du malware

4 Analyse Statique (Kali Linux)

L'analyse des chaînes de caractères (`strings`) a révélé que le malware est encapsulé ("packed") dans un installateur **Inno Setup**.

```
1 This installation was built with Inno Setup.  
2 FileDescription: Audio Book Conversion Setup
```

Listing 3 – Indicateurs Inno Setup

Cette technique de "Trojan" vise à tromper l'utilisateur en se faisant passer pour un logiciel légitime de conversion audio.

5 Analyse Dynamique (Windows 10)

L'échantillon a été exécuté sur la VM Windows 10 instrumentée.

5.1 Chaîne d'Exécution (Process Monitor)

L'outil **Process Monitor** a permis de tracer l'infection :

1. **Dropper** : Le faux installateur s'exécute.
2. **Installation** : Il déploie la charge utile dans %LOCALAPPDATA%.
3. **Persistance** : L'exécutable audiobookconversion14230.exe est lancé.

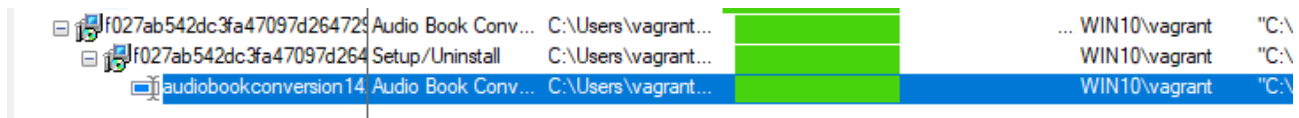


FIGURE 3 – Visualisation de l'arbre des processus infectieux

5.2 Activité Botnet (Relais Proxy)

L'analyse prolongée du trafic réseau (fichier CSV exporté depuis Wireshark) a révélé que la machine compromise est activement utilisée comme nœud de sortie (Proxy) pour le botnet.

Nous avons observé la séquence suivante :

1. Une IP externe (94.26.38.3) envoie une requête HTTP GET /rand à notre machine victime.
2. Immédiatement après, notre machine relaie cette même requête vers une cible tierce (45.11.182.186).
3. La réponse de la cible est ensuite renvoyée à l'IP initiale.

Cette activité confirme que le malware **Socks5Systemz** transforme l'hôte en relais pour masquer les activités illicites des attaquants.

148	41.626975	10.0.2.15	94.26.38.3	TCP	66	49700 → 2024	[SYN]	Seq=0	Win=64240	Len=0	MSS=1460	WS=256	SACK_PERM
149	41.719993	94.26.38.3	10.0.2.15	TCP	60	2024 → 49700	[SYN, ACK]	Seq=0	Ack=1	Win=65535	Len=0	MSS=1460	
150	41.720089	10.0.2.15	94.26.38.3	TCP	54	49700 → 2024	[ACK]	Seq=1	Ack=1	Win=64240	Len=0		
151	41.720137	10.0.2.15	94.26.38.3	TCP	55	49700 → 2024	[PSH, ACK]	Seq=1	Ack=1	Win=64240	Len=1		
152	41.720392	94.26.38.3	10.0.2.15	TCP	60	2024 → 49700	[ACK]	Seq=1	Ack=2	Win=65535	Len=0		
153	41.720407	10.0.2.15	94.26.38.3	TCP	342	49700 → 2024	[PSH, ACK]	Seq=2	Ack=1	Win=64240	Len=288		
154	41.720775	94.26.38.3	10.0.2.15	TCP	60	2024 → 49700	[ACK]	Seq=1	Ack=290	Win=65535	Len=0		
155	41.837643	94.26.38.3	10.0.2.15	TCP	60	2024 → 49700	[PSH, ACK]	Seq=1	Ack=290	Win=65535	Len=2		
156	41.886160	10.0.2.15	94.26.38.3	TCP	54	49700 → 2024	[ACK]	Seq=290	Ack=3	Win=64238	Len=0		
10370	104.680245	10.0.2.15	94.26.38.3	TCP	66	49732 → 2024	[SYN]	Seq=0	Win=64240	Len=0	MSS=1460	WS=256	SACK_PERM
10371	104.687140	94.26.38.3	10.0.2.15	TCP	60	2024 → 49732	[SYN, ACK]	Seq=0	Ack=1	Win=65535	Len=0	MSS=1460	
10372	104.687226	10.0.2.15	94.26.38.3	TCP	54	49732 → 2024	[ACK]	Seq=1	Ack=1	Win=64240	Len=0		
10373	104.687300	10.0.2.15	94.26.38.3	TCP	55	49732 → 2024	[PSH, ACK]	Seq=1	Ack=1	Win=64240	Len=1		
10374	104.687365	10.0.2.15	94.26.38.3	TCP	345	49732 → 2024	[FIN, PSH, ACK]	Seq=2	Ack=1	Win=64240	Len=291		

FIGURE 4 – Preuve du comportement de "Man-in-the-Middle" du botnet

6 Conclusion

L'analyse a confirmé la nature malveillante du fichier. Il s'agit d'un botnet Socks5Systemz utilisant un installateur légitime (Inno Setup) pour l'évasion. L'analyse dynamique a non seulement révélé l'exfiltration de données vers un C2, mais a également prouvé l'utilisation active de la machine compromise comme proxy SOCKS, transformant la victime en bouclier pour les activités des cybercriminels.