Authentification

1. Préambule

L'objectif de ce TP est d'explorer le schéma d'authentification le plus répandu sur les systèmes d'exploitation : l'authentification à base de mot de passe.

2. Prérequis

Deux machines virtuelles au format VirtualBox ont été préparées pour ce TP :

- SES_TP1_Kali : utilisation d'une distribution Kali Linux contenant notamment les outils John-The-Ripper et PwDump (ce dernier a été mis à jour à partir des sources[1] GitHub) ;
- SES_TP1_Win10 : dérivée de la machine virtuelle Windows 10[2] mises à disposition par Microsoft pour tester le navigateur Edge, le clavier a été passé en français pour éviter les désagréments de saisie avec un clavier QWERTY, l'antivirus Windows Defender a été désactivé et plusieurs outils ont été ajoutés (Mimikatz, John-The-Ripper et la suite Sysinternals).

Pour la suite des TP, il sera utile de récupérer un aide-mémoire sur John-The-Ripper (JtR) : https://countuponsecurity.com/2015/06/14/jonh-the-ripper-cheat-sheet/

3. Stockage des mots de passe

Exercice 3-1

Le mot de passe root de la VM Kali est toor (mot de passe courant sur cette distribution qui est tout sauf sûr).

JtR peut dériver une liste de mots de passe selon différentes règles. Comparer le nombre de mot de passe présents à l'origine dans le fichier /usr/share/john/password.lst et le nombre obtenu avec différents jeux de règles :

```
# john --wordlist=/usr/share/john/password.lst -rules --stdout > /dev/null

# john --wordlist=/usr/share/john/password.lst -rules:Jumbo --stdout >
/dev/null
```

Selon la machine utilisée, la ligne suivante peut prendre plusieurs minutes, il faut mieux la lancer avec une liste de mots de passe plus restreinte :

```
# john --wordlist=/usr/share/john/password.lst -rules:All --stdout > /dev/null
```

1 https://github.com/Neohapsis/creddump7/blob/master/framework/win32/hashdump.py
2 https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/

A l'aide de JtR, trouver les mots de passe de :

- James (fichier /root/hash1)
- Jessica (fichier /root/hash2)
- Bill (fichier /root/hash3)
- Fabrice (fichier /root/hash4) dont le mot de passe est constitué de 5 caractères (1 chiffre suivi de 4 lettres minuscules)

Comparer le temps d'obtention des mots de passe en fonction des algorithmes utilisés, ainsi que les mots de passe découverts pour james et jessica (le même mot de passe a été saisi à la création) du compte,

Exercice 3-2

Cet exercice vise à récupérer les mots de passe contenus dans la base SAM de la VM Windows 10. Habituellement, 2 fichiers sont récupérés en accédant hors-ligne à une partition système d'un Windows (par exemple, avec un live CD) :

- C:\Windows\System32\config\SYSTEM
- C:\Windows\System32\config\SAM

A titre d'information (ces fichiers sont déjà disponibles dans la VM Kali), on peut sous réserve d'avoir des privilèges administrateur générer ces mêmes fichiers avec les commandes suivantes :

```
reg save HKLM\SYSTEM system_w10.hiv

reg save HKLM\SAM sam_w10.hiv
```

Dans la VM Kali, taper la commande suivante :

```
# pwdump system_w10.hiv sam_w10.hiv > hash_w10
```

A partir du fichier hash_w10 obtenu, chercher le mot de passe du compte IEUser. Attention, le fichier généré contient 2 types de hash : LM et NTLM. Le premier est désactivé sur la machine d'origine et les hashs LM présents dans le fichier hash_w10 correspondent à un mot de passe vide.

4. Récupération d'éléments d'authentification en mémoire Windows

Exercice 4-1

Le mot de passe récupéré à l'exercice précédent permet d'ouvrir une session sur la machine virtuelle Windows 10.

L'outil Mimikatz permet de chercher des éléments d'authentification (par exemple, des hashs NTLM ou des tickets Kerberos) dans la mémoire du processus LSASS. Il est également possible de faire cette recherche dans un *dump* mémoire de ce processus obtenu à l'aide de l'utilitaire Syinternals Procdump :

```
procdump -ma lsasss.exe
```

Lancer Mimikatz (situé dans C:\Tp\Mimikatz) dans une invite de commande administrateur et charger un *dump* provenant d'une machine intégrée à un domaine Active Directory (mettant donc en œuvre de l'authentification Kerberos).

```
mimikatz # sekurlsa::minidump C:\Users\IEUser\Documents\lsass.dmp

mimikatz # sekurlsa::tickets
```

Chercher le nom d'un utilisateur du domaine LAB-AD dans les différents tickets.

Chercher si cet utilisateur identifié possède un hash NTLM à l'aide d'une des 2 commandes suivantes.

```
mimikatz # sekurlsa::msv
```

```
mimikatz # sekurlsa::logonPasswords
```

Si c'est le cas, peut-on retrouver son mot de passe à l'aide de JtR (installé dans C:\TP\john180j1w\run) ?

_____

_____

Mimikatz possède une fonction pour calculer le hash d'un mot de passe :

```
mimikatz # kerberos::hash /password:lemotdepassetrouvé
```

Si l'obtention du mot de passe n'est pas possible (i.e. mot de passe complexe), il reste souvent possible de réaliser une attaque de type Pass-The-Hash pour une machine en réseau.

Faite le même type d'opération sur la machine virtuelle. Il faut alors changer le contexte de Mimikatz :

```
mimikatz # sekurlsa::process
```

A l'issue, testez le mécanisme de protection LSA (https://technet.microsoft.com/en-us/library/dn408187(v=ws.11).aspx). Pour cela, il faut naviguer dans le registre (regedit.exe) jusqu'à la clé : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa, puis créer une valeur nommée RunAsPPL, de type DWORD, avec comme donnée 0x1. Une fois la valeur créée, il faut redémarrer la machine virtuelle.

Peut-on réaliser les mêmes actions ?

Au sein de Mimikatz, il est possible de désactiver la protection LSA à l'aide du driver fourni.

```
mimikatz # privilege::debug

mimikatz # !+

mimikatz #!processprotect /process:lsass.exe /remove
```

Vérifier que l'analyse de la récupération des éléments d'authentification est de nouveau possible.

Le chargement du driver Mimikatz n'est pas une opération discrète, il est possible d'en retrouver la trace dans les journaux Windows avec les commandes Powershell suivantes :

```
Get-EventLog System -Source 'Service Control Manager' | Where-Object
{ $_.EventID -eq 7045 } | Select-Object -first 5 TimeGenerated, Message | fl
```

5. Politique de mot de passe personnalisée sous Windows

Exercice 5-1

Pour cet exercice sera utilisé l'outil développé dans l'article suivant : https://blog.scrt.ch/2017/08/23/passfilt-dll-complexifier-sa-politique-de-mot-de-passe-windows/

Le code source est disponible sur GitHub, Le code utile est présent dans le fichier suivant : https://github.com/julesduvivier/PasswordFilter/blob/master/PasswordFilter/dllmain.cpp

Pour réaliser cet exercice, il faut désactiver de façon pérenne la protection LSA en supprimant la valeur RunAsPPL. En effet, la dll utilisée n'est pas signée (cf. l'article Technet sur la protection LSA).

Configurer le filtre à l'aide du programme C:\tp\PasswordFilter\PasswordFilterService.exe puis redémarrer pour que LSASS charge le filtre.

Exécuter ProcessExplorer (C:\tp\SysinternalsSuite\procexp64.exe) en tant qu'administrateur et vérifier (CTRL+D) les dll chargées dans le processus lsass.exe. Une dll nommée PasswordFilterx86_64_v1-2.dll doit y figurer.

Configurer la politique de mot de passe pour incorporer un dictionnaire (par exemple, celui fournit avec J-t-R C:\TP\john180j1w\run\password.lst) puis tester que changement de mot de passe. Les refus de changement de mot de passe sont consignés dans un fichier journal propre au filtre.

6. Authentification Kerberos

Exercice 6-1

A l'aide de Wireshark, analyser les fichiers « auth_krb_1.pcapng » et « auth_krb_2.pcapng ». Identifier les points suivants :

- nom du client
- SPN du TGS
- SPN du serveur

- algorithmes utilisés pour pour le TGT et l'*authenticator*

Pour le fichier « auth_krb_2.pcapng », quelle différence peut-on faire entre les 2 messages KRB_AS_REQ envoyés par le client ?

_____

_____