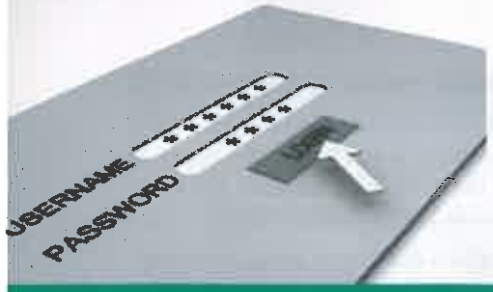


ÉDITION OFFLINE DE L'ACTIVE DIRECTORY

Nicolas RUFF – nicolas.ruff@eads.net – EADS Innovation Works

mots-clés : SAM / ACTIVE DIRECTORY / ESE / JET BLUE / MOTS DE PASSE / AUDIT WINDOWS / CHIFFREMENT RÉVERSIBLE

Autant le fichier SAM (qui gère les comptes locaux sous Windows) a été largement disséqué, autant les techniques d'attaque sur le fichier NTDS.DIT (qui gère les comptes Active Directory) restent un mystère pour la majorité des auditeurs en sécurité, même 10 ans après la sortie de Windows 2000. Pourtant, des travaux notables sur le sujet existent « dans la nature ».

1 Historique

Active Directory est le nom donné par Microsoft à son annuaire d'authentification accessible au travers du protocole LDAP et apparu avec Windows 2000. En effet, le mécanisme de gestion des comptes utilisateurs développé pour les versions antérieures de Windows NT, à savoir le fichier SAM, ne passait pas à l'échelle dans les très gros réseaux. Les limitations de ce fichier sont en grande partie liées aux limitations de la base de registre **[1]**, sur laquelle il s'appuie.

Pour stocker les données de l'*Active Directory*, Microsoft a recyclé une technologie déjà éprouvée : le moteur du logiciel Exchange, appelé ESE **[2]** (*Extensible Storage Engine*) ou « Jet Blue » **[3]**. Cette technologie est bien plus performante que la base de registre pour la gestion d'accès concurrents à des millions d'objets de toute taille.

Le qualificatif d'*Extensible* est lié au fait que le schéma peut être étendu (ex. ajout de nouveaux types d'objets), mais pas réduit. L'installation de tout logiciel qui va étendre le schéma Active Directory (ex. Microsoft Exchange) va « tatouer » de manière définitive la forêt Windows.

Le stockage physique des données Active Directory s'effectue dans un répertoire choisi lors de la promotion du contrôleur de domaine (commande `DCPROMO`). Plusieurs fichiers de *checkpointing* sont présents dans ce répertoire, ESE étant un moteur transactionnel. Mais seul le fichier `NTDS.DIT` contient effectivement les données à jour.

Il est très facile d'obtenir une copie de ce fichier, même sur un système *live*. Ne vous jetez pas sur HoboCopy **[4]** et consorts, il suffit d'utiliser la commande native `NTBACKUP` par exemple. Il est beaucoup plus difficile de l'interpréter...

Car la migration vers Active Directory pose un problème aux administrateurs, auditeurs en sécurité et *hackers* de tout poil : si le format « base de registre » utilisé par le fichier SAM était parfaitement compris (particulièrement depuis cette thèse **[5]**), ce qui autorisait la création d'outils puissants pour la manipulation des mots de passe qui y sont contenus **[6]**, le format ESE reste quant à lui beaucoup plus obscur...

2 Travaux antérieurs

2.1 Tibor Biro

Les premiers travaux publics sur le sujet dont j'ai pu avoir connaissance sont ceux du site TBIRO **[7]**, à savoir :

- RADPass : outil permettant de supprimer le mot de passe d'un compte Active Directory ;
- SHEdit : outil permettant d'éditer l'attribut « SID History » attaché à un compte Active Directory.

Les codes sources ne sont malheureusement pas fournis. D'autre part, ces outils requièrent la présence de la bibliothèque `ESENT.DLL`. L'auteur s'est probablement basé sur la documentation Microsoft **[8]** **[9]** ainsi que le SDK fourni pour ESE, et tout aussi probablement sur du *reverse engineering* de cette bibliothèque, la documentation étant peu fournie dans les années de création des outils (vers 2002/2003).

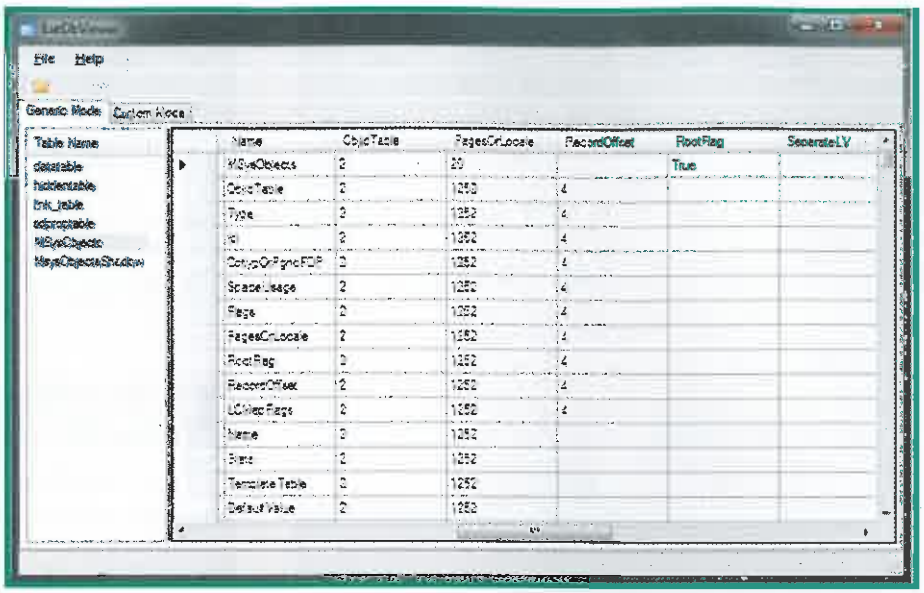
2.2 Incise sur le « SID History »

L'attaque sur l'attribut « SID History » est passée relativement inaperçue - elle a pourtant fait l'objet d'un correctif de sécurité (MS02-001) et d'une fonctionnalité supplémentaire appelée « SID Filtering » **[10]**.

L'attribut « SID History » a été conçu pour faciliter la migration des comptes utilisateurs d'un domaine Windows NT4 vers un domaine Windows 2000. En effet, le changement de domaine implique le changement de préfixe des SID **[11]** - ce qui a pour conséquence que toutes les listes de contrôle d'accès (ACL) existantes doivent être mises à jour.

Afin d'éviter cet écueil, il est possible d'ajouter des attributs à un compte utilisateur, qui décrivent les SID antérieurs dont il disposait.

Si un administrateur de domaine (ou un attaquant ayant un accès physique au fichier Active Directory d'un contrôleur de domaine quelconque) peut s'ajouter le SID d'un compte administrateur d'entreprise, cela pose un problème de sécurité évident... et heureusement corrigé.



EseDbViewer vs. Windows 2000 Active Directory sur un Windows Seven

2.3 Alexander Lahin

À peu près à la même date (2003), un auteur russe a publié sur le site http://www.void.ru/ une série d'articles très détaillés sur la manière dont sont stockés les mots de passe dans Active Directory, et particulièrement les mots de passe « en clair » lorsque l'option ***Store password using reversible encryption*** est activée pour un compte utilisateur... ou pour le domaine entier !

NOTE

Cette option est nécessaire si les utilisateurs sont amenés à s'authentifier par d'autres algorithmes que LM et NTLM - typiquement CRAM-MD5 ou CHAP. Le système doit alors conserver les mots de passe « en clair » [12], car les empreintes LM et NTLM ne lui sont d'aucune utilité dans ce cas.

Cet auteur se repose lui aussi sur les bibliothèques fournies par Microsoft (principalement `ESENT.DLL`) pour accéder aux données Active Directory et les exporter au format XML. Il ne rentre pas dans les détails d'implémentation du fichier `NTDS.DIT`. La lecture des mots de passe est donc possible, mais pas leur modification.

Malheureusement, tous les travaux de cet auteur ont disparu et ne sont accessibles que grâce à archive.org **[13]** - ce qui n'inclut pas les codes sources. À ce sujet, si un lecteur dispose d'une copie des fichiers attachés à ces articles, je suis preneur !

2.4 Digression sur les mots de passe en clair

Une présentation ultérieure lors de la conférence *HAR* 2009 **[14]** a définitivement réglé le problème des mots de passe « en clair ».

La réponse se trouve dans la fonction `RetrieveCleartextPassword()` de la bibliothèque `RASSFM.DLL`. Cette fonction utilise grosso modo un sel, une clé en dur, un secret de la LSA, et la fonction de chiffrement RC4 pour stocker les mots de passe de manière réversible dans l'attribut `userParameters` de chaque utilisateur. L'outil `RevDump` a été publié pour inverser cette opération.

Enfin... rien n'est jamais définitif dans le monde Windows, puisque les méthodes de stockage et de chiffrement ont évolué dans Windows 2008, et que l'outil susmentionné ne fonctionne plus!

2.5 EseDbViewer

L'outil open source **[15]** EseDbViewer **[16]** permet d'éditer les fichiers Windows Mail, Windows Desktop Search et Windows Live Messenger, qui font usage de la même technologie de stockage.

Cet outil n'est pas capable d'ouvrir un fichier Active Directory de type `NTDS.DIT`.

Ou du moins n'était pas capable, car cet outil repose lui aussi sur les bibliothèques fournies par Microsoft (à savoir `ESENT.DLL`). Or après quelques tests, il s'avère que la version de cette bibliothèque livrée avec Windows Vista et Windows Seven supporte également le format Active Directory, contrairement aux versions antérieures (ex. Windows XP).

La cause de ce comportement subtil n'a pas été investiguée : il est plus simple de réécrire le logiciel en partant des API Jet* **[17]** que de le déboguer.

Compte tenu des noms charmants assignés aux 300 000 colonnes de cette « base de données », il faut un certain temps (ou quelques recherches Google) pour retrouver où sont stockés les *hash* :

- ATTk589879 = Hash LM ;
- ATTk589914 = Hash NTLM ;
- ATTk589918 = Historique du hash NTLM ;
- ATTk589984 = Historique du hash LM.

Ces colonnes ne contiennent pas les hash « en clair », mais une forme chiffrée (à l'aide d'une clé protégée à son tour par la SYSKEY du DC) - sinon un simple « grep » dans le fichier `NTDS.DIT` aurait suffi pour extraire les hash :)

Rien d'insurmontable non plus, mais ceci est une autre histoire...

3 Travaux actuels

Tous les travaux antérieurs reposent sur les bibliothèques fournies par Microsoft, principalement `ESENT.DLL`. Aucun ne s'est attaqué au format interne du fichier `NTDS.DIT`.

Toutefois, plusieurs publications récentes pourraient changer la donne.

3.1 Documentation Microsoft

Dans le cadre de l'initiative « Open Specification Promise » **[18]**, Microsoft a publié un grand nombre de documents techniques auparavant internes. Les spécifications publiées couvrent essentiellement les protocoles de communication, ainsi que quelques formats de fichiers (comme Microsoft Office).

Le cas d'Active Directory est couvert par les documents suivants :

- [MS-ADTS] : *Active Directory Technical Specification* ;
- [MS-ADA1] : *Active Directory Schema Attributes A-L* ;
- [MS-ADA2] : *Active Directory Schema Attributes M* ;
- [MS-ADA3] : *Active Directory Schema Attributes N-Z* ;
- [MS-ADSC] : *Active Directory Schema Classes* ;
- [MS-ADLS] : *Active Directory Lightweight Directory Services Schema*.

Les formats de fichier et les détails d'implémentation ne font pas partie des documents disponibles actuellement.

3.2 libesedb

La vraie nouveauté est la publication il y a quelques mois de la bibliothèque open source `libesedb` **[19]** par un expert dans le domaine du *forensics*. Cette bibliothèque permet enfin d'entrevoir la possibilité d'une édition *offline* du fichier Active Directory de manière complètement indépendante des bibliothèques Microsoft.

Pour l'instant, l'ouverture d'un fichier Active Directory déclenche encore quelques bogues dans le code disponible, mais gageons que ces bogues sont en passe d'être corrigés (si ça n'est pas déjà le cas au moment où cet article sera disponible en kiosque).

Conclusion

L'édition offline d'un fichier Active Directory ne change pas radicalement la donne dans le domaine du *pentest* : les outils existant actuellement sont déjà parfaitement capables d'extraire les condensats des mots de passe depuis un système *live*, ce qui est le cas le plus courant (il est rare que le contrôleur de domaine soit éteint lors d'un pentest :)).

L'intérêt majeur des outils offline est leur fiabilité totale : il est pour ainsi dire impossible de crasher le système audité lorsqu'on travaille sur une copie de sauvegarde de données...

Il faut toutefois remarquer que l'état de l'art dans le domaine des outils publiquement disponibles a très peu évolué ces dernières années : le 64 bits est à peine supporté, personne ne s'est intéressé aux mots de passe stockés en clair dans Active Directory, ni aux autres secrets potentiellement intéressants (ex. DPAPI, BitLocker).

Le véritable apport de l'édition offline, c'est à mon sens la possibilité de modifier le contenu du fichier Active Directory au-delà des limites imposées par les API Microsoft. Ce qui conduit à des scénarios de compromission intéressants, réalisés depuis des contrôleurs de domaine isolés sur des sites distants et moins bien protégés physiquement, par exemple.

Microsoft ne s'y est pas trompé et propose avec Windows 2008 le rôle de *Read-Only Domain Controller* (RODC **[20]**), qui limite le nombre de secrets répliqués localement et empêche la propagation dans la forêt de toute modification apportée à la base Active Directory locale. Preuve qu'un vrai risque a été identifié. ■

■ REMERCIEMENTS

Aurélien B., pour sa connaissance insondable de Windows.

Fabrice D., pour sa bonne humeur inextinguible.

J.-B. pour ses idées et ses tuyaux.

RÉFÉRENCES

[1] http://support.microsoft.com/kb/256986

[2] http://fr.wikipedia.org/wiki/Extensible_Storage_Engine

[3] À ne pas confondre avec le moteur de Microsoft Access, appelé « Jet Red » :)

[4] http://alt.pluralsight.com/wiki/default.aspx/Craig/HoboCopy.html

[5] http://amnesia.gtisc.gatech.edu/~moyix/suzibandit.ltd.uk/MSc/

[6] http://pogostick.net/~pnh/ntpasswd/

[7] http://tbiro.com/

[8] http://msdn.microsoft.com/en-us/library/ms684493(VS.85).aspx

[9] http://msdn.microsoft.com/en-us/library/aa964813(VS.85).aspx

[10] http://technet.microsoft.com/fr-fr/library/cc772633(WS.10).aspx

[11] Le format des SID est détaillé sur le site Microsoft : http://msdn.microsoft.com/en-us/library/aa379597(v=VS.85).aspx

[12] Pour ceux qui n'ont pas le courage d'utiliser Google Translate, la réponse courte semble être RetrieveCleartextPassword() dans RASSFM.DLL :)

[13] http://web.archive.org/web/20051031110814/www.void.ru/content/1081, http://web.archive.org/web/20051031110929/http://www.void.ru/content/1090

[14] ... ou du moins compilé en bytecode .NET non obfusqué :)

[15] http://blog.teusink.net/2009/08/passwords-stored-using-reversible.html

[16] http://www.woany.co.uk/esedbviewer/

[17] http://www.microsoft.com/interop/osp/default.mspx

[18] Exemple ici : http://blogs.msdn.com/b/windowssdk/archive/2008/10/23/esent-extensible-storage-engine-api-in-the-windows-sdk.aspx

[19] http://sourceforge.net/projects/libesedb/

[20] http://technet.microsoft.com/en-us/librarycc732801%28WS.10%29.aspx