Mots de passe sous Windows NT

Pierrot Nicolas - Reis Zéférino

13 mai 2002

Table des matières

1 Introduction

Les mots de passe sont le coeur de la sécurité dans tous les réseaux, aussi bien locaux que distants. En effet, ce sont eux qui nous permettent de nous identifier, et de garder à l'abri les données sensibles.

Cependant, ils reposent principalement sur l'utilisateur, et s'ils sont mals choisis, leur "cassage" est simple, car de nombreux logiciels permettent de les trouver en un temps assez court.

Ce rapport présente donc les mécanismes de stockage des mots de passe sous Windows NT, leur encryptage, mais aussi, et malheureusement, les différentes et nombreuses attaques qui permettent de se faire passer pour un utilisateur donné, en suptilisant son passe.

2 Définition d'une empreinte

Une empreinte résulte du hachage d'un mot de passe par un algorithme non réversible.

Cette opération permet de sauvegarder dans le système une "image" du mot de passe et empêche quiconque d'accéder à celui-ci en clair.

Pour des raisons de sécurité, il parait par conséquent naturel de n'avoir, sur n'importe quel type de plateforme, aucun mot de passe stocké directement en clair sur le disque dur. Seules des empreintes sont donc gardées. Elles correspondent, comme sous les systèmes *nix, aux mots de passe cryptés par des fonctions de hachage. Sous Windows, ces fonctions sont MD4 et DES.

On peut considérer un mot de passe "hashed" comme un cryptage du mot de passe que personne ne peut décrire. Si une personne connaît le mot de passe crypté d'un autre compte, elle ne pourra pas s'en servir directement pour se connecter.

Le principal lieu de stockage de Windows pour les mots de passe cryptés se trouve dans la base de registres, ce qui les empêche d'être accessibles. Cependant, ces mots de passe peuvent aussi être obtenus dans d'autres endroits.

2.1 Cryptage avec Syskey

Il est intéressant d'utiliser la commande " SYSKEY ", afin d'augmenter la protection des mots de passe stockés localement. Par défaut, une version cryptée des mots de passe utilisateurs est stockée dans le Registre SAM auquel seuls les administrateurs globaux ont accès.

Cette commande permet de configurer le système pour que le mot de passe crypté de l'utilisateur soit à nouveau crypté avec un algorithme de 128-bits. Ceci est donc mis en oeuvre dans le seul but d'accroître la protection initiale.

A part un défaut éventuel dans l'algorithme, il a pour réputation d'être absolument indestructible. L'utilisation de ce dernier peut s'éffectuer dans l'un de ces deux cas :

- **Auto Boot** : Le système génère alors une clé de cryptage interne brouillée et finit par la conserver sur le système. Ceci permet d'effectuer des démarrages automatiques sans intervention de l'utilisateur.

Mais attention, si à l'origine la base SAM est peu sécurisée, alors l'utilisation de la clé l'est aussi.

- **Floppy Boot** : Le système crée une clé complexe, aléatoire et la copie sur une disquette qui doit être insérée pour lancer le système. L'intérêt est que la clé n'est pas stockée sur le système.

La taille du mot de passe détermine la puissance finale du cryptage face aux attaques brutales. Un mot de passe de 14 caractères alphanumériques crypté avec SYSKEY produit une clé dont la taille est d'environ 82-bits.

Toutefois, le cryptage DES 56-bits reste une valeur sur. Bien qu'il ait été cassè par une attaque de type " brute force ", cela a nécessité les ressources combinées de milliers d'ordinateurs sur Internet. Le cryptage 82-bits est à peu près 16 millions de fois plus complexe que le 56-bits.

2.2 Cryptage avec DES

DES (Data Encryption Standard) connu aussi sous le nom de DEA (ANSI) et DEA-1 (ISO) est un standard depuis 25 ans. DES est issu d'une longue réflexion des instances d'état et du privées afin de définir un ensemble de critères, pour obtenir un algorithme qui :

- assure un niveau de sécurité élevé
- contienne une spécification compléte et facile à comprendre
- ait une sécurité qui réside dans la clé et pas dans le secrêt de l'implémentation
- soit disponible pour tous les utilisateurs
- soit applicable dans différents domaines
- puisse être implanté dans du matériel électronique à moindres coûts
- soit performant
- puisse être valider
- doit être exportable

Beaucoups de polémiques autour de DES et les manières de casser ce code ont eu lieu. Le coût d'une machine étant capable de casser DES en 3.5 heures avait été estimé à 1 million de dollars en 1993. De nos jours, ce coût décroit

rapidement. De plus, il parait que la NSA possède une machine capable de casser DES dans un temps entre 3 et 11 minutes. Cette machine est de type " Cray Y-MP ".

2.3 Hachage par MD4 (Message Digest #4)

Une fonction de hachage peut être aussi appelée fonction de hachage à sens unique ou "one-way hash function" en anglais. Ce type de fonction est énormément utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par la fonction de cryptage. Naturellement, la caractéristique principale d'une fonction de hachage est de produire un haché des données, c'est-à-dire un condensé de ces données. Ce condensé de taille fixe a une valeur qui diffère suivant la fonction utilisée.

MD4 est une fonction de hachage à sens unique de 128 bits conçue par Ron Rivest, utilisant un ensemble simple de manipulations de bits sur des opérandes de 32 bits. Elle a été développée pour être une alternative beaucoup plus rapide à MD2. Cette derniere est décrite dans le RFC 1320.

Mais attention, il a été démontré qu'il est possible de trouver deux fichiers produisant la même empreinte numérique MD4 sans avoir à faire de recherche exhaustive. A vrai dire, il serait possible de trouver des collisions en moins d'une minute sur un PC typique de 1998, cette fonction doit par conséquent être considérée comme caduque et ne plus être employée pour un usage "cryptographique".

3 Localisation des empreintes

Ces empreintes sont stockées dans un fichier nommé SAM (Security Account Manager). Il correspond à une base de données dans laquelle sont répertoriés les mots de passe et noms d'utilisateur. De plus, au démarrage de Windows, ce fichier est aussi stocké dans la base de registre.

3.1 Différentes sauvegardes de la base SAM

Toutefois, il est à noter que la base SAM est sauvegardée dans différents endroits :

- Cette dernière se trouve dans le répertoire " /Win/system32/config/ ". Le fichier est cependant inaccessible, même pour l'administrateur système, et on ne peut donc ni le lire ni le copier.
- Elle est aussi stockée dans la base de registre à l'endroit suivant : " HKEY_LOCAL_MACHINE/SAM/SAM ". Mais cette clé est, comme le fichier de base SAM, illisible et non modifiable.
- Cependant, une base SAM ne comprenant que le mot de passe Administrateur se trouve dans le répertoire " /Win/repair/ ". Il correspond au mot de passe entré lors de l'installation de Windows. Cette base ne sera jamais modifiée, même si l'Administrateur venait à changer son password.
- Il est possible d'obtenir une copie compressée de la base SAM par l'intermédiaire de l'utilitaire de réparation de disque de Windows. Elle est placée dans le répertoire " /Win/repair/ ", sous le nom " SAM._ ". Il est donc intelligent de penser à effacer ce fichier le plus rapidement possible.

3.2 Contenu de la base SAM

Les empreintes sont coupées en deux parties qui sont chacune cryptée à l'aide de l'algorithme DES, ce qui permet à deux utilisateurs ayant le même mot de passe de ne pas avoir la même entrée dans la base SAM.

Les deux parties se décompose de la manière suivante :

- La première partie est chiffrée avec le RID de l'utilisateur qui est composé des 32 bits de poids faible du SID.
- La seconde partie est chiffrée avec une valeur calculée à partir du RID de l'utilisateur.

3.3 Récupération de la base SAM

Une manière simple de récupérer le contenu de la base SAM est de " booter " directement à partir d'un autre système d'exploitation (Unix par exemple). Par conséquent, les protections sur ces données deviennent alors fictives. L'utilisateur n'a plus qu' à monter la partition Windows, et à recopier le contenu de la base SAM.

4 Calcul des empreintes

En réalité, deux empreintes sont stockées. Une correspond au mot de passe qui est rentré au démarrage de Windows (NT Hash), et l'autre correspond aux autorisations de connexions distantes sur le réseau (LM Hash).

4.1 LM Hash

Ce mot de passe a une taille maximum de 14 caractères. Lors de la demande de connexion, le serveur envoie au client un challenge (8 octets de long) pour hasher son mot de passe. Quand ce challenge est reçu, le client effectue plusieurs opérations :

- Il commence par transformer le mot de passe en ASCII unicode. Ensuite, deux parties de 8 octets chacune sont crées. Dans la première, on retrouve les 7 premiers caractères du mot de passe, concaténé d'un octet. Dans la seconde, le même travail est effectué, avec les 7 derniers caractères du mot de passe. De plus, si le mot de passe fait moins de 7 caractères, la deuxième partie est alors toujours identique, et vaut, après hachage, " 0xAAD3B435B51404EE ".
- Après avoir hacher le mot de passe, le client rajoute, s'il le faut, des " 0 " en fin de chaîne. Il fait ensuite 3 paquets de 8 octets chacun (7 octets résultant du hachage + un octet de parité). Ensuite, ces trois parties servent à encrypter le challenge reçu par le serveur. Ces 3 nouvelles parties (cryptages du challenge) sont concaténées et envoyées au serveur.
- De son côté, le serveur réalise les mêmes opérations, c'est-à-dire le hachage et cryptage du mot de passe. Il compare ensuite les deux " versions ", et si la comparaison des deux chaînes cryptées est bonne, le mot de passe est accepté.

4.2 Exemples sur le LM Hash

Pour mieux comprendre, prenons deux exemples.

4.3 L'explication par l'exemple

Supposons que le challenge envoyé par le serveur est (en hexadécimal) : " 0x0001234567890123 ". Le client hache son mot de passe, et l'on suppose que le résultat vaut : " 0x823E1AA8A1E5665fABD7B435B51404EE ". A

cela, on concatène 5 " 0 " pour pouvoir ensuite faire 3 paquets de 7 octets, ce qui nous donne " 0x823E1AA8A1E566 0x5fABD7B435B5140 0x4EE00000 ".

A ces paquets de 7 octets, on rajoute un 8ème octet de parité. On obtient donc : 1_paquet_8_octets 1_paquet_8_octets 1_paquet_8_octets

Supposons que le cryptage du 1er paquet par le challenge
" 0x0001234567890123 " donne comme résultat
" 0xAAAAAAAAAAAAAAAA ", le 2ème
" 0xBBBBBBBBBBBBBBBB ", et enfin le 3ème
" 0xCCCCCCCCCCCCCCCC ".

Maintenant, le client n'a plus qu'à envoyer au serveur la chaîne concaténée, donc la chaîne :
" 0xAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBCCCCCCCC
CCCCCCCC " !

4.4 Cas d'un mot de passe de moins de 7 caracteres

La première chose à faire est de tester si le mot de passe à moins de 8 caractères. Pour cela, on prend seulement 7 octets du 3ème paquet (" 0x4EE00000 "), et on lui rajoute le 8ème octet de parité. Si, après cryptage par le challenge, on retrouve la chaîne " 0xCCCCCCCCCCCCCCCC ", alors nous sommes presque sur d'avoir un mot de passe de moins de 8 caractères.

4.5 NT Hash

En théorie, la taille maximale de ce mot de passe est de 128 caractères. Mais en pratique, seulement 14 caractères peuvent être utilisés. Le principe de ce hachage est simple : le mot de passe est converti en code Unicode, avec 2 octets par caractères (pour conserver les caractères étendus). A ce résultat, on applique la fonction de hachage MD4. La chaîne obtenue fait 16 octets, et est envoyée au serveur.

5 Les différentes attaques possibles

5.1 Pwdump et ses variantes

5.1.1 Petit descriptif de l'utilitaire

A l'heure actuelle, il existe déjà trois versions du logiciel libre, développé en GPL. Ce dernier a pour objectif majeur de permettre l'extraction des empreintes de la base SAM sous Windows.

Cet outil, à la fois pratique et commode, extrait par conséquent le résultat de la fonction de hachage des mots de passe (base SAM), à partir de la base de registre qui se tient dans
" HKEY_LOCAL_MACHINE/SECURITY/SAM/Domains/Account/Users ".

Ces informations sont stockées dans un fichier au format smbpasswd valide. Ceci devrait donc être une aubaine pour les administrateurs SAMBA, avec une portabilité assurée entre les différentes plateformes (Windows et Unix) grâce au format.

Une fois sauvegardé, le fichier répertoriant les entrées des mots de passe NT se présente de la manière suivante :
<user> :<id> :<lanman pw> :<NT pw> :comment :homedir :

<user> représente le nom de l'utilisateur sous Windows NT, <id> est l'identifiant qui caractérise la version de Windows (RID : Release ID), et <lanman pw> révèle le mot de passe crypté de l'utilisateur lanman. <NT pw> indique le mot de passe crypté de l'utilisateur Windows NT (md4). Il est important de noter que si l'utilisateur n'a pas de passe, alors il est remplacé par la chaîne " NO PASSWORD***** ". L'attribut 'commentaire' est la concaténation du nom complet de l'utilisateur sous Windows NT et du champ description dans le programme de gestion des utilisateurs de Windows. Le champ 'homedir' ne peut contenir le caractère ' :', car il est utilisé comme séparateur dans le fichier smbpasswd.

Néanmoins, il est significatif de dire qu'il existe des versions différentes de ce programme, suivant la version du Windows utilisé :

- Pwdump.exe pour un NT version USA
- Pwdumpf.exe pour un NT version Française

5.1.2 Comment l'utiliser ?

Il est préférable de dupliquer la base de données des comptes de la machine NT, et de créer un utilisateur UNIX, en l'ajoutant dans le fichier (/etc/passwd) avec le même numéro de compte Unix que celui de NT RID, ce qui fera une copie du fichier smbpasswd plus simple pour plus tard. Ce compte devra prohiber toute connexion via telnet sur la plateforme Unix (ceci est similaire à une suppression des droits sur le fichier log en local sur le serveur NT). Mais cette manipulation ne doit pas interdire d'utiliser SAMBA comme serveur. Le fichier crée smbpasswd doit être copié dans le fichier " *SAMBA/private/smbpasswd* ", où " SAMBA " désigne le nom du répertoire dans lequel il fut installé.

Si SAMBA a été installé et configuré avec un niveau de sécurité utilisateur et des mots de passe encryptés, alors on obtient ce qui suit : " security = user encrypted passwords = yes " dans le fichier smb.conf.

Les utilisateurs NT logués sur un domaine NT devraient par conséquent être capable d'avoir un accès transparent aux ressources.

L'utilitaire pwdump.exe prend comme argument un nom de machine, et récupère une base de données contenant les mots de passe de la machine spécifiée, si l'utilisateur a les droits nécessaires sur la machine distante. Ce programme doit donc être exécuté en tant qu'administrateur. Par défaut, il récupère les empreintes de la base de registres en local.

Il est important de signaler que ce logiciel ne décrypte pas les empreintes récupérées de la base de registres. Mais il a néanmoins subi quelques évolutions.

5.1.3 Quelles évolutions ont été apportées au programme original ?

Pwdump2 a vu son apparition avec quelques changements majeurs, qui sont les suivants :

- Il est désormais capable de dupliquer les mots de passe dupliqués à partir de " Active Directory ", ce qui était inconcevable par la version originale.
- Il est capable de déterminer automatiquement le pid de lsass, par conséquent il n'est plus nécessaire de le fournir sur la ligne de commande.

Cette application fonctionne avec ou sans avoir activer " SYSKEY " sur le système, mais simplement en local. Le fichier en sortie suit le même format, que son prédécesseur (par Jeremy Allison), et peut être utilisé comme entrée par le programme " l0phtcrack ", ou utilisé par SAMBA. Il est nécessaire d'avoir certains privilèges, acquis de base par l'administrateur.

Pour l'utiliser, il est demandé à l'utilisateur de placer ces deux fichiers " pwdump2.exe " et " samdump.dll " dans le répertoire contenant le système de fichiers NT. En exécutant le programme, le contenu de la base SAM devrait être écrit sur la console. Il suffit alors de rediriger le tout dans un fichier, afin d'avoir une sauvegarde.

Afin de pouvoir fonctionner convenable, cet utilitaire utilise une technique connue, appelée injection d'une dll. De manière générale, le processus " pwdump2.exe " force un autre processus (ici " lsass.exe ") à charger une dll (" samdump.dll "), et à exécuter le code à partir de la dll dans l'espace mémoire de l'autre processus, en l'occurrence " lsass.exe ".

Dans ce cas précis, la dll " samdump.dll " est chargée par l'utilitaire " lsass.exe ", lequel fera appel à la fonction appropriée pour lire les empreintes.

Ce programme a toutefois des limites : il ne duplique pas le nom complet de l'utilisateur juste le nom du compte. Il est par conséquent intéressant d'utiliser cette technique dans les contrôleurs de domaines, afin d'obtenir tous les utilisateurs du réseau en question.

Quant a lui, Pwdump3 fournit une protection accrue de l'information cryptée du mot de passe, par le cryptage des données avant de les envoyer sur le réseau. Il utilise l'algorithme " Diffie-Hellman ", conformément à des accords, pour générer une clé partagée qui n'est pas véhiculée à travers le réseau, mais utilisée par les applications de " Windows Crypto ". Contrairement à la version 2, ce dernier fonctionne à travers le réseau avec ou sans " SYSKEY " actif. De plus, il permet à l'administrateur de récupérer les mots de passe à partir d'une machine distante, évitant ainsi une exécution en local sur chaque machine du réseau.

5.2 SMB Capture

Samba est utilisé sur des réseaux Windows/*nix. Il permet soit de permettre aux utilisateurs *nix d'accéder à des fichiers/répertoires partagés sous

Windows, soit de gérer une base de données d'utilisateurs Windows. Dans tous les cas, le serveur envoie au client un challenge permettant de crypter le mot de passe. En récupérant les trames du protocole Samba, on peut, en utilisant diverses attaques, avoir les mots de passes des utilisateurs.

Voici des exemples de captures de trames Samba réalisées par " L0phtCrack ". Le format est le suivant :

DOMAIN/username :3 :challenge :hachage crypté LANMAM :hachage crypté NTLM.

Les captures suivantes montrent une machine NT se connectant à une autre machine NT. Les challenges envoyés par chacun sont différents, bien évidemment.

DOMAIN/user :3 :c21ee5e0c1a8ae89 :626cc3ec9f8f1849bbd6455414 77be48bf261b486 9c36e7a :f9dfdb9ee9d1705a4fd45a0ed5f2c62e0c7a 957860a59559

DOMAIN/user :3 :ce16b6d32eee2e29 :8f96e377f2b9670fa425c4e52a e4ae6ae3e23f693 d518719 :d9a3180ce6e30f8a12d46703847147b70066 dbaf9a5b654e

Les 2 captures suivantes montrent une machine NT se connectant à une autre machine Windows 98. Cette fois ci, le challenge est le même dans les deux cas (la station NT ouvrant la connexion).

DOMAIN/user :3 :8f2eceae79b55000 :43caa3ff5c793d04bbbe2332e89 18bf80735b0100 89dc573 :1c592e5dcf78cf658829d0cbe61c0e4c32b5 ed7a87f5097e

DOMAIN/user :3 :8f2eceae79b55000 :43caa3ff5c793d04bbbe2332e89 18bf80735b0100 89dc573 :1c592e5dcf78cf658829d0cbe61c0e4c32b5 ed7a87f5097e

La capture suivante montre une autre machine NT se connectant à une autre machine Windows 98. Cette fois ci, le challenge est le même dans les deux cas, mais est surtout identique à la précédente capture !

DOMAIN/user :3 :8f2eceae79b55000 :43caa3ff5c793d04bbbe2332e89
18bf80735b0100 89dc573 :1c592e5dcf78cf658829d0cbe61c0e4c32b5
ed7a87f5097e

Il est à noter que dans les 3 dernières captures, si le nom d'utilisateur et le challenge sont les mêmes, alors les paquets cryptés hachés sont identiques.

5.3 Attaques dictionnaires

Par défaut, les mots de passe de la plupart des logiciels sont cryptés et stockés dans un fichier. Afin d'obtenir un mot de passe, il suffit simplement de récupérer ce fichier et d'executer un logiciel de " dictionary cracking ". Ce procédé consiste à utiliser un logiciel qui s'occupera de tester un nombre limité de mots, dont le but majeur sera de trouver un mot de passe.

Chaque mot représente un mot d'usage courant, un prénom, un nom, ou une abréviation. La totalité des mots utilisés est regroupée dans un seul fichier : un dictionnaire. De là est venu le nom de cette attaque. Cependant, une phase de calcul doit être réalisée en amont de l'attaque, puisque chaque mot d'un dictionnaire est chiffré et sauvegardé dans une base de données. Par conséquent, certaines contraintes existent :

- Nécessite une phase de préparation longue.
- Utilisation de mémoire de masse importante.

En contre partie, la phase de crackage est très rapide puisqu'elle est limitée à des recherches dans une base.

Afin de cibler et donc d'accroître la recherche, il existe plusieurs dictionnaires différents, suivant l'utilisation :

- Mots courants de la langue natale de l'utilisateur.
- Prénoms et noms de même origine que l'utilisateur.
- Noms de personnages et d'acteurs.
- Vocabulaires propres à des livres, films, séries, jeux ...

De plus, toute personne est apte à pouvoir utiliser cette attaque. Pour ce faire, il est nécessaire de récupérer un logiciel permettant de faire cette attaque et le fichier de mots de passe. Il paraît donc évident que le mot de passe sera rapidement trouvé s'il est emprunté du langage courant.

Pour se prévenir contre une telle attaque, il y a quelques règles de base à respecter :

- N'utilisez pas de mot de passes signifiant quelquechose.
- Utilisez plutôt une combinaison de chiffres et de lettres.
- Changer vos mots de passe régulièrement...

D'autres techniques peuvent etre utilisées pour la protection par mots de passe. Ces dernières sont vivement recommandées. Un des programmes qui a déjà fait ses preuves est le fameux " Crack ". Il a été conçu par et pour les administrateurs systèmes. Il a pour but de permettre à ces derniers d'être aussi bien équipés que les 'craqueurs' en la matière.

Il est capable de deviner le mot de passe d'un utilisateur en se basant sur les combinaisons de lettres et de chiffres de son nom de session, de son nom, et de mots des dictionnaires que l'on veut bien lui fournir.

5.4 Attaques hybrides

L'attaque hybride se base à quelques nuances près comme l'attaque dictionnaire sur une restriction des mots du langage. En clair, elle se base sur une liste de mots, comme les dictionnaires de la vie courante.

Mais des modifications sont apportées à ces dictionnaires, pour cibler ou élargir la recherche. Ces dictionnaires particuliers sont en fait la base pour les attaques dites " hybrides ".

Ces dernières se présentes donc de la manière suivante :

- Zéro, une, toutes les lettres en majuscule.
- Mot renversé ou dupliqué.
- Suffixer et/ou préfixer un chiffre et/ou un caractère de ponctuation.
- Substitution de lettres par des chiffres.

5.5 Attaques brute force

5.5.1 Principe

Cette méthode consiste à tester de façon exhaustive toutes les combinaisons possibles de caractères (alphanumériques + symboles), de manière à trouver au moins un mot de passe valide.

Une attaque brute force est rien de plus que la détection d'une paire identifiant utilisateur/mot de passe sur un service qui cherche à authentifier l'utilisateur avant de lui accorder l'accès.

Cette attaque est basée sur le fait que n'importe quel mot de passe est crackable. Ce n'est qu'une histoire de temps, tout en sachant que la puissance des machines double tous les deux ans. Enfin, les crackers n'hésitent pas à fabriquer des cartes électroniques de cracking, ce qui améliore en conséquence la rapidité de la machine, et par conséquent les chances de trouver un mot de passe valide.

En général, ce procédé est emprunté lorsque la méthode du " dictionnary cracking " a échoué.

5.5.2 Parade

La meilleure défense contre la détection brute force est l'utilisation de mots de passe puissants et difficiles à décrypter. Un mécanisme de mots de passe à utilisation unique est le plus souhaitable. Par conséquent, il parait nécessaire de mettre en place de bonnes procédures de gestion de mots de passe. En voici, les plus intéressantes :

- Vérifier que tous les utilisateurs ont un mot de passe valide.
- Forcer le renouvellment de mot de passe tous les trentes jours pour les comptes à droits protégés et tous les soixante jours pour les utilisateurs courants.
- La longueur minimale d'un mot de passe doit être d'au moins six caractères alphanumériques, huit de préférence.
- Mettre en place des outils de composistion de mots de passe qui interdisent à l'utilisateur de choisir un mot de passe faible.
- Utiliser un mot de passe différent sur tous les systèmes auxquels une connection peut être établie.

5.6 Attaque Man in the middle

Le principe de cette attaque est de faire croire au client qui veut se connecter sur un serveur que le pirate est ce serveur. Le pirate écoute sur le réseau les requêtes ARP qui sont broadcastées. Quand il reçoit une requête, une réponse avec sa propre adresse Ethernet est envoyée au client pour lui faire croire que le serveur est le pirate. Bernée, la victime enverra dorénavant ses paquets à la machine du pirate. Il n'a plus qu'à re-router les paquets entre les deux machines pour qu'elles ne voient pas la différence.

Maintenant que le pirate s'est fait passer pour le serveur, il reçoit tous les paquets transitant entre les deux machines. On se retrouve donc proche du cas de SMB Capture, où il ne reste plus qu'à tenter de trouver le mot de passe, par " brute force " par exemple.

5.7 Attaque Passing the Hash

Le hachage des mots de passe est stocké dans la base SAM. Quand un utilisateur veut se connecter, il entre son passe, qui est haché, puis crypté avec le challenge envoyé par le serveur. Le tout est envoyé au serveur, qui vérifie en comparant les haches.

Légende :

- H est une fonction de hachage MD4.
- E et D sont des fonctions de cryptage/décryptage DES.
- P est le mot de passe en clair.
- S est le mot de passe haché (S = H(P)), stocké dans la base SAM.
- N est le challenge.
- A est le client, B le serveur.

Résumons la demande de connexion :

- A→B : demande de connexion
- B→A : envoie de N
- A→B : E(N, H(P))

Le serveur peut vérifier que S = D(N, E(N, H(P))) ou que
E(N, S) = E(N, H(P)).

Maintenant, avec " passing the hash " :

- A→B : demande de connexion
- B→A : envoie de N
- A→B : E(N, S)

Bien sûr on a S = D(N, E(N, S)), et la connexion est ouverte. En fait, le problème est de passer l'étape H(P).

SMBproxy a besoin de connaître l'adresse IP du serveur, ainsi qu'un fichier de mot de passe au format PWDump3.

Si quelqu'un se logue sur le Proxy, celui-ci renverra la requête au serveur, en utilisant le login donné dans la demande de connexion, mais le Proxy va changer le hache NTLM par celui se trouvant dans le fichier de passes.

5.8 Keyloggers

Les keyloggers sont par définition des enregistreurs de touches. Par extension, ils permettent d'enregistrer les touches utilisées par l'utilisateur sur son clavier et tous les évènements déclenchés.

Dangereux, ces derniers ne sont tout de meme pas répertoriés parmis les virus, vers, ou chevaux de Troie, puisqu'ils n'ont pas pour habitude d'altérer quoi que se soit dans la machine cible. Ils ont pour simple objectif de permettre l'enregistrement d'informations.

Ces petits logiciels, invisibles et faciles d'emploi, sont une manière détournée de récupérer des mots de passe. Installés sur une machine, ils permettent de sauvegarder dans un fichier texte tout ce qui est tapé sur un clavier, et peuvent même l'envoyer par email. Il est très facile d'obtenir les informations recherchées dans ces conditions !

Toutefois, meme face à la multitude de keyloggers existants, le mode opératoire de ces derniers ne diffère point. Ils peuvent etre installés directe-

ment en local par le pirate ou via une connexion internet par un cheval de Troie.

Il est coutûme que les keyloggers soient lancés automatiquement au démarrage de la machine hôte, afin de filtrer tout ce qui est réalisé. De plus, une multitude d'options peuvent etre configurées par le pirate, comme :

- La date de sauvegarde.
- Les applications ouvertes à cette periode.
- Le choix de cryptage des données.
- L'auto-desctruction du keylogger, le rendant ainsi invisible.
- La date d'execution de ce dernier.

Il est important de noter qu'un mot de passe est réclamé, ce qui confère au pirate la certitude que seul lui pourra lire (voire décrypter) le fichier obtenu. L'utilisateur se verrait donc dans l'incapacité de déchiffrer le contenu de celui-ci.

Néanmoins, des programmes comme " Procdump " permettent de signaler à l'utilisateur la présence de keyloggers à l'utilisateur.

5.9 NT scheduler

Une autre technique pour obtenir facilement la base SAM des mots de passe est de passer par le Scheduler.

En effet, on sait que la base n'est pas accessible, même pour l'administrateur système. Mais, il est possible pour l'utilisateur SYSTEM d'avoir accès à la base SAM dans la base de registres.

Pour cela, il suffit de programmer le Scheduler pour qu'il lance à une heure donnée l'éditeur de base de registres "Regedit.exe". Ainsi, la base de registres sera ouverte par l'utilisateur SYSTEM, et il ne reste plus qu'à aller copier la base SAM !

6 Exploits

6.1 Redirection de ports SMB

Cet exploit permet à un utilisateur sans privilège de rediriger les services SMB sur n'importe quel autre serveur SMB. Il permet de rediriger l'accès aux fichiers, aux imprimantes et à l'authentification réseau.
http ://archives.indenial.com/hypermail/ntbugtraq/1998/February1998/0000.html

6.2 NetBios

Cet article explique de A à Z comment trouver des partages NetBios, comment les utiliser et comment les contourner.
http ://packetstormsecurity.nl/NT/docs/NTExploits.txt

6.3 Cassage des mots de passe NT

Incontournable, www.phrack.org est LA référence sur les exploits. En voici un qui permet de récupérer les mots de passe de Windows NT !
http ://www.phrack.org/phrack/50/P50-08

7 Conclusion

Dans tous les systèmes, se sont les mots de passe qui permettent de nous identifier, et donc de mettre à l'abri des regards indiscrets les données sensibles.

Cependant, mêmes cryptés, ces mots de passe peuvent être récupérés par des personnes malveillantes. Et ce n'est pas leur hachage qui les protègent !

Le fait que la base SAM soit protégée en lecture montre donc que Windows ne s'est pas donné les moyens d'élaborer une polique sérieuse de mots de passe, le nombre de techniques permettant de les récupérer montre à quel point ce système est vulnérable à ce sujet.

Il est même stupéfiant de voir sur Internet tant de post relatant des failles de Windows, et le plus souvent, Microsoft ne veut pas écouter les alertes que les "crackers" leur font part...

8 Bibliographie

LM et NT hash :
http ://www.insecure.org/sploits/l0phtcrack.lanman.problems.html
http ://www.insecure.org/sploits/WinNT.passwordhashes.deobfuscation.html

Le logiciel Pwdump :
http ://www.webspan.net/~tas/pwdump2/
http ://www.polivec.com/pwdump3.html

Sites sur la scurité :
http ://www.securite.org/db/securite/systemes/windows
http ://www.insecure.org
http ://www.securite.org
http ://www.securityfocus.com
http ://www.securiteam.com/
http ://www.ussrback.com/