Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S.G.D.S.N
Agence nationale de la sécurité des systèmes d'information
CERT-FR

Paris, le 25 juillet 2014
N° CERTFR-2014-ACT-030

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2014-ACT-030

1 - Changement illégitime de mot de passe en environnement Active Directory par attaque de type « Pass-The-Key »

Introduction

Récemment, la société Aorato a publié un article [1] décrivant une attaque consistant à changer le mot de passe d'un utilisateur sans en connaître l'actuel. L'attaque décrite repose sur l'utilisation du protocole Kerberos et la compromission de l'empreinte NTLM d'un utilisateur. L'article d'Aorato insiste sur l'utilisation des empreintes au format NTLM (utilisées comme clé de chiffrement RC4 des messages Kerberos). Cependant, le protocole de changement de mot de passe via Kerberos ne nécessite que la présentation d'un ticket valide, peu importe la suite de chiffrement utilisée. Cet article se propose de préciser la vulnérabilité et d'apporter des recommandations adaptées à l'attaque décrite.

Description

Il existe actuellement deux protocoles majeurs d'authentification sous Windows :

- NTLM, protocole historique, dont Microsoft déconseille désormais l'utilisation [2] ;
- Kerberos, disponible depuis Windows 2000, pour lequel plusieurs algorithmes de chiffrements sont disponibles [3] : DES, RC4 (utilisant l'empreinte NTLM pour des raisons historiques de migration) et AES.

De même, il existe actuellement deux types d'opérations permettant de modifier un mot de passe d'un utilisateur dans un Active Directory :

- « *Set password* », qui permet de réinitialiser un mot de passe. Cette opération permet de spécifier, pour un utilisateur donné, un nouveau mot de passe arbitrairement sans nécessairement connaître l'actuel. Son utilisation nécessite des droits particuliers et n'est accordée, par défaut, qu'à un ensemble restreint de groupes d'administration au sein d'un Active Directory. Nous ne l'aborderons donc pas dans cet article ;
- « *Change password* », qui correspond au changement, par l'utilisateur, de son mot de passe. Par défaut, les interfaces de Windows nécessitent toujours de saisir le mot de passe actuel ainsi que le nouveau.

Avec le protocole NTLM, toutes les opérations permettant de changer le mot de passe d'un utilisateur nécessitent effectivement de connaître le mot de passe actuel [4]. Or, avec Kerberos, ce prérequis n'est pas nécessaire. Les premières spécifications de Kerberos spécifiaient que seul le nouveau mot de passe devait être renseigné dans le message Kerberos utilisé lors d'un changement de mot de passe [5]. Ceci a par la suite été repris par Microsoft dans son implémentation [6], à des fins de compatibilité.

Cependant, pour pouvoir changer son mot de passe, un utilisateur doit s'authentifier en obtenant un ticket pour le service de changement de mot de passe (`kadmin/changepw`). Pour obtenir un tel ticket, il doit disposer d'au moins un type de clé : DES (désactivé par défaut depuis Windows 7 et 2008R2), RC4 (c'est-à-dire

l'empreinte NTLM) ou AES. Ainsi, un attaquant ayant compromis une des clés d'un utilisateur (DES si toujours utilisé, RC4/NTLM ou AES) se voit en mesure :

- de s'authentifier à distance avec l'identité de l'utilisateur compromis (attaque de type « Pass-The-Key ») afin de demander un ticket pour le service `kadmin/changepw` ;
- d'utiliser le ticket récupéré pour demander, sur le service kadmin d'un contrôleur de domaine (tcp ou udp port 464), le changement du mot de passe de l'utilisateur en spécifiant uniquement le nouveau mot de passe.

Dans la pratique, ces clés sont présentes et récupérées dans la mémoire du processus LSASS lorsqu'un utilisateur a ouvert une session sur un système Windows. Ainsi, comme le mentionne Aorato, la compromission d'une empreinte NTLM permet de changer le mot de passe d'un utilisateur sans connaître le mot de passe actuel. Mais cette opération est également possible avec une clé Kerberos au format DES et AES.

Impact

Le changement du mot de passe d'un utilisateur ne présente a priori pas d'intérêt direct pour un attaquant souhaitant s'authentifier à distance sur un système, puisque les attaques de type « Pass-The-Hash » avec NTLM ou « Pass-The-Key » avec Kerberos le lui permettent déjà. La seule exception concerne le protocole RDP, utilisé en particulier pour le service de bureau à distance, qui demande, quant à lui, le mot de passe en clair de l'utilisateur. Un changement de mot de passe avec l'attaque présentée ci-dessus permettrait donc à l'attaquant de s'authentifier avec le protocole RDP.

Conclusion et recommandations

La vulnérabilité présentée par Aorato se concentre uniquement sur l'utilisation de l'empreinte NTLM comme clé de chiffrement RC4 dans les tickets Kerberos. Dans ce cadre précis uniquement, cette utilisation peut être empêchée en ajoutant les utilisateurs à protéger au groupe de sécurité « Protected Users » [7] disponible depuis Windows Server 2012 R2, dont les membres ne peuvent pas utiliser les suites de chiffrement DES et RC4.

Cependant, étant donné que la vulnérabilité n'est pas liée uniquement à l'utilisation des empreintes NTLM, mais au fonctionnement intrinsèque du protocole Kerberos, cette recommandation s'avère insuffisante en pratique. En effet, le mécanisme de changement de mot de passe reste accessible à un attaquant ayant compromis la clé AES d'un utilisateur. Des opérations de durcissement des systèmes Windows doivent donc être mises en œuvre pour éviter la compromission de ces clés (utiliser des comptes utilisateurs n'étant pas administrateurs locaux de leur poste de travail, vérifier la bonne application du correctif de sécurité "KB2871997" [8] [9], mettre en place des « protected process », etc.)

Enfin, une opération de changement de mot de passe génère l'évènement « Microsoft-Windows-Security-Auditing/4723 » (« Une tentative de modification de mot de passe d'un compte a été effectuée. »). Cependant, cette journalisation n'est effectuée que si la sous-catégorie de journalisation « Auditer la gestion des comptes d'utilisateur » de la catégorie « Gestion du compte » est activée.

Documentation

[1] Active Directory Vulnerability Disclosure: Weak encryption enables attacker to change a victim's password without being logged :
http://www.aorato.com/blog/active-directory-vulnerability-disclosure-weak-encryption-enables-attacker-change-victims-password-without-logged/

[2] NT LAN Manager (NTLM) Authentication Protocol : 5.1 Security Considerations for Implementers :
http://msdn.microsoft.com/en-us/library/cc236715.aspx

[3] Kerberos Protocol Extensions : 3.1.5.2 Encryption Types :
http://msdn.microsoft.com/en-us/library/cc233952.aspx

[4] Security Account Manager (SAM) Remote Protocol (Client-to-Server) : 3.1.5.10 Change Password Pattern :
http://msdn.microsoft.com/en-us/library/cc245705.aspx

[5] Kerberos Change Password Protocol :
http://tools.ietf.org/html/draft-ietf-cat-kerb-chg-password-02

[6] Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols :
http://tools.ietf.org/html/rfc3244.html

[7] Protected Users Security Group :
http://technet.microsoft.com/en-us/library/dn466518.aspx

[8] An Overview of KB2871997 :
http://blogs.technet.com/b/srd/archive/2014/06/05/an-overview-of-kb2871997.aspx
[9] Amélioration de la protection des secrets d'authentification sur les systèmes Windows :
http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-020/index.html

2 - Analyse rapide de fichiers suspects (première partie)

Cet article a pour objectif de présenter une méthodologie d'analyse rapide permettant d'établir un premier diagnostic concernant un fichier suspect. Le but n'est pas d'effectuer une analyse exhaustive, mais plutôt de lever un doute quant au caractère malveillant d'un fichier, et d'identifier les premières actions à mener consécutives à l'ouverture de ce fichier.

L'article ne s'attarde pas sur le vecteur de compromission, on suppose qu'il s'agit ici d'un fichier ouvert manuellement par la victime (exemple : ouverture d'une pièce jointe d'un message électronique). Sous cette hypothèse, l'article se limite aux trois formats de fichiers les plus couramment utilisés par les attaquants, à savoir le format exécutable PE, les documents Office et les documents PDF. Cette première partie va se concentrer sur les fichiers de format PE. La deuxième partie s'attardera sur les formats PDF et Office.

Pour tous les types de fichiers, on peut tout d'abord s'appuyer sur les logiciels antivirus dans leur version « scan à la demande ». Dans le cas où le résultat est négatif, il n'est cependant pas possible de conclure quant au caractère malveillant du fichier. En revanche, les faux-positifs sont moins fréquents. Ainsi un résultat positif d'un ou plusieurs antivirus permettra de confirmer le caractère malveillant d'un fichier suspect.

Des plateformes en ligne permettant une analyse automatisée par de nombreux moteurs antivirus existent, la plus connue étant Virus Total [1]. Ces plateformes ont l'avantage d'affranchir l'utilisateur de l'étape d'installation des antivirus et de manipulation d'une grande quantité de moteurs. Néanmoins, elles ont l'inconvénient de mettre les fichiers analysés à disposition des éditeurs antivirus et d'entreprises clientes de leur service. La confidentialité des données transmises est donc un élément à prendre en compte lors de l'analyse de ce type de document. Par ailleurs, chaque rapport d'analyse de fichier est disponible publiquement et permet sans restriction de rechercher un ancien résultat à partir du condensat d'un fichier. Une victime peut ainsi en profiter pour accéder à un résultat d'analyse d'un fichier préalablement analysé, sans avoir à le transmettre à la plateforme. De la même manière, un attaquant peut procéder à une veille sur ces services et vérifier régulièrement si l'un de ses fichiers malveillants a été soumis par un tiers, ce qui indiquerait que la compromission a été identifiée d'une manière ou d'une autre. Le CERT-FR a déjà constaté ce type de veille dans le cadre d'incident de sécurité ciblé.

L'utilisation de ce type de service doit donc être mûrement réfléchie.

Fichiers PE (Portable Executable)

Ce format binaire est le standard utilisé par Microsoft pour de nombreux fichiers exécutables, les plus courants étant les fichiers d'extension EXE, DLL et SYS. D'autres extensions moins communes telles que les SCR (écrans de veille) ou CPL (éléments du panneau de configuration) respectent également ce standard. Les fichiers contiennent directement du code machine référencé par un pointeur nommé « Original Entry Point » ou OEP qui sera exécuté au lancement du programme. Parfois, ce code ne sera pas directement du code machine mais du code interprétable par une machine virtuelle, comme dans le cas du .NET. Les risques liés à ces fichiers ont été évoqués dans un précédent article [2].

Analyse statique

Afin d'analyser un binaire au format PE, il est plus commode d'utiliser un « parseur » de PE tel que « CFF Explorer » [3]. Ce type d'outil s'appuie sur les spécifications du format PE pour parcourir un fichier passé en entrée et afficher les différents éléments du format tels que la table des imports, la date de compilation, etc. Le code présent dans le binaire n'est exécuté à aucun moment, d'où le terme d'analyse statique.

Les éléments suivants sont intéressants dans le cadre d'une analyse rapide de fichier et peuvent suffire à établir le caractère malveillant d'un fichier :

- absence quasi complète de chaînes de caractères, montrant l'utilisation d'un « packer » ;
- section ayant une taille sur le disque de zéro octets, car cela indique une section qui ne contient aucune donnée utile initialement et qui sera probablement utilisée au moment de l'exécution pour stocker des données déchiffrées ou décompressées par le code malveillant ;

- date de compilation incongrue, ce qui prouve une modification manuelle par un attaquant souhaitant cacher cette information ;
- section ou ressource ayant une entropie très élevée, ce qui peut indiquer la présence de code ou de données chiffrés qui seront déchiffrés à l'exécution ;
- point d'entrée référençant une adresse n'étant pas située dans une section standard telle que .text ou .code, montrant ainsi l'utilisation d'un « packer » ;
- entrées TLS, car le code présent dans ces entrées est exécuté avant le point d'entrée et cette fonctionnalité est rarement utilisée par les programmes légitimes ;
- présence de fonctions suspectes dans la table des imports n'ayant pas de lien avec les fonctionnalités supposées de l'exécutable analysé, telles que des fonctions de communication réseau pour un logiciel d'utilisation purement locale et sans mécanisme de mise à jour ;
- présence de noms de fonctions incohérents ou aléatoires dans la table des exports dans le cas de l'analyse d'une bibliothèque DLL ;
- présence de sections ayant un nom inhabituel. Les noms courants sont `.text`, `.code`, `.rsrc`, `.data`, `.rdata`, `.idata`, `.edata`, `.pdata`, `.reloc`.
- incohérence entre les informations présentes dans le binaire telles que l'éditeur, le nom du produit, la version et les informations réelles, ou incohérence avec la signature Authenticode (binaire supposé de Microsoft signé par un éditeur tiers).

Plusieurs outils d'analyse de PE recherchent des caractéristiques similaires à la liste ci-dessus et affichent toutes les anomalies rencontrées. Les scripts python exescan.py [4] et peframe.py [5] fournissent des résultats pertinents et ont l'avantage de permettre un traitement automatisé de plusieurs fichiers PE. Ils détectent également l'utilisation d'un « packer » à partir d'une base de données de signatures connues, sachant que peu de logiciels légitimes utilisent ce type d'outils (Skype en était un exemple célèbre).

Analyse dynamique

Une autre approche d'analyse consiste à lancer l'exécutable dans un environnement cloisonné pour observer les différentes actions réalisées par celui-ci. Cet environnement est généralement une machine virtuelle pour éviter de compromettre le poste hôte dans le cas d'un fichier malveillant.

Plusieurs outils permettent d'enregistrer les événements consécutifs à l'ouverture d'un fichier exécutable, tels que Process Monitor [6]. L'avantage de cet outil est la possibilité de filtrer des opérations particulières, par exemple uniquement les écritures dans la base de registre ou dans un fichier, et ainsi identifier plus facilement des actions malveillantes. A titre d'exemple, un exécutable qui ajoute une entrée dans la clé de registre `HKEY_LOCAL_MACHINE` indique une volonté de se lancer au démarrage de la machine, ce qui peut être une action illégitime.

Le trafic réseau généré pendant l'exécution du programme peut quant à lui être analysé par un outil tel que Wireshark [7], de préférence depuis la machine hôte, afin d'identifier d'éventuelles connexions vers des domaines suspects ou malveillants. De nombreuses solutions telles que Cuckoo Sandbox [8] réalisent toutes ces tâches automatiquement puis produisent un rapport d'exécution, affranchissant l'utilisateur d'une exécution manuelle. En revanche, il convient tout de même d'installer et de configurer les différents outils et paramètres proposés par ces solutions. Sinon, il existe des plateformes entièrement en ligne dans lesquelles l'utilisateur peut directement envoyer le fichier suspect et récupérer le rapport d'analyse quelques minutes plus tard.

Les réserves quant à la confidentialité des données évoquées dans le cas des plateformes d'analyse antivirale s'appliquent aussi pour ces solutions. Il existe de nombreux services de ce type, mais on peut citer Anubis [9], Malwr [10] (basé sur Cuckoo Sandbox) ou encore Joe Sandbox [11].

Les limites de l'analyse dynamique sont nombreuses, et il faut les prendre en compte lors de l'interprétation des résultats :

- un exécutable malveillant peut détecter l'environnement cloisonné de façon plus ou moins sophistiquée et modifier son comportement si un tel environnement est identifié, par exemple en se terminant ou en n'exécutant aucune action malveillante ;
- un exécutable malveillant peut détecter les solutions de traçage évoquées précédemment et modifier son comportement en conséquence ;
- un fichier malveillant peut attendre plusieurs minutes avant d'exécuter sa charge malveillante dépassant ainsi la limitation de temps d'exécution des plateformes d'analyse automatique ;
- un exécutable malveillant peut attendre une action utilisateur avant d'exécuter sa charge malveillante, telle que des mouvements de souris ou un clic sur un bouton, ce que n'émulent pas la plupart des plateformes d'analyse automatique.

Documentation

[1] Virus Total :
http://www.virustotal.com
[2] Risques liés au format « Portable Executable » (PE) :
http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-022/
[3] CFF Explorer :
http://www.ntcore.com/exsuite.php
[4] exescan.py :
http://securityxploded.com/exe-scan.php
[5] peframe.py :
https://github.com/guelfoweb/peframe
[6] Process Monitor :
http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx
[7] Wireshark :
http://www.wireshark.org
[8] Cuckoo sandbox :
http://www.cuckoosandbox.org
[9] Anubis :
https://anubis.iseclab.org
[10] Malwr :
https://malwr.com
[11] Joe Sandbox :
http://www.joesecurity.org

3 - Rappel des avis émis

Dans la période du 18 au 24 juillet 2014, le CERT-FR a émis les publications suivantes :

- CERTFR-2014-AVI-329 : Multiples vulnérabilités dans F5 BigIP Analytics
- CERTFR-2014-AVI-330 : Multiples vulnérabilités dans phpMyAdmin
- CERTFR-2014-AVI-331 : Multiples vulnérabilités dans les produits Mozilla

Gestion détaillée du document

25 juillet 2014 version initiale.

Conditions d'utilisation de ce document :	`http://cert.ssi.gouv.fr/cert-fr/apropos.html`
Dernière version de ce document :	`http://cert.ssi.gouv.fr/site/CERTFR-2014-ACT-030`