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 02 novembre 2015
N° CERTFR-2015-ACT-044

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2015-ACT-044

1 - Service de compatibilité des programmes

Présentation du service de compatibilité des programmes

Depuis Windows 2000, Microsoft a déployé le service de compatibilité des programmes afin de permettre à des applications compilées pour une version antérieure de Windows de fonctionner sur un système d'exploitation récent, sans recompilation de celles-ci. Ainsi, un éditeur de logiciel peut utiliser l'outil Microsoft Application Compatibility Toolkit pour spécifier les modifications qui devront être apportées à l'exécutable lors de son lancement. Ces modifications peuvent être de plusieurs types, comme par exemple la réécriture de certaines zones mémoires contenant du code par un code corrigé ou encore la modification à la volée d'arguments passés à un binaire. Le lanceur d'applications de Windows doit également savoir dans quel cas il doit appliquer ces modifications. Pour cela, l'éditeur peut spécifier une ou plusieurs conditions, telles qu'un nom ou une taille de fichier, un condensé de fichier exécutable ou encore des métadonnées spécifiques (éditeur, version...). L'ensemble de ces conditions et des modifications associées est stocké dans un fichier binaire au format SDB (Shim DataBase), qui n'est pas documenté par Microsoft.

Un système Windows par défaut dispose de plusieurs fichiers SDB qui contiennent des milliers de modifications génériques. Un éditeur peut les réutiliser pour sa propre application. Ces fichiers sont stockés dans le répertoire C:\Windows\AppPatch\et C:\Windows\AppPatch\AppPatch64. Le répertoire "Custom" contient les fichiers SDB réalisés par les éditeurs tiers. Certains éditeurs stockent directement leurs fichiers SDB dans le répertoire C:\Program Files. Pour activer un fichier SDB, il faut qu'il soit répertorié dans la base de registre. Les clés de registre suivantes contiennent la liste des fichiers SDB tiers qui seront parcourus par le lanceur d'application de Windows :

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\InstalledSDB
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Custom

L'outil *sdbinst.exe* de Microsoft permet notamment de manipuler ces clefs de registre.

Depuis quelques années, Microsoft fait usage de cette technologie à d'autres fins que la compatibilité, par exemple pour appliquer des correctifs temporaires (Fix-It) ou pour complexifier les exploitations de vulnérabilités via son outil EMET.

Détournement du service de compatibilité des programmes à des fins malveillantes

Une personne malveillante peut détourner ce service en référençant son propre fichier SDB, directement dans la base de registre. Pour ce faire, il doit toutefois disposer des privilèges administrateur.

Des cas publics font notamment état de l'utilisation de fichiers SDB pour assurer la persistance d'un fichier malveillant au redémarrage de la machine au travers d'une instruction forçant le chargement d'une DLL lors du lancement d'un exécutable donné, ou encore via l'écrasement du code d'une DLL légitime de Windows pour détourner le flux d'exécution vers du code malveillant.

Par ailleurs, sans l'installation du KB3045645, l'outil *sdbinst.exe* peut être utilisé pour contourner l'UAC.

Investigations

Le cache du service de compatibilité des programmes fait partie des artefacts couramment utilisés lors d'une analyse forensics pour déterminer les programmes lancés par l'attaquant.

Afin d'identifier une compromission utilisant des fichiers SDB a posteriori, il convient de vérifier les clés de registres listées précédemment. Il est également pertinent de rechercher les exécutions de *sdbinst.exe*.

Idéalement, il est souhaitable de vérifier que les fichiers SDB existants sont légitimes. Toutefois, la démarche est complexe dans la mesure où les fichiers SDB par défaut font l'objet de plusieurs milliers de modifications régulières légitimes. En outre, ils ne sont pas signés.

Quelques outils existent pour convertir les fichiers SDB dans un format XML plus compréhensible. Cependant, ils ont pour la plupart été réalisés par rétro-ingénierie et il n'est donc pas garanti qu'ils couvrent tous les cas de figure possibles :

- sdb2xml
- sdb-explorer
- python-sdb

Documentation

- Déplacement latéral au sein d'un réseau Microsoft Windows (Première partie)
  http://www.cert.ssi.gouv.fr/site/CERTFR-2014-ACT-037/CERTFR-2014-ACT-037.html
- COMMENT FAIRE : Activer la technologie Mode compatible pour les applications sous Windows 2000 Service Pack 2
  https://support.microsoft.com/fr-fr/kb/279792
- Kit de déploiement et d'évaluation Windows (Windows ADK) pour la Mise à jour Windows 8.1
  https://www.microsoft.com/fr-fr/download/details.aspx?id=39982
- Correctif Microsoft KB3045645 :
  https://support.microsoft.com/en-us/kb/3045645
- Analyzing Gootkit's persistence mechanism (new ASEP inside!)
  http://blog.cert.societegenerale.com/2015/04/analyzing-gootkits-persistence-mechanism.html
- The Real Shim Shady
  http://files.brucon.org/2015/Tomczak_and_Ballenthin_Shims_for_the_Win.pdf
- Shims
  http://sdb.tools/resources.html
- Mettez à jour pour forcer une invite UAC quand un fichier personnalisé .sdb est créé dans Windows
  https://support.microsoft.com/fr-fr/kb/3045645
- Shim Database to XML
  http://blogs.msdn.com/b/heaths/archive/2007/11/02/sdb2xml.aspx
- Tool to view and create Microsoft shim database files (SDB).
  https://github.com/evil-e/sdb-explorer
- Pure Python parser for Application Compatibility Shim Databases (.sdb files)
  https://github.com/williballenthin/python-sdb

2 - Gestion des polices de caractères sous Windows 10

Les vulnérabilités liées à la gestion des polices de caractères TrueType et OpenType sous Windows sont fréquentes. Entre janvier et octobre 2015, six des dix dernières vagues de correctifs publiées par Microsoft comprennent un bulletin les concernant.

Jusqu'à Windows 8.1, une vulnérabilité de type corruption de mémoire dans le traitement des polices de caractères impliquait souvent une possibilité d'élévation de privilèges en mode noyau. En effet, le code de traitement des polices résidait en noyau, dans le module `win32k.sys` ou dans le module `ATMFD.DLL` (*Adobe Type Manager Font Driver*), fourni par Adobe.

Avec Windows 10, Microsoft a refactorisé le code de gestion des polices et a déporté une partie de leur traitement en mode utilisateur. L'ancien code noyau est toujours présent, et utilisé dans certains cas, mais il se double d'une implantation en mode utilisateur nommée `fontdrvhost.exe` (*Font Driver Host*), dont l'usage est préféré en temps normal.

Comme auparavant, les fonctions GDI de chargement de polices, telles que `AddFontResource`, ou de calcul sur les polices, telles que `GetTextExtentPoint`, invoquent des services système de `win32k.sys` (scindé sous Windows 10 en `win32kbase.sys` et `win32kfull.sys`).

Lorsqu'un tel service est invoqué, un nouvel objet du noyau, nommé `UmfdHostLifeTimeManager`, s'assure de la disponibilité du pilote de police en mode utilisateur, `fontdrvhost`, avant de lui relayer la requête correspondante. S'il ne tourne pas encore, celui-ci est chargé à la volée. Pour ce faire, le noyau envoie un message au processus `Winlogon`, lui signifiant de lancer le pilote.

`Winlogon` démarre alors `fontdrvhost` dans un conteneur `AppContainer`, tournant avec un niveau d'intégrité bas, de façon à limiter l'impact d'une éventuelle compromission du processus sur l'intégrité du reste du système (sur ce sujet, voir aussi l'article sur les mécanismes de type "bac à sable" du bulletin d'actualité CERTFR-2014-ACT-26).

Une fois le pilote `fontdrvhost` démarré, celui-ci traite les requêtes en provenance du noyau. Elles sont prises en compte par quatre threads, qui communiquent avec le noyau par le biais d'appels système dédiés.

Le pilote `fontdrvhost` duplique à la fois le code de gestion des polices TrueType de `win32k.sys` et celui gérant les polices OpenType de `ATMFD.DLL`. En outre, cette instance du code OpenType en mode utilisateur bénéficie également de la compilation avec l'option CFG *Control Flow Guard* du compilateur, contrairement au pilote noyau `ATMFD.DLL` (au sujet du mécanisme CFG, voir l'article du bulletin d'actualité CERTFR-2015-ACT-015).

Cette tentative de sécurisation du code de gestion des polices de caractères sous Windows 10 est un effort louable. L'avenir seul dira si ce nouveau mécanisme suffira à détourner les attaquants de cette cible. Comme souvent, cette amélioration s'inscrit dans une démarche de défense en profondeur : l'ajout d'un obstacle supplémentaire, même s'il n'est pas infranchissable, tend à complexifier le travail de l'attaquant.

Documentation

- http://cert.ssi.gouv.fr/site/CERTFR-2014-ACT-026/
- http://cert.ssi.gouv.fr/site/CERTFR-2015-ACT-015/

3 - Évaluer la menace liée au typosquatting de nom de domaine

Il est devenu courant d'évaluer le niveau de sécurité de son entreprise en effectuant des audits et des tests d'intrusion de ses sites Web et de ses services. Certaines attaques ne portent pas directement sur les services offerts par l'entreprise, mais sur les fautes de frappe que pourraient commettre ses clients ou ses partenaires.

Le risque porté par ces attaques est souvent oublié ou ignoré lors des audits ou tests d'intrusion, mais si les ressources des attaquants sont utilisées pour mettre en oeuvre ces attaques, leur retour sur investissement est probablement significatif.

Le *typosquatting* tire parti des fautes de frappe commises lors de l'écriture du nom du site avec lequel l'utilisateur souhaite communiquer. Dans les vecteurs d'attaque habituellement identifiés se trouvent les noms de sites Internet, les adresses de courriel, mais également tout terme pouvant être référencé par un moteur de recherche. Le but de l'attaquant est de répondre présent lorsqu'un utilisateur fera une erreur en tapant l'adresse.

Par exemple, l'entreprise *Mon Entreprise* dispose d'un site internet `www.monentreprise.internet` (sans tiret) et d'adresses de courriel comme `contrats@monentreprise.internet`. Un attaquant achètera les noms de domaines suivants :

- mon-entreprise.internet
- mon.entreprise.internet
- monetnreprise.internet
- monenterprise.internet
- moneentreprise.internet
- monentreprse.internet
- mone-ntreprise.internet
- etc.

Ainsi, un utilisateur faisant une faute de frappe lors de l'envoi d'un courriel à `contrats@mon-entreprise.internet` enverra les documents potentiellement confidentiels, souvent mal protégés, directement à l'attaquant. Il est également difficile pour un utilisateur de choisir entre l'adresse légitime de l'entreprise (`...@monentreprise.internet`) et d'autres variantes acceptables avec des tirets en séparation de mot par exemple.

Cette attaque facilite également le *hameçonnage* en déjouant la vigilance de l'utilisateur. L'attaquant émet des courriels aux salariés de l'entreprise ciblée ou à ses clients, incluant un lien vers le site *typosquatté*. Ceux-ci se voient offrir le formulaire d'authentification identique à l'original sur un nom de domaine visuellement très proche de l'original. Comme ce formulaire est hébergé sur le site de l'attaquant, ce dernier reçoit l'identifiant et le mot de passe des utilisateurs visés. Il peut également mettre à disposition des documents malveillants à la place des originaux.

Plusieurs dizaines de noms de domaines peuvent être ainsi achetés afin d'exploiter le maximum d'erreurs, dans le but d'augmenter la chance qu'un courriel arrive dans la mauvaise boite. Dans la recherche de ces usurpations de nom, il convient d'évaluer les variations suivantes :

- Césure ou sous-domaine : découpage des mots par tirets ou points
- Omission : oubli d'une lettre
- Insertion ou Répétition : ajout d'une lettre, lettre simple doublée ou lettre double triplée
- Transposition : échange de deux lettres dans le nom
- Remplacement ou Homoglyphe : substitution d'une lettre par une autre lui ressemblant, éventuellement à partir d'un jeu de caractères étrangers (unicode)

Les attaques par homoglyphes sont les plus subtiles, les domaines les plus souvent rencontrés sont ceux remplaçant la lettre *m* par les lettres *r* et *n*, et échangeant les lettres *i* et *l*. Cette version du *typosquatting* est utilisée lorsque l'attaquant incite l'utilisateur à cliquer sur un lien envoyé par courriel par exemple. En fonction de la police d'écriture, certains noms sont particulièrement difficiles à distinguer :

- monentreprise.internet (original)
- monentreprlse.internet
- rnonentreprise.internet

Identifier clairement les noms proches ayant ou n'ayant pas encore été usurpés, permet à la fois d'anticiper ces attaques et de communiquer vers ses employés, clients ou partenaires, sur les sites Internet et de courriel appartenant réellement à l'entreprise.

Il convient donc, pour chaque nom de domaine acheté, de déterminer les noms de domaine les plus proches et de les faire enregistrer, avant que quelqu'un d'autre ne le fasse. Il existe également des outils et offres de services pour identifier et alerter sur ces attaques.

4 - Rappel des avis émis

Dans la période du 26 octobre au 01 novembre 2015, le CERT-FR a émis les publications suivantes :

- CERTFR-2015-AVI-451 : Multiples vulnérabilités dans Apple Safari
- CERTFR-2015-AVI-452 : Multiples vulnérabilités dans Apple iOS
- CERTFR-2015-AVI-453 : Multiples vulnérabilités dans Apple OSX
- CERTFR-2015-AVI-454 : Multiples vulnérabilités dans Apple OSX Server
- CERTFR-2015-AVI-455 : Vulnérabilité dans Adobe Shockwave Player
- CERTFR-2015-AVI-456 : Multiples vulnérabilités dans Squid
- CERTFR-2015-AVI-457 : Vulnérabilité dans OpenOffice

Gestion détaillée du document

02 novembre 2015 version initiale.