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 28 février 2014
N° CERTFR-2014-ACT-009

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2014-ACT-009

1 - Vulnérabilité dans l'implémentation du protocole SSL/TLS de certains produits Apple

Le 21 février 2014, Apple a publié des mises à jour de sécurité corrigeant une vulnérabilité dans les produits Apple TV versions 6.x, Apple iOS versions 6.x et versions 7.x. Quelques jours plus tard, une mise à jour de sécurité est publiée pour corriger la même vulnérabilité dans le système d'exploitation OS X Mavericks versions 10.9.x.

Cette vulnérabilité impacte le processus de vérification du certificat envoyé par un serveur durant l'initialisation d'une session SSL/TLS. Un attaquant situé en position d'homme du milieu (MITM) pourrait, lors de la négociation SSL/TLS, remplacer le certificat légitime du serveur par un certificat dont la clé privée correspondante est connue par l'attaquant. La vérification du faux certificat par un client vulnérable ne déclencherait alors aucune erreur. L'attaquant pourrait ainsi procéder au déchiffrement voire à la modification des données échangées entre le client et le serveur une fois la session SSL/TLS établie.

La connexion à des réseaux Wi-Fi public non sécurisés peut permettre la réalisation d'une telle attaque, en particulier si l'isolation entre les clients n'est pas garantie.

Description de la vulnérabilité

La vulnérabilité est située dans la fonction `SSLVerifySignedServerKeyExchange` présente dans le fichier source `sslKeyExchange.c`. Cette fonction est responsable de la vérification de la signature du certificat envoyé par le serveur.

```
static OSStatus SSLVerifySignedServerKeyExchange(...) {
    OSStatus err;

    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    ...

    err = sslRawVerify(...)
```

```
    ...

  fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

L'instruction `goto fail` à la ligne 6 est exécutée si l'appel à la fonction `SSLHashSHA1.update` ligne 5 retourne une valeur différente de 0 correspondant à un code d'erreur. Dans ce cas, le programme continue son exécution à l'étiquette `fail` pour libérer les ressources allouées et retourner la valeur de la variable `err`.

Dans le cas contraire, la variable `err` vaut alors 0 ; mais le programme exécute inconditionnellement l'instruction `goto fail` à la ligne 7, ce qui résulte en une sortie prématurée de la fonction. Celle-ci retourne alors 0, bien que les fonctions `SSLHashSHA1.final` et `sslRawVerify` ne soient jamais appelées : les paramètres de la négociation SSL/TLS ne sont donc pas vérifiés.

Recommandations

Le CERT-FR recommande :

- l'application des correctifs de sécurité dès que possible ;
- de ne pas faire transiter de données sensibles durant l'utilisation de réseaux Wi-Fi publics (gares, aéroports, ...) ;
- de désactiver la connexion automatique aux réseaux Wi-Fi.

Documentation

- Détails concernant la vulnérabilité SSL/TLS :
  http://www.crowdstrike.com/blog/details-about-apple-ssl-vulnerability-and-ios-706-patch/index.html
- Avis du CERT-FR concernant la vulnérabilité SSL/TLS dans les produits Apple TV et iOS :
  http://www.cert.ssi.gouv.fr/site/CERTFR-2014-AVI-081/index.html
- Avis du CERT-FR concernant la vulnérabilité SSL/TLS dans Apple OS X Maverick :
  http://www.cert.ssi.gouv.fr/site/CERTFR-2014-AVI-095/index.html
- Code source vulnérable du fichier sslKeyexchange.c :
  http://opensource.apple.com/source/Security/Security-55471/libsecurity_ssl/lib/sslKeyExchange.c

2 - Vulnérabilités logicielles sous Windows dues à la duplication non sécurisée de handle

L'ANSSI a récemment été amenée à auditer une suite logicielle. L'audit a permis de mettre en évidence une vulnérabilité due à une duplication non sécurisée de `handle`.

L'application est composée de deux parties :

- un service Windows tournant sous l'identité `SYSTEM` ;
- un processus s'exécutant sous l'identité de l'utilisateur courant.

Pour des besoins internes à l'application, le service duplique un handle sur un de ses threads et le fournit au processus de l'utilisateur courant. Or, les droits d'accès associés au handle vers le thread ne sont pas restreints lors de la duplication. Cela confère ainsi à l'utilisateur standard le contrôle total sur un thread s'exécutant dans le service `SYSTEM`. Ceci permet alors de réaliser une élévation locale de privilèges.

Mécanismes d'objets et de handles

Les systèmes Windows décrivent de nombreuses ressources manipulées par le noyau de manière générique en utilisant le principe d'«objet». Un objet est une structure de données pouvant représenter un fichier, un processus, un thread, un périphérique, etc.

Les applications souhaitant manipuler ces ressources n'accèdent pas directement aux structures de données stockées dans l'espace mémoire du noyau, mais doivent obtenir un «handle» sur l'objet. Celui-ci est ensuite fourni

aux API de manipulation du type d'objet concerné. Ce handle est propre au processus courant, et correspond en fait à un index dans un tableau des objets ouverts par ce processus (table des handles).

Lors de l'ouverture d'un objet, un processus spécifie un masque d'accès contenant les différents droits d'accès désirés. Ce masque est confronté au contexte de sécurité du processus (access token) et au descripteur de sécurité de l'objet pour valider ou refuser l'ouverture de ce dernier. Si l'ouverture est accordée avec les droits demandés, ceux-ci seront associés au handle retourné.

Ce mécanisme d'objets et de handles a notamment pour avantage de permettre :

- à Microsoft de mettre à jour les structures de données internes aux objets, tout en assurant la rétrocompatibilité, sans modifier les interfaces de programmation (API) proposées aux utilisateurs ;
- d'affecter des droits d'accès différents à des handles sur un même objet, selon les besoins (par exemple des processus peuvent ouvrir deux handles sur un même fichier, l'un en lecture uniquement et l'autre en écriture).

Exemples de duplications dangereuses et impacts associés

L'API `DuplicateHandle` permet de dupliquer un handle accordé à un premier processus vers un second (il est nécessaire pour cela que le processus effectuant la duplication possède le droit PROCESS_DUP_HANDLE sur les deux processus). Les deux processus possèdent alors un handle désignant un seul et même objet, qu'ils peuvent chacun manipuler. Cette API permet de spécifier, via son paramètre dwDesiredAccess, un masque contenant les droits d'accès qui seront associés au handle dupliqué. Si la duplication est utilisée pour transmettre un handle vers un processus de contexte de sécurité moins privilégié, sans restriction des droits d'accès, elle peut s'avérer dangereuse et représenter une vulnérabilité pour certains types d'objets.

Par exemple, peuvent poser problème :

- pour un objet Processus, les droits d'accès PROCESS_CREATE_THREAD, PROCESS_QUERY_INFORMATION, PROCESS_VM_OPERATION, PROCESS_VM_WRITE et PROCESS_VM_READ, ou le droit d'accès PROCESS_ALL_ACCESS les cumulant. Ils permettent l'utilisation des API `CreateRemoteThread`, `WriteProcessMemory`, ou `ReadProcessMemory` permettant de créer un nouveau thread dans le contexte du processus ou d'accéder en lecture ou en écriture à sa mémoire ;
- pour un objet Thread, les droits d'accès THREAD_GET_CONTEXT, THREAD_SET_CONTEXT, ou le droit d'accès THREAD_ALL_ACCESS les cumulant. Ils permettent l'utilisation des API GetThreadContext ou `SetThreadContext` permettant d'obtenir ou de définir la valeur des registres CPU du thread visé et donc de modifier arbitrairement son flot d'exécution.

Dans le cas du programme audité, il a été possible, au moyen du handle vers le thread, de définir un contexte d'exécution permettant la réutilisant de portions de code présents dans des bibliothèques chargées par le service Windows vulnérable et de lui faire exécuter un appel à l'API `LoadLibrary`. Cela permet de charger et d'exécuter une bibliothèque arbitraire et de prendre le contrôle du processus SYSTEM.

Recommandations

Les deux exemples ci-dessus, concernant les objets Processus et Thread, ne sont pas les seuls dangereux. Il est possible de dupliquer des handles désignant d'autres types d'objets différents en particulier Event, Pipe, Token, Device, Job, Desktop, etc.

Durant le développement d'une application, il est nécessaire de sécuriser l'utilisation de l'API `DuplicateHandle`, en connaissant le contexte de sécurité du processus de destination et en appliquant le principe du moindre privilège.

Il faut pour cela étudier le masque d'accès minimum nécessaire selon l'usage, s'assurer qu'il ne contient pas de droits pouvant s'avérer dangereux, et le spécifier via le paramètre dwDesiredAccess de l'API `DuplicateHandle`.

Enfin, le paramètre dwOptions permet de spécifier l'option dangereuse DUPLICATE_SAME_ACCESS, qui rend inopérant le paramètre dwDesiredAccess et associe au handle dupliqué les mêmes droits d'accès que le handle existant, sans que ceux-ci ne soient explicites. Cette option est donc à proscrire.

Documentation

1 http://msdn.microsoft.com/en-us/library/windows/desktop/ms724251.aspx

3 - Le format SWF (Flash) et les risques liés à son utilisation

Le format SWF a été créé en 1996 par Adobe Systems. Il est le format de description et de transport des applications à destination du lecteur Adobe Flash Player, logiciel gratuit servant à la diffusion de contenu multimédia Flash.

Les utilisations sont assez variées et vont de gadgets très simples sur une page Web à des applications beaucoup plus complexes (jeux vidéos, plateformes de formation en ligne, applications métier, etc.). SWF est un format ouvert, c'est-à-dire que les spécifications sont disponibles en libre accès. Cependant, Adobe Systems reste détenteur des droits. Ce modèle permet la création d'une multitude d'outils d'édition ou d'analyse du format SWF. Malgré cela, les alternatives au lecteur officiel se font rares, et sont souvent très en retard voire à l'abandon en matière de fonctionnalités. Portable, le lecteur Flash fonctionne sur les principaux systèmes d'exploitation des ordinateurs actuels (Windows, Linux, Mac OS X). Il est installé selon Adobe sur plus de 1,3 milliard de terminaux, avec des versions adaptées aux mobiles (Android, Apple iOS, BlackBerry OS, Symbian OS) et aux consoles de jeux (Sony PlayStation et Nintendo Wii notamment). Adobe annonce par ailleurs avoir un taux de pénétration de 90% des postes connectés à l'Internet.

Cela explique en partie pourquoi le format SWF constitue un vecteur d'attaque privilégié par les attaquants. Plusieurs éléments rendent le format SWF potentiellement dangereux. Tout d'abord, lorsqu'un utilisateur navigue sur le Web, son navigateur exécute par défaut automatiquement les applications Flash présentes sur le site sans aucune action particulière ou approbation explicite de la part de l'utilisateur, ni information préalable de celui-ci.

Ensuite, le format SWF offre un nombre significatif de fonctionnalités natives sur lesquelles un attaquant peut s'appuyer, notamment :

- la lecture et le décodage de nombreux formats (MP3, FLV, JPEG, XML, SWF, MP4, etc.) ;
- la communication via de nombreux protocoles (TCP, HTTP, RTMP, SSL, etc.) ;
- l'exécution de scripts de type « ActionScript ».

De plus, les fichiers SWF sont rarement utilisés tels quels, comme cela pourrait par exemple être le cas pour des formats de documents texte. Ils se retrouvent le plus souvent contenus dans des sites Web, mais peuvent aussi être embarqués dans des documents courants, comme les fichiers Word ou PDF. Ils peuvent donc assez facilement et de façon discrète être utilisés pour accéder à des informations telles que des identifiants de session ou d'autres informations contenues par exemple dans les fichiers parents.

La plupart des fichiers SWF que l'on trouve sur l'Internet sont compressés au format LZ777 (ZIP) dans le but de réduire la bande passante nécessaire pour distribuer ces fichiers. Les spécifications du format autorisent aussi l'utilisation de LZMA pour la compression, mais celui-ci est généralement peu utilisé dans les applications légitimes. Cela a notamment permis par le passé à un certain nombre de campagnes d'attaques d'échapper à de nombreux anti-virus et solutions de détection, qui ne prenaient pas en compte ce type de compression.

Exemples de vulnérabilités

La CVE-2007-0071 est une vulnérabilité assez simple et couramment exploitée. Elle repose sur la mauvaise interprétation d'une valeur dans le fichier (débordement de nombre entier), entraînant une allocation mémoire bien trop importante pour pouvoir réussir. Cette allocation laisse alors le lecteur Adobe Flash Player dans un état non désiré permettant ainsi l'exécution de code arbitraire à distance. Un exemple connu d'exploitation de cette vulnérabilité mène au scénario d'attaque suivant :

- téléchargement d'un fichier contenant une liste d'URL puis,
- téléchargement et exécution des fichiers pointés par ces URL.

La CVE-2014-0497 repose sur l'utilisation d'un script ActionScript visant à construire dynamiquement un exécutable peu complexe permettant le téléchargement et l'exécution de code arbitraire à distance. Un exemple connu d'exploitation de cette vulnérabilité mène au téléchargement de deux fichiers exécutables malveillants permettant de :

- voler des mots de passe (e.g. FireFox, Opera, Safari, OperaMail, Thunderbird, Pidgin, etc.) et des données des différents formulaires remplis sur des sites bien connus et fréquemment visités, comme Google, Twitter, Facebook, etc. ;
- établir un canal de Commande et Contrôle avec un serveur sur l'Internet maîtrisé par l'attaquant (cf. "Codes malveillants de type RAT", CERTA-2013-ACT-050), afin de mener à bien la récupération de ces informations à distance.

Durcissement de la configuration de Flash et bonnes pratiques

Afin de prévenir au maximum l'exploitation par des attaquants du format SWF, le CERT-FR recommande de :

- désactiver l'exécution automatique des greffons dans le navigateur (onglet « about:permissions » pour FireFox, « Options Internet » puis « Securité » pour IE) ;
- utiliser des greffons de navigateurs permettant de bloquer l'exécution automatique de Flash (NoScript, Flashblock, etc.) ;
- protéger par mot de passe les trousseaux de clefs utilisés par les navigateurs (ceci n'est d'ailleurs pas uniquement valable pour Flash) ;
- durcir la configuration du lecteur Flash, en éditant le fichier de configuration situé dans WINDIR\System32\Macromedia\, WINDIR\SysWow64\Macromedia ou USERPROFILE\AppData\Local\Google\Chrome\pour les systèmes utilisant Google Chrome ;
- sensibiliser les utilisateurs quant aux risques que présente le format SWF.

Un ensemble de variables de configuration sont accessibles, dont certaines permettant de désactiver les fonctionnalités de communication (téléchargement, téléversement, accès aux périphériques) ou d'activer le mode protégé. Par ailleurs, la seule désactivation de l'exécution automatique du greffon Flash n'est pas suffisante. En effet, une fenêtre surgissante sera quoi qu'il arrive proposée à l'utilisateur pour l'inviter à activer le greffon. Dans un tel cas, l'autorisation de l'utilisateur peut temporairement voire définitivement outrepasser la protection. Enfin, il convient de noter que s'il peut exister des alternatives au lecteur officiel Adobe Flash Player (c'est le cas de Gnash par exemple), elles ne sont pas forcément recommandables car pas nécessairement plus exemptes de défauts et ne sont généralement pas au même niveau en matière de fonctionnalités. De plus, elles ne disposent pas forcément d'un support de sécurité aussi réactif que celui fourni par Adobe Systems. Il semble donc plus pertinent, lorsque l'utilisation de Flash est souhaitable, d'utiliser le lecteur officiel avec un suivi aussi réactif que possible des mises à jour de sécurité.

Rappel des avis et alertes émis sur Flash

Depuis Juillet 2013, le format SWF a fait l'objet des avis suivants :

- http://www.cert.ssi.gouv.fr/site/CERTFR-2014-AVI-078
- http://www.cert.ssi.gouv.fr/site/CERTFR-2014-AVI-052
- http://www.cert.ssi.gouv.fr/site/CERTA-2014-AVI-019
- http://www.cert.ssi.gouv.fr/site/CERTA-2013-AVI-636
- http://www.cert.ssi.gouv.fr/site/CERTA-2013-AVI-509
- http://www.cert.ssi.gouv.fr/site/CERTA-2013-AVI-405

4 - Pistage de documents numériques

Certaines entités diffusant des documents numériques souhaitent parfois savoir, pour diverses raisons, si les destinataires les ont effectivement reçus ou consultés. Cet article vise à mettre en lumière les mécanismes présents dans certains types de documents, pouvant provoquer une communication vers un serveur distant parfois à l'insu de l'utilisateur.

Par exemple, une méthode répandue utilisant le courrier électronique consiste à inclure une image invisible dans un message HTML. Lors de la consultation du message, cette image est téléchargée depuis un serveur. Il suffit alors à l'emetteur de consulter les journaux du serveur pour être informé de l'ouverture du message.

Afin de se prémunir contre cette pratique, il est généralement possible de configurer le client de messagerie pour désactiver le chargement automatique des images. Toutefois, d'autres méthodes peuvent être utilisées suivant le type de document, certaines étant notamment basées sur des fonctions de sécurité.

Un fichier PDF peut, par exemple, être signé numériquement et inclure un certificat X.509. L'objectif de cette signature est de permettre la vérification de l'authenticité ou l'intégrité du fichier. Si le certificat possède une extension indiquant un serveur de CRL ou OCSP, le lecteur PDF va contacter ce serveur pour connaître le statut de révocation du certificat. Notons que, par défaut, dans le cas d'*Adobe Reader*, la vérification de la signature est effectuée automatiquement à l'ouverture d'un document et le statut de révocation est vérifié si le certificat remonte à un certificat racine de confiance. Il est cependant possible de modifier ce comportement grâce aux options disponibles dans « Edition », « Préférences », « Authentification ».

Microsoft Office possède, quant à lui, une fonctionnalité appelée *Information Rights Management* (IRM) permettant de contrôler l'accès aux documents Microsoft office. Lorsqu'un document, protégé à l'aide de cette fonctionnalité, est ouvert pour la première fois par un utilisateur, un serveur RMS (*Rights Management Services*) est contacté afin de télécharger une licence d'utilisation et déterminer le niveau d'accès accordé à cet utilisateur pour ce document. Une organisation peut, par ailleurs, disposer de son propre serveur RMS.

Ces communications automatiques peuvent fournir des informations aux personnes en mesure de les surveiller, comme le fait que le document ait été consulté, l'adresse IP publique de l'utilisateur ayant consulté le document, et parfois même une indication sur la version du lecteur de document utilisé.

Le CERT-FR conseille bien sûr d'utiliser les mécanismes de sécurité présentés ici lors de l'émission de documents si cela est jugé utile, mais l'objectif de cet article est de sensibiliser les utilisateurs aux actions automatiques qui se produisent lors de la réception et de la consultation d'un document venant de l'extérieur.

Documentation

- Article sur Adobe Reader et les CRL :
  http://blog.didierstevens.com/2013/05/13/adobe-reader-and-crls/
- Présentation de la fonctionnalité *Information Rights Management* dans Office 2010 :
  http://office.microsoft.com/en-us/excel-help/information-rights-management-in-office-2010-HA010354260.aspx

5 - Rappel des avis émis

Dans la période du 21 au 27 février 2014, le CERT-FR a émis les publications suivantes :

- CERTFR-2014-AVI-078 : Multiples vulnérabilités dans Adobe Flash Player
- CERTFR-2014-AVI-079 : Multiples vulnérabilités dans Google Chrome
- CERTFR-2014-AVI-080 : Multiples vulnérabilités dans PostgreSQL
- CERTFR-2014-AVI-081 : Vulnérabilité dans les produits Apple
- CERTFR-2014-AVI-082 : Multiples vulnérabilités dans HP Service Manager
- CERTFR-2014-AVI-083 : Vulnérabilité dans Cisco Unified SIP Phone 3905
- CERTFR-2014-AVI-084 : Multiples vulnérabilités dans Cisco IPS Software
- CERTFR-2014-AVI-085 : Vulnérabilité dans Cisco Unified Computing System
- CERTFR-2014-AVI-086 : Vulnérabilité dans Cisco Firewall Services Module
- CERTFR-2014-AVI-087 : Multiples vulnérabilités dans HP Application Information Optimizer
- CERTFR-2014-AVI-088 : Vulnérabilité dans McAfee ePolicy Orchestrator
- CERTFR-2014-AVI-089 : Multiples vulnérabilités dans HP XP P9000 Performance Advisor Software
- CERTFR-2014-AVI-090 : Multiples vulnérabilités dans Apple QuickTime
- CERTFR-2014-AVI-091 : Multiples vulnérabilités dans IBM AIX
- CERTFR-2014-AVI-092 : Multiples vulnérabilités dans Apple Safari
- CERTFR-2014-AVI-093 : Vulnérabilité dans Cisco Prime Infrastructure
- CERTFR-2014-AVI-094 : Multiples vulnérabilités dans IBM Content Navigator

Durant la même période, les publications suivantes ont été mises à jour :

- CERTFR-2014-ALE-001-001 : Vulnérabilité dans Microsoft Internet Explorer (ajout du correctif provisoire.)

Gestion détaillée du document

28 février 2014 version initiale.