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 27 juin 2014
N° CERTFR-2014-ACT-026

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2014-ACT-026

1 - Évolution de la politique de support de Microsoft suite à la publication de Windows 8.1 update 1

Microsoft accélère son rythme de lancement de produits, et les publications récentes de Windows 8.0, 8.1 et récemment 8.1 Update en sont l'illustration.

Malheureusement, la lisibilité de la politique de support de Microsoft est, quant à elle, mise à mal par cette accélération.

Jusqu'à présent, le cycle de vie des systèmes d'exploitation Microsoft se lisait comme suit :

- les mises à jour de sécurité sont publiées pendant 10 ans après la mise à disposition pour les versions majeures de Windows ;
- un niveau de service pack est supporté deux ans après la sortie de son successeur.

Or, nous remarquons que le cycle de vie de Windows 8.1 n'est pas celui d'une version majeure de Windows (10 ans) mais plutôt celui d'un service pack (8.0 expire 24 mois après la mise à disposition de 8.1).

De plus, nous avons désormais l'Update (mise à jour cumulative) de Windows 8.1 qui propose quelques ajustements d'interface graphique mais surtout, pour ce qui nous concerne, introduit un nouveau concept au cycle de vie des produits Microsoft : la mise à jour cumulative supportée 30 jours après sa publication (portée récemment à 120 jours pour les réseaux d'entreprise utilisant WSUS, Windows Intune et SCCM).

Ainsi, les mises à jour de sécurité publiées après le 13 mai 2014 ne seront pas proposées aux clients Microsoft Update ne disposant pas de l'Update de Windows 8.1. Les clients WSUS se verront proposer les mises à jour de Windows 8.1 sans la mise à jour cumulative jusqu'au 12 août 2014.

Il s'agit là d'un changement important de la politique de support des produits Microsoft qu'il convient de prendre en compte dans une politique de gestion des correctifs de sécurité.

Documentation

- Windows 8.1 Support Lifecycle Policy FAQ :
  http://support.microsoft.com/gp/lifecycle-Windows81-faq
- Windows 8.1 Update: WSUS Availability, Extended Deployment Timing :
  http://blogs.windows.com/windows/b/springboard/archive/2014/04/16/windows-8-1-update-and-wsus-availability-and-adjusted-timeline.aspx
- Quelles sont les nouveautés de mise à jour Windows 8.1 et de mise à jour Windows RT 8.1 ? :
  http://windows.microsoft.com/fr-fr/windows-8/whats-new

2 - Mécanismes de protection de type « sandbox » sous Windows (partie 2)

L'article de la semaine précédente s'était concentré sur deux mécanismes en particulier pour mettre en place des protections de type « sandbox », à savoir l'utilisation d'un jeton d'accès utilisateur limité et l'utilisation d'objets de type « Jobs ». Le présent article se focalise sur trois autres mécanismes :

- l'utilisation d'une station et d'un bureau spécifique ;
- l'utilisation des niveaux d'intégrité (à partir de Windows Vista) ;
- l'utilisation de l'API SetProcessMitigationPolicy.

Utilisation d'une station et d'un bureau spécifique

Windows offre différentes catégories d'objets permettant de gérer l'interface graphique utilisateur. La station et le bureau en sont deux composants. La station est un objet conteneur qui, entre autres, gère le presse papier ainsi que les `atoms` globaux. Une station peut gérer un voire plusieurs bureaux.

Le bureau quant à lui est chargé de l'affichage à l'écran et gère les objets graphiques (fenêtres, menus, ...). Il sert aussi de délimiteur pour les messages graphiques envoyés par les applications : ces dernières doivent être rattachées au même bureau afin de pouvoir communiquer entre elles via les messages graphiques.

Ces deux types d'objets possèdent des ACE (Access Control Entries) ; il est donc possible, pour un processus, de créer des objets avec des permissions d'accès limités.

La création d'un processus dans une station et un bureau spécifique permet d'isoler ce dernier du reste des processus exécutés sur la machine et ce de plusieurs manières :

- Chaque station possède son propre presse-papier ainsi qu'une table d'`atom` globale spécifique. Cette isolation empêche donc le processus sandboxé d'accéder au presse-papier ou à la table d'`atom` globale de la station principale de l'utilisateur. Ainsi il est possible d'éviter que le contenu du presse papier, ou les chaînes stockées dans la table d'`atom` globale, soient écrasés par un processus malveillant. La création d'une station est possible via l'appel à la fonction `CreateWindowStation`.
- L'isolation du processus dans un bureau spécifique permet d'empêcher l'interaction du processus avec d'autres applications (via des appels aux fonctions `SendMessage` ou `PostMessage`). Cette protection permet d'éviter l'implémentation d'attaques de type « Shatters Attack ». Ces attaques permettent à un processus possédant moins de privilèges d'envoyer (via le mécanisme de communication des interfaces graphiques de Windows) un message dont le but est d'insérer des données (provenant du processus contrôlé par l'attaquant) dans le processus ciblé privilégié. Une fois les données insérées, l'attaquant va chercher (toujours via le même mécanisme) à les exécuter. La création d'un Bureau est possible via l'appel à la fonction `CreateWindowsDesktop`.

Utilisation du contrôle d'intégrité obligatoire (MIC)

Depuis Windows Vista, un nouveau mécanisme de sécurité a fait son apparition dans les systèmes Windows : les niveaux d'intégrité. Il est à noter que chaque objet (dont les processus) possède un niveau d'intégrité.

Les niveaux de confiance (ou d'intégrité) les plus courants sont les suivants (du plus faible au plus élevé) :

- Untrusted ;
- Low integrity ;
- Medium integrity ;
- High integrity ;
- System integrity.

Ce mécanisme confronte, lors de la demande d'accès à un objet, le niveau d'intégrité du processus faisant la demande d'accès à celui de l'objet ciblé. Si ce dernier a un niveau d'intégrité plus élevé que le processus faisant la demande alors le mécanisme de vérification se réfère à un ensemble de politiques standards afin de déterminer les restrictions qui seront appliquées au processus.

Ces politiques sont au nombre de trois:

- `NO_READ_UP` : le processus avec un niveau d'intégrité inférieur ne peut pas lire l'objet cible ;
- `NO_WRITE_UP` : le processus avec un niveau d'intégrité inférieur ne peut pas écrire l'objet cible ;
- `NO_EXECUTE_UP` : le processus avec un niveau d'intégrité inférieur ne peut pas exécuter l'objet cible.

Par défaut tous les objets du système ont la politique `NO_WRITE_UP`, les processus (entre autres) ont en plus la politique `NO_READ_UP`. De plus, tous les objets qui n'ont pas de niveau d'intégrité défini auront par défaut le niveau Medium, correspondant au niveau de confiance d'un utilisateur connecté au système. Ainsi, ce dernier pourra accéder de manière transparente à ses objets.

Lorsqu'un processus protégé par une sandbox se voit attribuer un niveau d'intégrité faible (Low integrity), il ne peut donc écrire que dans les emplacements d'un niveau d'intégrité inférieur ou égal.

Par défaut, les éléments associés à ce niveau d'intégrité sont les suivants :

- pour les clefs de registre : HKEY_CURRENT_USER\Software\AppDataLow et ses sous-clefs ;
- pour les dossiers : %USER PROFILE%\AppData\LocalLow et ses sous dossiers.

Ce mécanisme vient renforcer celui des jetons restreints : même si un dossier reste accessible au jeton restreint (par exemple DACL à NULL), le processus ne pourra pas y accéder à cause de son niveau d'intégrité plus faible. De même, l'application des mécanismes de contrôle d'intégrité à l'envoi de messages graphiques (via des appels aux fonctions `SendMessage` ou `PostMessage`) empêche l'envoi de messages permettant l'écriture dans un processus de niveau supérieur, ce qui a pour effet de bloquer les attaques de type Shatters Attack.

Utilisation de l'API SetProcessMitigationPolicy

Depuis Windows 8, une nouvelle API est apparue : `SetProcessMitigationPolicy`. Cette fonction permet, entre autres, d'interdire à un processus de réaliser des appels aux fonctions du pilote « Win32k.sys ». Ce pilote, notamment responsable de la gestion de l'affichage côté système, a connu par le passé de multiples vulnérabilités. Ce nouveau mécanisme permet d'empêcher qu'un processus puisse "s'évader" de sa sandbox en utilisant une vulnérabilité présente dans ce pilote.

Enfin, il est à noter que depuis Windows 8, Microsoft a introduit un nouveau mécanisme de sécurité dans son système d'exploitation. Ce mécanisme, appelé `AppContainer` et utilisé par toutes les applications de type Métro, s'apparente à une protection de type sandbox. En effet, toutes les applications compatibles Métro utilisent une API spécifique, WinRT. Cette dernière se charge, lors d'appels à des fonctions spécifiques, de communiquer avec un broker (RuntimeBroker.exe) qui va déterminer si l'application possède les droits suffisants pour effectuer l'opération demandée.

Ces droits, inscrits dans la base de registre lors de l'installation de l'application, ont été précédemment définis par le développeur au travers du fichier `AppxManifest.xml` et auront été validés par Microsoft lors de l'ajout de l'application sur le marché Microsoft. Ce modèle de sécurité de déclaration des droits nécessaires à l'application s'apparente à celui que l'on peut trouver sur les magasins d'applications d'iOS ou d'Android.

Le lecteur attentif se sera aperçu que, parmi les fonctionnalités offertes par ces différents mécanismes de sécurité, certaines se recoupent et d'autres sont complémentaires. Le développeur est donc libre de faire ses choix afin d'implémenter le ou les mécanismes correspondants à ses besoins.

Documentation

- Page MSDN concernant les stations et les bureaux :
  - http://msdn.microsoft.com/en-us/library/windows/desktop/ms687096%28v=vs.85%29.aspx
  - http://msdn.microsoft.com/en-us/library/windows/desktop/ms682573%28v=vs.85%29.aspx
- Page MSDN concernant les niveaux d'intégrité :
  http://msdn.microsoft.com/en-us/library/bb625964.aspx

3 - Rappel des avis émis

Dans la période du 20 au 26 juin 2014, le CERT-FR a émis les publications suivantes :

- CERTFR-2014-AVI-277 : Multiples vulnérabilités dans le noyau Linux d'Ubuntu
- CERTFR-2014-AVI-278 : Multiples vulnérabilités dans le noyau Linux de Red Hat
- CERTFR-2014-AVI-279 : Multiples vulnérabilités dans Juniper Junos OS
- CERTFR-2014-AVI-280 : Vulnérabilité dans Huawei eSap Platform
- CERTFR-2014-AVI-281 : Multiples vulnérabilités dans phpMyAdmin
- CERTFR-2014-AVI-282 : Multiples vulnérabilités dans VMware vCenter Operations Management Suite
- CERTFR-2014-AVI-283 : Vulnérabilité dans le noyau Linux de SUSE

Gestion détaillée du document

27 juin 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-026