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 avril 2015
N° CERTFR-2015-ACT-017

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2015-ACT-017

1 - Recrudescence des attaques par macro « Visual Basic for Applications » dans les documents Microsoft Office

Depuis plusieurs semaines, il est observé sur Internet une recrudescence des attaques s'appuyant sur des macros VBA dans les documents Microsoft Office (Word, PowerPoint, Excel, etc.). Ces attaques sont relativement simples, car elles n'exploitent aucune vulnérabilité : elles reposent sur une mauvaise configuration du logiciel, ainsi que sur le comportement à risque des utilisateurs. En effet, la plupart des documents Microsoft Office peuvent comporter des macros VBA qui seront exécutées à l'ouverture, à la fermeture du document, ou suite à une action de la part de l'utilisateur (par exemple un clic sur un objet).

L'objectif de ces macros malveillantes est généralement de télécharger un programme à l'insu de l'utilisateur et de l'exécuter. Pour ce faire, l'attaquant dispose de plusieurs fonctionnalités qui lui permettent d'arriver à ses fins :

1. tout d'abord, afin que son code s'exécute sans action volontaire de la part de l'utilisateur, il faut créer une fonction appelée `Auto_Open`, `Workbook_Open` ou encore `Document_Open`, qui contiendra le code à exécuter à l'ouverture du document ;
2. une fois la macro exécutée automatiquement, il est possible de télécharger le programme malveillant en important des fonctions de la bibliothèque `wininet.dll` (par exemple `InternetOpenUrl`) ;
3. la dernière étape consiste à exécuter le binaire, avec les droits de l'utilisateur courant, grâce à la fonction `VBA Shell`.

Dans le logiciel Office, il est possible d'accéder à l'éditeur de macro en cliquant sur `Alt+F11`, mais la lecture de la plupart des macros malveillantes est protégée par mot de passe. Il est cependant possible de les récupérer, puisqu'elles sont stockées en clair dans le fichier. Dans le cas de l'ancien format OLE/CFB1, on pourra utiliser l'outil `oledump.py2` de Didier Stevens, développé en Python. Dans le cas du nouveau format OOXML, qui est en fait un fichier ZIP classique, on pourra chercher à extraire les fichiers `vbaProject.bin`, qui correspondent à des fichiers OLE/CFB.

Cependant, il n'est pas rare que le code VBA soit offusqué pour éviter d'être lu et détecté. Pour cela, les techniques classiques à base de `xor`, `rot` ou encodage hexadécimal sont utilisées, mais aussi les fonctions comme `Chr` et `ChrW` qui permettent de transformer un entier en caractère. Par exemple, les instructions ci-dessous se traduisent par la chaîne « `http` » :

```
Chr(104) & Chr(116) & Chr(116) & Chr(112)
```

Enfin, il arrive qu'une partie du code soit également reconstruite dynamiquement, à l'exécution, pour diminuer l'efficacité de l'analyse statique. Une partie de ces documents malveillants est par ailleurs diffusée dans le format MSO, qui est assez peu répandu (probablement afin de leurrer les outils de détection). Il consiste en un fichier XML contenant notamment des données encodées en base64 associées à un élément `<w:binData>`. Ces données, une fois décodées, commencent par la chaîne `ActiveMime` [3]. Un flux compressé avec l'algorithme deflate se trouve à l'octet 50. Ce flux correspond à un fichier Office classique.

Afin de se protéger contre ce type de menace, les conseils habituels sont rappelés :

– ne pas ouvrir les documents ou les pièces jointes non sollicités ;
– désactiver l'exécution automatique des macros dans la suite bureautique ;
– maintenir le système d'exploitation et l'anti-virus à jour.

La désactivation de l'exécution automatique des macros se paramètre dans le menu suivant :

```
Fichier
   > Options
       > Centre de gestion de la confidentialité
           > Paramètre du Centre de gestion de la confidentialité
               > Paramètres des macros
                   > Désactiver toutes les macros avec notifications
```

Néanmoins, beaucoup d'attaquants comptent sur le comportement de l'utilisateur et lui demandent d'activer les macros, afin de visualiser correctement le document. Ce comportement à risque est bien entendu à éviter. Il convient donc de rester attentif aux demandes d'activation de contenu et de sensibiliser régulièrement les utilisateurs quant à la gestion de la messagerie électronique.

Documentation

1. MS-CFB :
   https://msdn.microsoft.com/en-us/library/dd942138.aspx
2. oledump.py :
   http://blog.didierstevens.com/programs/oledump-py/
3. Return of the Mime MSO, now with Macros :
   http://blog.malwaretracker.com/2015/03/return-of-mime-mso-now-with-macro.html
4. MS-OVBA :
   https://msdn.microsoft.com/en-us/library/cc313094.aspx

2 - CVE-2015-1863 : dépassement de tampon dans wpa_supplicant

Le 22 avril 2015, une vulnérabilité de type dépassement de tampon a été corrigée dans le logiciel `wpa_supplicant`, pouvant causer une corruption du tas.

Ce logiciel est une implémentation libre de la norme IEEE 802.11i (protocole de sécurité des réseaux sans-fil aussi connu sous le nom WPA2), utilisé notamment dans les systèmes d'exploitation de type Android, Linux, Unix et BSD. Une version expérimentale pour Microsoft Windows est également fournie par l'éditeur.

Vulnérabilité

Cette vulnérabilité, identifiée sous la référence CVE-2015-1863, est due à une mauvaise vérification de la taille des données reçues lors d'une liaison directe entre deux appareils (mode P2P aussi appelé Wi-Fi Direct). En effet, lors du traitement de certaines actions dans ce mode, `wpa_supplicant` va écrire en mémoire le SSID du réseau (*Service Set IDentifier*, c'est-à-dire le nom du réseau) : la variable qui va recevoir cette valeur est limitée à 32 octets alors que la variable qui la lui transmet peut contenir jusqu'à 255 octets. Sans vérification de la taille, cette écriture peut conduire à un dépassement de tampon.

Suite à la réception d'un paquet en mode P2P (par exemple lors d'un balayage des appareils disponibles à proximité), le logiciel va ajouter l'émetteur du paquet à la liste des appareils disponibles. Pour cela, il utilise la fonction `p2p_add_device`. Dans cette fonction, il copie notamment le SSID reçu (champ de la structure du type `p2p_message`) dans le champ de la structure `p2p_device` prévu à cet effet :

```
int p2p_add_device(struct p2p_data *p2p, const u8 *addr, int freq,
            struct os_reltime *rx_time, int level, const u8 *ies,
            size_t ies_len, int scan_res){
    struct p2p_device *dev;
    struct p2p_message msg;
...
    if (msg.ssid &&
    (msg.ssid1 != P2P_WILDCARD_SSID_LEN ||
        os_memcmp(msg.ssid + 2, P2P_WILDCARD_SSID, P2P_WILDCARD_SSID_LEN)
```

```
    != 0)) {
        os_memcpy(dev->oper_ssid, msg.ssid + 2, msg.ssid1);
        dev->oper_ssid_len = msg.ssid1;
    }
...
}
```

Dans la structure `p2p_message`, le champ `ssid` est déclaré en `const u8 *ssid;` et peut contenir jusqu'à 255 octets de données. Or, dans la structure `p2p_device`, le champ `oper_ssid` est un tableau dont la taille est fixée à 32 octets (`u8 oper_ssid32;`). Aucune vérification sur la taille du champ `p2p_message.ssid` n'étant faite avant la copie, cela rend possible un dépassement de 223 octets sur les champs contigus à `oper_ssid` dans la structure `p2p_device`. De plus, d'après le bulletin de l'éditeur, ce débordement peut atteindre jusqu'à 150 octets supplémentaires dans le tas, en fonction de l'architecture matérielle utilisée.

L'exploitation de cette vulnérabilité nécessite que l'appareil ciblé soit en mode P2P et que l'attaquant soit situé dans le périmètre de réception radio de la victime. La corruption du tas engendrée peut aboutir à un comportement inattendu au niveau du programme, un déni de service, une fuite d'informations ou potentiellement une exécution de code à distance. Le contexte d'usage du mode P2P, moins fréquent que celui du mode infrastructure, limite les risques liés à cette vulnérabilité.

Depuis la découverte de cette vulnérabilité, une rustine logicielle a été appliquée au code de `wpa_supplicant`. Désormais, une vérification de la cohérence entre la taille des différentes variables contenant le SSID est effectuée avant l'écriture :

```
---
src/p2p/p2p.c | 1 +
1 file changed, 1 insertion(+)

diff --git a/src/p2p/p2p.c b/src/p2p/p2p.c
index f584fae..a45fe73 100644
--- a/src/p2p/p2p.c
+++ b/src/p2p/p2p.c
@@ -778,6 +778,7 @@ int p2p_add_device(struct p2p_data *p2p, const u8 *addr, int freq,
if (os_memcmp(addr, p2p_dev_addr, ETH_ALEN) != 0)
os_memcpy(dev->interface_addr, addr, ETH_ALEN);
if (msg.ssid &&
+ msg.ssid[1] <= sizeof(dev->oper_ssid) &&
(msg.ssid[1] != P2P_WILDCARD_SSID_LEN ||
os_memcmp(msg.ssid + 2, P2P_WILDCARD_SSID, P2P_WILDCARD_SSID_LEN)
!= 0)) {
--
```

Systèmes impactés

Cette vulnérabilité touche de nombreux systèmes d'exploitation, notamment Android, dont toutes les versions inférieures à la 5.1 sont vulnérables. Concernant Ubuntu, un correctif est disponible et distribué pour les versions 14.04, 14.10 et 15.04. Il en est de même pour Debian, les versions Squeeze et Wheezy disposant également d'un correctif.

Recommandations

Le CERT-FR recommande de mettre à jour les composants logiciels basés sur `wpa_supplicant` (composants système, pilotes, etc.) dès que le correctif aura été publié par leur éditeur. Dans l'attente de ce correctif, le CERT-FR recommande de ne pas utiliser le mode Wi-Fi Direct.

Documentation

- Bulletin de sécurité et correctif `wpa_supplicant` :
  http://w1.fi/security/2015-1/
- Référence CVE CVE-2015-1863 :
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-1863

3 - Rappel des avis émis

Dans la période du 20 au 25 avril 2015, le CERT-FR a émis les publications suivantes :

- CERTFR-2015-AVI-186 : Vulnérabilité dans Mozilla Firefox
- CERTFR-2015-AVI-187 : Multiples vulnérabilités dans PHP
- CERTFR-2015-AVI-188 : Multiples vulnérabilités dans Wordpress
- CERTFR-2015-AVI-189 : Multiples vulnérabilités dans le noyau Linux Red Hat
- CERTFR-2015-AVI-190 : Vulnérabilité dans le noyau Linux Red Hat

Gestion détaillée du document

27 avril 2015 version initiale.