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 13 septembre 2017
N° CERTFR-2017-ACT-036

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2017-ACT-036

1 - Recommandations concernant la détection de documents malveillants au format RTF

Rich Text Format est un format de document spécifié par Microsoft, dont la révision actuelle est la 1.9.1.

Plusieurs vulnérabilités ont été découvertes au cours des dernières années dans les parseurs de ce format, comme par exemple les CVE 2010-3333 et 2014-1761, ou dans l'interprétation du contenu embarqué du fichier dans le cas des CVE 2012-0158, 2015-1641, et plus récemment 2017-0199 et 2017-8759. Les documents RTF sont ainsi couramment utilisés dans des attaques basées sur l'ouverture de documents malveillants.

Nous proposons dans ce bulletin de revenir sur le format RTF et de montrer les limites d'approches visant à détecter des documents malveillants l'employant.

Format RTF

Le format RTF est extrêmement souple.

Prenons l'exemple illustratif suivant :

```
{\rtf0{\*\ignorezmoi{\controlwordavecparam12345 Ceci est un ensemble de données}}}
```

Le format est composé de différents types d'éléments principaux que nous retrouvons dans l'exemple :

- des **ControlWord**, qui sont des mots-clefs composés uniquement de lettres précédées d'une barre oblique inverse (\) (par exemple `\rtf`) servant à spécifier la mise en page du document. Les **ControlWord** sont éventuellement suivis d'un paramètre numérique signé, comme par exemple `\height1234` ou `\xpos-123`, et délimités entre eux soit par un autre élément du format, soit par un espace, soit par un caractère non alphanumérique ;
- des **ControlSymbol**, qui sont des caractères spéciaux précédés d'une barre oblique, permettant de spécifier un comportement associé à un **ControlWord**. Dans notre exemple le symbole ***** indique au parseur d'ignorer le **ControlWord** placé après le symbole s'il n'est pas reconnu (comme `\ignorezmoi` ici, qui n'est pas valide) ;
- les autres caractères, qui ne sont pas des **ControlWord** ou des **ControlSymbol**, correspondent à des données à afficher dans le document ;
- et enfin, les données, les **ControlWord** et leurs **ControlSymbol** peuvent être organisés en **Group**, délimités par des accolades **{** et **}**, qui rassemblent plusieurs éléments du document pour la mise en forme (par exemple, pour les notes de bas de page, les images, ...).

Idéalement, un document RTF standard est composé uniquement de caractères ASCII-7, ce qui ne donne pas accès à l'ensemble des valeurs prises par un octet (0 à 127 au lieu de 0 à 255). Cependant, il a été rendu possible dans des versions successives du format d'introduire du contenu binaire à l'aide de méthodes d'encodage (par

exemple `\'e3` ou `\u1234`) ou du *ControlWord* **\bin**, suivi du nombre d'octets dans le document à interpréter comme du binaire (par exemple `\bin42` suivi de 42 octets interprétés comme binaires).

Analyse statique

La souplesse du format est utilisée par les attaquants pour rendre très coûteuses les analyses statiques et les détections basées sur des signatures antivirales, voire permettre d'y échapper totalement.

Voici une liste non exhaustive d'exemples permettant d'échapper à une détection par une signature :

- l'ajout de caractères de retour à la ligne ou d'espaces inutiles entre les éléments du format, et de préférence pour l'attaquant entre les parties importantes utilisées pour une exploitation de vulnérabilité ;
- l'insertion d'accolades ouvrantes ou fermantes visant à empêcher un parseur de bien interpréter le contenu ;
- le réencodage de certaines parties déterminantes, grâce à `\bin` ou d'autres méthodes d'échappement ;
- l'insertion de **ControlWord** légitimes à des emplacements invalides (comme `\tab` au milieu d'un objet OLE placé dans `\objdata`) ;
- l'insertion de groupes avec pour seul contenu des **ControlWord** invalides qui seront ignorés par le parseur, car précédés du **ControlSymbol *** (exemple: `{\*\jesuisinvalide1234}` sera finalement interprété comme un groupe vide) ;
- dans une variante du cas précédent, il est également possible de cacher des données grâce aux paramètres : `{\*\jesuisinvalide1234e5}` deviendra `{e5}` car le **ControlWord** `jesuisinvalide` sera ignoré avec son paramètre numérique `1234`, dont `e5` ne fait pas parti : la lecture du paramètre s'arrête au caractère non numérique 'e'.

On notera enfin que Microsoft Office ne suit pas à la lettre la spécification du format, permettant encore d'autres comportements inattendus. Par exemple, bien que la spécification indique que la taille maximale d'un **ControlWord** soit de 32 caractères, le tampon alloué pour sa recopie par le parseur dans Microsoft Office est de 255 octets, qui devient alors la vraie limite de taille.

L'ensemble des possibilités offertes par le format ainsi que les différences d'implémentation sont autant d'opportunités pour les attaquants d'échapper à la détection.

Analyse dynamique

Il est possible de se tourner vers de l'analyse dynamique pour combler les lacunes de l'analyse statique.

Un exemple d'approche pouvant être employée est alors d'ouvrir un document dans une machine virtuelle bac à sable non connectée à Internet, et d'y ouvrir le document dans un lecteur vulnérable tout en collectant une trace des différentes opérations survenant sur le système et sur le réseau. La trace collectée est alors analysée pour y rechercher un comportement attendu caractéristique d'une exploitation du lecteur.

Dans le cas de l'exploitation de la CVE-2017-0199 au sein de Microsoft Office Word, on recherchera par exemple une tentative de téléchargement d'un document non légitime et l'exécution d'un processus `mshta.exe` destiné à interpréter ce document.

L'analyse dynamique est généralement plus fiable que l'analyse statique dans le cas des documents RTF à cause des nombreuses évasions statiques possibles. Cependant elle présente également des lacunes :

- le comportement malveillant supposé caractéristique par le défenseur et recherché dans la trace d'exécution peut être modifié par l'attaquant pour utiliser une autre méthode aboutissant au même résultat ;
- selon la connectivité ou la configuration de la machine, une partie des opérations peuvent être absentes ;
- une détection du bac à sable par la charge malveillante n'est pas à exclure.

Recommandations

Le CERT-FR recommande :

- de mettre à jour les logiciels manipulant des fichiers RTF pour réduire le risque de compromission par ce vecteur d'attaque ;
- pour l'analyse statique de documents RTF, l'utilisation de parseurs supportant diverses heuristiques de reformattage des documents avant d'appliquer des signatures de détection ;

– pour l'analyse dynamique, de tenir les heuristiques comportementales adaptées au contexte dans lequel s'exécute le document (connectivité, configuration, version des logiciels, ...) ;
– de tenir les heuristiques des outils et équipements à jour avec les différentes évolutions des menaces.

Documentation

– Avis CERT-FR CERTFR-2015-AVI-098
http://cert.ssi.gouv.fr/site/CERTFR-2015-AVI-098/index.html
– Avis CERT-FR CERTFR-2015-AVI-151
http://cert.ssi.gouv.fr/site/CERTFR-2015-AVI-151/index.html
– Avis CERT-FR CERTFR-2014-AVI-157
http://cert.ssi.gouv.fr/site/CERTFR-2014-AVI-157/index.html
– Avis CERT-FR CERTFR-2014-ACT-012
http://cert.ssi.gouv.fr/site/CERTFR-2014-AVI-012/index.html
– Spécification du format RTF 1.9.1
https://www.microsoft.com/en-us/download/details.aspx?id=10725
– Article FireEye - How RTF malware evades static signature-based detection
https://www.fireeye.com/blog/threat-research/2016/05/how_rtf_malware_evad.html
– Article CISCO Talos - How malformed RTF defeats security engines
http://blog.talosintelligence.com/2017/03/how-malformed-rtf-defeats-security.html

2 - Rappel des avis émis

Dans la période du 04 au 12 septembre 2017, le CERT-FR a émis les publications suivantes :

– CERTFR-2017-AVI-282 : Multiples vulnérabilités dans le noyau Linux de Suse
– CERTFR-2017-AVI-283 : Multiples vulnérabilités dans Google Android (Nexus)
– CERTFR-2017-AVI-284 : Multiples vulnérabilités dans Google Chrome
– CERTFR-2017-AVI-285 : Vulnérabilité dans Apache Struts
– CERTFR-2017-AVI-286 : Multiples vulnérabilités dans les produits Cisco
– CERTFR-2017-AVI-287 : Multiples vulnérabilités dans le noyau Linux de RedHat
– CERTFR-2017-AVI-288 : Multiples vulnérabilités dans le noyau Linux de Suse

Gestion détaillée du document

13 septembre 2017 version initiale.