L'ACTU SÉCU 10

JANVIER 2007

La frustration des RSSI...

xmco Partners

L'heure est aux voeux, de toutes parts. Chacun souhaite à ses proches, ses collaborateurs, son entourage, ses partenaires, ses clients, une très bonne année. Tout ceci va durer encore pendant quelques jours.
Sans forcément qu'il n'y ait de rapport avec cette effusion de sentiments, je me suis mis à réfléchir à la frustration au travail, et particulièrement à celle que ressentent souvent les Responsables Sécurité que je rencontre. A ma connaissance, la sécurité informatique ne constitue aujourd'hui qu'une problématique technique pour la plupart des gens. Une des dimensions ingrates de la sécurité informatique est réellement négligée. Sans vouloir faire de la psychologie de comptoir, j'ai pu me rendre compte, au cours de ma carrière, des innombrables difficultés auxquelles les RSSI ont à faire face : absence de budgets, positions hiérarchiques floues, difficultés de sensibiliser les collaborateurs, les Directions Générales, sentiment de lutter contre les éléments, contre tous les autres...

Ce qui est assez remarquable est que cette situation est particulièrement répandue dans les entreprises françaises, au sein desquelles on retrouve régulièrement les mêmes travers. Il serait d'ailleurs intéressant d'étudier la situation des RSSI dans d'autres pays d'Europe ou bien aux Etats-Unis, afin de déterminer si ces problèmes existentiels proviennent du poste ou d'autres facteurs externes.

Nous avons beaucoup réfléchi sur les solutions qui peuvent permettre aux RSSI de sortir de leur enfermement. L'une des premières pistes à explorer réside dans la stratégie de communication autour de la sécurité en entreprise. On constate souvent que plus la sécurité est perçue comme une contrainte technique, moins elle est considérée comme stratégique. Inversement, Les RSSI qui arrivent à peser sur les décisions stratégiques de l'entreprise sont ceux qui ont réussi à "valoriser" leurs effort, d'un point de vue Marketing, auprès de leur direction.

L'exercice de communication ne constitue pas une épreuve facile, a fortiori dans un domaine dont les connaissances paraissent si éphémères. Résoudre la quadrature du cercle revient à convertir chaque maillon de la chaîne en un élément propre du système de sécurité de l'entreprise : renforcer la vigilance de chacun, choisir un mot de passe robuste, protéger ses données, ne pas oublier son badge... constituent un ensemble d'actions si simples à énoncer, et si ardues à voir mises en oeuvres...

J'aimerais toutefois conclure sur une note d'optimisme : cette situation n'est pas une fatalité ! Il existe des moyens simples qui permettent de sortir de ces situations d'isolement, parfois difficiles à assumer. Il est possible d'entrevoir la sécurité autrement que comme un frein, une gêne, un obstacle. Je le sais, car je l'ai vu, nous l'avons fait, nous avons accompagné des clients dans ces démarches enrichissantes qui permettent de savourer le plaisir d'avoir atteint son objectif.

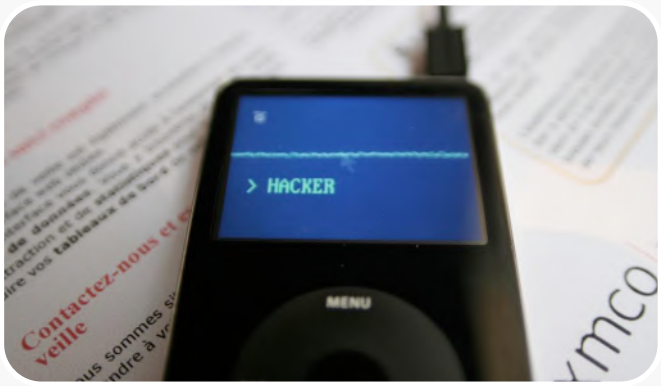
Je vous souhaite bien entendu, à vous, ainsi qu'à vos proches, une très bonne année. Mais surtout, je vous souhaite de vous sentir utile, dans votre métier, dans votre quotidien.

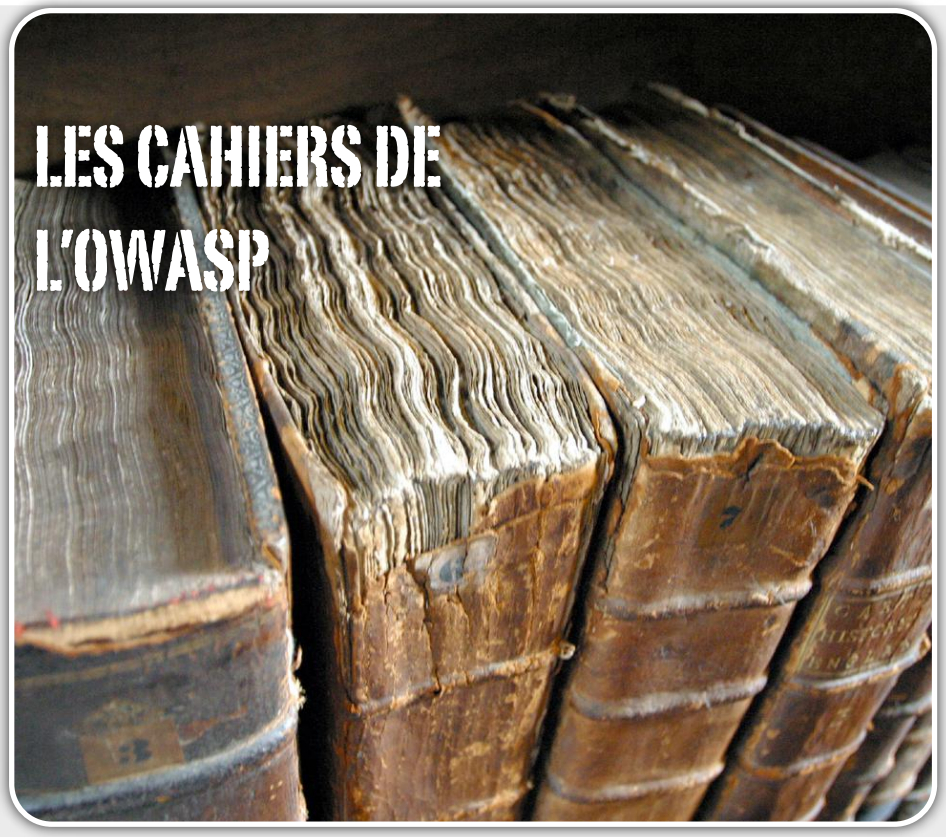
Marc Behar
Président du cabinet

Décembre 2006

Nombre de bulletins Microsoft : 7
Nombre d'exploits dangereux : 26

Top 5 des virus
1. 35,2% Dref
2. 22,2% Netsky
3. 10,7% Mytob
4. 7,8% Stratio
5. 5,2% Bagle





L'Encodage

Ce mois-ci, nous avons choisi de vous présenter le chapître « Canoncalization » de l'Owasp qui présente les différentes techniques d'encodage.

Nous essaierons d'éclaircir ou d'enrichir vos connaissances afin de connaître les parades et d'éviter de devenir la cible d'utilisateurs mal intentionnés.

XMCO | Partners

L'objectif de ce document est de vous présenter deux formes d'attaques moins connues mais néanmoins très efficaces. La compréhension de ces techniques vous permettra d'implémenter des parades simples pour garantir la robustesse de vos applications lorsque celles-ci seront confrontées à des données encodées, internationalisées et uni-encodées (Unicode).

Définition du problème

Lors d'une communication entre deux systèmes, l'ensemble des caractères reçus par un système doit être analysé et compris unilatéralement. Aucun des navigateurs, des serveurs web, des pare-feux applicatifs ou autres agents HTTP ne traite les spécificités d'encodage de la même manière. La « canonicalization » consiste en la conversion des données d'un format vers un autre.
Peu de personnes prennent conscience de ce problème et cela engendrent des erreurs d'implémentation qui peuvent avoir une incidence sur la sécurité de l'application.
Beaucoup de plates-formes web sont vulnérables car les exploits unicode sont rarement testés.

« Canonical » signifie la forme la plus simple ou la plus standard pour une donnée. Le fait de convertir une entrée vers sa forme la plus standard constitue donc un enjeu majeur.

Les applications web sont en permanence confrontées à des problèmes de conversion, par exemple en ce qui concerne les différentes façons d'encoder des URLs ou des adresses IP. Toutes les données reçues sous des formes inattendues doivent être traitées de la même manière.

NB: Se protéger à l'égard des attaques de type « canocalization » ou « Unicode » ne signifie pas que toutes les applications doivent être internationalisées, mais plutôt que toutes les applications doivent savoir gérer les données malformées ou sous forme Unicode.

Le codage

Tout d'abord, voici un petit rappel des normes de codages (ASCII, Unicode, UTF-8) qui sont souvent mélangées :

ASCII

L'ASCII (American Standard Code for information interchange) est une norme qui longtemps a été la plus utilisée en informatique. Elle comprend 128 caractères et est codée en binaire sur un octet. Elle rassemble tous les caractères les plus courants.

L'Unicode

L'Unicode est une norme informatique qui fut développée par le Consortium Unicode (1991). Elle permet de spécifier un numéro unique pour chaque caractère, *quelle que soit la plate-forme, quel que soit le logiciel et quelle que soit la*

langue. Dotée de 65535 codes *elle établit donc* une correspondance entre n'importe quel caractère d'écriture (toutes langues et tous signes) et un identifiant numérique.

Ce standard est aujourd'hui utilisé par la plupart des systèmes d'exploitation, dans tous les navigateurs récents et dans les applications client-serveur. Ainsi, cette implémentation permet d'échanger des données provenant de plusieurs langues et de plusieurs pays.
Généralement en Unicode, un caractère est codé sur 2 octets ce qui prend deux fois plus de place qu'en ASCII (1 octet). Etant donné que la majorité des caractères européens et américains utilisent seulement le code ASCII, un autre code a donc été implémenté afin de minimiser l'espace alloué à chaque caractère tout en étant capable d'en gérer une grande variété : l'UTF8.

UTF8

UTF-8 possède la propriété de préserver l'intégralité du format ASCII américain (US-ASCII). Il est compatible avec les systèmes de fichiers, les parsers et autres logiciels basés sur des données US-ASCII. Par contre, il est transparent pour toutes les autres valeurs.
Un texte UTF-8 est donc composé de caractères ASCII et peut également comprendre de l'unicode pour définir uniquement les caractères spéciaux.

Un caractère Unicode est, quant à lui, composé d'une suite d'un ou de quatre octets. Les caractères de numéro 0 à 127 sont codés sur un octet dont le bit de poids fort est toujours nul.

Les caractères de numéro supérieur à 127 sont codés sur plusieurs octets. Dans ce cas, les bits de poids fort du premier octet forment une suite de 1 de longueur égale au nombre d'octets utilisés pour coder le caractère, les octets suivants ayant 10 comme bits de poids fort.

Représentation binaire	Signification
0xxxxxxx	1 octet codant 1 à 7 bits
110xxxxx 10xxxxxx	2 octets codant 8 à 11 bits
1110xxxx 10xxxxxx 10xxxxxx	3 octets codant 12 à 16 bits
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	4 octets codant 17 à 21 bits

Figure 1 : Définition du nombre d'octet utilisé en UTF-8

Caractère	Numérotation décimal	Représentation binaire
A	65	01000001
é	233	11000011 10101001
€	8364	11100010 10000010 10101100
𝄢	119070	11110000 10011101 10000100 10011110

Figure 2 : Exemple de codage UTF-8

Un problème de sécurité évident

L'UTF8 présente donc un avantage considérable : il permet de représenter des milliers de caractères tout en optimisant l'espace utilisé. En revanche, il engendre l'apparition d'un problème majeur. En effet, les programmes mal écrits peuvent accepter un certain nombre de représentations UTF-8 et les convertir comme un seul et unique caractère.

Prenons l'exemple de la chaîne suivante : « /../ », qui est utilisée par les pirates afin de remonter une arborescence. La forme canonique du caractère ASCII "." est un point encodé « 2 E » en ASCII). Mais il est également possible de l'utiliser sous la forme UTF-8 sur 2 octets. Nous obtenons alors une représentation étendue telle que : C0 AE.
2 E (hexa)=0010 1111=46 en décimal
C0 AE (UTF-8)= 11000 000 1010 1111=46 en numérotation décimale

La chaîne « /../ », composée de 4 caractères peut donc se coder « 2F 2E 2E 2F » et « 2F C0 AE 2E 2F »

De même, il existe encore d'autres formes de représentations étendues pour le point : E0 80 AE, F0 80 80 AE, F8 80 80 80 AE et FC 80 80 80 80 AE.

Si l'analyseur syntaxique n'est pas soigneusement écrit pour rejeter ce type de chaînes, alors une faille de sécurité potentielle est ouverte. Un virus a exploité ce problème en 2001 en attaquant plusieurs serveurs web (nimda). Toutes les entrées utilisateur (URL, formulaires…) peuvent également être saisies sous forme Unicode pour camoufler un code malicieux et permettre ainsi une grande variété d'attaques. La RFC 2279 référence un grand nombre de façons d'encoder du texte comme nous allons vous le montrer dans la suite de cet article.

L'importance du format UTF-8 vient du fait que les serveurs web et les applications exécutent certaines étapes de leurs traitements dans ce format. L'ordre dans lequel ces traitements sont réalisés est important pour la sécurité de l'application. Fondamentalement, les étapes « URL decoding », « UTF-8 decoding » et les étapes intermédiaires sont des étapes au sein desquelles le contrôle sur les données reçues est critique pour la sécurité du reste de l'application.

Par exemple, si le contrôle qui consiste à rechercher la chaîne ".." est réalisé avant l'étape "UTF-8 decoding", il est possible d'injecter la chaîne ".." sous la forme étendue du format UTF-8 et donc de contourner le contrôle. Même si le contrôle détecte quelques variantes pour écrire un point sous une forme non-canonique, il est toujours possible que certaines formes d'encodage ne soient pas traitées.

INFO...

Quand IIS s'emmêle....

Prenons l'exemple d'une faille de sécurité découverte sur Microsoft IIS 4.0/5.0. Le serveur web recherchait la chaîne « /../ » avant d'effectuer le décodage UTF-8. Toutes les variantes UTF-8 n'étaient donc pas prises en compte.
En soumettant une url de ce type : http://victime/../../winnt/system32/cmd.exe, IIS identifiait bien l'attaque et générait une erreur.
Cependant, si l'attaquant remplaçait la chaine « ../.. » avec une version UTF-8 « ..%C1%9C.. », l'attaque de Directory Transversal pouvait être menée sans problème.

Quelques Exemples...

De nombreuses variantes d'une même url peuvent donc être formées et piéger les serveurs web. Le but de notre attaque est de lister le répertoire /bin du serveur web à l'aide d'une attaque de "Directory Transversal" :

Version non codée :
http://www.example.com/cgi-bin/bad.cgi?foo=../../bin/ls%20-al

Version URL-encodée de l'exemple :
http://www.example.com/cgi-bin/bad.cgi?foo=..%2F../bin/ls%20-al

Version Uni-encodée de l'exemple :
http://www.example.com/cgi-bin/bad.cgi?foo=..%c0%af../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%9c../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%pc../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c0%9v../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c0%qf../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%8s../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%1c../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%9c../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%c1%af../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%e0%80%af../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%f0%80%80%af../bin/ls%20-al
http://www.example.com/cgi-bin/bad.cgi?foo=..%f8%80%80%80%af../bin/ls%20-al

Les formats d'entrée (Input Formats)

Les applications web travaillent généralement sur un seul format de données : ASCII, ISO8859-1 ou Unicode (les programmes Java utilisent UTF-16). Vos visiteurs peuvent utiliser d'autres formats (ou « locales ») et les attaquants peuvent choisir arbitrairement l'utilisation de n'importe quel code « local ».

Le double encodage (Double encoding)

La plupart des applications web n'effectuent qu'une seule fois l'opération de décodage des données reçues. Malheureusement, un attaquant peut encoder deux fois les données envoyées (encodage de l'encodage). Cette technique permet d'induire en erreur le contrôle et d'insérer des caractères qui seront utilisés pour l'attaque.

Tester votre application

Identifier l'encodage supporté par votre application

Une technique simple et rapide permet de tester rapidement l'encodage utilisé par votre application. En vous envoyant une requête HEAD sur le port 80 de votre site web avec l'outil telnet, vous obtiendrez en réponse le type d'encodage utilisé dans les champs « content-type ».

```
xmcopartners:~# telnet www.xmcopartners.com
Trying 66.94.237.76...
Connected to premium.geo.yahoo.akadns.net.
Escape character is '^]'.
HEAD / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Wed, 17 Jan 2007 18:26:58 GMT
Connection: close
Content-Type: text/html; charset=iso-8859-1

Connection closed by foreign host.
xmcopartners:~#
```

Figure 3 : tentative de connexion via l'outil telnet sur le port 80 du site xmcopartners.com

Double encodage : votre application est-elle vulnérable ?

Comme nous l'avons présenté, le double encodage est une technique utilisée pour contourner le mécanisme de validation de votre serveur web.
Il est important de vérifier si votre application web gère ce genre d'astuces. Vous pouvez utiliser les outils de conversion mis à votre disposition à la fin de l'article « XSS Cheat Sheet » du site suivant :
http://ha.ckers.org/xss.html

Si l'injection réalisée sur votre application web fournit un résultat, votre application est alors vulnérable.

Comment se protéger ?

Utiliser les standards

Différentes solutions peuvent être implémentées afin de valider correctement les caractères envoyés par un utilisateur de votre application web.
Tout d'abord, une forme canonique doit être choisie et toutes les entrées de l'utilisateur doivent être converties vers cette forme avant d'êtres prises en compte par l'application. Les contrôles de sécurité (sanitization) doivent être réalisés après avoir correctement effectué le décodage UTF-8.
Vous devez analyser les contrôles effectués par votre application afin d'être certain qu'un format précis de données a été défini par les développeurs.

Si aucun format n'est défini, les formats (par défaut) suivants sont conseillés :

-Pour les requêtes « HTTP POST » : Le format ISO 8859-1 est recommandé. Ce format ne gère pas forcément les caractères encodés sur 2 octets (double byte character). Vous devez tester votre application avec les navigateurs clients officiellement supportés afin de vous assurer que les caractères encodés sur 2 octets soient correctement supportés.
-Pour les requêtes « HTTP GET » : Le format par défaut dépend du navigateur client et du format utilisé par la précédente requête GET. L'encodage de l'URL ne gère pas forcément correctement les caractères encodés sur 2 octets. Microsoft Internet Explorer peut être contraint d'utiliser exclusivement le format UTF-8, qui est un format canonique.

-.NET: Unicode (little endian*) doit être utilisé

-Implémentations JSP, à l'instar de Tomcat : ce genre d'applications utilise le format UTF8 (Reportez-vous à la section "javaEncoding" du fichier de configuration web.xml pour tous les conteneurs de servlets).

-Java: Unicode (UTF-16, généralement en « big endian »* mais cela peut varier en fonction de l'OS natif exécutant la machine virtuelle Java).

-PHP: Par défaut dans php.ini, le format est ISO 8859-1.

NB: Plusieurs fonctions PHP réalisent des assomptions (parfois erronées) sur le format utilisé ce ce qui entraîne des problèmes lors du changement de format.

Définition explicite des formats et locales utilisés

Le serveur web doit toujours définir et annoncer la locale et le code pays, comme par exemple : "en_US", "fr_FR", "zh_CN". De la même manière, le ou les formats attendus par l'application doivent être clairement définis. L'utilisation des standards HTML précisés plus haut sont recommandés. Ainsi les entrées et les sorties seront correctement traitées par tous les navigateurs, serveurs ou autres applications.

Une fois que chacun de ces points est correctement mis en oeuvre, il est fortement conseillé de tester l'application avec divers encodages.

** Little endian : ou petit-boutiste. L'octet de poids faible est stocké avant l'octet de poids fort. Utilisé chez Intel. À la place de « octet », on peut avoir un Word, un DWord ou le couple Segment-Offset pour une adresse (dans ce cas l'offset précède le segment).*

** Big Endian : ou gros-boutiste, Description d'une façon (parmi d'autres) dont on stocke les nombres dans plusieurs octets : l'octet de poids fort est stocké avant l'octet de poids faible. Utilisé sur la famille des 680x0 de Motorola, c'est la façon habituelle de voir un nombre.*

Bibliographie

- IDS Evasion using Unicode
http://online.securityfocus.com/print/infocus/1232

- W3C Internationalization Home Page
http://www.w3.org/International/

- XSS Cheat Sheet
http://ha.ckers.org/xss.html

- RFC2279
http://www.ietf.org/rfc/rfc2279.txt

INFO...

Windows Vista déjà piraté

Quelques jours après la sortie de Windows Vista, des pirates ont découvert comment activer frauduleusement le nouveau système d'exploitation. La sécurité de Vista, qui selon Microsoft, implémente des protections inviolables, est remise en cause.
Le nouveau système utilise une méthode d'authentification à clefs appelée Key Management Service (KMS). Au sein d'une entreprise, chaque copie de Vista doit être activée séparément par un serveur dédié.

Les attaquants ont réussi à créer un serveur similaire et proposent aux internautes de télécharger une machine virtuelle Vmware qui aura pour rôle d'activer les licences.

D'autre part, une seconde solution a été trouvée. Elle permet d'arrêter ce compte à rebours afin d'obtenir une version "illimitée" pour les versions x86.
En modifiant quelques lignes dans les dossiers Licensing de Vista et en utilisant les fichiers "pkeyconfig.xrm-ms" et "tokens.dat", une licence définitive est alors activée.

La fin d'année ne sourit pas à Microsoft qui s'apprête à sortir, en janvier, la version grand public de son nouveau système...

UN POINT SUR LA CRYPTOLOGIE

La sécurité absolue

La cryptographie est de plus en plus utilisée. Elle permet de résoudre de nombreux problèmes de sécurité.

Cependant, son utilisation est parfois obscure ou mal implémentée. L'étude de son histoire permet de mieux comprendre les recommandations actuelles telles que le choix de la longueur des clés.

XMCO | Partners

Rappel

Petit Lexique

Le préfixe « crypto » provient du grec et signifie « caché ». Le terme « cryptologie » constitue l'ensemble de la cryptographie et de la cryptanalyse.

La cryptographie est l'art de chiffrer/déchiffrer une information. En contrepartie, la cryptanalyse est l'art de casser un algorithme, c'est-à-dire de découvrir complètement ou partiellement l'information originelle sans en connaître la clé. La cryptanalyse a toujours été utilisée pour décrypter les langues mortes.

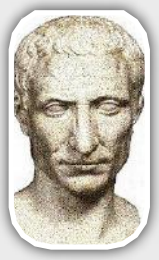
Confusion

Le verbe « crypter » est à bannir car il provient d'un abus de langage lié à son homologue anglais « to encrypt ».
Ainsi, un utilisateur peut chiffrer ou déchiffrer un message avec une clé tandis qu'un pirate peut décrypter un message sans disposer d'aucune information.
Enfin, un algorithme est dit « cassé/cracké », lorsqu'il existe une méthode permettant de déchiffrer un texte sans retrouver la clé ou de découvrir la clé utilisée sans « brute-force ».

Un peu d'Histoire

Les prémices

L'histoire retient Jules César comme précurseur et utilisateur régulier de la cryptographie pour envoyer des messages confidentiels tout en se protégeant d'éventuels messagers malhonnêtes.



Or, bien avant, entre le Xe et VIIe siècle av. J.-C., les Grecs utilisaient déjà une technique de chiffrement par transposition qui permettait de modifier la disposition des lettres dans un message. Ils se servaient d'une scytale (cf. Figure 1), appelée bâton de « Plutarque », autour duquel ils enroulaient une bande de cuir pour y inscrire le message. Une fois déroulé, le message est envoyé au destinataire qui possède un bâton de diamètre identique pour être à même de le déchiffrer.

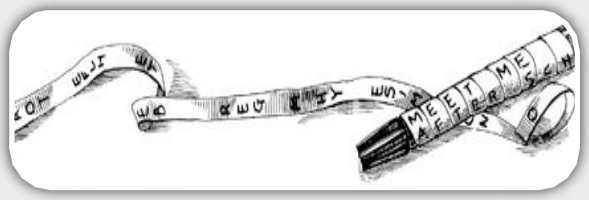


Figure 1 : La Scytale

Ensuite, un historien grec, Polybios, inventa une technique complètement différente. L'historien inséra l'alphabet au sein d'un tableau à deux entrées, ce qui lui permit d'encoder un caractère par un nombre (cf. Figure 2).

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Texte clair	X	M	C	O
Texte chiffré	53	32	13	34

Figure 2 : Chiffrement de Polybios

Jules César n'employait, quant à lui, qu'un simple décalage de 3 caractères alphabétiques (cf. Figure 2).

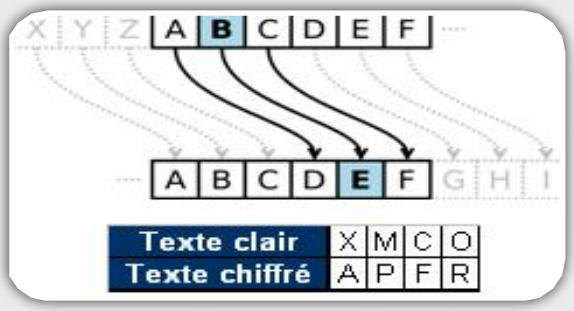


Figure 3 : Chiffrement de César (la lettre « A » devient « D »)

Les évolutions de la cryptographie symétrique

L'histoire de la cryptologie fût également marquée par les messages de Mary Stuart, exécutée en 1587 pour avoir participer à un complot qui visait à assassiner la reine d'Angleterre Elizabeth. L'accusée utilisait un algorithme de substitution, comme des hiéroglyphes, pour toutes ses correspondances (cf. Figure 3).

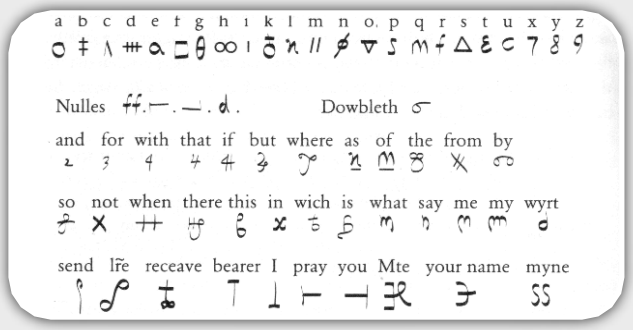


Figure 4 : Chiffrement de Mary Stuart

Durant la même période, le diplomate français Blaise de Vigenère présentait, dans son livre « Traicté des chiffres ou secrètes manières d'escrire » [2], une technique de chiffrement par substitution poly-alphabétique qui ne sera décrypté qu'en 1854. L'algorithme utilise une clé, dont chaque lettre définit le décalage alphabétique à appliquer sur le texte en clair (cf. Figure 5).

Texte clair	X	M	C	O	P	A	R	T	N	E	R	S
Clé	S	E	C	R	E	T	S	E	C	R	E	T
Texte chiffré	P	Q	E	F	T	T	J	X	P	V	V	L

Figure 5 : Chiffrement de Vigenère

Un dérivé de cette technique a été publiée en 1917 par Gilbert Vernam. Le chiffrement de Vernam est parfaitement sûr et a été prouvé par Shannon en 1949. La principale différence réside dans l'utilisation d'une clé unique et de taille équivalente à celle de l'information à chiffrer et dont chaque élément est choisi aléatoirement. Cet algorithme de chiffrement est utilisé au plus haut niveau : il assure la confidentialité du célèbre « Téléphone Rouge » entre Washington et Moscou.

Une dernière innovation, plus récente, fût l'utilisation de la machine « Enigma » par l'armée Allemande, durant la Seconde Guerre Mondiale.

Cette machine permettait de chiffrer toutes les communications radio ou télégraphiques . Le système était simple mais efficace [1]. Chaque lettre est substituée par une autre, avec des résultats qui changent à chaque fois. La machine est alimentée par une pile électrique. Elle contient un mécanisme de rotors qui modifie le circuit électrique à chaque frappe. Lorsqu'un utilisateur appuie sur une touche du clavier, un circuit électrique se ferme et une lampe s'allume en indiquant le caractère à utiliser (cf. Figure 6).

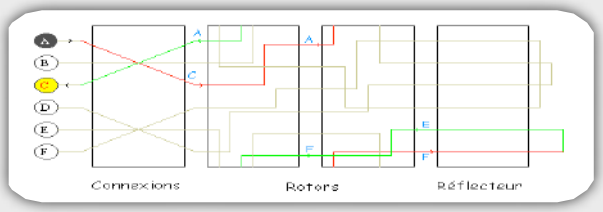


Figure 6 : Chiffrement de « Enigma »

Jusqu'au début des années 1970, la cryptographie était principalement utilisée à des fins malicieuses ou militaires. Par la suite, un nouveau besoin est apparu, notamment dans le domaine industriel pour protéger les flux interbancaires. Jusqu'à cette période, les méthodes de chiffrements ne réalisaient que le remplacement d'un caractère par un autre sans altérer l'ordre des informations. Ainsi, de nombreuses méthodologies de cryptanalyse se sont reposées sur l'occurrence d'une valeur en fonction de chaque langue (exemple, fréquence de la lettre « e » en français).

Les nouvelles générations d'algorithmes implémentent plusieurs systèmes de chiffrement. Par exemple, l'algorithme DES découpe l'information en plusieurs blocs puis utilise des permutations et des substitutions. L'information source est ainsi totalement obscurcie. La seule faiblesse de cet algorithme est l'utilisation d'une clé de petite taille (56 bits). DES a donc été remplacé par AES, algorithme qui n'est toujours pas cassé à ce jour, qui utilise des méthodes similaires mais avec des tailles de clés plus longues et des recommandations strictes. AES est aujourd'hui la référence pour les chiffrements symétriques.

Le besoin de la cryptographie asymétrique

De nos jours, l'utilisation massive d'Internet nécessite l'introduction de cryptographie asymétrique. Cette méthodologie permet d'assurer des fonctionnalités de confidentialité, d'intégrité, d'authentification et de non-répudiation. Les algorithmes comme RSA sont basés sur une paire de clés (publique/privée) et nécessitent des ressources importantes. Ils sont principalement utilisés pour l'échange de clés symétriques et pour l'authentification des parties.

Synthèse

La preuve par l'image

La plupart des algorithmes présentés chiffrent l'information sans modifier la disposition des données. Ainsi seule une taille de clé très longue permet d'obtenir un chiffrement solide. Afin d'illustrer plus précisément ces mécanismes, nous allons chiffrer une image (cf. Figure 7) avec quelques méthodes [3] que nous avons décrites dans ce sujet.

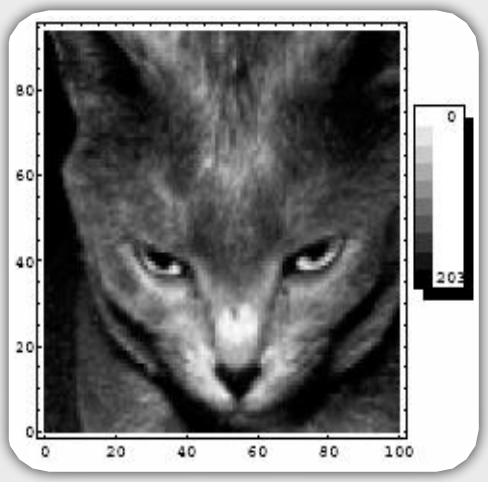


Figure 7 : Image initiale

En analysant la Figure 8, nous pouvons remarquer que le chiffrement de César ne masque en rien l'information, en effet, l'image est toujours visible, seules les couleurs ont été modifiées.

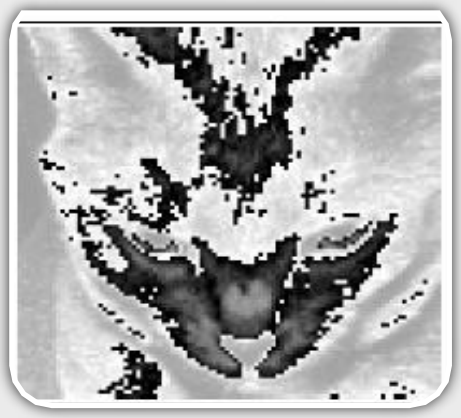


Figure 8 : Chiffrement de César

Les figures 9 et 10 démontrent la nécessité d'utiliser des clés extrêmement longues lors de l'utilisation d'un simple algorithme de permutation ou de substitution. Il est même intéressant de voir que l'utilisation de Vigenère avec une clé de 9400 bits ne camouflent pas toutes les informations, en effet, la silhouette du chat est toujours visible.

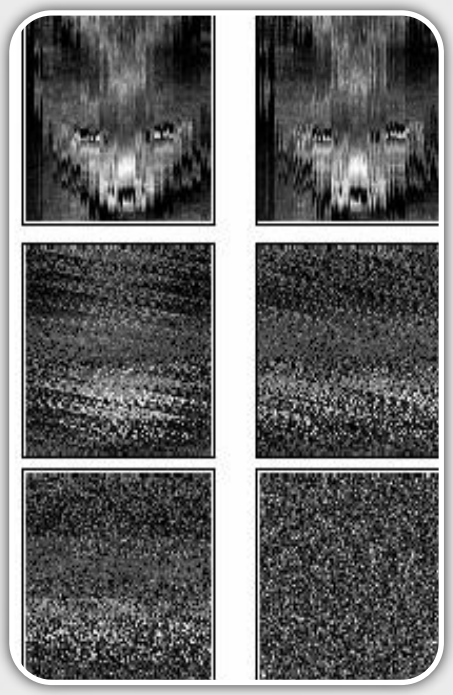


Figure 9 : Chiffrement par permutation (longueur de clé : 5, 10, 47, 94, 940, 9400)

Toutefois, la combinaison d'opérations de permutations et de substitutions effectuées par un algorithme comme DES permet d'obtenir de bien meilleurs résultats. En effet, même avec une clé de 56 bits, l'image est totalement obscurcie (cf. Figure 11).

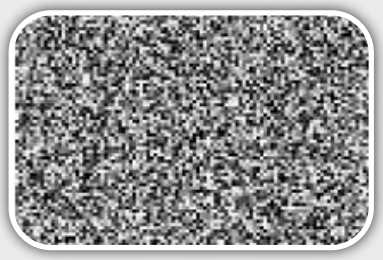


Figure 11 : Chiffrement DES

Conclusion

Les principales faiblesses de la cryptographie sont la mauvaise implémentation des algorithmes ainsi que le choix d'une taille de clé trop petite. L'exemple le plus médiatique est le protocole WEP qui permettait de sécuriser les échanges des réseaux sans-fil Wifi. Le WEP a été remplacé par le WPA qui respecte les recommandations d'utilisation de l'algorithme RC4 (clé minimale de 128 bits et vecteurs d'initialisation de 48 bits).

La cryptographie est, aujourd'hui, un outil indispensable qui connaît cependant ses limites. En effet, les algorithmes que nous utilisons reposent sur des problèmes mathématiques (difficulté de factorisation de nombres très grands, Théorème de Fermat). Pour information, en 2005, le plus grand nombre factorisé était de 663 bits or l'utilisation de RSA recommande des clés de 1024 ou 2048 bits.
A ce jour, seule l'apparition d'un ordinateur quantique permettrait de résoudre ces problèmes et rendrait alors obsolète toutes nos architectures de sécurité. D'ailleurs, l'algorithme quantique existe déjà… [4]

Bibliographie

- [1] Simulation du fonctionnement de Enigma (Applet Java)
http://www.bibmath.net/crypto/debvingt/enigmasimul.php3

- [2] Livre de Vigenère
http://gallica.bnf.fr/notice?N=FRBNF31575919

- [3] Démonstration réalisé par Robert ERRA sous l'environnement « Mathematica ».
http://www.esia.fr

- [4] Algorithme de Shor
http://fr.wikipedia.org/wiki/Algorithme_de_Shor

INFO...

Le chef de la mafia "has been" !!

On pourrait penser que les grands secrets de notre société sont bien gardés par des méthodes de chiffrement sécurisées. Et bien détrompez-vous!

Dernier exemple en date, l'erreur de débutant du chef de la mafia lui même... Le 18 avril 2006, Bernardo Provenzano a été arrêté après plusieurs décennies de cavale. Il utilisait un algorithme célèbre... ce n'était ni le fameux AES, ni l'algorithme RSA mais un des plus vieux chiffrement du monde : l'algorithme de César.
Comme nous vous l'expliquions dans cet article, il suffit de décaler l'alphabet de 3 lettres et de le noter sous une forme numérique.

Nous avons pu trouver une partie du message qui a permis à la Police Italienne de retrouver le chef de la Cosa Nostra :

"...I met 512151522 191212154 and we agreed that we will see each other after the holidays..."

Même des novices auraient pu trouver cette énigme. Je vous laisse un indice, on sépare les chiffres de la manière suivante : 5 12 15 15 22 19 12 12 15 4. Sauriez-vous retrouver les deux mots manquants ??

Réponse : On convertit cette suite de nombre en lettre (A=1, B=2...), puis on revient 3 lettres en arrière dans l'alphabet :
5 : E -->B
12 : L --> I
15 : O --> N
15 : O --> N
22 : V --> U
19 : S --> R
12 : L --> I
12 : L --> I
15 : O --> N
4 : D --> A

On obtient "BINNU RIINA" qui une fois entré dans Google, nous donne un lien vers Wikipédia :
"Grand ami de Frank Coppola, il fut le mentor de Salvatore Riina dit Toto Riina et de Bernardo Provenzano, dit u binnu".

Cette personne était donc un ami de Provenzano ce qui a permis de démasqué le parain sicilien...

NOUVELLES TENDANCES

Les vers javascript

Nous vous parlions, le mois précédent, des attaques de Cross Site Scripting dont les pirates raffolent. Nous avons choisi de vous présenter, cette fois-ci, les vers Javascript qui se sont développés depuis 2005 et qui ont attaqué quelques grands sites comme MySpace, Yahoo ou encore Google.

Comment fonctionnent-ils ? Comment se propagent-ils ? Petite présentation des derniers vers qui tendent à se développer sur la Toile.

XMCO | Partners

Définitions et présentations

Qu' est-ce que le Javascript ?

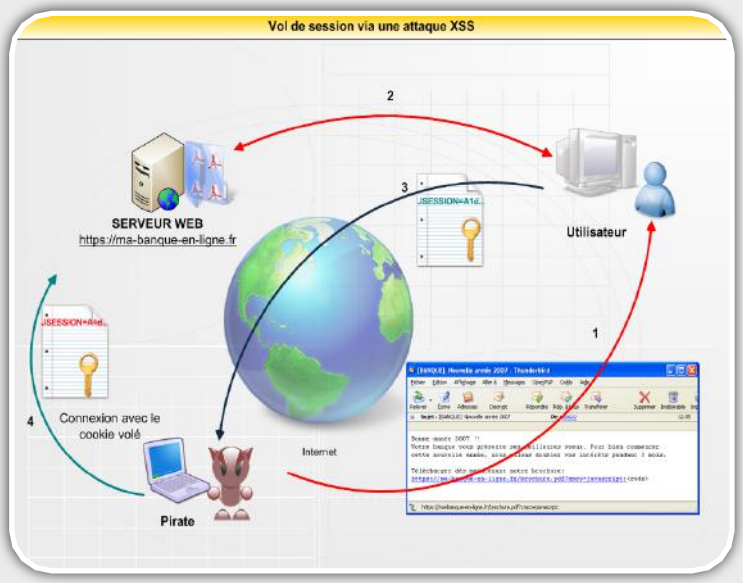
Le Javascript est un langage de programmation principalement utilisé dans les pages HTML. Il permet d'apporter certaines améliorations au format HTML et d'exécuter des commandes diverses sur le poste client (et non sur le serveur web) : écriture sur une page HTML (document.write), récupération des informations relatives au document (document.cookie), chargement d'une autre page (document.location), affichage de popup (alert())…

Les codes Javascript sont lisibles par tous les internautes. En effet, aucune confidentialité du code n'est possible et toutes les portions de code sont accessibles.

Qu'est-ce que le XSS ?

Le XSS est la contraction de Cross Site Scripting. Ce terme désigne une insertion imprévue de code Javascript au sein d'une page renvoyée par le serveur web et exécutée du coté client par les navigateurs web.

Le Cross Site Scripting est donc une attaque menée dans le but d'exécuter un code Javascript directement sur le poste de la victime. Cette technique a été particulièrement utilisée par les pirates pour voler la session d'un utilisateur connecté. Une fois le serveur web vulnérable identifié, le script est inséré au sein de l'URL et est envoyé la victime. Si cette dernière est connectée sur le site web vulnérable, elle enverra immédiatement, et à son insue, son cookie de session au pirate. L'attaquant pourra ensuite rejouer ce cookie et voler la session de la victime pour mener des actions frauduleuses.



On distinguent deux grandes familles de Cross Site Scripting : le XSS permanent et le XSS non permanent.
Le XSS non permanent est retourné immédiatement et ne reste pas sur le serveur web. La victime est touchée personnellement et l'attaque est souvent menée grâce à un paramètre non contrôlé dans l'URL. De son côté, le XSS permanent est une technique employée pour laisser le code malicieux sur le serveur web pour que chaque client, qui visite la page vérolée, devienne une nouvelle victime.

Le code réside alors sur le serveur web et sera exécuté par le navigateur. C'est cette méthode que les vers XSS utilisent.

Si vous souhaitez en savoir plus sur les techniques XSS non permanentes, nous vous proposons de vous référer à notre newsletter « Actu-Secu » de Décembre 2006.

Comment cette vulnérabilité est-elle exploitée par les vers ?

Comme les parasites qui doivent s'accrocher à un corps vivant, les vers XSS doivent s'accrocher à une page web pour vivre. Pour cela, les vers utilisent des sites web où les utilisateurs sont autorisés à insérer leurs propres balises HTML (où il est possible d'insérer du code Javascript). Cela concerne essentiellement les sites communautaires, les forums, les blogs, etc.

Les développeurs web ne prennent pas souvent la peine de vérifier chacune des entrées utilisateurs et laissent ainsi une porte ouverte aux pirates. Les failles XSS sont assez courantes et les avis sur la gravité de cette faille sont plutôt partagés. Certains (administrateurs, DSI…) ne prennent pas cette vulnérabilité au sérieux car ce genre d'attaque nécessite une intervention relativement difficile à réaliser pour l'utilisateur car elle doit être menée à grande échelle.

D'autres, dont les experts en sécurité, abordent le sujet sous un autre angle. Ils ont conscience du danger. En effet, cette faille de sécurité est bien plus grave qu'elle n'y paraît.
Les pirates l'utilisent le plus souvent pour voler le cookie de session d'un utilisateur. Cela signifie qu'un attaquant peut prendre le contrôle de votre session lorsque vous êtes connecté sur le site de votre banque en ligne. Baptisée « le nouveau débordement de tampon » par certains spécialistes, elle offre d'autres possibilités dont celle d'exécuter n'importe quel code Javascript. Vous êtes donc à même d'imaginer les risques qu'une simple faille de Cross Site Scripting peut engendrer : scan d'un réseau interne, propagation de vers, etc.…
Un ver XSS peut se propager par l'intermédiaire de plusieurs vecteurs. Les exemples qui suivent, présentent les différentes voies de contamination possibles.

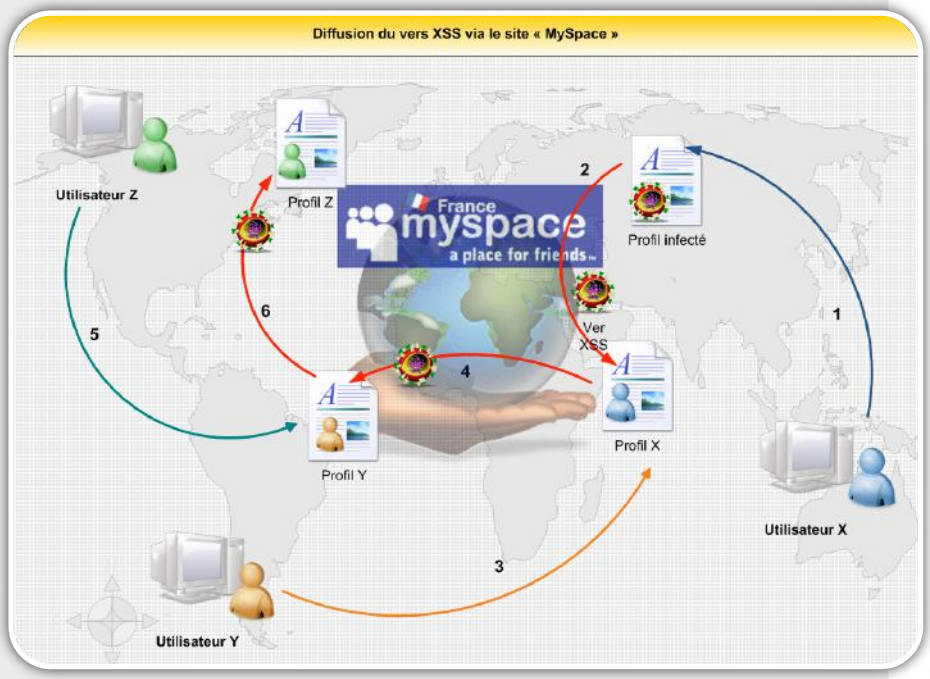
Samy : premier ver pour MySpace

Le premier ver du genre fut créé en 2005 par un adolescent. Le ver n'avait aucun impact sérieux, il se contentait d'ajouter le profil « Samy » dans la liste d'amis de l'utilisateur abusé . Il se répliquait ensuite pour finalement contaminer près de 1 000 000 profils.

Le jeune pirate a, tout d'abord, analysé la sécurité du site en question et a constaté que l'insertion de balises HTML au sein de son profil était relativement simple. Bien que le site filtrât correctement plusieurs mots utilisés par le langage « Javascript » (ce qui restreint à priori ce genre d'attaque), l'insertion d'un tel code était toujours possible en utilisant des méthodes de contournement ingénieuses…(voir cadre a la fin de l'article)
Samy a trouvé une méthode relativement simple qui consistait à scinder le mot « Javascript » en 2 (retour chariot) afin de tromper une grande partie des navigateurs web. L'attaque pouvait alors commencer.

A l'aide d'un code Javascript particulièrement travaillé, le ver envoyait silencieusement des requêtes POST et GET via Ajax pour contaminer chacun des utilisateurs et des contacts visitant le profil « Samy ». La méthode utilisée était particulièrement astucieuse puisque l'auteur a réussi à générer des requêtes http via Ajax ainsi qu'un clic de validation sans la moindre intervention de l'utilisateur.

La création d'un tel ver n'est pas évidente car les plus grands sites prennent généralement des précautions afin d'éviter l'exécution de code Javascript. L'auteur de ce ver a donc du trouver des moyens pour contourner les contrôles de MySpace.



Des exemples de vers sur des sites connus : Yahoo et MySpace

Yammaner : le premier ver sur une plateforme web-mail

Yammaner fut le premier ver du genre. Identifié en juin 2006, ce virus était inséré directement dans le corps d'un e-mail et s'exécutait dès qu'un utilisateur de la web-mail de Yahoo ouvrait l'e-mail vérolé. Ce dernier avait pour but de chercher tous les contacts du carnet d'adresses pour se diffuser par la suite par l'intermédiaire des requêtes Ajax. L'exécution de code Javascript n'était pas interdite par Yahoo. N'importe quel code pouvait donc être inséré dans un e-mail et s'exécuter dans le contexte de la victime. Une fois l'e-mail ouvert, le ver se propageait en envoyant une copie de lui-même à tous les contacts de la victime (avec un champ « From » correspondant à un des contacts).

Dernier exemple en date : Quickspace MySpace

MySpace est un site de rencontre utilisé par près de 73 millions de personnes dans le monde. La moindre faille pourrait affecter le profil de nombreux utilisateurs, ce qui fut le cas au mois de décembre dernier. En effet, les pirates ont utilisé une fonction intégrée de Quicktime et une faille de Cross Site Scripting (corrigée maintenant) pour voler un grand nombre de comptes.

Le ver était écrit en Javascript et camouflé au sein d'une vidéo « .mov ».
Le principe est relativement simple. La vidéo malicieuse est glissée sur une page de profil MySpace. Un usager MySpace se connecte alors sur le profil qui « héberge » la vidéo pirate.
Dès l'ouverture de la vidéo avec le lecteur Quicktime, le ver utilise une fonction nommée « HREF Track », qui oblige le lecteur à utiliser un code JavaScript (voir plus bas) afin de charger des pages web. Ce code téléchargeait un script « .js » aux adresses suivantes :

http://almobty.com/css/[REMOVED].js
http://www.cake.fi/images/[REMOVED].js
http://www.daviddraftsystem.com/images/[REMOVED].js
http://www.tm-group.co.uk/images/[REMOVED].js

Le contenu du profil de l'utilisateur abusé était alors modifié. Une fausse barre de navigation remplaçait la barre officielle et pointaient vers des sites pirates qui affichaient une fausse page de login MySpace. Par ailleurs, il subtilisait les mots de passe et les login entrés par les victimes. Quant à la vidéo infectée, elle apparaissait sur le profil MySpace de l'utilisateur qui devenait alors le nouvel hôte pour diffuser le ver.

Enfin le code qui contenait aussi une fonction de spamming était envoyé par e-mail à tous les contacts du compte infecté.

Figure 2 : Fausse barre de navigation qui pointe vers une site pirate aux couleurs de MySpace

Comme nous vous le disions au préalable, le problème vient du logiciel Quicktime et de la fonction « HREF track ». D'après Apple, cette fonctionnalité permet de définir des fonctions Javascript ou bien des pages web qui chargeront une "frame" ou fenêtre précise. En d'autres termes, une page web spécifiée dans une balise « HREF Track » s'ouvrira dès le lancement de la vidéo.

La fonction de Quicktime mise en cause n'a pas alerté Apple qui a délivré un correctif temporaire ne prévoyant pas cependant de supprimer cette fonction.

La faille PDF au centre d'interrogations...

Après l'insertion de code Javascript au sein de fichiers « .Mov » et « .swf » qui aident au développement de vers, un autre format de fichier extrêmement utilisé inquiète les experts en sécurité informatique. Le format PDF peut également exécuter du Code Javascript à l'aide d'une simple URL. En ajoutant un morceau de code après le nom du fichier PDF dans l'URL, une attaque de Cross Site Scripting est possible. Tous les sites qui hébergent un tel fichier, sont alors vulnérables ce qui laisse un très grand nombre de cibles pour les développeurs de malwares.

Un article détaillé a, d'ailleurs, été rédigé par nos consultants. Il est disponible à l'adresse citée en référence.

Bibliographie :

- Explication du code du ver Samy : http://namb.la/popular/tech.html
- Article Xmco : La faille d'Acrobat Reader qui inquiète les DSI http://www.mag-securs.com/article.php3?id_article=6940
- Description du ver Yamanner par F-Secure http://www.f-secure.com/v-descs/yamanner_a.shtml

ASTUCE...

Comment l'auteur du ver Samy a contourné les contrôles du site MySpace ?

Les techniques de contournement pour insérer du code Javascript sont nombreuses. Petit aperçu des astuces utilisées par l'auteur du ver Samy pour contourner les protections « anti-Javascript ».

Utiliser les balises autorisées pour insérer du Javascript

Comme nous vous l'indiquions, la première grande étape fut de trouver un moyen pour insérer des balises Javascript. Seules les balises <a>, <img> et étaient autorisées sur le site MySpace. Aucune validation au sein des tags CSS n'était alors implémentée. Par conséquent, le code Javascript pouvait être inséré de la manière suivante:

Exemple : `<div style="background:url('javascript:alert(1)')">`

Insérer de code au sein d'une variable

Après avoir trouvé le moyen d'insérer du code Javascript, un autre problème se posait. En effet, il était impossible de saisir des guillemets au sein de la balise DIV (qui en contient par défaut). La seule solution était de faire appel a une expression qui aurait pour rôle de stocker le code de notre virus comme la ligne suivante. Pour insérer des guillemets double une simple conversion "décimal->ASCII" avec la fonction adéquate suffisait...

Exemple : insérer du code au sein d'une balise div : `<div id="mycode" expr="alert('hah!')" style="background:url('javascript:eval(document.all.mycode.expr)')">`

Exemple : insérer des guillements doubles : `<div id="mycode" expr="alert('double quote: ' + String.fromCharCode(34))" style="background:url('javascript:eval(document.all.mycode.expr)')">`

Insérer le mot "Javascript", nécessaire à l'exécution du code malicieux

Le code Javascript pouvait alors être introduit mais pas encore exécuté car le serveur web rejetait le mot « Javascript ». L'auteur du ver a donc eu l'idée de scinder le mot en deux avec le caractère de retour chariot qui n'est pas interprété par certains navigateurs web (IE, Safari).

Exemple : `<div id="mycode" expr="alert('hah!')" style="background:url('javascript:eval(document.all.mycode.expr)')">`

Phase finale : récupération d'informations diverses pour la duplication du ver

La partie « préparation » était alors en place.

L'étape suivante consistait à poster le code malicieux sur le profil de l'utilisateur. Le code source de la page était donc nécessaire. La simple fonction « document.body.innerHTML » permettait de récupérer ce dernier ainsi que le numéro d'identifiant de la victime (ID). Une fois de plus, les développeurs avaient pris soin de contrôler toute insertion de la chaîne « innerHTML ». Cependant, l'utilisation de la fonction « eval » qui permet d'exécuter le code Javascript fourni en paramètre n'était pas filtrée.

Exemple : `alert(eval('document.body.inne' + 'rHTML'));`

Notre fonction requise pouvait alors être insérée.

Après avoir également obtenu plusieurs informations indispensables (numéro d'identifiant de la victime, listes des contacts de la victime, hash pour la confirmation de l'ajout…) l'envoi de la requête était possible et l'auteur du ver pouvait ainsi contaminer un autre profil en copiant le code du ver sur la page de la première victime. La réplication s'effectuait lors de chaque visite du profil infecté.

Tendance de l'activité malicieuse d'Internet :

L'activité malicieuse du mois de décembre reflète, à peu de chose près, celle de l'année 2006. En effet, nous avons assisté à :

- la publication de nombreux exploits sans innovation particulière.
- La spécialisation et la complexification des programmes malicieux qui visent les plates-formes mobiles.
- Le maintien de l'activité virale.
- L'exposition des programmes les plus utilisés à la recherche de vulnérabilités.

La publication continue d'exploit

Une année qui finit comme elle a commencé :

Plus de neuf exploits Microsoft ont été publiés ce mois-ci. Ces derniers permettent à un attaquant distant de causer le déni de service d'un postes de travail qui implémente un système Windows. Cependant, l'utilisation de ces programme requiert un serveur malicieux ou un accès local à la machine ciblée. De ce fait, leurs portées diminuent sensiblement.

Par ailleurs, un exploit qui vise des produits Oracle a également été publié à la fin de l'année 2006. Ce code malveillant exploite une ancienne faille Oracle, corrigée en août 2004.
Il s'agit, plus précisément, d'une faille de type "Directory Transversal" du composant "extproc" des versions Oracles 9i et 10g. Ce dysfonctionnement permettait à un attaquant distant d'accéder à des librairies présentes hors du répertoire "$ORACLE_HOME\bin". Un pirate, qui disposait des privilèges "CREATE [ANY] LIBRARY", pouvait, en exploitant cette faille, exécuter des commandes arbitraires sous les privilèges de l'utilisateur "DBMS".

Les exploits ne sont pas les seuls à profiter des failles de sécurité. En effet, les virus et les vers en ont aussi besoin pour se diffuser.

La publication de plusieurs virus SymbianOS

La mutation du comportement :

De nombreux virus qui visent les téléphones portables, en particuliers ceux qui implémentent un système SymbianOS, ont été publiés durant le mois de décembre.

Ces différents programmes malicieux sont de plus en plus performants et leurs auteurs ont acquis une certaine expérience dans ce domaine.
Le nombre de programmes malicieux de ce type devrait croître. En effet, l'agrégation de fonctionnalités et l'omniprésence des terminaux mobiles dans notre vie quotidienne en font des objets sensibles. Les pirates l'ont bien compris et les schémas d'attaques connus sur les infrastructures IP devraient se décliner sur ces nouveaux supports. Nous en avons eu quelques illustrations à la fin de l'année 2006.

Par exemple, le manque de vigilance des usagers peut être exploité. Le cheval de Troie "Skull.AF" l'illustre parfaitement. Il se diffuse sous la forme d'un fichier SYS non certifié. Ce troyen se diffuse massivement, alors qu'il est recommandé de ne pas installer de programme non certifié.

Une fois qu'il a infecté un terminal, il remplace les applications systèmes par des versions non fonctionnelles. Ainsi , toutes les fonctionnalités du téléphone sont désactivées et

toutes les icônes sont alors remplacées par une image qui présentent une tête de mort.

Les anciens programmes malveillants sont améliorés afin de contourner les restrictions qui leur avaient été érigées. C'est le cas du cheval de Troie Pbleaster.G qui infecte les téléphones équipés du système Symbian OS Series 60.
Ce programme copie la totalité des contacts stockés sur le téléphone infecté et transmet ces informations à l'attaquant via Bluetooth.

La multiplicité des vecteurs de communication des téléphones mobiles profite également aux pirates. En effet, cette diversité élargit la surface d'attaque.

Un autre code malicieux a été identifié ce mois-ci sur les téléphones qui implémentent le système Symbian. Ce dernier se nomme SymbOS/Mobispy.A et se charge de récupérer toutes les informations à caractère confidentiel comme les sms, les appels effectués, les notes, etc. Toutes les informations récoltées par ce biais sont envoyées vers un serveur pirate.

Comme nous le disions au début de ce paragraphe, ces attaques se complexifient, s'améliorent et sont de plus en plus ciblées. La recrudescence de ce type de code, d'exploits et de virus permet de prédire une augmentation de l'activité virale mobile en 2007.

L'activité virale surfe sur l'actualité

Comment pousser les internautes à la faute :

A chaque événement planétaire, le volume d'e-mails vérolés augmente considérablement. La période des fêtes de fin d'année constitue une aubaine pour les pirates. En effet, ces derniers espèrent glisser quelques pièces jointes malicieuses dans la masse d'e-mails envoyés. Parmi ces e-mails malveillants, nous retrouvons **Trojan-Spy.Win32.Ardamax.e** qui se diffuse sous la forme de l'exécutable "**Christmas_Puzzle.exe**". Le nom contextuel a été choisi pour obtenir un maximum de chance de tromper un internaute. Une fois installé, ce malware se dissimule à l'aide d'un rootkit qui rend sa détection très laborieuse.

De nombreux e-mails, ayant comme sujet "happy New year" , ont été expédiés récemment. Certains d'entre eux possédaient une pièce jointes malveillante "**postcard.exe**". L'ouverture de cet e-mail donnait instantanément à l'expéditeur la possibilité d'effectuer des actions arbitraires sur le poste de l'internaute abusé.

Ce malware pèse 170 ko. Il est connu sous les noms suivants:

NOM	EDITEUR
TR/Dldr.Tibs.jy	AntiVir
Downloader.Generic3.EIY	AVG
Trojan.Downloader-388	ClamAV
Win32/Luder.I	CA
Trojan.DownLoader.17085	DrWeb
W32/Tibs.RA	F-Prot
Trojan-Downloader.Win32.Tibs.jy	F-Secur
Trojan-Downloader.Win32.Tibs.jy	Kaspersky
Win32/Nuwar.M	NOD32
W32/Dref-U	Sophos

Une présentation powerpoint contrefaite sur le thème de Noël, a profité de l'atmosphère festive de la fin de l'année pour infecter quelques postes. Ce programme malveillant est connu sous le nom de "Exploit.MSPPoint.Agent.g". Il exploite la vulnérabilité Microsoft MS06-012.

Enfin, un autre executable "CHRISTMAS.EXE" est envoyé avec une image de Noël. Ce dernier est un bot qui essaie de télécharger des fichiers malveillants sur les deux serveurs suivants :

- waiguadown.008.net
- user.free.77169.net

INFO...

La disparition des virus...pas si sûr

A peine sorti en version professionnelle, le futur système d'exploitation de Microsoft est déjà critiqué par les professionnels de la sécurité.

En effet, certains annonçaient ce nouvel opus comme un OS sécurisé qui n'a nullement besoin d'un antivirus. Cette affirmation a été contredite par des éditeurs d'antivirus.

Il a été démontré que même si les virus "Net-sky-D" et "MyDoom-O" sont effectivement bloqués par l'outils de messagerie, ces programmes malveillants peuvent infecter Vista si l'utilisateur consulte un e-mail vérolé via une webmail.

La sécurité de Vista n'a donc pas fini de faire parler d'elle...

OUTILS LIBRES

Les des outils bien utiles :

Chaque mois, nous vous présentons les outils libres qui nous paraissent indispensables. Les logiciels abordés sont variés : utilitaires de sécurité et autres programmes utiles, voir indispensables, en entreprise.

Ce mois-ci, nous avons choisi d'analyser les logiciels suivants :

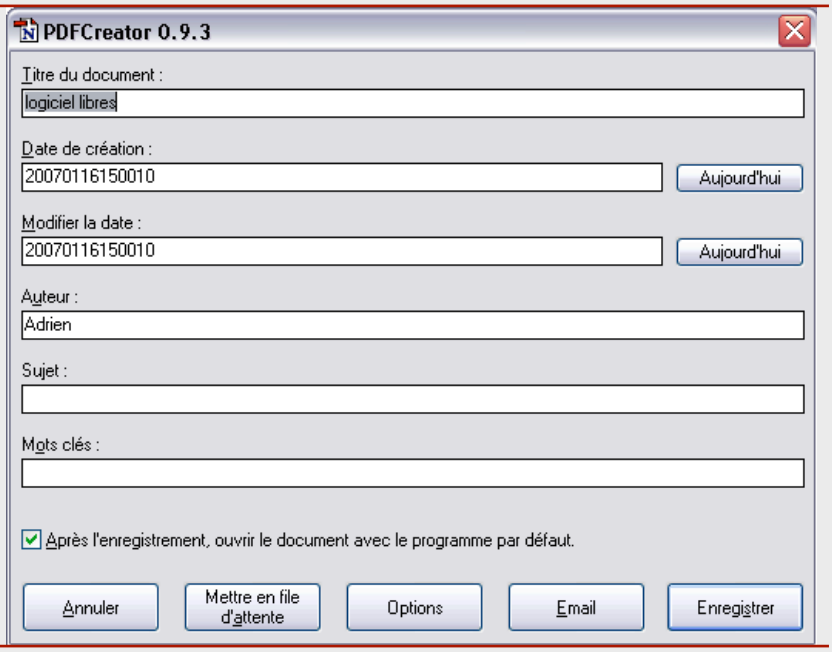
- PDFCreator : logiciel de création de fichiers PDF
- 7-zip: Outil de compression
- PowerToys : Série d'outils
- Supercopier : Logiciel de copie

Vous trouverez à la fin de cette section un tableau récapitulatif des versions de tous les logiciels présentés lors des précédents numéros d' « Actu Sécurité ».

XMCO | Partners

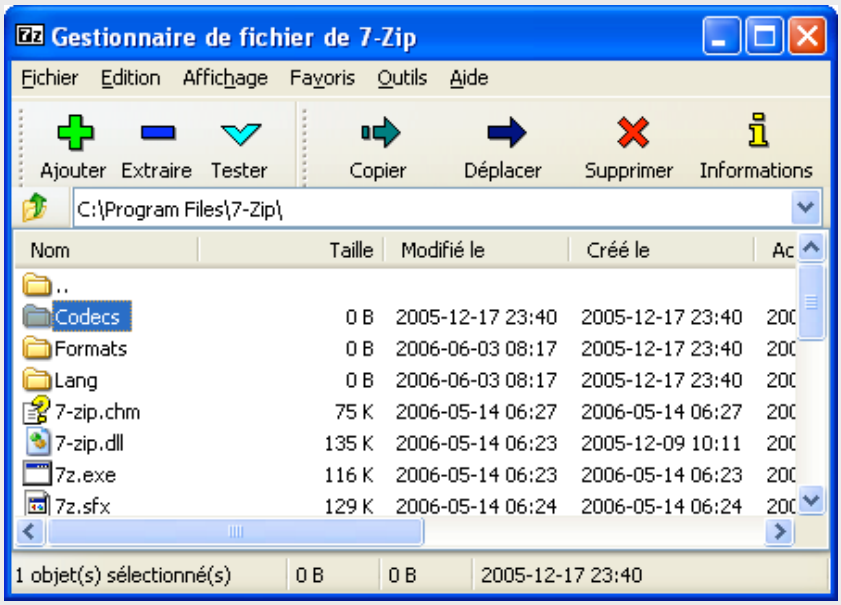
PdfCreator

Creation de fichiers PDF

Version actuelle	0.9.3
Utilité	★★★★★
Type	Logiciel Bureautique
Description	PDFCreator est un logiciel libre qui permet de créer des fichiers imprimables en fichiers PDF. Ce logiciel est entièrement configurable (résolution du document, version d'Acrobat Reader pour la compatibilité...). Il s'installe comme une imprimante et est donc lancé à partir de l'option « Imprimer » de n'importe quel logiciel. Simple et pratique, c'est l'un des seuls freeware du genre.
Capture d'écran	
Téléchargement	Ce logiciel est disponible pour les plates-formes Windows à l'adresse suivante : https://sourceforge.net/project/showfiles.php?group_id=57796
Sécurité de l'outil	Aucune faille n'a été publiée à ce jour.
Avis XMCO	Les solutions gratuites pour convertir des documents Word et Excel en PDF ne sont pas nombreuses. PDFCreator est extrêmement simple. En quelques clics, l'imprimante virtuelle est installée et la conversion s'effectue rapidement.

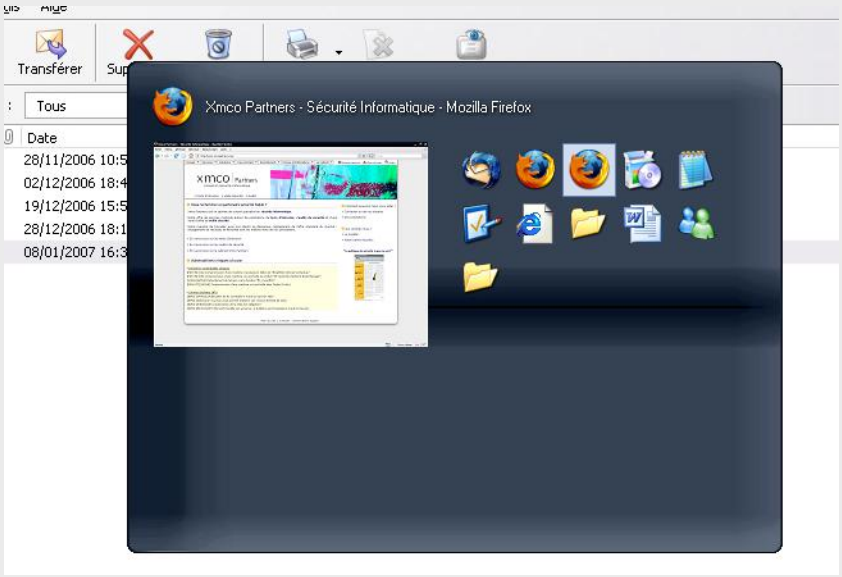
7-Zip

Logiciel de compression/décompression

Version actuelle	4.42
Utilité	★★★★★
Type	Logiciel Bureautique
Description	7-zip est un logiciel de compression méconnu qui apporte cependant tous les avantages des leaders du marché. Il supporte près de 63 formats (ZIP,TAR, RAR, ARJ, CAB, CHM, CPIO, DEB, ISO, LZH, NSIS, RAR, RPM…) et s'intègre dans le menu contextuel de Windows comme ses aînés.
Capture d'écran	
Téléchargement	Ce logiciel est disponible pour les plates-formes Windows à l'adresse suivante : http://www.7-zip.org/fr/
Sécurité de l'outil	Aucune faille n'a été publiée à ce jour.
Avis XMCO	Les fameux Winzip et Winrar sont, incontestablement, les plus utilisés du marché. Cependant, ces derniers restent des Shareware et ne peuvent être utilisés sans l'achat d'une licence ou en dehors de la période d'évaluation. Ce logiciel est aussi efficace que ses aînés. Il améliore même la compression de 2 à 10%. 7-zip sera donc le parfait outil de compression pour tous vos formats Windows et Unix.

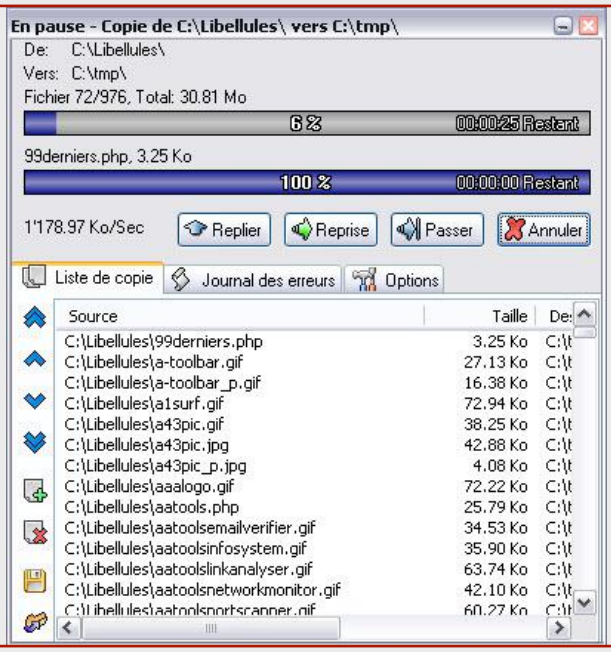
PowerToys

Utilitaires pour Windows

Version actuelle	Juillet 2002
Utilité	★★★★½
Type	Logiciel Bureautique
Description	PowerToys est une série d'outils développés pour Windows dans le but de s'intégrer directement aux fonctionnalités natives de l'OS de Microsoft. Plusieurs logiciels sont donc proposés par Microsoft. Ils s'intègrent au sein des menus contextuels de Windows. Voici la liste de ces outils. Certains changeront votre façon de travailler et vous feront gagner du temps.
Capture d'écran	
Téléchargement	Les Powertoys sont disponibles à l'adresse suivante : http://www.microsoft.com/france/windows/xp/home/utilisez/info/info.asp?mar=/france/windows/xp/pro/utilisez/info/20021015-powertoys.html
Sécurité de l'outil	Aucune faille n'a été publiée à ce jour.
Avis XMCO	PowerToys est une suite de petits logiciels extrêmement utile. Certains vous paraîtront inutiles, d'autres comme Image Resizer ou Open Command Windows Here vous changeront la vie !

Supercopier

Copieur de fichiers

Version actuelle	0.9.3
Utilité	
Type	Logiciel Bureautique
Description	La copie sous Windows n'a jamais été optimale : aucune reprise en cas d'arrêt, de lenteur, ou autres problèmes divers… SuperCopier est un logiciel libre qui améliore la copie de fichiers. Il s'intègre dans le menu contextuel de Windows et ajoute les fonctionnalités suivantes : suspension, reprise des transferts, réglages du taux de transfert, copie de nombreux fichiers (jusqu'à 4 To de fichiers), visualisation et modification de la liste des fichiers à copier en cours de copie, réglage du degré de priorité du processus de copie.
Capture d'écran	
Téléchargement	Ce logiciel est disponible pour toutes les versions de Windows à l'adresse suivante : http://sourceforge.net/projects/supercopier/
Sécurité de l'outil	Aucune faille n'a été publiée à ce jour.
Avis XMCO	Ce logiciel peut être utile pour toutes les grosses copies de fichiers. En effet, la reprise en cas d'échec peut s'avérer intéressante et pratique…

Suivi des versions

Version actuelle des outils libres présentés dans les numéros précédents

NOM	DERNIÈRE VERSION	DATE	LIEN
Debian Sarge	Version stables 3.1 r2	19/04/2006	http://www.debian.org/CD/netinst/
Snort	2.6.11	22/11/2006	http://www.snort.org/dl/
MySQL	5.1.14		http://dev.mysql.com/downloads/mysql/5.1.html
	5.0.27		http://dev.mysql.com/downloads/mysql/5.0.html
	4.1.22		http://dev.mysql.com/downloads/mysql/5.1.html
Apache	2.2.3		http://httpd.apache.org/download.cgi
	1.3.37		http://httpd.apache.org/download.cgi
Nmap	4.2	11/2006	http://www.insecure.org/nmap/download.html
Firefox	2.0	06/2006	http://www.mozilla-europe.org/fr/products/firefox/
Thunderbird	1.5.0.8	11/2006	http://www.mozilla-europe.org/fr/products/thunderbird/
Spamassassin	3.1.7	10/2006	http://spamassassin.apache.org/downloads.cgi?update=200603111700
Putty	0.58		http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
ClamAV	0.88.7	11/12/2006	http://www.clamav.net/stable.php#pagestart
Ubuntu	6.10 Edgy Eft	10/2006	http://www.ubuntu-fr.org/telechargement
Postfix	2.3	06/06/2006	ftp://ftp.club-internet.fr/pub/mirrors/ftp.porcupine.org/postfix-release/index.html
Squid Stable 14	2.5	29/05/2006	http://www.squid-cache.org/Versions/v2/2.5/
Filezilla	2.2.29	1/11/2006	http://filezilla.sourceforge.net/
OpenSSH	5.5	7/11/2006	http://www.openssh.com/
Search & Destroy	1.4		http://www.safer-networking.org/fr/download/index.html
ARPWatch			ftp://ftp.ee.lbl.gov/arpwatch.tar.gz
GnuPG	1.4.6	11/2006	http://www.gnupg.org/(fr)/download/
BartPE	3.1.10a	6/10/2003	http://severinterrier.free.fr/Boot/PE-Builder/
TrueCrypt	4.2a		http://www.truecrypt.org/downloads.php

NOM	DERNIÈRE VERSION	DATE	LIEN
Back-Track	2.0	10/2006	http://www.remote-exploit.org/index.php/BackTrack_Downloads
MBSA	2.0.1	10/08/2006	http://www.microsoft.com/technet/security/tools/mbsahome.mspx
Ps-Exec	1.73	04/12/2006	http://www.microsoft.com/technet/sysinternals/utilities/psexec.mspx
Helios	v1.1a	6/10/2003	http://helios.miel-labs.com/2006/07/download-helios.html
Opera	9.02		http://www.opera.com/download/
Internet Explorer	IE 7		http://www.microsoft.com/windows/ie/downloads/default.mspx
Outils de suppression de logiciels malveillants	1.21	10/10/2006	http://www.microsoft.com/downloads/details.aspx?FamilyID=ad724ae0-e72d-4f54-9ab3-75b8eb148356&DisplayLang=fr
F-Secure Blacklight	Blacklight Beta		http://www.f-secure.com/blacklight/try_blacklight.html
Writely	Writely beta		http://www.writely.com
Nessus	3.0.4	11/2006	http://www.nessus.org/download
Windows Services for Unix	3.5		http://www.microsoft.com/france/windows/sfu/decouvrez/detail.mspx
VNC	4.1.2/4.2.7		http://www.realvnc.com/cgi-bin/download.cgi
Vmware Player	1.0.2		http://www.vmware.com/download/player/
Sync Toy	1.4		http://www.microsoft.com/downloads/details.aspx?FamilyID=E0FC1154-C975-4814-9649-CCE41AF06EB7&displaylang=en
MySQL Front	3.0		http://www.clubic.com/lancer-le-telechargement-9175-0-mysql-front.html
Winscp	3.8.2		http://winscp.net/eng/download.php
Lcc		23/11/2006	http://www.q-software-solutions.de/downloaders/show_download_locations
Cain	2.0		http://www.oxid.it/cain.html
RSS Bandits	1.3.0.42	25/11/2006	http://www.rssbandit.org/
Netmeeting			
OpenOffice	2.1		http://www.download.openoffice.org/index.html
Pspad	4.5.2	20/10/2006	http://pspad.com/fr/download.php
Cygwin	1.5.23-2		http://www.cygwin.com

NOM	DERNIÈRE VERSION	DATE	LIEN
Aircrack	0.6.2		http://airtcrack-ng.org/download.php