Les beaux jours des **pirates** informatiques

Malgré la multiplication des antivirus en tout genre, jamais nos ordinateurs n'ont été aussi menacés par les infections informatiques. Loin de s'améliorer, la situation se dégrade inexorablement...

Éric Filiol
est directeur du laboratoire de virologie et de cryptologie de l'École supérieure et d'application des transmissions.
efiliol@esat.terre.defense.gouv.fr

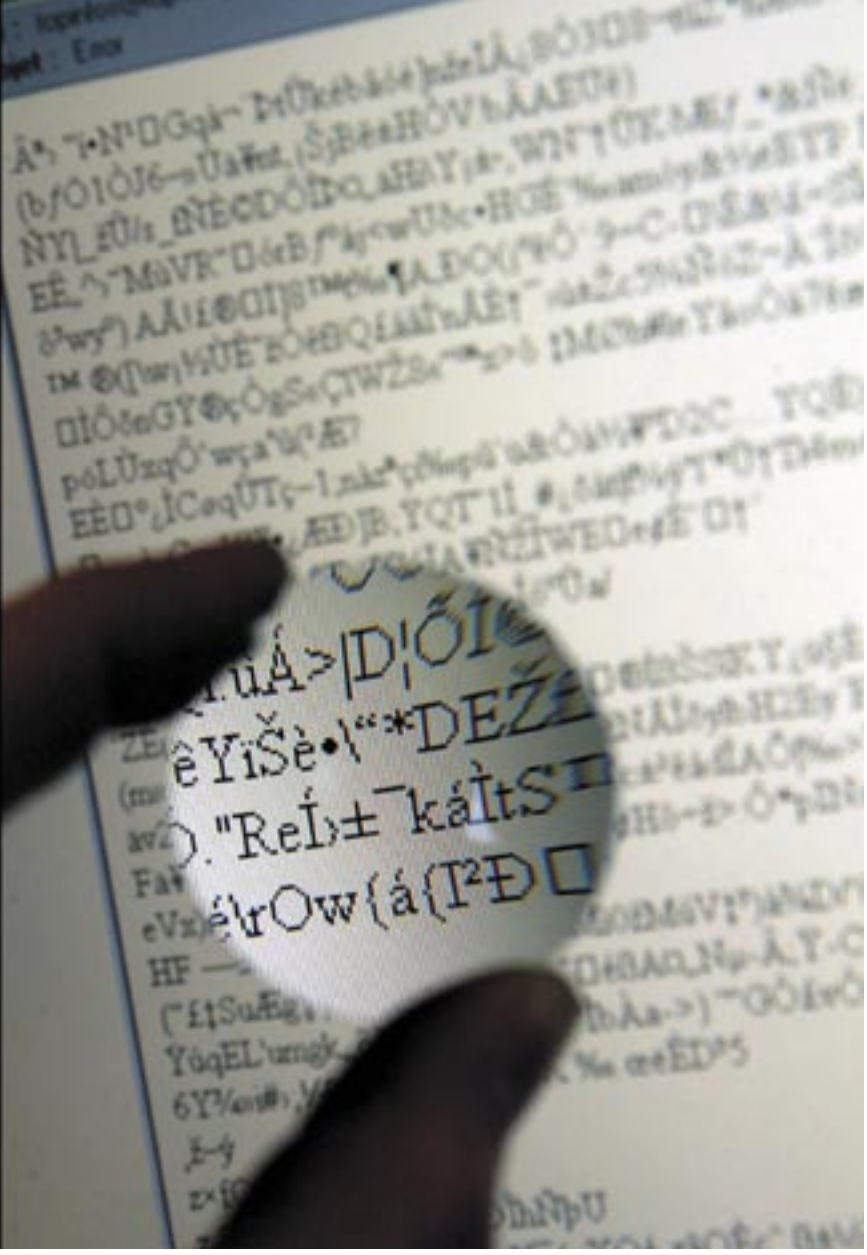
Depuis 1999, le monde est confronté à un nouveau type d'épidémies. Comme pour la grippe, les responsables sont des virus. Mais leur génome est fait de lignes de code et non de matériel génétique, et les victimes sont des ordinateurs et non des êtres vivants.

Plusieurs épidémies ont déjà eu une ampleur planétaire : Melissa (1999), IloveYou (2000), CodeRed, Sircam, Nimda et BadTrans (2001), Klez, BugBear (2002), Blaster (2003) et Sasser (2004). La plupart ont entraîné un fort ralentissement du réseau Internet et ont occasionné des dégâts (pertes de production, heures de travail, etc.) estimés à plusieurs milliards de dollars à chaque fois.

Depuis 2004, ces grandes épidémies semblent avoir disparu. L'activité virale aurait-elle cessé ? Pas du tout : le nombre de nouveaux codes malveillants reste aussi élevé que par le passé – entre 600 et 1 200 par mois. Mais la menace a pris un nouveau visage, se faisant plus discrète et plus pernicieuse. Les infections se propagent plus rapidement [fig. 1], au point que les antivirus sont systématiquement pris de cours (un processus de mise à jour réclame au minimum douze heures). L'accélération est due à de nouvelles techniques de recherche des adresses IP à infecter. À titre d'exemple, citons le ver Slammer. En janvier 2003, il lui a fallu trente minutes pour atteindre les quatre coins de la planète (illustration, p. 85) : selon des résultats obtenus dès l'année suivante, un tel ver aurait besoin, avec ces nouvelles techniques, de seulement quelques secondes pour aboutir au même résultat.

En quoi consiste l'évolution des codes malveillants ? Pour le comprendre, il faut tout d'abord rappeler ce que l'on entend par ces mots. La notion d'infection informatique – héritée du concept de programme autoreproducteur (capable de produire une réplique identique de lui-même) énoncé par le mathématicien américain John von Neumann en 1948 – est née en 1985 des travaux de Fred Cohen, informaticien à l'université de Californie du Sud, aux États-Unis [1]. Considéré comme le père de la virologie informatique moderne, celui-ci est le premier à avoir formalisé et étudié mathématiquement la notion de virus, puis à avoir étudié la détection des codes malveillants.

La principale contribution de Fred Cohen est d'avoir prouvé le résultat fondamental suivant : « *Le problème de la détection virale est un problème généralement indécidable* * » (lire « Le virus contradictoire », p. 84). En d'autres termes, de même que



© 6YS DANNY/PHOTO NEWS/GAMMA

le problème de la quadrature du cercle n'a pas de solution, il est impossible, en pratique, de détecter tous les virus. Un peu plus tard, en 1989, l'informaticien américain Leonard Adleman – à qui l'on doit notamment le terme de virus, inspiré du monde vivant – a généralisé les travaux de Fred Cohen à toutes les formes de codes malveillants [2]. De cette généralisation, on a établi une classification aujourd'hui adoptée dans toute la virologie informatique (lire « La classification des infections », p. 85).

Redoutable complexité

Très peu de travaux théoriques sur la détection des codes malveillants ont été menés par la suite. Quelques classes connues de virus ont été étudiées du point de vue de la théorie de la complexité*, en particulier par Diomidis Spinellis à l'université d'Athènes, en Grèce [3], Zhihong Zuo et Mingtian Zhou de l'université de Chengdu, en Chine [4] et par Jean-Yves Marion et son équipe du Laboratoire lorrain de recherche en informatique et ses applications, à Nancy [5]. Leurs travaux aboutissent tous à une même conclusion : la détection de ces classes de virus est un problème d'une complexité telle qu'un « bon » programmeur parviendra toujours à mettre en défaut un antivirus…

Comment fonctionnent les antivirus actuels ? Il reposent sur « l'analyse de forme », plus connue sous le terme impropre de « recherche de signatures » : on passe au crible le code du virus et non son activité (on analyse la forme et non le fond). Une démarche possible, car les virus ont tous des « signatures » caractéristiques, plus ou moins compliquées. Ces signatures sont consignées dans des bases de données. Conséquence : si le code est inconnu, il ne sera pas détecté.

PASSER AU CRIBLE LE CODE EXÉCUTABLE d'une infection informatique à la recherche d'une signature caractéristique est aujourd'hui la seule technique dont on dispose pour protéger, ensuite, un ordinateur contre cette infection.

Des travaux récents menés dans notre laboratoire ont permis non seulement de modéliser et d'unifier toutes les techniques de détection par analyse de forme, mais également de montrer comment elles peuvent être contournées en pratique [6]. Ainsi, à l'aide de la notion de schéma de détection*, on a montré qu'il suffit de modifier un seul octet pour rendre un code indétectable.

Pour aller au-delà, il faut changer son fusil d'épaule et faire appel à un tout autre concept : « l'analyse comportementale ». Il s'agit ici non plus de se pencher sur le code de l'infection informatique, mais sur ses actions. Cette approche n'est hélas pas encore réellement efficace, car il est très difficile de distinguer un comportement normal du système d'un comportement malicieux. Malgré les dires des éditeurs, elle n'est encore utilisée que de manière marginale [7].

Doit-on se résigner à notre vulnérabilité ? La réalité est plus complexe, et surprenante : les techniques antivirales véritablement efficaces existent, mais elles ne sont pas commercialement viables. Pourquoi ? Parce qu'elles imposent de consacrer peu ou prou toute la puissance de l'ordinateur à l'antivirus… En effet, d'après des résultats théoriques, la détection se décompose en actions ▷

[1] F. Cohen, « Computer Viruses », Thèse de doctorat, université de Californie du Sud, 1985.
[2] L. Adleman, *Advances in Cryptology – Crypto'88, LNCS, 403*, 353, 1989.
[3] D. Spinellis, *IEEE Transactions in Information Theory, 49*, 280, 2003.
[4] Z. Zuo et M. Zhou, *The Computer Journal, 47*, 627, 2004.
[5] G. Bonfante, et *al., Journal in Computer Virology, 1*, 45, 2006.

***Indécidable** se dit d'un problème lorsqu'il n'existe aucun programme permettant de le résoudre.

***La théorie de la complexité** est une discipline de l'informatique théorique qui s'attache à classer les problèmes selon le nombre d'opérations qu'il faut effectuer pour les résoudre.

***Un schéma de détection** consiste en un motif plus ou moins complexe d'octets et en une fonction de détection modélisant la façon de rechercher ce motif.

Fig. 1 La vitesse de propagation des vers

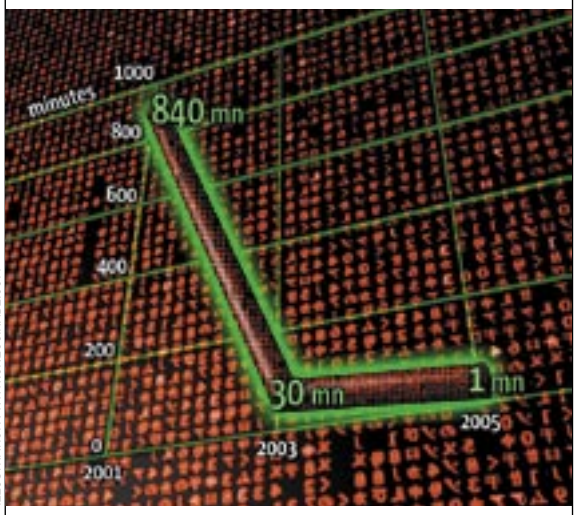


ILLUSTRATION : WWW.GREGCIRADE.COM

UN VER COMME CODERED, en 2001, s'est propagé à l'ensemble du réseau Internet en quatorze heures, alors que Slammer, en janvier 2003, n'aura eu besoin que de trente minutes. Selon des études théoriques conduites en 2004, une minute suffirait avec la technologie « Flash Worm » !

▷ qui sont, pour certaines, d'une complexité trop importante pour être accomplies en un temps acceptable pour l'utilisateur (il s'agit notamment de problèmes issus de la théorie des graphes) [1, 2, 3, 5]. Les éditeurs n'ont d'autre choix que de faire un compromis entre la qualité de la détection et la viabilité commerciale (l'utilisateur ne doit pas se rendre compte de l'activité de son antivirus). Conscients des faiblesses de leurs adversaires, les programmeurs de codes malveillants ont développé des techniques de lutte « anti-antivirale ». Celles-ci reposent sur trois « piliers » : la furtivité, le polymorphisme et le blindage.

Travailler sur la furtivité, c'est permettre au code malveillant de se déployer sans se faire repérer. Il peut faussement déclarer des secteurs du disque dur comme défectueux pour s'y dissimuler : répertoriés comme inutilisables, ils ne peuvent plus être contrôlés par le système d'exploitation. Il peut aussi effacer lui-même sa présence de la liste des processus. Exemple : le virus Stealth, qui « dérobe » de la mémoire vive au système et s'y cache ensuite.

Le polymorphisme est une technique destinée à contourner la recherche de signatures. Le code se transforme en permanence afin de limiter le nombre d'éléments fixes. Ces « mutations » – pour poursuivre l'analogie avec les virus biologiques – se produisent soit *via* des techniques de réécriture (une même action pouvant être réalisée par des codes différents), soit par le chiffrement du code.

Enfin, le blindage consiste à doter le code de fonctionnalités qui vont perturber ou retarder son analyse par l'antivirus. Le code malveillant dispose ainsi de plus de temps pour agir et retarde d'autant la mise à jour de l'antivirus. Un virus comme Whale a nécessité, en 1991, plusieurs jours d'analyse (un code très élaboré pourrait en demander autant aujourd'hui).

Algorithmes élaborés

Ces techniques permettent de contrarier la lutte antivirale, mais l'analyse de code finit toujours par déboucher sur l'identification de caractéristiques qui permettront la détection ultérieure de l'infection. On peut même dire que la plupart des programmeurs de codes malveillants ne se sont pas fatigués jusqu'ici à écrire des codes véritablement sophistiqués (sans compter qu'il est courant qu'ils fassent des erreurs algorithmiques). Les antivirus sont, eux aussi, de faible niveau technique [6,7]. Le duel « épée contre bouclier » a donc été jusque-là relativement équilibré, à la mise à jour de l'antivirus près. Mais les programmeurs adoptent de plus en plus une véritable démarche intellectuelle et font appel à des techniques algorithmiques et des résultats mathématiques parmi les plus élaborés. La situation risque donc de changer rapidement – elle a peut-être même déjà changé – avec l'apparition de nouvelles techniques virales.

En quoi consistent-elles ? On a récemment identifié des techniques liées à des cas particuliers difficiles, pour lesquelles il n'existe aucun algorithme efficace, voire indécidables. La seule parade est donc d'éviter que ces codes ne parviennent à infecter le système… Pour l'heure, aucune de ces nouvelles infections virales n'a encore été observée « dans la nature » : nous en avons fabriqué dans notre laboratoire de recherche où elles restent confinées, et nous ignorons si d'autres équipes en ont aussi obtenu.

On peut les classer en quatre grandes catégories (certaines infections appartenant à plusieurs d'entre elles). Il y a tout d'abord les codes qui agissent « en groupes ». Les fonctions virales ne sont plus concentrées dans un seul code, mais réparties sur plusieurs codes différents. Cette « segmentation » permet de contourner tous les antivirus. Des études menées au laboratoire de virologie et de cryptologie ont démontré toute l'efficacité de cette approche, avec des codes « preuve de concept » (créés à des seules fins de recherche puis détruits), comme POC_Serial ou POC_Parallèle_4.

Il y a ensuite les codes « métamorphes ». Le métamorphisme est semblable au polymorphisme

LE VIRUS CONTRADICTOIRE

© ASP

Pour illustrer sa démonstration de l'indécidabilité du problème de la détection virale, Fred Cohen (ci-contre) imagina un virus qu'il baptisa « virus contradictoire ». De quoi s'agit-il ? Une procédure de détection D (autrement dit, un antivirus) détermine si un programme P est infecté ou non. Le code *virus_contradictoire* fait appel à D pour l'appliquer sur lui-même. Si D répond « non infecté », alors le virus lance l'infection. Il y a contradiction, d'où l'indécidabilité.

La classification des infections

Parmi les infections informatiques, on distingue tout d'abord les « codes simples » (parmi lesquels les logiciels espions ou spywares) des « codes autoreproducteurs ». Les infections simples regroupent les codes qui s'installent dans un système et s'y dissimulent avant de passer à l'attaque. Plusieurs cas peuvent se présenter. Soit l'attaque débute lorsqu'une condition particulière est remplie (par exemple, une date et une heure) ou une action est réalisée (envoi d'un fichier) ; on parle alors de « bombe logique ». Soit le code va ouvrir une porte cachée sur l'extérieur pour faciliter la pénétration du système et son attaque ultérieure ; on parle alors de « cheval de Troie ».

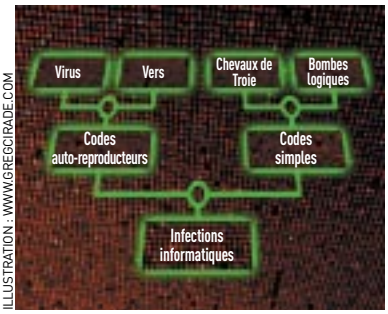


ILLUSTRATION : WWW.GREGCIRADE.COM

La catégorie des codes autoreproducteurs comprend les virus et les vers. Ils ont la capacité d'infecter et d'infester un système en multipliant les copies d'eux-mêmes. Les vers sont des virus capables d'utiliser un réseau pour se propager *(via* différents protocoles comme TCP/IP, la messagerie, les connexions peer-to-peer, etc.), ce qui leur permet d'infecter plusieurs millions de machines en quelques minutes.

à la différence que les procédures de mutation varient elles aussi. Le code est donc capable de se réécrire dans sa totalité. Par exemple, W32.Metaphor, preuve de concept publiée en 2003.

Troisième technique, la furtivité à base de machines virtuelles. Des travaux récents menés par les laboratoires de Microsoft et de l'université du Michigan, aux États-Unis [8], ainsi que ceux de Joanna Rutkowska à Singapour, en Asie [9], ont permis de montrer qu'il est possible d'être totalement invisible dans un système. Le principe général des technologies développées (respectivement SubVirt et BluePill), consiste à basculer le système d'exploitation, à son insu, dans une machine virtuelle, c'est-à-dire que le code malveillant prend le contrôle d'une émulation de la machine sans que l'utilisateur ne s'en rende compte.

Quatrième technique : le blindage total. Il permet d'interdire l'analyse de code et non plus simplement de la retarder. Un code comme Bradley a prouvé son efficacité par des techniques de chiffrement et de gestion de clé de chiffrement sophistiquées et dans un contexte d'attaques relativement ciblées [10].

Communauté artisanale

Malgré la médiatisation de la « menace virale », celle-ci semble promise à des jours tranquilles, car, depuis vingt ans, aucune recherche digne de ce nom n'existe dans ce domaine. La communauté antivirale a directement œuvré pour qu'elle reste l'apanage des sociétés éditrices de logiciels antivirus. Et ces dernières se sont contentées d'analyser les codes malveillants identifiés et de mettre à jour inlassablement leurs produits.

Toute autre recherche, notamment universitaire a été marquée du sceau de l'opprobre, et publier un quelconque résultat significatif est encore de nos jours relativement difficile. Ainsi, en 2002, lorsque l'université de Calgary, au Canada, a tenté de créer un cours de virologie, la communauté antivirale a usé de tellement de pressions que ce cours n'a jamais vu le jour. Aujourd'hui, on peut dire que les concepteurs de codes malveillants disposent d'une avance conséquente. En particulier, ils ont « diversifié » les cibles de leurs attaques : téléphones mobiles (près de 300 codes à ce jour), pocketPC, consoles de jeux... tout ce qui est assimilable à un ordinateur est potentiellement infectable par un code malveillant. ■ É. F.


© 2003 THE REGENTS OF UNIV. OF CALIFORNIA D.R.

LE VER SLAMMER a touché toute la planète, en 2003. Le diamètre des cercles bleus traduit (en échelle logarithmique) le nombre de machines infectées trente minutes seulement après le début de l'épidémie.

POUR EN SAVOIR PLUS

▷ Éric Filiol, *Les Virus informatiques : théorie, pratique et applications*, collection IRIS, Springer France, 2004.
▷ Éric Filiol et Philippe Richard, *Cybercriminalité – Enquête sur les mafias qui envahissent le Web*, éditions Dunod, 2006.
▷ Éric Filiol, *Techniques virales avancées*, collection IRIS, Springer France, janvier 2007.

[6] E. Filiol, *Journal in Computer Virology, Eicar Conference 2006 Special Issue*, V. Broucek et P. Turner (dir.), *2*, 35, 2006.
[7] E. Filiol, *et al.*, *Journal in Computer Virology*, TCV 2006 Special Issue, G. Bonfante et J.-Y. Marion *(dir.)*, 3, à paraître.
[8] S. Kind et *al.*, www.eecs.umich.edu/virtual/papers/king06.pdf, 2006.
[9] J. Rutkowska, SysCan 2006 Conference, http://invisiblethings.org/papers.html, 2006
[10] E. Filiol, *Proceedings of the 14th EICAR Conference*, 216, 2005.