L'évolution des idées en virologie informatique

Éric Filiol

École Supérieure et d'Application des Transmissions
Laboratoire de virologie et de cryptologie
B.P. 18, 35998 Rennes - France

Mots-cles : virologie informatique, virus, antivirus, sécurité informatique.

1 Introduction

La sécurité concerne l'ensemble des techniques et mesures destinées à envisager, gérer et contrer les atteintes de nature malveillante, contre un système. La notion de « nature malveillante » décrit, en fait, la capacité d'adaptation de l'attaque à la défense. Cette définition générale s'applique dans tous les domaines et pour les systèmes confrontés aux attaques.

Concernant les systèmes d'information et de communication, la virologie informatique est probablement le domaine scientifique et technique qui illustre le mieux cette problématique générale de la sécurité informatique : le fameux duel lance-cuirasse.

Les évolutions techniques en matière de virus ont toujours été directement dictées par l'évolution des techniques antivirales, lesquelles en réaction ont toujours été contraintes d'évoluer en fonction du risque viral lui-même. Ainsi la naissance du polymorphisme, la plus connue des techniques virales, est née de la volonté de contourner les techniques de détection par signatures. A travers cet exemple, presque caricatural maintenant, c'est l'histoire toute entière de la virologie informatique qui s'y trouve résumée.

L'histoire de la virologie est malheureusement encore très lacunaire tant dans le fond que dans la forme, et ceci pour plusieurs raisons :

— la virologie informatique est un domaine dans lequel tout n'est pas publié ni médiatisé et quand publication il y a, c'est le plus souvent de manière clandestine (le fameux « *underground* ») voire détournée, par la diffusion d'une

infection informatique. De plus, établir une chronologie, dater des événements est difficile dans la mesure où le laps de temps s'écoulant entre la naissance d'une technique et sa première concrétisation par une infection connue, ce laps de temps peut être plus ou moins grand.

— les quelques tentatives de créer une histoire de la virologie [5, 10, 12] se réduisent à énumérer de manière fragmentaire, les exemples les plus connus, les plus médiatisés, de virus informatiques. En outre, cette perspective historique se fait toujours linéairement, comme une succession sans véritable fil conducteur qui permettrait d'avoir une vision cohérente de ce domaine et des axes de réflexion susceptibles de permettre une meilleure perception des enjeux futurs.

— la virologie informatique souffre d'un déficit d'image : les virus sont perçus comme des programmes obligatoirement dangereux et leurs concepteurs, voire ceux qui les étudient, vus comme des acteurs constestables d'une science qui ne le serait pas moins. La récente évolution de la loi française [4, article 46] en est la meilleure illustration. La conséquence a été que, marquée du sceau de l'opprobe, la virologie informatique est privée d'une partie du débat technique : celui traitant de l'« attaque ». Amputée de cet aspect pourtant fondamental, toute tentative de réflexion historique s'en trouve forcément amoindrie.

— Enfin, et ce n'est pas là la moindre des raisons, la virologie informatique est très vite devenue un enjeu économique et commercial où les aspects techniques et recherche, quand ils ne sont pas considérés comme quantité négligeable, sont le fait quasi-exclusif des sociétés éditrices d'antivirus. Ces dernières, dont le principal souci est de conserver le monopole technique de cette science, tout en tentant de survivre dans un secteur où la guerre économique fait rage, ne manifestent aucun intérêt pour une vision épistémologique de la virologie informatique, se contentant d'énumérer de manière chronologique la suite sans fin des virus et autres codes malveillants qui frappent jour après jour.

Les premières tentatives de formalisation de la virologie informatique, par Fred Cohen, en 1986 [1], postérieures à l'apparition des premiers virus – officiellement en 1981– ont esquissé, de manière très sommaire et sur le seul plan technique, l'évolution à attendre dans ce domaine de connaissance, entre la défense et l'attaque. Le résultat théorique certainement le plus important de la virologie informatique montre que l'interdépendance entre ces deux facettes est totale.

Le but de cet article est de présenter les évolutions techniques de la virologie depuis vingt cinq ans, en la replaçant dans cette perspective et en considérant les exemples les plus marquants. Cela permettra de montrer comment une « vision historique » de la virologie informatique peut aider la lutte antivirale. Dans une première partie, nous présenterons ce que fut la « préhistoire » de la virologie puis dans la section 3,

nous verrons comment avec l'apparition du polymorphisme, de la furtivité et des techniques de blindage de code, défense et attaque se sont organisées l'une contre l'autre. Dans une troisième partie, nous présenterons ce que l'on peut considérer comme la seconde « révolution culturelle » de la virologie informatique, avec l'apparition des virus dits « de documents ». Enfin, nous verrons comment l'évolution des systèmes informatiques vers le « tout réseau » a vu presque simultanément et tout naturellement l'apparition des vers informatiques.

2 Préhistoire : des origines à Fred Cohen (1936 - 1986)

Le terme de virus informatique est né, curieusement, à la fin de ce que l'on peut considérer comme la préhistoire de la virologie informatique, en 1984. Ce terme est dû, officiellement, à Fred Cohen, à l'époque chercheur à l'université de Californie du sud et qui préparait sa thèse sur ce sujet.

L'histoire avant cette date est assez floue et aucune mention significative de codes viraux ou assimilés n'est présente. Fred Cohen est souvent cité, officiellement, comme le premier auteur, en 1983, d'un code viral. Rien n'est moins sûr, tant l'Histoire dans ce domaine et à cette époque est pauvre en sources ouvertes. Quelques informations parcellaires et vagues (dont une allusion claire dans les remerciements de Fred Cohen en préambule de sa thèse) semblent accréditer le fait qu'en réalité, au moins aux Etats-Unis, une réflexion sérieuse était déjà menée dans ce domaine, et en particulier par le Pentagone. Cela n'a rien détonnant quand on sait que les deux autres pères fondateurs, Alan Turing et John von Neumann, ont largement été impliqués à la fois dans les fondements de la virologie informatique et des projets classifiés[1]

L'apparition du terme virus a déclenché, en quelque sorte, les hostilités et a mis ce domaine de connaissance sous les « feux de la rampe », lui donnant par là même un statut sulfureux. Avant ce coup de « marketing » aux effets désastreux, près de cinquante ans de réflexion et de formalisation sereines ont permis de jeter les bases de la virologie informatique, même si pendant cette période, des préoccupations scientifiques différentes ont présidé à ces travaux.

2.1 Alan Mathison Turing (1912 - 1954)

Aucune histoire de l'informatique, ou de l'une quelconques de ses branches, ne serait complète sans la mention des travaux d'Alan Turing 1. Ils seront le point de

1. Alan Turing a, pendant la seconde guerre mondiale, largement et significativement contribué à la cryptanalyse de la machine Enigma et von Neumann, entre autres projets, au projet Manhatttan, consacré à la réalisation de la bombe atomique américaine.

FIGURE 1 – Alan Mathison Turing (1912 - 1954)

départ et le passage obligé de tous les travaux ultérieurs menés dans cette discipline. Nous ne rentrerons pas dans les détails techniques – le lecteur consultera [3] pour une introduction plus complète des travaux de Turing et [13] pour l'article original – et l'apport de Turing, pour ce qui nous concerne, peut se résumer aux points suivants :

— la formalisation des mécanismes viraux utilise essentiellement la notion de *machine de Turing*. Une machine de Turing est la représentation abstraite et générale d'un ordinateur et des programmes susceptibles d'être exécutés sur cet ordinateur. Ce modèle théorique a permis de répondre à de nombreux problèmes fondamentaux parmi lesquels :

  – Soit une fonction f donnée. Cette fonction est-elle « effectivement » calculable ? En d'autres termes, existe-t-il un algorithme permettant de réaliser, de calculer f ?

  Pour ce qui nous intéresse – les virus informatiques – la fonction f est celle de l'*auto-reproduction*. Un programme peut-il se reproduire lui-même ? Les travaux de Turing et ceux de ses exégètes ne se sont pas intéressés à ce problème particulier

— là où une machine de Turing ne traite que d'un seul problème particulier (il y a donc autant de machine de Turing que de problèmes), Turing a ensuite défini la notion de *machine de Turing universelle* – en d'autres termes, une

machine capable de décrire toutes les autres machines, ou de manière équivalente, tous les autres problèmes[2].

— le problème de l'arrêt (*décidabilité*). Il s'agissait là d'étudier de « calculabilité effective », autrement dit à quelles conditions un programme s'arrête-t-il et produit-il un résultat. Une fonction (correspondant à un problème donné) effectivement calculable est appelée une *fonction récursive*.

Ce dernier est fondamental pour notre propos car il pose d'emblée le problème de fond de la virologie informatique : les programmes viraux sont des programmes qui « s'arrêtent » (programmes décidables) alors qu'il n'existe aucune machine s'arrêtant, dédiée au problème général de la détection antivirale.

2.2 Stephen Cole Kleene (1909 - 1994)

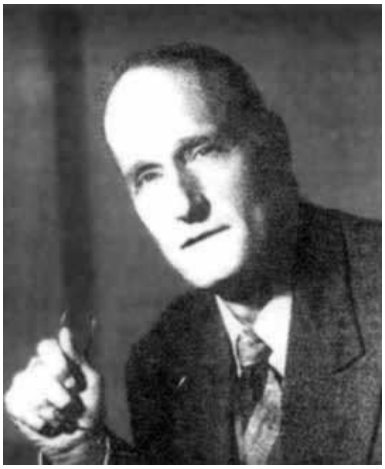

FIGURE 2 – Stephen Cole Kleene (1909 - 1994)

Les travaux de Kleene concernent la logique mathématique [6, 7]. Cependant, un de ses théorèmes, connu sous le nom de *théorème de récursion de Kleene* et qui date de 1938, en a fait indirectement l'un des pères fondateurs de la virologie informatique. Ce résultat est en fait la première formalisation de la notion d'auto-reproduction pour un programme informatique. Et l'auto-reproduction est une propriété inhérente à tout virus.

2. La notion de machine de Turing universelle préfigure celle de générateurs de virus qui feront leur apparition en 1992 avec des programmes comme *Virus Creation Lab*, suivi de bien d'autres par la suite.

Plus précisément, le théorème de récursion de Kleene indique pour une même fonction f, il existe plusieurs programmes, de codes différents, permettant de la calculer. Si la fonction f est la fonction Identité ($f(x) = x$), nous avons même des codes identiques, et de là, implicitement, la notion d'auto-reproduction, autrement dit, de virus simple. Pour toute fonction f, différente de l'identité, le théorème de récursion décrit simplement le mécanisme de polymorphisme, près de 50 ans avant les travaux de Cohen, et sa première réalisation pratique au début des années quatre-vingt dix.

2.3 John von Neumann (1903 - 1957)

FIGURE 3 – John von Neumann (1903 - 1957)

L'apport de von Neumann a été capital pour la virologie informatique et ce, encore une fois, bien avant que le terme ne naisse. Génie dans de nombreux domaines, père de l'un des premiers ordinateurs, l'ENIAC, ce mathématicien a surtout prouvé de manière pratique qu'un programme pouvait être auto-reproducteur.

La théorie des automates cellulaires[3] est née en 1948 de la tentative de von Neumann de trouver un modèle réductionniste pour décrire les processus d'évolution biologique, en particulier celui de l'auto-reproduction [14, 15].

3. Le terme *cellulaire* tire son origine des travaux de von Neumann, qui a considéré pour ces objets un plan divisé en cellules carrées, chacune contenant un automate fini.

Après avoir défini et analysé de manière théorique son modèle, von Neumann l'a réalisé en pratique. Von Neumann s'est posé la question de savoir s'il était possible de construire effectivement une « machine » auto-reproductrice, capable de construire, sans perte de complexité, d'autres machines, et en particulier elle-même.

Von Neumann pensait qu'il devait exister un algorithme permettant de décrire les mécanismes complexes (biologiques et biochimiques) d'une « machine biologique » donnée. Si un tel algorithme existe, il en est de même, par conséquent, pour une machine de Turing universelle permettant de le réaliser, autrement dit, en corollaire de s'auto-reproduire. A l'inverse, si des machines de Turing universelles existent, alors, il en a déduit que les mécanismes du vivant sont descriptibles par des machines.

L'automate complet final, réalisé par von Neumann est extrêmement complexe et nécessite plusieurs dizaines de pages pour être décrit. C'est la première réalisation pratique d'un programme se reproduisant. En 1984, Fred Cohen nommera cela, à juste titre, un virus.

2.4 Fred Cohen

FIGURE 4 – Fred Cohen

L'œuvre de Fred Cohen, et en particulier sa thèse [1], contient à elle toute seule l'histoire de la virologie informatique. Cela fait de lui à la fois un père fondateur et un visionnaire de la virologie informatique. Une grande partie de ce que nous connaissons aujourd'hui dans ce domaine a été envisagé (quelquefois très succintement, il est vrai). Parmi ses nombreux résultats, citons les plus importants :

— définition rigoureuse de la notion de virus,
— définition théorique du polymorphisme,
— étude et expérimentation de la propagation virale,
— définition de modèles de sécurité,
— et surtout démonstration mathématique de l'indécidabilité du problème général de la détection virale.

Ce dernier résultat est fondamental car il prouve que toute protection antivirale peut être contournée. Le duel lance-cuirasse pouvait commencer. Toute la suite depuis cette époque ne sera que l'illustration de ce théorème fondamental.

3 Antivirus contre virus : une lutte sans fin (1986 - 1996)

Le duel virus/antivirus ne s'est pas instauré dès l'apparition des premiers virus[4], au début des années quatre-vingts. L'informatique était peu diffusée et réservée au monde de la recherche et de l'industrie. Aussi l'apparition des premiers virus ne pouvait que rester assez discrète. Perçus plus comme des exercices de style, il est fortement probable que les premiers virus n'aient pas été considérés comme une menace vraiment importante, du moins jusqu'aux années quatre-vingt dix. De manière logique, les premières antivirus n'ont été eux-mêmes que des exercices de style pour répondre à un challenge purement intellectuel. Cela explique pourquoi pendant les années quatre-vingts, les logiciels antivirus étaient soit des *freeware* ou des *shareware*. L'intérêt économique de la lutte antivirale était loin évident. Ce n'était clairement pas la motivation initiale.

Les choses commencèrent à changer dès 1988, année lors de laquelle, les premiers éditeurs d'antivirus firent leur apparition. Cependant, les virus, en nombre encore assez réduit, n'étaient toujours pas perçus comme une véritable menace[5] et les ventes de logiciels antivirus ne décollaient pas – surtout si l'on considère qu'une offre gratuite non négligeable de ces produits existait encore.

C'est dans ce contexte que le duel virus/antivirus s'est progressivement développé. Il est intéressant de noter que certains des premiers virus, dès 1987, comportaient déjà des fonctionnalités destinées à contrer les premières techniques antivirales connues. Il est probable que les auteurs de ces virus aient pris connaissance des travaux de Fred Cohen, ou à défaut aient été conscients de l'existence de techniques antivirales, mêmes simples. Il n'est pas exagéré d'affirmer que les premiers auteurs

4. A part le virus *Apple* en 1981, peu d'exemples de virus « dans la nature » sont connus. Il faudra attendre les années 1987/1988 pour voir apparaitre des virus, mais encore en très petit nombre.

5. Il faudra attendre l'année 1991 pour que commence à naître une certaine psychose, certes créée et entretenue par certains éditeurs d'antivirus, non sans succès. Pour plus de détails, lire [11, Chap. 1].

de virus ont simultanément imaginé les premières techniques antivirales en cherchant à les contourner.

Au tout début, les techniques de détection se résumaient à l'analyse du virus (le plus souvent après avoir obtenu le code source directement auprès des auteurs ou d'autres programmeurs de virus [11]). Cette analyse permet alors d'isoler une chaine binaire identifiant spécifiquement le virus : la signature. Il s'agit en quelque sorte d'une « empreinte virale » analogue aux empreintes digitales rechercheées sur le lieu d'un crime.

3.1 La furtivité

C'est la plus ancienne des techniques utilisées par les virus pour contrer les antivirus. Dans un premier temps la furtivité était essentiellement de nature statique. Dans la mesure où les antivirus recherchaient une signature, les virus ont très vite cherché à échapper au parcours de leur code en le cachant dans des zones soit inaccessibles nativement à l'antivirus (par exemple, la piste 0 de la tête 0 du disque dur, entre autres possibilités) soit ignorées par lui (secteurs défectueux du disque dur), ou encore en modifiant certaines données systèmes susceptibles de trahir l'action du virus (la taille et la date de dernière modification du fichier infecté, par exemple). L'exemple le plus représentatif est le virus *Brain*, apparu en 1987.

Très rapidement, les antivirus se sont adaptés et au fur et à mesure ont pris en compte les techniques furtives statiques au fur et à mesure qu'elles étaient identifiées. De plus, les antivirus, pour tenter de prendre un temps d'avance face à l'impossibilité de prévoir toutes les signatures éventuelles, ont cherché à détecter les virus non plus seulement par empreinte mais lors de leur action. En effet, certains événements ou actions sont le fait exclusif ou plus fréquent de virus.

En réponse, très vite, les virus ont opté pour des techniques de furtivité dynamiques consistant à créer des jeux de miroirs ou des systèmes de double-fond pour leurrer les antivirus en leur renvoyant l'image d'un système sain. Un bon exemple est le virus *Stealth*, créé en 1991 par Mark Ludwig.

Les techniques furtives depuis n'ont cessé de se développer et sont toujours actuelles, mêmes les innovations dans ce domaine sont de nos jours moins fréquentes. La lutte antivirale repose sur la capacité à analyser très rapidement un code viral novateur, à en comprendre le fonctionnement et à modifier l'antivirus en conséquence.

3.2 Le polymorphisme

Les techniques furtives tentent de dérober le code viral à l'analyse de l'antivirus qui y cherche une signature. L'idée de code évolutif – idée évoquée de manière théo-

rique et rapide, par Fred Cohen dans sa thèse – a très vite été mise en pratique pour contrer l'analyse directe du code viral. L'idée est cette fois de faire varier le code viral, lors de l'infection, de manière fréquente de sorte à interdire ou au minimum à très fortement limiter la présence d'éléments fixes comme une signature : le polymorphisme viral entre en scène.

Le premier exemple connu date de 1990 avec quatre variantes du virus *Vienna*, créée par Mark Washburn, suivi en 1991 du virus suisse *Tequila*[6]. Mais cette tentative, utilisant le chiffrement de la plus grande partie du virus, était encore timide, quoique intéressante. La détection ne pouvait qu'utiliser la procédure de chiffrement/déchiffrement, en clair en tête du code. Mais cette dernière prend un très grand nombre de formes possibles, rendant la détection, à l'époque, très difficile et génératrice d'un grand nombre de fausses alarmes. A la fin de 1991, une quinzaine de virus polymorphes avaient été identifiés.

L'année 1992 fut un véritable cauchemard pour les concepteurs de logiciels antivirus. Ce fut l'année où un pirate bulgare surnommé *Dark Avenger*[7] fit « passer à l'ouest » le premier moteur de mutation dénommé *MtE* (*the Mutation Engine*), suivi très rapidement d'un second *Commander bomber*. Compilé avec le code d'un virus, ce moteur génère une forme « mutée » du virus. L'auteur annonçait près de quatre milliards de formes possibles. La réponse antivirale se fit assez rapidement et le détecteur du moteur *Mte* permettait de détecter également toutes les formes mutées qui en étaient issues.

Très vite, de nombreux autres moteurs de mutation firent leur apparition (*TPE, NED, DAME*). Plus évolués, plus efficaces, ils réussirent beaucoup mieux là où *MtE* avait échoué. La détection se révéla beaucoup plus difficile et bien moins fiable (notamment en provoquant un nombre de fausses alarmes plus élevé). De nombreux virus apparurent, générés à partir de ces moteurs. Ces moteurs accrurent le risque dans la mesure où ils étaient utilisables par de simples programmeurs de virus, sans les compétences techniques nécessaires pour maitriser le polymorphisme. Il en résulta un effet démultiplicateur grâce à ce polymorphisme « prêt à utiliser ». Parallèlement, d'autres virus, dotés de capacités polymorphes très évoluées continuèrent de frapper[8].

6. *Tequila* était non seulement polymorphe mais aussi furtif.

7. Une hypothèse récurrente mais non prouvée affirme que *Dark Avenger* était en fait un membre des services spéciaux bulgares.

8. L'un des plus célèbres est le virus *Tremor* qui s'est propagé essentiellement en Allemagne, via une chaine TV satellite.

3.3 Le blindage

Le polymorphisme, à partir de la moitié des années quatre-vingt dix, est relativement bien géré mais toujours avec retard. L'analyse du virus, elle-seule, permet de comprendre comment s'opère le processus de mutation et ensuite d'améliorer l'antivirus. Comme pour la furtivité, tout est une question de rapidité de réaction.

En 1990, apparut le virus *Whale* qui illustra une nouvelle approche dans la lutte anti-antivirale : le blindage. Les succès des antivirus, jusque là, résidaient sur la capacité à disposer d'au moins un fichier infecté, à l'analyser après une phase de désassemblage et à en comprendre le fonctionnement. Les techniques de blindage viral ont pour but de rendre sinon impossible du moins très complexes les opérations de désassemblage et d'analyse. Au minimum, il s'agit de retarder le plus possible la lutte afin de favoriser la propagation du virus.

Si les techniques de blindage initiées par le virus *Whale* se révélèrent finalement inefficaces, elles ont montré le chemin et initiée une réflexion à ce sujet. D'autres virus utiliseront par la suite (*StarShip* par exemple) et utilisent encore – plus ou moins efficacement– des techniques de blindage viral.

4 Les virus de documents : une révolution culturelle (depuis 1996)

Le duel lance-cuirasse évoqué dans la section précédente a fini par se traduire par une certain équilibre où l'attaque conserve une légère avance et un petit avantage sur la défense. Cette situation de *statu quo*, somme toute, convenait assez bien aux éditeurs d'antivirus, qui parvenaient à gérer la menace tout en assurant leurs profits. L'antivirus est devenu un rempart indispensable à tout utilisateur soucieux de sa sécurité. L'analyse de chaque virus contenait la solution au problème.

Les attaquants, eux, ne se sont pas satisfaits de cette situation et la volonté de nuire ou le goût du challenge, selon les motivations, réclamait d'innover pour mettre à mal les protections antivirales tout en accroissant le risque. Bref, il s'agissait de bouleverser ce *statu quo* obtenu au milieu des années quatre-vingt dix – même si les techniques présentées précédemment sont restées d'actualité, y compris de nos jours.

En 1996, il se produisit alors une véritable récolution culturelle dans le monde de la virologie informatique qui amplifia considérablement le risque. Jusqu'à cette époque, la notion de virus (ou autre infection informatique) était indissociablement attachée à celle de programme exécutable. Autrement dit, tout code mal-

veillant ne peut se propager et s'exécuter que par l'exécution d'un code binaire compilé[9].

Or, dans un environnement informatique, il existe d'autres types de fichiers ou de données et notamment les documents selon différents formats (texte simple, postscript, format bureautique type word...). Au milieu des années quatre-vingt dix, ces fichiers documents, n'étant pas exécutables – du moins en apparence – ne représentaient aucun risque. La notion de virus infectant des documents et se propageant par eux étaient considérée comme un mythe voire un canular.

En réalité, l'évolution de l'informatique vers toujours plus d'ergonomie avait modifié grandement la situation. Deux éditeurs de logiciels ont notamment largement contribué à cette petite révolution : *Microsoft* avec ses logiciels bureautiques comme *Word* ou *Excel* et *Adobe* avec un langage comme le Postscript. Peu à peu, la frontière entre exécutable pur et document pur s'est progressivement estompée : la plupart des formats de documents incorporent depuis cette époque – et cette évolution n'a jamais cessé – des fonctions, transparentes le plus souvent, d'exécution (l'exemple le plus connu est le langage *Visual Basic pour Applications* présent nativement dans les applicatifs se la suite bureautique *Office* ; nous pourrions également citer les langages de scripts comme le VBS qui donneront un second souffle aux virus de documents au tout début des années 2000). Pratiquement tous les formats sont désormais concernés. La facilité d'apprentissage de la plupart des langages attachés[10] et la propension à échanger en un nombre sans cesse croissant de documents ont considérablement accru le risque viral.

Le problème est que cette évolution, au début, n'a pas été perçue comme un facteur de risque viral. Aucune réflexion n'avait été menée et l'apparition du premier virus de document pour *Word* en 1996, le virus *Concept*, a provoqué une véritable révolution dans le monde de la sécurité informatique[11]. La simple ouverture de ce qui paraissait n'être qu'un simple document provoquait l'infection de l'applicatif et de là, celle de tout document produit ou consulté par son intermédiaire. L'action du virus – des successeurs de *Concept* comme le virus *Colors*, l'ont montré – peut dépasser les limites de l'applicatif pour agir sur la totalité de l'environnement.

9. La compilation est l'opération permettant de produire à partir d'un code source, écrit dans un langage compréhensible par un être humain, un code dit binaire compréhensible par la machine. Dans le cas d'un code dit *interprété*, le code source est exécuté directement par l'intermédiaire d'un *interpréteur*.

10. Quiconque ayant appris l'assembleur ou connaissant un peu ce langage ne peut que reconnaitre qu'en comparaison des langages comme le *Visual Basic for Applications* sont faciles à maitriser. Il a donc fallu peu de temps à un plus grand nombre de programmeurs d'accéder à un langage plus « ergonomique » permettant de réaliser de manière efficace des virus.

11. En réalité, il semblerait que *Concept* n'est pas été le premier. Un virus pour AMI-Pro en janvier 1996 et un autre pour le logiciel *Lotus-1-2-3* sont quelquefois mentionnés comme légérement antérieurs. L'Histoire n'a retenu que *Concept*, apparu en mars 96.

Depuis le cas du virus *Concept*, la menace a été prise en compte mais de manière encore très imparfaite – même de nos jours, le plus souvent en recyclant les anciennes techniques antivirales – alors que les auteurs de virus rivalisent d'ingéniosité pour exploiter la moindre des fonctionnalités susceptibles de permettre un processus d'exécution, aussi limité soit-il. A l'exception de formats vraiment inertes (**.txt, *.rtf, *.csv*...), le risque viral dit « de document » n'a jamais été aussi grand. En exploitant l'ergonomie procurée par ces fonctionnalités, et de fait le refus des utilisateurs de s'en priver, les pirates ont bousculé le *statu quo* obtenu au début des années quatre-vingt dix et accru de manière sensible leur avantage. Le retour a une situation plus équilibrée entre l'attaque et la défense semble encore loin.

5 Apparition des vers : le nombre et la vitesse pour arme (depuis 1999)

Les infections informatiques jusque-là visaient uniquement les postes informatiques (clients ou serveurs). Le risque infectieux étaient par conséquent localisé à un pays ou une région. Il s'agissait d'endémies localisées et non d'épidémies à l'échelle mondiale. Cette situation concourrait à rendre la lutte antivirale relativement aisée. De plus, la progression d'un virus était relativement lente[12]

L'émergence des réseaux dans le grand public date à peu près de la fin des années quatre-vingt dix. Très vite, les programmeurs de virus ont compris l'intérêt de ce nouvel environnement pour propager leurs créations et ainsi créer des difficultés nouvelles, et de taille, aux antivirus, lesquels ont également mis un certain temps s'adapter, d'une manière qui pourrait être, encore de nos jours, jugées non satisfaisantes. Les vers ont fait leur apparition.

La technologie des vers ne date cependant pas de 1999 et quelques exemples, certains très célèbres, doivent être cités : le ver Xerox[13] (expérience de Scoch et Hupp [9]) en 1982, le ver *Christma Exec* en 1987, le ver Internet de 1988, le ver *Wank* de 1989. Mais ces exemples sont restés, du moins à l'époque de leur apparition, limités au monde professionnel et aux utilisateurs, peu nombreux, de réseaux ; ainsi ont-ils connu peu ou prou de médiatisation sur le moment.

Il a fallu attendre en réalité une dizaine d'années – quand les réseaux sont devenues accessibles à tous – pour voir apparaitre le véritable risque lié aux vers informatiques : vers de type simple exploitant des failles logicielles de sécurité, ou virus

12. Une exception notable est le virus *Concept* qui s'est très vite répandu [2]. Sa propagation rapide est due à la commercialisation de trois CDROMs édités Microsoft et infectés par *Concept*.

13. Le ver Xerox doit être considéré à part car il ne s'agit pas, contrairement aux autres, d'une attaque mais d'une expérimentation contrôlée en laboratoire de calcul distribué sur un réseau.

visant une ou plusieurs applications orientées réseau (macro-vers ou vers de courrier électronique) exploitant soit l'ignorance ou la naïveté des utilisateurs soit des failles dans ces applications. Le ver permet alors une action massive, efficace et extrêmement rapide : le nombre et la vitesse deviennent des armes propices à compliquer la lutte antivirale. Le risque devient alors planétaire. De quelques centaines à quelques milliers de machines, la menace s'etend à des millions de machines dans le monde (*Melissa* en 1999, *ILoveYou* en 2000) en un temps record (dix minutes ont suffi au ver *Slammer* pour infecter 70,000 serveurs dans le monde en janvier 2003).

Face à cette nouvelle menace, les antivirus ont dû, encore une fois s'adapter et inclure des protections vis-à-vis de l'extérieur. Les filtres de messagerie sont apparus et l'usage des pare-feux est devenu plus répandu, quoique de gros progrès restent encore à faire dans ce domaine. Un grand nombre de machines pouvant être infectées, la proportion d'utilisateurs encore inconscients du risque et qui ne disposent d'aucune protection, cette proportion est importante et contribue à entretenir un niveau de risque encore élevé que nul antivirus ou autre protection ne parviendra à gérer en totalité. La loi du nombre est du côté des attaquants.

La vitesse de propagation est telle que les antivirus doivent encore plus fréquemment qu'auparavant mettre à jour leurs différentes bases (signatures et comportements) et ne peuvent donc gérer qu'avec retard, au minimum de quelques heures, les attaques qui se succèdent à un rythme qui parait toujours plus intense. Il est donc devenu nécessaire, évolution récente de ces produits, d'être connecté, sinon de manière permanente du moins fréquemment pour disposer des mise-à-jour. Etre connecté signifie donc permettre aux vers d'agir. Les attaquants ont contraint la défense à être constamment sur la brèche.

Cette dictature du nombre et de la vitesse a obligé également les produits antiviraux – notamment si l'on considère la concurrence effrénée dans ce secteur économique – a faire un certains nombre de compromis techniques, pour disposer de produits toujours plus fluides, plus réactifs (du moins en apparence) : amaigrissement des bases de signature par exemple. Ces compromis contribuent à accroitre le risque contre lequel les produits sont sensés lutter.

Les vers qui nous frappent depuis 1999 s'adaptent sans cesse et évoluent afin d'accroitre la pression sur les éditeurs d'antivirus et utiliser au mieux tout ce que la technologie nous offre ou ce que l'environnement informatique recèle de faiblesses :

— combinaison de différentes techniques : des vers comme *Badtrans* (fin 2001) ou *Scob* (juin 2004) incorporent des fonctionnalités de chevaux de Troie ;
— utilisation de plus en plus raffinée de techniques d'ingénierie sociale ou de manipulation psychologique de l'utilisateur pour l'inciter à déclencher le ver, le plus souvent par l'intermédiaire d'une pièce jointe ;

— attitude agressive vis-à-vis des antivirus : des vers comme *Klez* ou *Bug-Bear* (2002) désactivent ou désinstallent les antivirus ou les pare-feux qui les gênent ;
— utilisation de plus en plus rapide des failles logicielles touchant à la sécurité : alors qu'il pouvait s'écouler plusieurs semaines entre la publication d'une faille et son exploitation éventuelle par un ver, ce délai tend à être de l'ordre de quelques jours (vers *Blaster* (2003) ou *Sasser* (2004))...

Au final, l'accélération du rythme avec lequel les sociétés modernes fonctionnent et les faiblesses qui en résultent sont optimalement utilisées par les vers qui parviennent sur ce plan, déjà, à mettre les antivirus en difficulté. Si le nombre et la vitesse sont des armes efficaces pour les vers, ils sont plutôt une faiblesse pour la lutte antivirale : plus nombreux sont les cibles à traiter, à protéger et à éduquer, moins de temps est disponible pour le faire. Le succès des vers n'est pas tant dans leur niveau de technicité – même si certains sont très élaborés – mais dans la fragilité de nos sociétés.

6 Conclusion

Le duel virus/antivirus dure depuis environ une quinzaine d'années. Il est sans fin. Les résultats de Fred Cohen ont montré qu'il n'existe aucune protection absolue. Mais ce que l'Histoire nous montre, c'est qu'il n'existe, non plus, aucune arme absolue. Tout le jeu est affaire de connaissance et de compétences. L'équilibre est maintenu par le transfert des connaissances du virus à l'antivirus. Diffuser un virus revient à diffuser également le savoir permettant de lutter contre.

L'autre aspect important est celui des compétences techniques. Concevoir un virus vraiment évolué, qui mettra à mal les protections connues est une chose difficile et l'apanage de programmeurs consciencieux et extrêment compétents. Ils sont peu nombreux. En revanche, les imitateurs eux le sont. Ils se contentent de réutiliser, en les combinant quelquefois des techniques connues. Cet aspect là est important pour comprendre en quoi la lutte antivirale parvient toujours finalement à limiter les dégats, même si cela se fait avec retard.

Au final, gérer au mieux le fléau viral informatique nécessite de considérer en permanence le passé (la vision historico-technique) et le futur pour tenter de prévoir les évolutions techniques en matière de risque : c'est la raison pour laquelle la recherche en virologie informatique est vitale.

Références bibliographiques

[1] Cohen F. (1986) *Computer Viruses*, Ph. Thesis, University of Southern California.

[2] Filiol E. (2002), Le virus Concept. *Journal de la sécurité informatique MISC* n° 4, décembre.

[3] Filiol E. (2003) *Les virus informatiques : théorie, pratique et applications*, coll. Iris, Springer.

[4] Site consacré à la loi pour la confiance en l'économie numérique : `http://www.assemblee-nat.fr/12/dossiers/economie_numerique.asp`

[5] Harley D., Slade R., Gattiker U. E. (2002) *Virus : Définitions, mécanismes et antidotes*, Campus Press.

[6] Kleene S. C. (1936) General recursive functions of natural numbers, *Mathematische Annalen*, 112, p. 727-742.

[7] Kleene S. C. (1938) On Notation for ordinal numbers, *J. Symbolic Logic*, 3, 150-155.

[8] Ludwig M. A. (1993) *Computer Viruses and Artificial Life and Evolution*, American Eagle Press.

[9] Shoch J. F., Hupp J. A. (1982) The Worm programs - Early Experience with a Distributed Computation, *Communications of the ACM*, March, pp. 172-180.

[10] Slade R. (1992), *History of Computer Viruses*, `http://vx.netlux.org/lib/ars01.html`

[11] Smith G. C. (1994) *The Virus Creation Labs*, American Eagle Press.

[12] Solomon A. (1994), *A Brief History of PC Viruses*, `http://vx.netlux.org/lib/aas14.html`

[13] Turing A. M. (1936) On computable numbers with an application to the *Entscheidungsproblem*, *Proc. London Math. Society*, 2, 42, pp. 230-265.

[14] von Neumann J. (1951) The general and logical theory of automata, in Cerebral Mechanisms, *Behavior : The Hixon Symposium*, L.A. Jeffress ed., pp 1-32, Wiley.

[15] von Neumann J. (1966) *Theory of Self-reproducing Automata*, edited and completed by Burks, A. W., University of Illinois Press, Urbana and London.