Rapport de sécurité

Sylvain Rouquette
Epitech 4

ANTIVIRUS

Contents

1 Introduction

Tout débuta en 1986, le premier virus s'installait dans le secteur boot, se chargeait en mémoire au démarrage et se copiait sur toutes les disquettes qui entraient. Au début cela était juste amusant de trouver des astuces dans l'ordinateur et ce premier virus n'était pas des dangereux. La création des virus provient de différentes motivations comme le défi, la démonstration des capacités, la vengeance, le plaisir de faire souffrir les ordinateurs, la volonté de faire passer un message, bref toutes ces motivations amènent parfois des programmeurs chevronnés à vouloir développer des virus. Avec le temps les virus sont devenus de plus en plus complexes et parfois sont très dangereux, pouvant même détruire des parties physiques de l'ordinateur. Vu les problèmes que pouvaient causer les virus, que ça soit matériel ou financier pour une entreprise si celui-ci détruit les données, des antivirus ont commencé à être développé.

2 Comment fonctionnent les virus

2.1 Comprendre leurs fonctionnements est la première étape pour se défendre.

Il y a des milliers de virus et beaucoup de différentes catégories de virus, mais généralement ils correspondent tous à une même définition. Un virus est un programme machine intentionnellement conçu pour s'associer à un autre programme machine de manière à ce que quand le programme original est exécuté, le virus est aussi exécuté, puis se répliques lui-même en s'attachant à d'autres programmes. Le virus s'associe au programme original en s'attachant à ce programme ou même en le remplaçant, et la réplique est parfois sous la forme modifiée du virus. Le programme infecté peut être une macro, ou le secteur de boot d'un disque, le tout premier programme chargé à partir d'un disque amorçable.

Notez la partie " intentionnellement conçue " de la définition. Les virus ne sont pas de simples accidents. Des programmeurs qualifiés les écrivent et les développent, puis trouvent des moyens de les déposer sur des ordinateurs. Et plus les antivirus sont performants, plus le travail d'auteurs de virus est dur pour venir à bout d'eux. Pour beaucoup d'auteurs de virus, la chose est simplement un défi. Pour d'autres, c'est de se faire plaisir à rendre la vie des

ordinateurs incertaine ou malheureuse.

Les virus ont gagné une réputation pour être nocifs, mais en réalité beaucoup ne sont le pas. Oui, quelques fichiers endommagés ou exécutant d'autres formes de destruction, mais beaucoup sont bénins ou invisible pour la plupart des utilisateurs. Pour être considéré comme un virus, celui-ci à juste besoin de se répliquer lui-même, le reste est de l'extra.

Même les virus relativement bénins ne sont pas complètement inoffensifs, naturellement. Ils consomment l'espace disque, la mémoire, et les ressources processeur et affectent donc la vitesse et l'efficacité de votre machine. En outre, les programmes d'antivirus qui les détectent et les éliminent consomment également les ressources de l'ordinateur; beaucoup d'utilisateurs se plaignent des ralentissent occasionnés par les antivirus, plus que les virus eux-mêmes. En d'autres termes, les virus affectent la vie de votre ordinateur même lorsqu'ils ne sont pas réellement en train de faire quelque chose.

2.2 Virus et programmes agissant comme des virus

L'explication ci-dessus des virus est réellement plus spécifique que la manière dont nous tendons à employer le terme virus. D'autres types de programmes existent qui correspondent seulement à une partie de cette définition. Ce qu'ils ont en commun avec les virus est qu'ils agissent sans connaissance de l'utilisateur et commettent un certain genre d'acte à l'intérieur de l'ordinateur qu'ils sont intentionnellement conçus pour faire. Ces types incluent des worms (vers), des Trojan Horse (chevaux de Troie), et des dopplers (compte-gouttes). Tous ces programmes, y compris les virus, font partie d'une catégorie de programme connue sous le nom de malware, ou malicious-logic software.

Un worm est un programme qui se duplique mais n'infecte pas d'autres programmes. Il se copie sur des disquettes ou à travers des connections réseau, et parfois il utilise le réseau afin de se dupliquer. Un type de worm, le host worm, utilise le réseau seulement pour se dupliquer sur d'autres machines, tandis qu'un autre type, le network worm, répand des parties de lui-même à travers les réseaux et utilise sur les connections réseau pour exécuter ses divers parties. Les worms peuvent également exister sur ordinateur non connecté, dans ce cas ils se copient à divers endroits sur vos disques durs.

Le Trojan Horse vient de la mythologie grecque, racontée dans l'Odyssée, dans laquelle l'armée grecque a laissé un cheval en bois comme cadeau aux Troyens, cachant des troupes à l'intérieur du cheval pour pouvoir entrer dans la ville fortifié de Troie. Les grecs sont sortis et ont capturé la ville, finissant

le long siège. L'idée avec les ordinateurs est identique. Un Trojan Horse est un programme qui est caché à l'intérieur d'un programme apparemment inoffensif. Quand ce programme est exécuté, le Trojan Horse est aussi exécuté afin d'effectuer les actions que l'utilisateur ne veut pas. Les Trojan Horses ne se dupliquent pas. Une extension connu du Trojan Horse est Back Orifice, qui transforme votre machine en serveur administrable à distance, l'utilisateur ayant infecté votre machine à ainsi un contrôle total dessus. Les antivirus récents prévoient d'ailleurs un firewall embarqué pour surveiller l'activité réseau à cet effet.

Les dopplers sont des programmes conçus pour éviter la détection d'antivirus, habituellement par le chiffrage qui empêche l'antivirus de les détecter. Les fonctions typiques des dopplers sont de transporter et d'installer des virus. Ils attendent un événement spécifique que l'ordinateur déclenchera, et à ce moment là il exécutera le virus.

Lié à ces programmes est le concept de la bombe. Des bombes sont habituellement construites dans le malware en tant moyens pour l'activer. Des bombes sont programmées pour s'activer quand l'événement spécifique se produit. Quelques bombes s'activent à un instant spécifique, à l'aide typiquement de l'horloge de système. Une bombe a pu être programmée pour effacer tous les fichiers DOC de votre disque dur la veille de la nouvelle année ou faire apparaître un message le jour de l'anniversaire d'une personne célèbre. D'autres sont déclenchés par d'autres événements ou conditions: Une bombe pourrait attendre le vingtième démarrage d'un programme, par exemple, et effacez les dossiers de la taille du programme. Vu de cette façon, les bombes sont juste des scripts malveillants ou des taches planifiés.

Les virus peuvent être considérés comme exemples spéciaux impliquant un ou plusieurs de ces programmes de malware. Ils peuvent être répandu par des dopplers (bien qu'ils n'ont pas besoin d'être), et ils emploient l'idée de worm de se dupliquer. Tandis que les virus ne sont pas techniquement des Trojan Horse, ils agissent comme eux de deux manières:
ils font des choses que l'utilisateur n'a pas voulu
en s'attachant à un programme existant, ils transforment efficacement le programme original en Trojan Horse (ils se cachent à l'intérieur, se lancent au démarrage du programme, et provoquent des actions non désirés).

2.3 Comment fonctionne un virus

Les virus fonctionnent de différentes manières, mais voici le processus de base.

D'abord, le virus apparaît sur votre système. Il entre habituellement en tant qu'élément d'un exécutable infecté (COM, EXE, ou secteur de boot). Dans le passé les virus ont voyagé presque exclusivement par la distribution des disquettes infectées. Aujourd'hui, les virus sont fréquemment téléchargés sur les réseaux (Internet y compris) en tant qu'élément le plus téléchargé, ou de l'installation d'un logiciel d'essai, une macro pour un programme spécifique, ou un attachement sur un email.

Notez que l'email lui-même ne peut pas être un virus. Un virus est un programme, et il doit être exécuté pour devenir actif. Un virus livré comme attachement d'un email, donc, ne fait rien jusqu'à ce que vous le lanciez. Vous lancez ce genre de virus en lançant le fichier attaché, habituellement en double-cliquant dessus. Une manière d'aider à se protéger contre cette sorte de virus est de ne jamais ouvrir les attachements qui sont des exécutables (EXE ou COM) ou les fichiers de données pour des programmes, tels que les suites de bureau, qui fournissent des dispositifs de macro-écriture. Les fichiers graphiques, sonores, ou tout autre fichier de données est sûr.

Un virus commence sa vie sur votre PC, donc, comme un Trojan Horse. Il est caché dans un programme ou un fichier différent et se lance avec ce fichier. Dans un exécutable infecté, le virus a essentiellement modifié le programme original pour se diriger sur le code du virus et pour lancer ce code avec son propre code. Typiquement, il saute au code de virus, exécute ce code, et puis saute de nouveau au code du programme original. A ce moment le virus est en activité, et votre système est infecté.

Une fois actif, le virus effectue son travail immédiatement, si c'est un virus de type direct-action, ou se repose dans le fond comme programme résidant en mémoire, en utilisant le TSR (terminez et restez), une procédure permise par le système d'exploitation. La plupart sont de ce deuxième type et s'appellent des virus résidents. Etant donné la vaste gamme des activités permise par des programmes TSR, programmes de sauvegarde de fichier ou d'observation de l'activité du clavier ou de la souris, un virus résident peut être programmé pour faire autant de chose que le système d'exploitation le permet. En utilisant une bombe, il peut attendre des événements pour le déclencher, puis va travailler sur votre système. Une des choses qu'il peut faire est scanner votre disque ou vos disques réseaux pour rechercher d'autre

exécutables, il se copie alors dans ces programmes pour les infecter.

2.4 Types de virus

Les auteurs de virus expérimentent constamment de nouvelles manières d'infecter votre système, mais les types actuels de virus demeurent peu de temps. Ce sont des virus de secteur de boot, infecteurs de fichier, et macro virus. Il y a différents noms pour ces types et quelques sous-types, mais l'idée demeure la même.

Les virus ou les infecteurs de secteur de boot résident dans des secteurs spécifiques du disque dur, ceux qui sont lus et exécutés par l'ordinateur lors du démarrage. Les véritables virus de secteur de boot infectent seulement le secteur boot du DOS, alors qu'un sous-type appelé le virus MBR infecte le master boot record du disque. Ces deux parties du disque dur sont lues pendant le démarrage de l'ordinateur, pendant lequel le virus est chargé en mémoire. Les virus peuvent infecter les secteurs de boot des disquettes, mais typiquement une disquette de démarrage sans virus et protégée en écriture a toujours été une manière sûre de démarrer le système. Le problème, naturellement, est de garantir que la disquette elle-même est non infectée, et c'est la tâche que les antivirus tente de faire.

Les infecteurs de fichier, également appelés les virus parasites, sont des virus qui s'attachent aux exécutables, et ils sont les plus communs et les plus discutés. Un tel virus attend dans la mémoire que l'utilisateur exécute un autre programme, en utilisant un événement tel qu'un déclenchement pour infecter ce programme. Ainsi ils se dupliquent simplement par l'utilisation de l'ordinateur. Il y a différents types d'infecteurs de fichier, mais le concept est semblable dans tous.

Les macro virus, un type relativement nouveau, se servent du fait que beaucoup de programmes fonctionnent avec des langages de programmation intégrés. Les langages sont conçus pour aider les utilisateurs à automatiser les tâches en créant de petits programmes appelés les macros. Les programmes dans Microsoft Office, par exemple, se transportent avec un langage intégré, et en fait il fournit plusieurs de ses propres macros. Un macro virus est simplement une macro pour un de ces programmes, et en effet ce type de virus est devenu notoire grâce à l'infection de Microsoft Word. Quand un document contenant le macro virus est ouvert dans l'application cible, le virus s'exécute et fait ses dommages. En outre, il est programmé pour se copier dans d'autres documents, de sorte que l'utilisation du programme répande le

virus.

Un quatrième type, appelé le multipartite, combine l'infection de secteur de boot avec l'infection de fichier.

Pour une liste des virus avec les explications de ce qu'ils font, voyez la section Centre de Recherche Antivirus de Symantec.

2.5 De plus en plus subtile

Le concept du macro virus fonctionne parce que le langage de programmation permet d'accéder à la mémoire et aux disques durs. Ainsi, en fait, faites d'autres technologies récentes, incluant des contrôles ActiveX et des applets Java. Ceux-ci sont conçus pour protéger le disque dur contre les virus (Java mieux qu'ActiveX), mais le fait est que ces programmes peuvent s'installer sur votre ordinateur simplement parce que vous visitez un site Web. Évidemment, pendant que nous devenons de plus en plus câblés et pendant que nous nous attendons à des convenances telles que des mises à niveau du système d'exploitation en passant par Internet (Windows 98 et NT 5 le font), nous nous sommes mis à la porté de plus de virus et de tout autre malware.

Les auteurs de virus ne sont rien si non innovateurs, et ils proposent constamment de nouvelles manières de contrecarrer les antivirus. Les virus de type stealth, par exemple, trompent l'antivirus en lui faisant croire qu'il n'y a rien d'erroné. Essentiellement, le stealth virus maintient des informations sur les fichiers qu'il a infectés, puis attend dans la mémoire et intercepte l'antivirus qui recherchent les fichiers infectés. Il fournit aux antivirus l'ancienne information plutôt que la nouvelle. Les virus polymorphes se changent quand ils se dupliquent, de sorte que l'antivirus qui recherche les modèles spécifiques ne trouve pas toutes les duplications du virus, ceux qui survivent peuvent continuer à se dupliquer. Plusieurs autres types de virus futés apparaissent régulièrement, comme le jeu du chat et de la souris entre les auteurs de virus et les producteurs d'antivirus continue.

2.6 Comment un banal virus se répand

1. Une application affectée par un virus banal se charge en mémoire. Même lorsque l'application se termine, le virus reste dans la mémoire.

2. Le virus infecte une ou plusieurs autres applications. Quelques virus utilise le temps à processeur libre pour rechercher des applications, alors que

d'autres infectent des applications pendant qu'elles sont chargées en mémoire.

3. Ces applications infectées peuvent alors être transmises par l'intermédiaire des médias démontables, d'un réseau, ou de l'Internet.

3 Recherche d'une signature virale

La manière la plus simple pour détecter un virus est de rechercher dans un fichier susceptible d'être infecté s'il y a une signature virale, c'est-à-dire un bout de code qui ressemble à un code de virus présent dans la base de donnée de l'antivirus. Si une signature virale est détectée, le fichier pourrait être infecté. L'exemple standard est l'exemple d'EICAR, il suffit de copier les lignes suivantes dans un fichier texte et de renommer le fichier en .COM :

X5O!P%@AP[4\PZX54(P)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

L'antivirus devrait être capable de détecter le fichier comme étant un virus grâce à la signature virale.

4 Antivirus à technologie heuristique

D'une manière générale, il y à deux méthodes de base pour détecter des virus, spécifiques et génériques. La détection spécifique de virus exige de l'antivirus d'avoir quelques informations prédéfinies sur un virus spécifique (comme une chaîne à scanner). L'antivirus doit être fréquemment mis à jour afin de pouvoir détecter de nouveaux virus dés qu'ils apparaissent. Ces méthodes génériques de détection sont cependant basées sur des caractéristiques génériques du virus, donc théoriquement elles peuvent détecter chaque virus, y compris les nouveaux et les inconnus.

Pourquoi la détection générique prend-elle de l'importance? Il y a quatre raisons:

1) le nombre de virus augmente rapidement. Les études indiquent que le nombre total de virus double tous les neuf mois. La quantité de travail pour les chercheurs d'antivirus augmente, et les chances que quelqu'un sera frappé par un de ces nouveaux virus méconnaissables augmentent aussi.

2) le nombre de virus mutants augmente. Des codes sources de virus sont largement répandu et beaucoup de gens ne peuvent pas résister à la tentation d'expérimenter eux-mêmes, créant beaucoup de virus légèrement

modifiés. Ces virus modifiés peuvent ou peuvent ne pas être reconnus par les antivirus. Parfois ils sont, mais malheureusement souvent ils ne le sont pas.

3) le développement de virus polymorphes. Il est plus difficile de détecter des virus polymorphes comme MtE et TPE avec des scanneurs de virus. C'est souvent des mois après que le virus polymorphe a été découvert avant qu'un algorithme fiable de détection ait été développé. En attendant beaucoup d'utilisateurs ont une plus grande chance d'être infecté par ce virus.

4) les virus dirigés vers une organisation ou une compagnie spécifique. Il est possible que les individus utilisent des virus comme armes. En créant un virus qui travaille seulement sur des machines de l'organisation ou de la compagnie spécifique et il est très peu probable que le virus sorte de l'organisation. Ainsi il est très peu probable que n'importe quel antivirus pourra détecter le virus avant que celui-ci effectue son travail destructif et s'identifie.

Chacun de ces scénarios démontre le fait que les scanneurs de virus ne peuvent pas identifier un virus jusqu'à ce que le virus ait été découvert et analysé par un fournisseur d'antivirus.

Ces mêmes scénarios ne sont pas vrai pour les détecteurs génériques, et donc beaucoup de gens deviennent davantage intéressé dans les produits génériques d'antivirus. Des nombreuses méthodes de détection génériques, la recherche heuristique devient actuellement le plus important.

4.1 Emulateur pour une détection générique

En 1993, suite aux problèmes liés aux virus MTE (moteur pour créer des virus polymorphe), les sociétés de création d'antivirus se sont lancé dans une course de création d'un décrypteur générique, qui permettrait de détecter ce genre de virus. Le principe était donc d'émuler un ordinateur. Ce décrypteur générique est composé de quatre modules, un émulateur de processeur, de mémoire, de système, et d'un mécanisme de décision. La partie importante de ce décrypteur est le mécanisme de décision, le but n'étant pas d'émuler le programme jusqu'à la fin, ça prendrait beaucoup trop de temps, mais de voir comment il va réagir. Ce qui intéresse l'antivirus c'est que le virus se déchiffre pour pouvoir l'éradiquer, et celui-ci se déchiffre généralement avant le début de l'exécution du programme original. Mais il peut y avoir un problème avec ce genre de système. Le virus Cryptor générait son code en fonction de la machine sur laquelle il avait infecté le fichier, et il faisait planter l'émulateur du processeur si il n'était pas émulé avec le même processeur.

La détection en exécutant le fichier infecté dans univers émulé fonctionne bien pour les virus polymorphes et sera pratique pour participer à l'éradication de ce dernier.

4.2 Recherche heuristique

Une des tâches les plus longues qu'un chercheur de virus doit faire face est l'examen des fichiers. Les gens envoient souvent des fichiers aux chercheurs parce qu'ils croient que ces fichiers sont infectés par un nouveau virus. Parfois ils sont en effet infectés, parfois non. Chaque chercheur peut déterminer très rapidement ce qui se passe en chargeant le fichier suspecté dans un débuggeur. Quelques secondes sont souvent suffisantes, et beaucoup de chercheurs doivent s'être demandés comment ils pouvaient conclure aussi rapidement...

4.2.1 Intelligence artificielle

Certaines des nombreuses différences entre les virus et les programmes normaux sont que les programmes normaux commencent typiquement à rechercher la ligne de commande des options, effacer l'écran, etc.

Les virus cependant ne recherchent jamais les options ou l'espace libre de ligne de commande l'écran. Au lieu de cela ils commencent par une recherche d'autres fichier exécutables, en écrivant sur le disque, ou en se déchiffrant.

Un chercheur qui a chargé le fichier suspecté dans un débuggeur peut noter cette différence en un regard. La recherche heuristique est une tentative de mettre ces expérience et connaissance dans un scanneur de virus.

Le mot 'heuristique' signifie (selon un dictionnaire hollandais) 'la recherche de soi' et 'la connaissance pour déterminer quelque chose d'une manière méthodique'.

Un scanneur heuristique est une sorte de débuggeur ou de désassembleur. Les instructions sont désassemblées et leurs buts sont déterminés. Si un programme commence par une séquence MOVE AH, 5 INT 13h qui est une instruction BIOS pour formater le disque dur, le programme est suspecté, encore plus si le programme ne prend rien en ligne de commande ou n'interagit pas avec l'utilisateur.

4.2.2 Capacités suspectes

En réalité, l'heuristique est beaucoup plus compliquée. Les scanneurs heuristiques peuvent détecter des ordres d'instruction soupçonneux, comme la capacité de formater un disque, la capacité de rechercher d'autres exécutables, la capacité de rester résidents dans la mémoire, la capacité d'utiliser des appels système non standard ou non documentés, etc. Chacune de ces capacités a une valeur qui lui est assignée. Les valeurs assignées aux diverses capacités soupçonneuses dépendent de différentes choses. Une routine de formatage de disque n'apparaît pas dans beaucoup de programmes normaux, mais souvent dans les virus. Ainsi elle obtient une valeur élevée. La capacité de rester résidant en mémoire se trouve dans beaucoup de programmes normaux, ainsi outre le fait qu'elle apparaît également dans beaucoup de virus, elle n'obtient pas une valeur élevée. Si le total des valeurs pour un programme excède un seuil prédéfini, le scanneur lance l'alerte. Une simple capacité n'est jamais suffisante pour déclencher l'alarme. C'est toujours la combinaison des capacités suspectées qui convainquent le scanneur que le fichier est un virus.

4.2.3 Flags heuristiques

Quelques scanneurs ont placé un flag pour chaque capacité suspecte qui a été trouvées dans le fichier analysé. Ceci facilite l'explication de ce qui a été trouvé.

TbScan par exemple identifie beaucoup de séquence d'instruction. Chaque séquence d'instruction suspecte à un flag qui lui est assigné:

4.2.4 Description des flags

TbScan devrait sortir les flags suivants pour ces virus :

Plus il y a de flags de déclenchés par un fichier, plus il est probable que ce fichier soit infecté par un virus. Les programmes normaux déclenchent rarement un flag, alors qu'au moins deux flags sont exigés pour déclencher l'alarme. Pour la rendre plus compliquée, tous les flags n'ont pas le même poids.

4.3 Positifs faux

Juste comme toutes autres techniques génériques de détection, les scanneurs heuristiques blâment parfois des programmes innocents d'être contaminé par

Flag	Description
F	accès soupçonneux de fichier. Pourrait être capable d'infecter un fichier.
R	Relocator. Le code de programme sera réalloué d'une manière soupçonneuse.
A	Attribution De Mémoire Soupçonneuse. Le programme emploie une manière non standard de rechercher, et/ou d'assigner la mémoire.
N	nom de l'extension. L'extension est en conflit avec la structure du programme.
S	contient une routine pour rechercher des exécutables (COM ou EXE).
#	a trouvé une instruction de déchiffrage. C'est commun pour des virus mais également pour du logiciel protégé.
E	Point d'entrée flexible. Le code semble être conçu pour être lié dans n'importe quel endroit d'un exécutable. Commun pour des virus.
L	le programme emprisonne le chargement du logiciel. Pourrait être un virus qui arrête la charge de programme pour infecter le logiciel.
D	accès en écriture sur le disque. Le programme écrit sur le disque sans utiliser le DOS.
M	code résidant en mémoire. Ce programme est conçu pour rester dans la mémoire.
!	opcode inadmissible (instructions non-8088) ou branchement en dehors de la mémoire.
T	heure incorrecte. Quelques virus emploient ceci pour marquer les fichiers infectés.
J	construction soupçonneuse de saut. Point d'entrée par l'intermédiaire des sauts enchaînés ou indirects. C'est peu commun pour le logiciel normal mais commun pour des virus.
?	en-tête EXE contradictoire. Pourrait être un virus mais peut également être un bogue.
G	instruction poubelle. Contient du code qui semble n'avoir aucun but autre que le chiffrage ou à pour but d'éviter la reconnaissance par des scanneurs.
U	appel non documenté d'interruption DOS. Le programme pourrait être simplement rusé mais peut également être un virus en utilisant une manière non standard de se détecter.
Z	détermination d'EXE/COM. Le programme essaye de vérifier si un fichier est un fichier COM ou EXE. Les virus doivent faire ceci pour infecter un programme.
O	a trouvé le code qui peut être employé à réécrire ou déplacer un programme dans la mémoire.
B	retour au point d'entrée. Contient le code pour remettre en marche le programme après que des modifications au point d'entrée soient faites. Très habituel pour des virus.
K	pile peu commune. Le programme a une pile soupçonneuse ou une pile bizarre.

Virus	Flags
Jerusalem/PLO	FRLMUZ
Backfont	FRALDMUZK
Minsk_Ghost	FELDTGUZB
Murphy	FSLDMTUZO
Ninja	FEDMTUZOBK
Tolbuhin	ASEDMUOB
Yankee_Doodle	FN#ELMUZB

un virus. Ceci s'appelle " positif faux " ou " fausse alarme ".

La raison de ceci est simple. Quelques programmes s'avèrent justement avoir plusieurs capacités suspectes. Par exemple, le fichier de LOADHI.com de QEMM a les capacités suspectes suivantes (selon une version plus ancienne, pourtant obsolète de TbScan):

A = Attribution De Mémoire Soupçonneuse. Le programme emploie une manière non standard de rechercher, et/ou d'assigner la mémoire.

M = Code résidant en mémoire. Ce programme peut être un TSR mais également un virus.

U = appel non documenté d'interruption DOS. Le programme pourrait être simplement rusé mais peut également être un virus en utilisant une manière non standard de se détecter.

Z = détermination d'EXE/COM. Le programme essaye de vérifier si un fichier est un COM ou un EXE. Les virus doivent faire ceci pour infecter un programme.

O = a trouvé le code qui peut être employé à réécrire ou déplacer un programme dans la mémoire.

Toutes ces capacités sont disponibles dans LoadHi, et les flags sont suffisants pour déclencher l'alarme heuristique. Puisque que LoadHi est censé allouer de la mémoire supérieure, charger des programmes résidants dans la mémoire, les déplacer dans la mémoire supérieure, etc., toutes ces capacités suspectes peuvent facilement être expliquées et vérifiées. Cependant, le module de balayage ne peut pas savoir le but prévu du programme, et comme la plupart de ces capacités suspectes sont souvent trouvées dans les virus, il décrit juste le programme de LoadHi comme " un virus possible ".

4.3.1 Quelle est la gravité des fausses alarmes ?

Si un scanneur heuristique affiche un message: "ce programme peut formater un disque et peut résider dans la mémoire", et le programme est un outil de formatage de disque résidant en mémoire, est-ce vraiment une fausse alarme ? En fait, le scanneur a raison. Un utilitaire de formatage résidant contient évidemment le code pour formater un disque, et il contient le code pour rester résident en mémoire.

Le scanneur heuristique a complètement raison ! Vous pourriez appeler ça un faux soupçon, mais pas un positif faux. Le seul problème ici est que le scanneur indique que ce pourrait être un virus. Si vous pensez que le scanneur vous indique qu'il a trouvé un virus, il s'avère que c'est une fausse alarme. Cependant, si vous prenez cette information comme étant un moyen de vérifier si le fichier est réellement infecté, ça ne compterait pas comme fausse alarme. Le scanneur indique juste la vérité. Le principal problème ici est la personne qui doit prendre des décisions avec l'information fournie par le scanneur. Si c'est un utilisateur débutant, c'est un problème.

4.3.2 Eviter les fausses alarmes

Si nous l'appelons un positif faux ou un faux soupçon, cela importe peu. C'est ennuyeux que le scanneur détecte un virus à chaque recherche. Ainsi nous devons éviter cette situation. Comment réalisons-nous ceci ?

1) définition des (combinaisons) capacités soupçonneuses

Le scanneur ne reporte pas une alarme à moins qu'au moins deux capacités suspectes séparées aient été trouvées.

2) identification des codes communs de programme

Quelques codes connus de compilateur ou routines d'exécution de compression ou de déchiffrage peuvent causer les fausses alarmes. Ces codes spécifiques de compression ou de déchiffrage peuvent être reconnu par le scanneur pour éviter les fausses alarmes.

3) identification des programmes spécifiques

Quelques programmes qui posent normalement un problème (comme le programme de LoadHi utilisé dans l'exemple) peuvent être identifiés par le scanneur heuristique.

4) prétention que la machine n'est pas au initialement infectée

Quelques scanneurs heuristiques ont un mode d'apprentissage, c'est-à-dire qu'ils peuvent apprendre qu'un fichier causant une fausse alarme n'est

pas un virus.

4.3.3 Traiter les positifs faux

Quelques positifs faux ne sont pas facilement évités. Ainsi, l'utilisateur doit traiter une certaine quantité fausses d'alarmes, et doit prendre la décision finale de savoir si un fichier est infecté ou pas.

La plupart des personnes pensent qu'ils n'y a aucun moyen de savoir si un fichier est infecté ou pas. En fait il y a une manière de découvrir, mais ceci dépend du scanneur.

Le scanneur doit expliquer à l'utilisateur les raisons pour lesquelles le programme est suspect. 'Ce fichier pourrait contenir un virus' n'en dit pas beaucoup à l'utilisateur. C'est toujours exact. Chaque fichier pourrait contenir un virus, mais peut également être propre. Nous employons réellement un scanneur pour découvrir ! Qu'est-ce que l'utilisateur est censé faire avec ces informations ?

Cependant, si le scanneur indique qu'un certain programme peut rester résident dans la mémoire et est capable de formater un disque, l'utilisateur peut plus facilement comprendre ce qui se passe. Si un logiciel de traitement de texte donne une telle alarme, il est extrêmement probable que le programme soit vérolé, parce que les logiciels de traitement de texte ne peuvent généralement pas formater des disques et rester résidents dans la mémoire. Cependant, si le fichier suspecté est un outil de formatage de disque, toutes les capacités suspectes peuvent être expliqué grâce à l'utilisation prévue du programme.

Raison de soupçon: outil de formatage de disque résident en mémoire.

Programme	Probablement
Outil de formatage résidant	Pas de virus (innocent)
Editeur de texte	Suspect (virus)

Les deux programmes causent les mêmes alarmes heuristiques, mais la conclusion finale est différente.

Naturellement, elle exige d'un utilisateur avancé de tirer une conclusion pour la question " infectée ou pas ? ". Cependant, juger les résultats de n'importe quel scanneur (aussi scanneurs conventionnels) est une tâche pour un utilisateur expérimenté seulement. Si le scanneur a un mode d'apprentissage, c'est-à-dire qu'il peut se rappeler quels programmes causent une fausse alarme,

la recherche initiale devrait être exécutée par un utilisateur expérimenté, mais les recherches suivantes (quand on a éliminé les positifs faux possibles) peuvent être exécutés par un utilisateur débutant. C'est déjà pratique courante dans la plupart des organismes.

Quoi qu'il en soit, le résultat n'est pas si mauvais qu'il y parait, car toutes autres méthodes de détection (recherche de signature y compris) sont connues pour causer quelques fausses alarmes. L'heuristique cependant a l'avantage qu'il peut vous fournir assez d'information pour établir par vous-même si un fichier suspecté est probablement un virus ou pas.

4.4 Comment la recherche heuristique s'effectue ?

L'heuristique est une technique relativement nouvelle et toujours en cours de développement. Elle gagne cependant de l'importance rapidement. Ceci n'étonne pas car les scanneurs heuristiques peuvent détecter 90% des virus sans employer n'importe quelle information prédéfinie comme des signatures ou des valeurs de checksum. La quantité de positifs faux dépend du scanneur, mais un chiffre aussi bas que 0,1% peut être atteint facilement.

TbScan 6,02 employé sur la grande collection de virus de Vesselin Bontchev a montré les résultats suivants:

Recherche	7210	Détection
Méthode	Fichiers	Pourcentage
Conventionnelle	7056	97.86%
Heuristique	6465	89.67%

Un test positif faux est plus difficile à effectuer car il n'y a aucun résultat indépendant disponible.

4.5 Combinaison de la recherche conventionnelle et heuristique

Certains pensent que la recherche heuristique est un remplacement de la recherche conventionnelle. Ce n'est pas le cas. La recherche heuristique atteint un objectif très utile une fois utilisé en combinaison avec la recherche conventionnelle. Les résultats des deux méthodes de recherche peuvent être validés par eux-mêmes, réduisant de ce fait les positifs faux et également les négatifs faux.

Résultat combiné d'analyse:

Heuristique	Conventionnelle	Probabilité
Propre	Propre	Probablement propre
Propre	Virus	Certainement un positif faux
Virus	Propre	Certainement un négatif faux
Virus	Virus	Probablement infecté
Négatif faux : 10%	Négatif faux : 1%	Combiné : 0.1%
Positif faux : 0.1%	Positif faux : 0.001%	Combiné : 0.00001%

Les chances du scanneur heuristique et conventionnel d'échouer sont minimales. Si les deux méthodes de recherche ont les mêmes résultats, le résultat est presque sûr. Dans le cas ou les résultats ne s'accordent pas, une analyse supplémentaire est requise.

TbScan 6,02 employé sur la grande collection de virus de Vesselin Bontchev a montré les résultats suivants:

Recherche	7210	Détection
Méthode	Fichiers	Pourcentage
Conventionnelle	7056	97.86%
Heuristique	6465	89.67%
Combiné	7194	99.78%

4.6 L'avenir de la détection heuristique

4.6.1 Développement toujours en cours

La plupart des développeurs d'antivirus ne fournissent toujours pas un analyseur heuristique prêt à employer. Ceux qui ont l'heuristique déjà disponible l'améliorent toujours. Il est cependant peu probable que le taux de détection atteint 100% sans certaine quantité de positifs faux. D'autre part il est peu probable que la quantité de positifs faux n'atteint jamais 0%.

Serait-il possible d'avoir une détection correcte à 100% ? Il y a un grand secteur gris entre les virus et les non virus. Même pour des humains il est difficile de décrire ce qu'est un virus ou pas, une définition souvent utilisée

d'un virus d'ordinateur est ceci: " un virus est un programme qui peut se copier ". Selon cette définition le programme de DiskCopy.Com est un virus...

4.6.2 Et du côté des programmeurs de virus

Une question importante est l'effet sur les auteurs de virus. Il est probable qu'ils essayent d'éviter la détection par les scanneurs heuristiques. Jusqu'ici le but était d'éviter la détection par des scanneurs de signature, et c'était plutôt facile, car il suffisait juste de modifier seulement une petite partie d'un virus existant. Les adolescents avec de la compréhension de base de la programmation pouvaient le faire facilement. Éviter les scanneurs heuristiques cependant exige beaucoup plus la connaissance, si possible en tout. L'auteur de virus sera forcé d'écrire des virus plus complexes.

4.7 Nettoyage heuristique

Un virus se loge donc dans un exécutable et lors de l'exécution de se fichier, le code du virus est exécuté car se dernier avait placé un saut au début de l'EXE. Une fois le virus exécuté, il retourne à l'exécution normale du programme.

Pour nettoyer un programme infecté, il est important de reconstituer les octets recouverts par le saut vers le code du virus. Le virus doit reconstituer ces octets également, les octets originaux sont donc stockés quelque part dans le code du virus. L'antivirus recherche ces octets, tronque le fichier pour retirer le virus et remet les octets originaux.

4.7.1 Nettoyage conventionnel

L'antivirus doit savoir comment fonctionne le virus pour pouvoir l'éradiquer. Par exemple pour un fichier infecté par Jerusalem/PLO, l'antivirus s'y prendra comme ça :

Le virus à une taille de 1873 octets
Il a écrasé les 3 premiers octets pour effectuer sont saut
Les octets originaux se trouvent à l'octet 483 dans le code du virus
Copie des 3 octets pour restaurer les octets écraser par le saut
Suppression des 1873 octets du fichier

4.7.2 Imperfection des antivirus conventionnel

L'antivirus doit connaître le virus qu'il doit enlever. Il est impossible d'enlever un virus inconnu.

Le virus devrait être identique au virus connu. Si le virus utilisé dans l'exemple était modifié par exemple la taille passait de 1873 à 1869... L'antivirus enlèverait trop de données. Ce n'est pas une exception, mais cela se produit souvent puisqu'il y a tant de mutants. Par exemple, la famille de Jerusalem/PLO contient maintenant plus de 100 mutants!

Beaucoup de virus polymorphes ont des longueurs variables et maintiennent les instructions originales chiffrées. La plupart des antivirus conventionnels ne peuvent donc pas nettoyer des programmes infectés par MtE.

4.7.3 Le virus s'enlèvera avant l'exécution du programme

Nous avons vu au-dessus la façon dont un virus travaille. La partie intéressante est que quand le virus passe la main au programme original il reconstitue les octets originaux au début du programme et saute de nouveau au début le programme. Chaque virus peut réparer le programme original afin de le maintenir fonctionnel (excepté des virus de recouvrement, mais ces derniers ne peuvent pas être nettoyé de toute façon).

4.7.4 Laissez le virus effectuer le travail

Le principe de base du nettoyage heuristique est simple. L'antivirus heuristique charge le fichier infecté et commence à émuler le code du programme. Il emploie une combinaison de désassemblage, émulation et parfois exécution pour tracer l'évolution du virus, et pour émuler ce que le virus fait normalement. Quand le virus reconstitue les instructions originales et saute de nouveau au code original de programme, l'antivirus arrête le processus d'émulation. Le début maintenant du programme réparé est copié de nouveau dans le fichier sur le disque, et la partie du programme qui a gagné l'exécution sera coupée. Une analyse additionnelle du fichier nettoyé sera exécutée pour être sûr.

Notez que l'antivirus enlève réellement l'inconnu de l'inconnu. Aucune information prédéfinie sur le virus ou le dossier infecté n'est nécessaire.

Le processus de l'émulation est juste comme l'auto-stop. L'émulateur convainc le code viral qu'il s'exécute réellement, et il fait de l'auto-stop au point où le virus passe la main au programme original.

Cependant, le processus réel est très compliqué. Comme à l'auto-stop, beaucoup de choses peuvent être mal assorties:
le conducteur vous conduit au mauvais endroit. Le virus ne prévoit d'exécuter le programme original, mais il commence à faire des choses complètement différentes. Car le but de l'émulation est de reconstituer le programme original, nous n'atteignons jamais notre but.
le conducteur ne vous laissera pas dehors. Si le code viral exécute une boucle sans fin, le programme original ne sera jamais reconstitué ainsi l'antivirus pourrait attendre pour toujours.
le conducteur sort de la voiture. Une situation potentiellement dangereuse est que l'antivirus est trop ambitieux dans sa tâche d'émuler tout, et que le virus obtient la commande à l'intérieur de l'environnement émulé et s'échappe finalement de lui.
le conducteur frappe un arbre et vous tue aussi. Beaucoup de virus sont mal programmés. S'ils se brisent à l'intérieur de l'émulateur, les chances sont que l'émulateur se brise aussi.

4.7.5 Avantages et inconvénients

Avantages Aucun besoin d'identifier des mutants
Aucuns problèmes avec les virus polymorphes
Peut nettoyer les 'futurs' virus
L'utilisateur dépend moins des mises à jour de produit

Inconvénients Aucune copie exacte de l'original
Il nettoie tout: nettoie même les fichiers non infectés

Le nettoyage heuristique a besoin d'améliorations additionnelles. Quelques virus emploient les dispositifs d'anti-débuggeur qui font échouer l'émulation. Il est également encore possible qu'un virus détecte qu'il est émulé, et il peut simplement refuser de coopérer. Plus l'émulateur exécute correctement, moins la probabilité que le virus le détecte est grande.

5 Conclusion

Les virus n'ont pas fini de proliférer et un nouvel horizon s'est ouvert grâce à l'engouement du public pour Internet ainsi que des nouveaux systèmes d'exploitations qui favorisent la prolifération de virus, tel que Windows 9x/ME/NT/XP et du problème lié à Outlook qui exécutait automatiquement les script Visual Basic. Les virus ne cesseront d'exister et les antivirus devront sans cesse se mettre à leur niveau. Les systèmes d'exploitation change la manière dont les virus peuvent attaquer, mais cela change aussi la manière dont les antivirus peuvent intervenir. Bref, la guerre n'est pas prête d'être fini. Pendant un temps, il y a eu une rumeur comme quoi c'étaient les développeurs d'antivirus qui créaient des virus.

6 Bibliographie

VDAT 2.000.2, The Computer Virus Database, Cicatrix, 15 septembre 2000.

“ How viruses work ”, Neil Randall, PC Magazine.

Heuristic Anti-Virus Technology, Frans Veldman.

Emulators for Generic Decryptors, Frans Veldman.