Logiciels Malveillants

Sommaire

- Scanners basés hôte
- Anti-virus basé sur la signature
- Approches de balayage du périmètre
- Approches de collecte de renseignements distribués

Ce chapitre examine les différentes menaces et des contre-mesures liées aux logiciels malveillants. Une étude des différents types de logiciels malveillants et une classification générale basée d'abord sur les moyens utilisés par les logiciels malveillants pour se propager, puis sur la variété des actions ou des charges virales utilisées une fois que le logiciel malveillant ait atteint une cible, sont traitées. Les mécanismes de propagation comprennent ceux utilisés par les virus, les vers et les chevaux de Troie. Les charges virales comprennent la corruption du système, les bots, le phishing, les logiciels espions et les rootkits. La discussion se termine par un examen des approches de contre-mesures.

Malware

Les logiciels malveillants, ou maliciels, constituent sans doute l'une des catégories les plus importantes de menaces pour les systèmes informatiques. La norme NIST SP 800-83 (Guide to Malware Incident Prevention and Handling for Desktops and Laptops, juillet 2013) définit le malware comme "un programme qui est inséré dans un système, généralement de manière cachée, dans l'intention de compromettre la confidentialité, l'intégrité ou la disponibilité des données, des applications ou du système d'exploitation de la victime ou de l'ennuyer ou de le perturber de toute autre manière". Nous sommes donc préoccupés par la menace que les logiciels malveillants font peser sur les programmes d'application, sur les programmes utilitaires, tels que les éditeurs et les compilateurs, et sur les programmes au niveau du noyau. Nous sommes également préoccupés par son utilisation sur des sites et serveurs web compromis ou malveillants, ou dans des courriers électroniques de spam ou autres messages spécialement conçus, qui visent à tromper les utilisateurs en leur faisant révéler des informations personnelles sensibles.

Terminologie

La terminologie dans ce domaine pose des problèmes en raison de l'absence d'un accord universel sur tous les termes et parce que certaines catégories se chevauchent. Le tableau 1 est un guide utile pour certains des termes utilisés.

Table 6.1 **Terminology for Malicious Software (Malware)**

Name	Description
Advanced Persistent Threat (APT)	Cybercrime directed at business and political targets, using a wide variety of intrusion technologies and malware, applied persistently and effectively to specific targets over an extended period, often attributed to state-sponsored organizations.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.
Attack kit	Set of tools for generating new malware automatically using a variety of supplied propagation and payload mechanisms.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Backdoor (trapdoor)	Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality in a program, or onto a compromised system.
Downloaders	Code that installs other items on a machine that is under attack. It is normally included in the malware code first inserted on to a compromised system to then import a larger malware package.
Drive-by-download	An attack using code on a compromised website that exploits a browser vulnerability to attack a client system when the site is viewed.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
Flooders (DoS client)	Used to generate a large volume of data to attack networked computer systems, by carrying out some form of denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Logic bomb	Code inserted into malware by an intruder. A logic bomb lies dormant until a predefined condition is met; the code then triggers some payload.
Macro virus	A type of virus that uses macro or scripting code, typically embedded in a document or document template, and triggered when the document is viewed or edited, to run and replicate itself into other such documents.
Mobile code	Software (e.g., script and macro) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Spammer programs	Used to send large volumes of unwanted e-mail.
Spyware	Software that collects information from a computer and transmits it to another system by monitoring keystrokes, screen data, and/or network traffic; or by scanning files on the system for sensitive information.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes it.
Virus	Malware that, when executed, tries to replicate itself into other executable machine or script code; when it succeeds, the code is said to be infected. When the infected code is executed, the virus also executes.

(continued)

Table 6.1 **Terminology for Malicious Software (Malware)** (*Continued*)

Name	Description
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network, by exploiting software vulnerabilities in the target system, or using captured authorization credentials.
Zombie, bot	Program installed on an infected machine that is activated to launch attacks on other machines.

A Broad Classification of Malware

Classification des Malwares

Par le passé, un certain nombre d'auteurs ont tenté de classer les logiciels malveillants, comme cela a été proposé par les auteurs de [HANS04]. Bien qu'un certain nombre de critères puissent être utilisés, une approche utile consiste à classer les logiciels malveillants en deux grandes catégories : En se basant d'abord sur la manière dont ils se propagent pour atteindre les cibles souhaitées, puis sur les actions ou les charges virales qu'ils déclenchent une fois la cible atteinte.

Les approches précédentes de la classification des logiciels malveillants faisaient la distinction entre ceux qui nécessitent un programme hôte, étant un code parasite comme les virus, et ceux qui sont des programmes indépendants et autonomes exécutés sur le système comme les vers, les chevaux de Troie et les bots. Une autre distinction a été faite entre les logiciels malveillants qui ne se répliquent pas, comme les chevaux de Troie et les courriers électroniques non sollicités, et les logiciels malveillants qui se répliquent, y compris les virus et les vers.

1. Type de Malwares

Les mécanismes de propagation comprennent l'infection du contenu exécutable ou interprété existant par des virus qui sont ensuite propagés à d'autres systèmes ; l'exploitation des vulnérabilités des logiciels, soit localement soit sur un réseau, par des vers ou des téléchargements "drive-by download" pour permettre aux logiciels malveillants de se répliquer ; et les attaques d'ingénierie sociale qui convainquent les utilisateurs de contourner les mécanismes de sécurité pour installer des chevaux de Troie, ou de répondre aux attaques de phishing ou hameçonnage.

Les actions de effectuées par les logiciels malveillants une fois qu'ils atteignent un système cible peuvent inclure la corruption du système ou des fichiers de données ; le vol de service afin de faire du système un agent d'attaque zombie dans le cadre d'un réseau

de zombies ; le vol d'informations du système, en particulier des noms de connexion, des mots de passe ou d'autres détails personnels par des programmes d'enregistrement de clés ou des logiciels espions ; et la dissimulation où le logiciel malveillant cache sa présence sur le système pour éviter les tentatives de détection et de blocage.

Alors que les premiers logiciels malveillants avaient tendance à utiliser un seul moyen de propagation pour fournir une seule , nous constatons, au fur et à mesure de leur évolution, une croissance de logiciels malveillants mixtes qui incorporent une série de mécanismes de propagation et de charges virales qui augmentent leur capacité à se propager, à se cacher et à effectuer une série d'actions sur des cibles. Une attaque mixte utilise plusieurs méthodes d'infection ou de propagation, afin de maximiser la vitesse de contagion et la gravité de l'attaque. Certains logiciels malveillants prennent même en charge un mécanisme de mise à jour qui leur permet de modifier la portée des mécanismes de propagation et des charges virales utilisés une fois qu'ils sont déployés. Dans les sections suivantes, nous passons en revue ces différentes catégories de logiciels malveillants, puis nous examinons les contre-mesures appropriées.

1.1 Kits d'Attaques

Initialement, durant la décennie des années 80, le développement et le déploiement de logiciels malveillants nécessitaient des moyens techniques considérables et la compétence des auteurs des malwares. Puis plus tard, des kits de développement et d'attaque plus généraux ont apparu dans la décennie 2000, qui a grandement contribué à l'expansion de ces logiciels malveillants [FOSS10].

Ces boîtes à outils, souvent connues sous le nom de logiciels criminels, comprennent maintenant une variété de mécanismes et de modules de que même les novices peuvent combiner, sélectionner et déployer. Ils peuvent également être facilement personnalisés avec les dernières vulnérabilités découvertes afin d'exploiter la fenêtre d'opportunité entre la publication d'une faille dans un système et le déploiement généralisé de patchs pour la sécuriser (fermer). Ces kits ont considérablement élargi la population des attaquants capables de déployer des logiciels malveillants.

Bien que les logiciels malveillants créés à l'aide de ces boîtes à outils soient généralement moins sophistiqués que ceux conçus à partir de zéro, le nombre de nouvelles variantes qui peuvent être générées par les attaquants à l'aide de ces boîtes à outils crée un problème important pour ceux qui se chargent de la sécurité des systèmes informatiques.

La boîte à outils Zeus est un exemple éminent de ce type d'attaque, qui a été utilisés pour générer un large éventail de logiciels malveillants très efficaces facilitant ainsi les activités de cyber criminalité, en particulier, la saisie et l'exploitation des références bancaires [BINS10].

Le Kit d'exploitation Anglers, apparu pour la première fois en 2013, a été le kit le plus actif en 2015, souvent distribué par le biais d'une publicité malveillante qui exploitait les vulnérabilités de Flash. Il est sophistiqué et techniquement avancés, tant dans les attaques exécutées que dans les contre-mesures déployées pour résister à la détection. Il existe un certain nombre d'autres kits d'attaque en usage, bien que les kits spécifiques

changent d'année en année, les attaquants continuent à évoluer et à s'améliorer [SYMA16].

1.2 Sources d'Attaques

Une autre évolution importante des logiciels malveillants au cours des deux dernières décennies est le passage d'attaquants individuels, souvent motivés pour démontrer leurs compétences techniques à leurs pairs, à des sources d'attaque plus organisées et plus dangereuses. Il s'agit notamment des attaquants à motivation politique, des criminels et du crime organisé, des organisations qui vendent leurs services aux entreprises et aux nations, et des agences gouvernementales nationales. Cela a considérablement modifié les ressources disponibles et les motivations à l'origine de l'essor des logiciels malveillants, et a conduit au développement d'une vaste économie souterraine impliquant la vente de kits d'attaque, l'accès à des machines hôtes compromises et à des informations volées.

2. Menaces Persistantes Avancées

Les menaces persistantes avancées (APT) ont pris de l'importance ces dernières années. Il ne s'agit pas d'un nouveau type de logiciels malveillants, mais plutôt de l'application persistante et bien équipée d'une grande variété de technologies d'intrusion et de logiciels malveillants à des cibles sélectionnées, généralement des entreprises ou des politiciens. Les APT sont généralement attribués à des organisations parrainées par les états, et même certaines attaques provenant probablement aussi d'entreprises criminelles.

Les APT se distinguent des autres types d'attaques par leur sélection minutieuse des cibles et par leur persistance, souvent furtive, des efforts d'intrusion sur de longues périodes. Un certain nombre d'attaques très médiatisées, dont Aurora, RSA, APT1 et Stuxnet, sont souvent citées en exemple.

Caractéristiques APT

C'est en raison de ces caractéristiques qu'ils sont nommés :

- **Avancé** : Utilisation par les attaquants d'une grande variété de technologies d'intrusion et de logiciels malveillants, y compris le développement de logiciels malveillants personnalisés si nécessaire. Les différents composants ne sont pas nécessairement techniquement avancés, mais ils sont soigneusement sélectionnés pour convenir à la cible choisie.
- **Persistant** : Application déterminée des attaques sur une période prolongée contre la cible choisie afin de maximiser les chances de succès. Diverses attaques peuvent être appliquées progressivement, et souvent furtivement, jusqu'à ce que la cible soit compromise.
- **Menaces** : Menaces contre les cibles choisies résultant de l'intention des attaquants organisés, capables et bien financés de compromettre les cibles spécifiquement choisies. La participation active des personnes dans le processus augmente

considérablement le niveau de menace par rapport à celui dû aux outils d'attaque automatisés, ainsi que la probabilité de réussite de l'attaque.

Attaques APT

L'objectif de ces attaques varie du vol de propriété intellectuelle ou de données relatives à la sécurité et aux infrastructures à la perturbation physique des infrastructures. Les techniques utilisées comprennent l'ingénierie sociale, le harponnage par courrier électronique et le téléchargement à partir des sites web compromis susceptibles d'être visités par le personnel de l'organisation cible. L'objectif est d'infecter la cible avec des logiciels malveillants sophistiqués ayant de multiples mécanismes de propagation et charges virales. Une fois qu'ils aient obtenu l'accès initial aux systèmes de l'organisation cible, une autre gamme d'outils d'attaque est utilisée pour maintenir et d'étendre leur accès.

En conséquence, il est beaucoup plus difficile de se défendre contre ces attaques en raison de ce ciblage spécifique et de leur persistance. Elles nécessitent une combinaison de contre-mesures techniques, comme nous le verrons plus loin dans ce chapitre, ainsi qu'une formation de sensibilisation pour aider le personnel à résister à ces attaques. Même avec les meilleures pratiques de contre-mesures actuelles, le recours aux exploits du jour zéro et aux nouvelles approches d'attaque signifie que certaines de ces attaques ont des chances de réussir [SYMA16, MAND13]. Ainsi, de multiples couches de défense sont nécessaires, avec des mécanismes pour détecter, répondre et atténuer ces attaques. Ces mécanismes peuvent inclure la surveillance du trafic de commande et de contrôle des logiciels malveillants et la détection du trafic d'exfiltration.

Les Virus

La première catégorie de propagation de logiciels malveillants concerne les fragments de logiciels parasites qui s'attachent à certains contenus exécutables (Programmes exécutables) existants. Le fragment peut être un code machine qui infecte une application, un utilitaire ou un programme système existant, ou même le code utilisé pour démarrer un système informatique (Bootstrap).

Les infections par des virus informatiques ont constitué la majorité des logiciels malveillants observés au début de l'ère des ordinateurs personnels (Décennie 1980) Le terme "virus informatique" est encore souvent utilisé pour désigner les logiciels malveillants en général, et pas seulement les virus informatiques en particulier. Plus récemment, le fragment de virus informatique a été une forme de code de script, généralement utilisé pour prendre en charge le contenu actif de fichiers de données tels que ceux de Microsoft Documents Word, feuilles de calcul Excel ou documents Adobe PDF.

Un virus informatique est un logiciel qui peut "infecter" d'autres programmes, ou même tout type de contenu exécutable, en les modifiant. La modification comprend l'injection

du code original avec une routine pour faire des copies du code du virus, qui peuvent ensuite infecter d'autres contenus. Les virus informatiques sont apparus au début des années 1980, et le terme lui-même est attribué à Fred Cohen. Cohen est l'auteur d'un livre révolutionnaire sur le sujet [COHE94]. Le virus du cerveau (Brain virus), apparu pour la première fois en 1986, a été l'un des premiers à cibler les systèmes MSDOS, et a entraîné un nombre important d'infections durant cette époque.

Les virus biologiques sont de minuscules bouts de code génétique - ADN ou ARN - qui peuvent s'emparer de la machinerie d'une cellule vivante et la tromper pour qu'elle fabrique des milliers de répliques parfaites du virus original. Comme son homologue biologique, un virus informatique porte dans son code d'instructions le mécanisme pour faire des copies parfaites de lui-même.
Le virus typique est intégré dans un programme, ou support de contenu exécutable, sur un ordinateur. Ensuite, chaque fois que l'ordinateur infecté entre en contact avec un morceau de code non infecté, une nouvelle copie du virus passe dans le nouvel emplacement. Ainsi, l'infection peut se propager d'un ordinateur à l'autre, avec l'aide d'utilisateurs peu méfiants, qui échangent ces programmes ou fichiers porteurs sur disques compacts ou clés USB ; ou qui se les envoient les uns aux autres via un réseau. Dans un environnement de réseau, la possibilité d'accéder à des documents, des applications et des services système sur d'autres ordinateurs fournit un environnement parfait pour la propagation de ce code viral.

Un virus qui s'attache à un programme exécutable peut faire tout ce que le programme est autorisé à faire. Il s'exécute secrètement lorsque le programme hôte est exécuté. Une fois que le code du virus est en cours d'exécution, il peut exécuter n'importe quelle fonction, telle que l'effacement de fichiers et de programmes, qui est autorisée par les privilèges de l'utilisateur actuel. L'une des raisons pour lesquelles les virus ont dominé la scène des logiciels malveillants au cours des années précédentes était l'absence d'authentification de l'utilisateur et les contrôles d'accès aux systèmes informatiques personnels (Personnel Computers). Cela a permis à un virus d'infecter tout contenu exécutable du système. La quantité importante de programmes partagés sur disquette (Support de stockage très utilisé à l'époque mais actuellement obsolète) a également permis sa propagation facile, bien que quelque peu lente. L'inclusion de contrôles d'accès plus stricts sur les systèmes d'exploitation modernes entrave considérablement la facilité d'infection de ces virus traditionnels, à code machine exécutable. Cela a conduit au développement de macro-virus qui exploitent le contenu actif de certains types de documents, tels que les fichiers Microsoft Word ou Excel, ou les documents Adobe PDF. Ces documents sont facilement modifiés et partagés par les utilisateurs dans le cadre de leur utilisation normale du système, et ne sont pas protégés par les mêmes contrôles d'accès que les programmes.

Actuellement, un mode d'infection viral est généralement l'un des nombreux mécanismes de propagation utilisés par les logiciels malveillants contemporains, qui peuvent également inclure des vers, des chevaux de Troie, etc.

Composants d'un Virus

Communément admis et comme indiqué dans [AYCO06] un virus informatique est composé de trois parties. Plus généralement, de nombreux types de logiciels malveillants contemporains comprennent également une ou plusieurs variantes de chacun de ces composants.

- Mécanisme d'infection : Moyen par lequel un virus se propage, lui permettant de se reproduire. Ce mécanisme est également appelé vecteur d'infection
- Déclencheur : L'événement ou la condition qui détermine le moment où la charge virale est activée ou livrée, parfois appelée "bombe logique".
- Charge virale : Ce que fait le virus, en plus de se propager. La charge virale peut impliquer des dommages ou une activité bénigne mais perceptible.

Phases d'évolution du Virus

Au cours de sa vie, un virus typique passe par les quatre phases suivantes :

- Phase de dormance : Le virus est inactif. Le virus sera éventuellement activé par un événement quelconque, tel qu'une date, la présence d'un autre programme ou fichier, ou la capacité du disque dépassant une certaine limite. Tous les virus n'ont pas cette phase.
- Phase de propagation : Le virus place une copie de lui-même dans d'autres programmes ou dans certaines zones du système sur le disque. La copie peut ne pas être identique à la version de propagation ; les virus se transforment souvent pour échapper à la détection. Chaque programme infecté contiendra alors un clone du virus, qui entrera lui-même dans une phase de propagation.
- Phase de déclenchement : Le virus est activé pour remplir la fonction pour laquelle il a été conçu. Comme pour la phase de dormance, la phase de déclenchement peut être provoquée par divers événements du système, y compris un comptage du nombre de fois que cette copie du virus s'est répliquée.
- Phase d'exécution : La fonction est exécutée. La fonction peut être inoffensive, comme un message sur l'écran, ou dommageable, comme la destruction de programmes et de fichiers de données.

La plupart des virus qui infectent les fichiers de programmes exécutables effectuent leur travail d'une manière spécifique à un système d'exploitation particulier et, dans certains cas, à une plate-forme matérielle particulière. Ainsi, ils sont conçus pour tirer parti des détails et des faiblesses de systèmes particuliers. Les macro-virus, quant à eux, ciblent des types de documents spécifiques, qui sont souvent pris en charge par divers systèmes.

Virus des macros et des scripts

Au milieu des années 1990, les virus à code de macro ou de script sont devenus de loin le type de virus le plus répandu. Le NISTIR 7298 (Glossaire des termes clés de la sécurité de l'information, mai 2013) définit un macro-virus comme un virus qui s'attache aux documents et utilise les capacités de macro-programmation de l'application du document pour s'exécuter et se propager.

Les macro-virus infectent le code de script utilisé pour prendre en charge le contenu actif dans divers types de documents utilisateur. Les macrovirus sont particulièrement menaçants pour plusieurs raisons :

1. Un macrovirus est indépendant de la plate-forme. De nombreux macrovirus infectent le contenu actif des applications couramment utilisées, telles que les macros dans les documents Microsoft Word ou d'autres documents Microsoft Office, ou le code de script dans les documents Adobe PDF. Toute plate-forme matérielle et tout système d'exploitation qui prend en charge ces applications peut être infecté.
2. Les macro-virus infectent les documents, et non les portions de code exécutables. La plupart des informations introduites dans un système informatique se présentent sous la forme de documents plutôt que de programmes.
3. Les macro-virus se propagent facilement, car les documents qu'ils exploitent sont partagés dans le cadre d'une utilisation normale. Une méthode très courante est le courrier électronique, d'autant plus que ces documents peuvent parfois être ouverts automatiquement sans que l'utilisateur n'y soit invité.
4. Comme les macro-virus infectent les documents des utilisateurs plutôt que les programmes du système, les contrôles d'accès traditionnels aux systèmes de fichiers sont d'une utilité limitée pour empêcher leur propagation, puisque les utilisateurs sont censés les modifier.
5. Les macro-virus sont beaucoup plus faciles à écrire ou à modifier que les virus exécutables traditionnels. Les macro-virus tirent parti de la prise en charge du contenu actif au moyen d'un script ou d'un langage macro, intégré dans un document de traitement de texte ou un autre type de fichier.

En général, les utilisateurs utilisent des macros pour automatiser des tâches répétitives et ainsi économiser la tâche de saisie de texte (Frappes).
Elles sont également utilisées pour prendre en charge le contenu dynamique, la validation des formulaires et d'autres tâches utiles associées à ces documents.

Les documents Microsoft Word et Excel sont des cibles courantes en raison de leur utilisation répandue. Les versions successives des produits MS Office offrent une protection accrue contre les macro-virus. Par exemple, Microsoft propose un outil optionnel de protection contre les macro-virus qui détecte les fichiers Word suspects et avertit le client du risque potentiel d'ouverture d'un fichier avec des macros. À partir d'Office 2000, on a amélioré la sécurité des macros en permettant que celles-ci soient signées numériquement par leur auteur, et que les auteurs soient répertoriés comme étant de confiance.
Les utilisateurs étaient alors avertis si un document ouvert contenait des macros non signées, ou signées mais non de confiance, et il leur était conseillé de désactiver les macros dans ce cas. Divers fournisseurs de produits antivirus ont également mis au point des outils pour détecter et supprimer les macrovirus.
Comme pour les autres types de logiciels malveillants, la course et la concurrence se poursuivent dans le domaine des macrovirus, mais ceux-ci ne constituent plus la menace prédominante des logiciels malveillants.

Les documents PDF d'Adobe constituent un autre hôte possible pour les logiciels malveillants de type macro-virus. Ceux-ci peuvent prendre en charge toute une série de composants intégrés, notamment Javascript et d'autres types de code de script. Bien que les récents lecteurs de PDF comportent des mesures visant à avertir les utilisateurs lorsqu'un tel code est exécuté, le message affiché à l'utilisateur peut être manipulé pour les amener à autoriser son exécution. Si cela se produit, le code pourrait potentiellement agir comme un virus pour infecter d'autres documents PDF auxquels l'utilisateur peut accéder sur son système. Il peut également installer un cheval de Troie ou agir comme un ver.

Macro Virus Melissa

Bien que les macros langages puissent avoir une syntaxe similaire, les détails dépendent de l'application qui interprète la macro, et donc cibleront toujours les documents pour une application spécifique. Par exemple, une macro Microsoft Word, y compris une macro virus, sera différente d'une macro Excel. Les macros peuvent soit être enregistrées avec un document, soit être enregistrées dans un modèle global ou une feuille de calcul. Certaines macros sont exécutées automatiquement lorsque certaines actions se produisent. Dans Microsoft Word, par exemple, les macros peuvent s'exécuter lorsque Word démarre, qu'un document est ouvert, qu'un nouveau document est créé ou qu'un document est fermé. Les macros peuvent effectuer un large éventail d'opérations, non seulement sur le contenu du document, mais aussi lire et écrire des fichiers et appeler d'autres applications.

La figure suivante présente un exemple de fonctionnement d'un macrovirus : le pseudo-code du macrovirus Melissa. Il s'agit d'un composant du ver de messagerie Melissa que nous décrirons plus en détail dans la section suivante. Ce code serait introduit dans un système en ouvrant un document Word infecté, probablement envoyé par courrier électronique. Ce code de macro est contenu dans la macro Document_Open, qui est automatiquement exécutée lors de l'ouverture du document. Il désactive d'abord le menu Macro et certaines fonctions de sécurité connexes ; ce qui rend plus difficile pour l'utilisateur l'arrêt ou la suppression de son fonctionnement. Ensuite, le virus macro vérifie s'il est exécuté à partir d'un document infecté et, le cas échéant, se copie dans le fichier modèle global. Ce fichier est ouvert avec chaque document suivant, et le macro-virus s'exécute, infectant ce document. Il vérifie ensuite s'il a déjà été exécuté sur ce système, en regardant si une clé spécifique "Melissa" a été ajoutée au registre. Si cette clé est absente, et que Outlook est le client de messagerie, le macrovirus envoie alors une copie du document infecté actuel à chacune des 50 premières adresses du carnet d'adresses de l'utilisateur actuel. Il crée ensuite l'entrée de registre "Melissa", ce qui ne se fait donc qu'une seule fois sur un système quelconque. Enfin, il vérifie la date et l'heure courantes pour une condition de déclenchement spécifique, qui, si elle est remplie, entraîne l'insertion d'une citation de Simpson dans le document actuel.

Une fois le code de macro virus terminé, le document continue de s'ouvrir et l'utilisateur peut alors le modifier comme d'habitude. Ce code illustre la manière dont un macro-virus peut manipuler le contenu du document et accéder à d'autres applications du système. Il montre également deux mécanismes d'infection, le premier infectant chaque document

ouvert ultérieurement sur le système, le second envoyant des documents infectés à d'autres utilisateurs par courrier électronique.

Un code de macro-virus plus sophistiqué peut utiliser des techniques furtives telles que le cryptage ou le polymorphisme, en changeant à chaque fois son apparence, pour éviter la détection par balayage.

```
macro Document_Open
    disable Macro menu and some macro security features
    if called from a user document
       copy macro code into Normal template file
    else
       copy macro code into user document being opened
    end if
    if registry key "Melissa" not present
       if Outlook is email client
          for first 50 addresses in address book
              send email to that address
              with currently infected document attached
          end for
       end if
       create registry key "Melissa"
    end if
    if minute in hour equals day of month
       insert text into document being opened
    end if
end macro
```

Exemple de fonctionnement du macro virus Melissa [Stallings 2018]

Classification des Virus

Une course effrénée à la domination et au triomphalisme continue de s'imposer entre les auteurs de virus et les auteurs de logiciels anti-virus et ce, depuis l'apparition des virus. À mesure que des contre-mesures efficaces sont mises au point pour les types de virus existants, de nouveaux types sont développés. Il n'existe pas de système de classification simple ou universellement reconnu pour les virus. Dans cette section, nous suivons [AYCO06] et classons les virus selon deux axes orthogonaux : le type de cible que le virus essaie d'infecter et la méthode que le virus utilise pour se dissimuler aux utilisateurs et aux logiciels antivirus. Une classification des virus par cible comprend les catégories suivantes :

- Virus infectant le secteur de Boot[strap] : Infecte un enregistrement d'amorçage principal ou un enregistrement d'amorçage et se propage lorsqu'un système est démarré à partir du disque contenant le virus.
- Virus infectant des fichiers : Infecte les fichiers que le système d'exploitation ou le Shell considère comme exécutables.
- Macro-virus : Infecte les fichiers avec un code de macro ou de script qui est interprété par une application.
- Virus multipartite : Infecte les fichiers de plusieurs façons. En général, le virus multipartite est capable d'infecter plusieurs types de fichiers, de sorte que l'éradication du virus doit traiter tous les sites d'infection possibles.

Une classification des virus par stratégie de dissimulation comprend les catégories suivantes :

- Virus crypté : Une forme de virus qui utilise le cryptage pour masquer son contenu. Une partie du virus crée une clé de cryptage aléatoire et crypte le reste du virus. La clé est stockée avec le virus. Lorsqu'un programme infecté est invoqué, le virus utilise la clé aléatoire stockée pour décrypter le virus. Lorsque le virus se réplique, une clé aléatoire différente est sélectionnée. Comme la majeure partie du virus est cryptée avec une clé différente pour chaque instance, il n'y a pas de modèle binaire constant à observer.
- Virus furtif : Une forme de virus explicitement conçue pour se cacher de la détection par un logiciel antivirus. Ainsi, c'est l'ensemble du virus, et non pas seulement la charge, qui est caché. Il peut utiliser des techniques de mutation de code, de compression ou de rootkit pour y parvenir.
- Virus polymorphe : Forme de virus qui crée des copies lors de la réplication qui sont fonctionnellement équivalentes mais qui ont des configurations binaires nettement différentes. Ce comportement est fondé sur la stratégie permettant de vaincre les programmes qui recherchent les virus. Dans ce cas, la "signature" du virus varie avec chaque copie. Pour obtenir cette variation, le virus peut insérer au hasard des instructions superflues ou intervertir l'ordre des instructions. Une approche plus efficace consiste à utiliser le cryptage. La stratégie du virus de cryptage est suivie. La partie du virus qui est responsable de la génération des clés et de l'exécution du cryptage/décryptage est appelée moteur de mutation. Le moteur de mutation lui-même est modifié à chaque utilisation.

- Virus métamorphique : Comme un virus polymorphe, un virus métamorphique mute à chaque infection. La différence est qu'un virus métamorphique se réécrit complètement à chaque itération en utilisant de multiples techniques de transformation, ce qui augmente la difficulté de détection. Les virus métamorphiques peuvent modifier leur comportement ainsi que leur apparence.

Les Vers

La catégorie suivante de propagation de logiciels malveillants concerne l'exploitation des vulnérabilités des logiciels, telles que celles dont nous parlons aux chapitres 10 et 11, qui sont couramment exploitées par les vers informatiques. Un ver est un programme qui recherche activement d'autres machines à infecter, puis chaque machine infectée sert de rampe de lancement automatique pour les attaques sur d'autres machines. Les programmes de vers informatiques exploitent les vulnérabilités des logiciels des programmes clients ou serveurs pour accéder à chaque nouveau système. Ils peuvent utiliser des connexions réseau pour se propager d'un système à l'autre. Ils peuvent également se propager par le biais de supports partagés, tels que les clés USB ou les disques de données CD et DVD. Les vers de messagerie électronique se propagent dans le code de macro ou de script inclus dans les documents joints aux courriers électroniques ou aux transferts de fichiers par messagerie instantanée. Lors de son activation, le ver peut se répliquer et se propager à nouveau. En plus de se propager, le ver transporte généralement une certaine forme de charge virale, comme celles dont nous en parlerons plus loin.

Le concept de ver informatique a été introduit en 1975 dans le roman de John Brunner, The Shockwave Rider. La première implémentation connue d'un ver a été réalisée dans les laboratoires Xerox Palo Alto au début des années 1980. Il était non malveillant et recherchait des systèmes inactifs pour exécuter une tâche informatique intensive.

Réplication des vers

Pour se répliquer, un ver utilise des moyens d'accéder à des systèmes distants. Parmi ces moyens, on peut citer les suivants, dont la plupart sont encore utilisés activement :

- Courrier électronique ou messagerie instantanée : Un ver envoie une copie de lui-même par courrier électronique à d'autres systèmes, ou s'envoie en pièce jointe via un service de messagerie instantanée, de sorte que son code soit exécuté lorsque le courrier électronique ou la pièce jointe ait été reçu ou consulté.
- Partage de fichiers : Un ver crée une copie de lui-même ou infecte d'autres fichiers appropriés comme un virus sur un support amovible tel qu'une clé USB ; il s'exécute ensuite lorsque la clé est connectée à un autre système en utilisant le mécanisme d'exécution automatique en exploitant une vulnérabilité du logiciel, ou lorsqu'un utilisateur ouvre le fichier infecté sur le système cible.

- Capacité d'exécution à distance : Un ver exécute une copie de lui-même sur un autre système, soit en utilisant une fonction explicite d'exécution à distance, soit en exploitant une faille de programme dans un service de réseau pour détourner ses opérations (comme nous le verrons dans les chapitres 10 et 11).
- Capacité d'accès ou de transfert de fichiers à distance : Un ver utilise un service d'accès ou de transfert de fichiers à distance vers un autre système pour se copier d'un système à l'autre, où les utilisateurs de ce système peuvent ensuite l'exécuter.
- Capacité de connexion à distance : Un ver se connecte à un système distant en tant qu'utilisateur et utilise ensuite des commandes pour se copier d'un système à l'autre, où il s'exécute ensuite. La nouvelle copie du programme du ver est ensuite exécutée sur le système distant où, en plus des fonctions utiles qu'il exécute sur ce système, il continue à se propager.

Un ver utilise généralement les mêmes phases qu'un virus informatique : dormance, propagation, déclenchement et exécution. La phase de propagation remplit généralement les fonctions suivantes :

- Recherche de mécanismes d'accès appropriés à d'autres systèmes à infecter en examinant les tables d'hôtes, les carnets d'adresses, les listes d'amis, les pairs de confiance et autres dépôts similaires de détails d'accès à distance au système ; en analysant les adresses d'hôtes cibles possibles ; ou en recherchant des supports amovibles appropriés à utiliser.
- Utilisez les mécanismes d'accès trouvés pour transférer une copie de lui-même au système distant, et faites en sorte que la copie soit exécutée. Le ver peut également tenter de déterminer si un système a déjà été infecté avant de se copier sur le système. Dans un système multiprogrammé, il peut également dissimuler sa présence en se désignant comme un processus système ou en utilisant un autre nom qui peut ne pas être remarqué par un opérateur système. Des vers plus récents peuvent même injecter leur code dans des processus existants sur le système, et s'exécuter en utilisant des threads supplémentaires dans ce processus, afin de dissimuler davantage leur présence.

3. Propagation - vulnérabilité - exploit - vers

Découverte de cibles

La première fonction d'un ver de réseau dans la phase de propagation est de rechercher d'autres systèmes à infecter, un processus connu sous le nom de balayage ou d'empreinte digitale. Pour ces vers, qui exploitent les vulnérabilités des logiciels dans les services de réseau accessibles à distance, il doit identifier les systèmes potentiels qui exécutent le service vulnérable, puis les infecter. Ensuite, en général, le code du ver installé sur les machines infectées répète le même processus de balayage, jusqu'à ce qu'un vaste réseau distribué de machines infectées soit créé. [MIRK04] énumère les types suivants de stratégies d'analyse d'adresses réseau qu'un tel ver peut utiliser :

- Aléatoire : Chaque hôte compromis sonde des adresses aléatoires dans l'espace d'adresses IP, en utilisant une semence différente. Cette technique produit un volume

élevé de trafic Internet, qui peut provoquer une perturbation généralisée avant même le lancement de l'attaque proprement dite.

- Hit-List : L'attaquant compile d'abord une longue liste de machines potentiellement vulnérables. Ce processus peut être lent et s'étaler sur une longue période afin d'éviter qu'une attaque soit détectée. Une fois la liste compilée, l'attaquant commence à infecter les machines de la liste. Chaque machine infectée reçoit une partie de la liste à scanner. Cette stratégie se traduit par une période d'analyse très courte, ce qui peut rendre difficile la détection de l'infection.
- Topologique : cette méthode utilise les informations contenues sur une machine victime infectée pour trouver d'autres hôtes à scanner.
- Sous-réseau local : Si un hôte peut être infecté derrière un pare-feu, cet hôte cherche alors des cibles dans son propre réseau local. L'hôte utilise la structure d'adresse du sous-réseau pour trouver d'autres hôtes qui seraient autrement protégés par le pare-feu.

Modèle de propagation des vers

Un ver bien conçu peut se propager rapidement et infecter un grand nombre d'hôtes. Il est utile de disposer d'un modèle général de la vitesse de propagation du ver. Les virus et les vers informatiques présentent un comportement d'autoréplication et de propagation similaire à celui des virus biologiques. Nous pouvons donc nous tourner vers les modèles épidémiques classiques pour comprendre le comportement de propagation des virus informatiques et des vers.

La figure suivante montre la dynamique de la propagation des vers à l'aide de ce modèle. La propagation se déroule en trois phases. Dans la phase initiale, le nombre d'hôtes augmente de façon exponentielle. Pour s'en rendre compte, considérons un cas simplifié dans lequel un ver est lancé à partir d'un seul hôte et infecte deux hôtes voisins. Chacun de ces hôtes infecte deux autres hôtes, et ainsi de suite. Il en résulte une croissance exponentielle. Au bout d'un certain temps, les hôtes infectants perdent du temps à attaquer des hôtes déjà infectés, ce qui réduit le taux d'infection. Pendant cette phase intermédiaire, la croissance est à peu près linéaire, mais le taux d'infection est rapide. Lorsque la plupart des ordinateurs vulnérables ont été infectés, l'attaque entre dans une phase de fin lente, car le ver cherche les hôtes restants qui sont difficiles à identifier.

Il est clair que l'objectif de la lutte contre un ver est d'attraper le ver dans sa phase de démarrage lent, à un moment où peu d'hôtes ont été infectés. Zou et al [ZOU05] décrivent un modèle de propagation des vers basé sur une analyse des attaques de vers de réseau à ce moment-là. La vitesse de propagation et le nombre total d'hôtes infectés dépend d'un certain nombre de facteurs, notamment du mode de propagation, de la ou des vulnérabilités exploitées et du degré de similitude avec les attaques précédentes. Pour ce dernier facteur, une attaque qui est une variation d'une attaque précédente récente peut être contrée plus efficacement qu'une attaque plus nouvelle.

Modèle classique de propagation virale

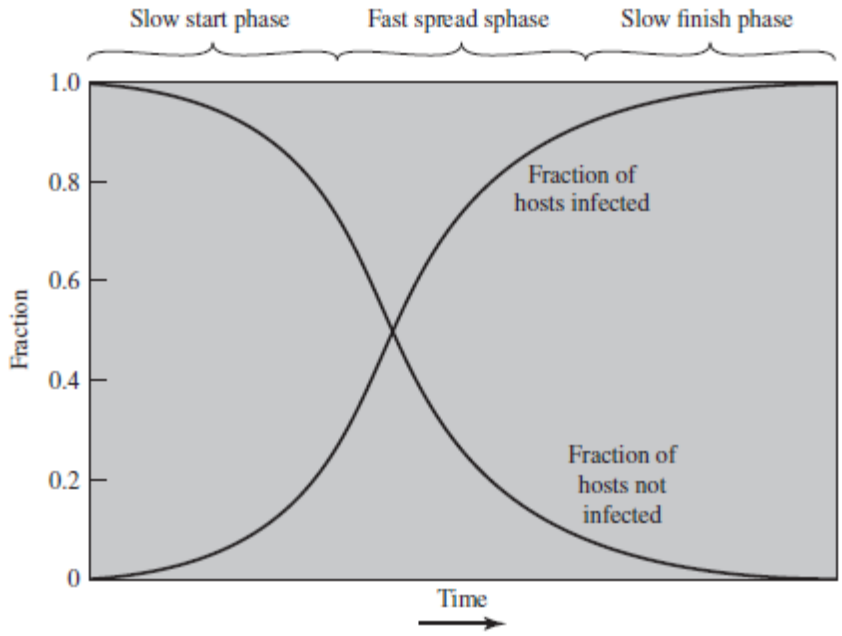
Un ver bien conçu peut se propager rapidement pour infecter un grand nombre de machines (Hosts). Il est donc utile d'avoir un modèle général du taux de propagation. Les virus et les vers suivent un comportement de réplication et propagation similaire aux virus biologiques. Il suffit d'examiner un modèle épidémique réel, par exemple covid-19, pour comprendre celui d'un malware. Ainsi, un modèle classique et simplifié d'une épidémie ou pandémie peut être exprimé comme suit :

$dI(t)/dt = \beta(t)*S(t)$ où I(t) = nombre d'individus infectés au temps t

S(t) = nombre d'individus susceptibles d'être infectés à l'instant t

β = Taux d'infection et N = taille de la population, N = I(t) + S(t)

La figure suivante montre l'évolution de la propagation d'un ver selon ce modèle. La propagation évolue en trois phases : Phase initiale avec progression exponentielle du nombre de machines infectées ; phase médiane avec progression presque linéaire mais avec un taux d'infection rapide. La phase finale correspond à une décroissance de la courbe d'infection car le ver arrive difficilement à trouver des hôtes non contaminés. La contre mesure consiste à identifier le malware dans la phase avec peu d'infections.



Modèle de propagation virale (Stallings 2018)

Première Infection Virale : Le ver Morris

On peut dire que la première infection importante, et donc bien connue, par un ver a été diffusée sur Internet par Robert Morris en 1988 [ORMA03]. Le ver Morris a été conçu pour se propager sur les systèmes UNIX et a utilisé un certain nombre de techniques de propagation différentes. Lorsqu'une copie commençait à s'exécuter, sa première tâche consistait à découvrir d'autres hôtes connus de cet hôte qui permettraient l'entrée de ce dernier. Le ver exécutait cette tâche en examinant diverses listes et tables, y compris des tables système qui déclaraient quelles autres machines étaient fiables pour cet hôte, des fichiers de transfert de courrier des utilisateurs, des tables par lesquelles les utilisateurs se donnaient la permission d'accéder à des comptes distants, et à partir d'un programme qui signalait l'état des connexions réseau. Pour chaque hôte découvert, le ver a essayé un certain nombre de méthodes pour y accéder :

1. Il a tenté de se connecter à un hôte distant en tant qu'utilisateur légitime. Dans cette méthode, le ver a d'abord tenté de craquer le fichier de mots de passe local, puis il a utilisé les mots de passe découverts et les identifiants d'utilisateur correspondants. L'hypothèse était que de nombreux utilisateurs utiliseraient le même mot de passe sur différents systèmes. Pour obtenir les mots de passe, le ver a lancé un programme de craquage de mots de passe qui a essayé :
   a. Le nom de compte de chaque utilisateur et les permutations simples de celui-ci
   b. Une liste de 432 mots de passe intégrés que M. Morris pensait être des candidats probables
   c. Tous les mots du dictionnaire du système local
2. Il a exploité un bug dans le protocole UNIX finger qui signale les déplacements d'un utilisateur distant.
3. Il a exploité une trappe dans l'option de débogage du processus distant qui reçoit et envoie du courrier. Si l'une de ces attaques réussissait, le ver parvenait à communiquer avec l'interpréteur de commandes du système d'exploitation. Il envoyait alors à cet interpréteur un court programme d'amorçage, émettait une commande pour exécuter ce programme, puis se déconnectait. Le programme d'amorçage a ensuite rappelé le programme parent et a téléchargé le reste du ver. Le nouveau ver a ensuite été exécuté.

Attaques significatives de vers

Le ver de messagerie Melissa, apparu en 1998, était le premier d'une nouvelle génération de logiciels malveillants comprenant des aspects de virus, de ver et de cheval de Troie dans un seul et même paquet [CASS01]. Melissa utilisait une macro Microsoft Word intégrée dans une pièce jointe. Si le destinataire ouvre la pièce jointe du courriel, la macro Word est activée. Ensuite, elle

1. Envoie une copie d'elle-même à tous les membres de la liste de diffusion dans le paquet de courrier électronique de l'utilisateur, se propageant comme un ver ; et
2. Provoque des dommages locaux sur le système de l'utilisateur, y compris la désactivation de certains outils de sécurité, et se copie également dans d'autres documents, se propageant comme un virus ; et
3. Si un temps de déclenchement était prévu, il affichait une citation de Simpson. En 1999, une version plus puissante de ce virus de courrier électronique est apparue. Cette version pouvait être activée simplement en ouvrant un e-mail contenant le virus, plutôt qu'en ouvrant une pièce jointe. Le virus utilise le langage de script Visual Basic pris en charge par le logiciel de messagerie électronique.

Melissa se propage dès qu'il est activé (soit en ouvrant une pièce jointe, soit en ouvrant le courrier électronique) à toutes les adresses électroniques connues de l'hôte infecté. Par conséquent, alors qu'auparavant, les virus mettaient des mois ou des années à se propager, cette nouvelle génération de logiciels malveillants peut le faire en quelques heures. [CASS01] note qu'il n'a fallu que trois jours à Melissa pour infecter plus de 100 000 ordinateurs, alors qu'il avait fallu des mois au virus Brain pour infecter quelques milliers d'ordinateurs une décennie auparavant.
Il est donc très difficile pour les logiciels antivirus de répondre à de nouvelles attaques avant que les dégâts ne soient importants.

Le ver Code Red est apparu pour la première fois en juillet 2001. Il exploite une faille de sécurité dans le serveur d'information Internet (IIS) de Microsoft pour pénétrer et se propager. Il désactive également le vérificateur de fichiers système dans Windows. Le ver sonde des adresses IP aléatoires pour se propager à d'autres hôtes. Pendant un certain temps, il ne fait que se propager. Il lance alors une attaque par déni de service contre un site web gouvernemental en inondant le site de paquets provenant de nombreux hôtes. Le ver suspend alors ses activités et se réactive périodiquement. Lors de la deuxième vague d'attaques, Code Red a infecté près de 360 000 serveurs en 14 heures. En plus des ravages qu'il a causés sur le serveur ciblé, Code Red a consommé d'énormes quantités de ressources Internet, perturbant ainsi le fonctionnement de ses services [MOOR02].

Code Red II est une autre variante distincte qui est apparue pour la première fois en août 2001 et visait également Microsoft IIS. Il a tenté d'infecter les systèmes du même sous-

réseau que le système infecté. De plus, ce nouveau ver installe une porte dérobée (Back door), permettant à un pirate informatique d'exécuter à distance des commandes sur les ordinateurs victimes.

Le ver Nimda, apparu en septembre 2001, présente également les caractéristiques d'un ver, d'un virus et d'un code mobile. Il s'est propagé en utilisant diverses méthodes de distribution :

- Le courrier électronique : Un utilisateur sur un hôte vulnérable ouvre une pièce jointe de courriel infectée ; Nimda recherche des adresses de courriel sur l'hôte et envoie ensuite des copies de lui-même à ces adresses.

- Partages de fichiers Windows : Nimda analyse les hôtes à la recherche des fichiers partagés Windows non sécurisés ; il peut ensuite utiliser NetBIOS86 comme mécanisme de transport pour infecter les fichiers sur cet hôte dans l'espoir qu'un utilisateur exécute un fichier infecté, ce qui activera Nimda sur cet hôte.
- Serveurs web : Nimda scanne les serveurs Web, à la recherche de vulnérabilités connues dans Microsoft IIS. S'il trouve un serveur vulnérable, il tente de transférer une copie de lui-même sur le serveur et l'infecte ainsi que ses fichiers.
- Clients Web : Si un client Web vulnérable visite un serveur Web qui a été infecté par Nimda, le poste de travail du client sera infecté à son tour.
- Portes dérobées : Si un poste de travail a été infecté par des vers antérieurs, tels que "Code Red II", Nimda utilisera la porte dérobée laissée par ces infections antérieures pour accéder au système.

Le ver SQL Slammer est apparu au début de 2003. Ce ver exploitait la vulnérabilité de débordement du tampon (Buffer overflow) dans le serveur SQL de Microsoft. Slammer était extrêmement compact et s'est propagé rapidement, infectant 90 % des hôtes vulnérables en 10 minutes. Cette propagation rapide a provoqué une importante congestion sur Internet.

Fin 2003, l'infection du ver Sobig.F est déclenchée, exploitant les serveurs proxy ouverts pour transformer les machines infectées en moteurs de spam. À son apogée, Sobig.F aurait représenté un message sur 17 transmis sur Internet et aurait produit plus d'un million de copies de lui-même dans les premières 24 heures.

Mydoom est un ver de messagerie électronique de masse apparu en 2004. Il a suivi une tendance croissante consistant à installer une porte dérobée dans les ordinateurs infectés, permettant ainsi aux pirates d'accéder à distance à des données telles que des mots de passe et des numéros de cartes de crédit. Mydoom se répliquait jusqu'à 1 000 fois par minute et aurait inondé l'Internet de 100 millions de messages infectés en 36 heures.

La famille de vers Warezov est apparue en 2006 [KIRK06]. Lorsque le ver est lancé, il crée plusieurs exécutables dans les répertoires du système et se met à fonctionner à chaque

démarrage de Windows en créant une entrée dans le registre. Warezov scanne plusieurs types de fichiers à la recherche d'adresses e-mail et envoie une copie de lui-même en tant que pièce jointe d'un courriel. Certaines variantes sont capables de télécharger d'autres logiciels malveillants, tels que les chevaux de Troie et les logiciels publicitaires. De nombreuses variantes désactivent les produits liés à la sécurité et/ou désactivent leur capacité de mise à jour

Le ver Conficker (ou Downadup) a été détecté pour la première fois en novembre 2008 et s'est rapidement propagé pour devenir l'une des infections les plus répandues depuis SQL Slammer en 2003 [LAWT09]. Il s'est d'abord propagé en exploitant la vulnérabilité de débordement de la mémoire tampon de Windows, alors que les versions ultérieures aient pu également se propager via les clés USB et les partages de fichiers sur le réseau. En 2010, elle constituait toujours la deuxième famille de logiciels malveillants la plus répandue observée par Symantec [SYMA16], même si des correctifs étaient disponibles auprès de Microsoft pour fermer les principales vulnérabilités qu'elle exploite.

En 2010, le ver Stuxnet a été détecté, bien qu'il ait été répandu discrètement depuis un certain temps déjà [CHEN11, KUSH13]. Contrairement à de nombreux vers précédents, il a délibérément limité sa vitesse de propagation pour réduire ses chances de détection. Il a également ciblé les systèmes de contrôle industriels, très probablement ceux associés au programme nucléaire iranien, dans le but probable de perturber le fonctionnement de leurs équipements. Il a soutenu une série de mécanismes de propagation, notamment via des clés USB, des partages de fichiers en réseau et en utilisant pas moins de quatre exploits de vulnérabilité inconnus et de type "zéro jour". La taille et la complexité de son code, l'utilisation de quatre exploits sans précédent de type "zero-day", ainsi que le coût et les efforts apparents dans son développement ont suscité un débat considérable.

Certains experts prétendent qu'il s'agissait de la première utilisation sérieuse d'une arme de cyberguerre contre l'infrastructure physique d'une nation. Les chercheurs de Symantec qui ont analysé Stuxnet ont noté que s'ils s'attendaient à trouver de l'espionnage, ils ne s'attendaient pas à voir des logiciels malveillants ayant pour but le sabotage ciblé. En conséquence, une plus grande attention s'est maintenant portée à l'utilisation des logiciels malveillants comme arme par un certain nombre de nations.

Fin 2011, le ver Duqu a été découvert, qui utilise un code apparenté à celui de Stuxnet. Son objectif est différent, il s'agit de cyber-espionnage, bien qu'il semble également viser le programme nucléaire iranien. Un autre ver de cyber-espionnage important et récent est la famille Flame, qui a été découverte en 2012 et semble cibler les pays du Moyen-Orient. Malgré les zones cibles spécifiques de ces différents vers, leurs stratégies d'infection ont été si efficaces qu'ils ont été identifiés sur des systèmes informatiques dans un très grand nombre de pays, y compris sur des systèmes maintenus physiquement isolés de l'Internet général. Cela renforce la nécessité d'améliorer considérablement les contre-mesures pour résister à ces infections.

Malware WannaCry

En mai 2017, l'attaque du logiciel malveillant de rançonnment WannaCry s'est propagée d'une extrême rapidité sur une période de quelques heures à quelques jours, infectant des centaines de milliers de systèmes appartenant à des organisations publiques et privées dans plus de 150 pays (US-CERT Alert TA17-132A) [BON17]. Le code malveillant s'est propagé comme un ver en analysant agressivement les réseaux locaux et les réseaux distants aléatoires, en essayant d'exploiter la vulnérabilité du service de partage de fichiers SMB sur les systèmes Windows non patchés. Cette propagation rapide n'a été ralentie que par l'activation accidentelle d'un domaine "kill-switch" par un chercheur en sécurité britannique, dont l'existence a été vérifiée dans les versions initiales de ce logiciel malveillant. Une fois installé sur les systèmes infectés, il a chiffré les fichiers, exigeant le paiement d'une rançon pour les récupérer.

Technologie des vers

L'état de l'art en matière de technologie des vers comprend les diverses caractéristiques suivantes communément admises :

- Multiplateforme : Les vers les plus récents ne se limitent pas aux machines fonctionnant sur les systèmes Windows, mais peuvent aussi attaquer une variété de plates-formes, en particulier les variétés bien connues des systèmes de la famille UNIX ; ou exploiter des macros ou des langages de script pris en charge dans les types de documents les plus courants.
- Multi-exploitation : Les nouveaux vers pénètrent dans les systèmes de diverses manières, en utilisant des exploits contre les serveurs Web, les navigateurs, le courrier électronique, le partage de fichiers et d'autres applications en réseau, ou via des médias partagés.
- Propagation ultra-rapide : Exploiter diverses techniques pour optimiser la vitesse de propagation d'un ver afin de maximiser ses chances de localiser le plus grand nombre possible de machines vulnérables dans un court laps de temps. Le succès remporté par un malware est d'autant plus important que sa rapidité de propagation et d'infection est fulgurante. Plus rapidement le malware est détecté, plus la fin de sa propagation est imminente. Par analogie, une fois le virus biologique Covid-19 a été identifié, un ensemble de mesures de protection ont été mises en place pour freiner sa progression ; et en attendant, un nombre important d'équipes de recherche se sont lancées à la recherche d'un vaccin pour pouvoir s'immuniser.
- Polymorphe : Pour échapper à la détection, passer les filtres et déjouer l'analyse en temps réel, les vers adoptent la technique du virus polymorphe. Chaque copie du ver

possède un nouveau code généré à la volée à l'aide d'instructions et de techniques de cryptage fonctionnellement équivalentes.

- Métamorphique : En plus de modifier leur apparence, les vers métamorphiques ont un répertoire de comportements qui se déclenchent à différents stades de la propagation.
- Vecteurs de transport : Comme les vers peuvent rapidement compromettre un grand nombre de systèmes, ils sont idéaux pour propager une grande variété de charges virales malveillantes, telles que les bots de déni de service distribués, les rootkits, les générateurs de courrier électronique indésirable et les logiciels espions.
- Exploit à zéro-jour: Pour obtenir un maximum de surprise et de distribution, un ver doit exploiter une vulnérabilité inconnue qui n'est découverte par la communauté générale du réseau qu'au moment du lancement du ver. En 2015, 54 exploits zéro-jour ont été découverts et exploités, soit beaucoup plus que les années précédentes [SYMA16]. Nombre de ces exploits concernaient des logiciels informatiques et mobiles courants. Certains, cependant, se trouvaient dans des bibliothèques et des progiciels de développement communs, et d'autres dans des systèmes de contrôle industriels. Cela indique l'éventail des systèmes visés.

Code mobile

La norme NIST SP 800-28 (Guidelines on Active Content and Mobile Code, mars 2008) définit le code mobile comme des programmes (Par exemple, un script, une macro ou un autre programme portable) qui peuvent être envoyés tels quels à un ensemble hétérogène de plateformes et exécutés avec une sémantique identique.

Le code mobile est transmis d'un système distant à un système local, puis exécuté sur le système local sans instruction explicite de l'utilisateur [SOUP13]. Le code mobile agit souvent comme un mécanisme permettant de transmettre un virus, un ver ou un cheval de Troie au poste de travail de l'utilisateur. Dans d'autres cas, le code mobile profite des vulnérabilités pour réaliser ses propres exploits, comme l'accès non autorisé aux données ou la compromission de la racine.

Les moyens de support les plus courants pour le code mobile sont les applets Java, ActiveX, JavaScript et VBScript. Les moyens les plus courants d'utiliser le code mobile pour des opérations malveillantes sur un système local sont les scripts intersites, les sites web interactifs et dynamiques, les pièces jointes aux courriels et les téléchargements de sites ou de logiciels non fiables.

Vers de Smartphones (Téléphones portables)

Les vers sont apparus pour la première fois sur les téléphones portables avec la découverte du ver Cabir en 2004, puis de Lasco et CommWarrior en 2005. Ces vers communiquent via des connexions sans fil Bluetooth ou via le service de messagerie multimédia (MMS). La cible est le smartphone, qui est un téléphone mobile permettant aux utilisateurs d'installer des applications logicielles provenant de sources autres que l'opérateur du réseau cellulaire. Tous ces premiers vers mobiles visaient les téléphones portables utilisant le système d'exploitation Symbian. Des logiciels malveillants plus récents ciblent les systèmes Android et iPhone.

Les logiciels malveillants des téléphones mobiles peuvent désactiver complètement le téléphone, supprimer des données sur le téléphone ou forcer l'appareil à envoyer des messages coûteux à des numéros surtaxés.

Le ver CommWarrior se réplique au moyen de Bluetooth vers d'autres téléphones dans la zone de réception. Il envoie également une copie de lui-même sous forme de fichier MMS vers les numéros du carnet d'adresses du téléphone et dans les réponses automatiques aux messages texte et MMS entrants. En outre, il se copie sur la carte mémoire amovible et s'insère dans les fichiers d'installation du programme sur le téléphone. Bien que ces exemples montrent que les vers de téléphones portables sont possibles, la grande majorité des logiciels malveillants de téléphones portables observés utilisent des applications de cheval de Troie pour s'y installer [SYMA16].

Téléchargements Furtifs

Une autre approche de l'exploitation des vulnérabilités des logiciels consiste à exploiter les bogues des applications des utilisateurs pour installer des logiciels malveillants. Une technique courante consiste à exploiter les vulnérabilités des navigateurs de sorte que lorsque l'utilisateur consulte une page web contrôlée par l'attaquant, celle-ci contient un code qui exploite le bogue du navigateur pour télécharger et installer un logiciel malveillant sur le système à l'insu de l'utilisateur ou sans son consentement. Cette technique est connue sous le nom de "drive-by-download" et constitue un exploit (Code spécifique pour vulnérabilité) courant dans les kits d'attaque récents. De multiples vulnérabilités dans les plugins Adobe Flash Player et Oracle Java ont été mis à profit par les attaquants depuis de nombreuses années, au point que de nombreux navigateurs en suppriment aujourd'hui la prise en charge. Dans la plupart des cas, ce logiciel malveillant ne se propage pas activement comme le fait un ver, mais attend plutôt que des utilisateurs sans méfiance visitent la page web malveillante pour se propager à leurs systèmes [SYMA16].

Attaques par trous d'eau (Watering-Hole Attacks)

En général, les attaques par téléchargement visent tout utilisateur qui visite un site compromis et qui est vulnérable aux exploits utilisés. Les attaques par trou d'eau en sont une variante utilisée dans les attaques très ciblées. L'attaquant fait des recherches sur ses victimes afin d'identifier les sites web qu'elles sont susceptibles de visiter, puis scanne ces sites pour identifier ceux dont les vulnérabilités permettent de les compromettre avec une attaque par téléchargement. Il attend ensuite que l'une de ses victimes se rende sur l'un des sites compromis. Leur code d'attaque peut même être écrit de manière à n'infecter que les systèmes appartenant à l'organisation cible, et ne prendre aucune mesure pour les autres visiteurs du site. Cela augmente considérablement la probabilité que la compromission du site ne soit pas détectée.

L'attaque de point d'eau ou trou d'eau qui utilise des moyens détournés pour parvenir à ses fins
est généralement employée contre des entreprises privées ou des institutions travaillant sur des secteurs sensibles et qui disposent de systèmes informatiques hautement protégés et difficiles à attaquer.

Malvertising ou Publicité Compromettante

Maladvertising est une autre technique utilisée pour placer des logiciels malveillants sur les sites web sans les compromettre réellement. L'attaquant paie pour des publicités qui ont de fortes chances d'être placées sur les sites web cibles et qui contiennent des logiciels malveillants. En utilisant ces annonces malveillantes, les attaquants peuvent infecter les visiteurs des sites qui les affichent. Là encore, le code malveillant peut être généré dynamiquement pour réduire les chances de détection ou pour infecter uniquement des systèmes spécifiques.

La Malvertising a connu une croissance rapide ces dernières années, car il est facile de les placer sur les sites web souhaités en posant peu de questions, et ils sont difficiles à suivre. Les attaquants placent ces publicités pour quelques heures seulement, alors qu'ils s'attendaient à ce que leurs victimes puissent naviguer sur les sites web ciblés, réduisant ainsi considérablement leur visibilité [SYMA16].

D'autres logiciels malveillants peuvent cibler les lecteurs de type PDF courants afin de télécharger et d'installer des logiciels malveillants sans le consentement de l'utilisateur lorsqu'il consulte un document PDF malveillant [STEV11]. Ces documents peuvent être diffusés par des courriers électroniques non sollicités ou faire partie d'une attaque de phishing ciblée.

Le détournement de clics

Le clickjacking, également connu sous le nom d'attaque de réparation de l'interface utilisateur (IU), est une vulnérabilité utilisée par un attaquant pour collecter les clics d'un utilisateur infecté. L'attaquant peut forcer l'utilisateur à faire diverses choses, allant du réglage des paramètres de son ordinateur à l'envoi involontaire de l'utilisateur vers des sites web qui pourraient contenir des codes malveillants. De plus, en tirant parti d'Adobe Flash ou de JavaScript, un attaquant peut même placer un bouton sous ou au-dessus d'un bouton légitime, ce qui le rend difficile à détecter pour les utilisateurs. Une attaque typique utilise plusieurs couches transparentes ou opaques pour inciter un utilisateur à cliquer sur un bouton ou un lien d'une autre page alors qu'il avait l'intention de cliquer sur la page de niveau supérieur. Ainsi, l'attaquant détourne les clics destinés à une page et les dirige vers une autre page, qui appartient très probablement à une autre application, à un autre domaine ou aux deux. En utilisant une technique similaire, les frappes peuvent également être détournées. Grâce à une combinaison soigneusement élaborée de feuilles de style, d'iframes et de zones de texte, un utilisateur peut être amené à croire qu'il tape le mot de passe de son courriel ou de son compte bancaire, mais qu'il tape au contraire dans un cadre invisible contrôlé par l'attaquant.

Il existe une grande variété de techniques pour réaliser une attaque de type "clickjacking", et de nouvelles techniques sont développées au fur et à mesure que des défenses sont mises en place contre les anciennes techniques. Voir [NIEM11] et [STON10] qui sont des références utiles.

L'ingénierie sociale

La dernière catégorie de propagation de logiciels malveillants que nous considérons concerne l'ingénierie sociale (Social engineering), qui consiste à piéger, ou induire en erreur, les utilisateurs pour les entrainer à compromettre leurs propres systèmes ou leurs informations personnelles. Cela peut se produire lorsqu'un utilisateur consulte et répond à un courrier électronique non sollicité, ou lorsqu'il autorise l'installation et l'exécution d'un programme ou d'un code de script de cheval de Troie. Cela peut se produire dans les situations suivantes :

1. Courrier électronique non sollicité (spam)

Avec la croissance explosive de l'internet au cours des dernières décennies (Ou de IET : Internet de toute chose : Internet of Every Thing ou IoT : Internet of Things, en voie de supplanter l'internet classique), l'utilisation généralisée du courrier électronique et le coût extrêmement faible requis pour envoyer de gros volumes de courrier électronique, s'est imposée l'émergence de l'augmentation du courrier électronique non sollicité en masse, communément appelé spam.

Les experts de la sécurité [SYMA16] note que plus de la moitié du trafic entrant du courrier électronique professionnel représente du spam, malgré un déclin progressif ces

dernières années. Cela impose des coûts importants à la fois sur l'infrastructure de réseau nécessaire pour relayer ce trafic, et sur les utilisateurs qui doivent filtrer leurs courriers électroniques légitimes de cette inondation. En réponse à cette croissance explosive, on a assisté à une croissance tout aussi rapide de l'industrie anti-spam qui fournit des produits permettant de détecter et de filtrer les courriers électroniques non sollicités. Cela a conduit à un défit, à relever à la fois, par les spammeurs qui mettent au point des techniques pour faire passer en douce leur contenu et, par les défenseurs, qui déploient les efforts pour les bloquer [KREI09]. Cependant, le problème du spam persiste, car les spammeurs exploitent d'autres moyens pour atteindre leurs victimes. Cela inclut l'utilisation des médias sociaux, ce qui reflète la croissance rapide de l'utilisation de ces réseaux. Par exemple, les auteurs de [SYMA16] ont décrit une campagne de spam réussie pour la perte de poids qui exploitait des centaines de milliers de faux comptes Twitter. Ces comptes se soutenaient et se renforçaient mutuellement, afin d'augmenter leur crédibilité et la probabilité que les utilisateurs les suivent, et tombent ensuite dans le piège de l'escroquerie. Les escroqueries sur les réseaux sociaux reposent souvent sur le partage des arnaques par les victimes crédules ou sur de fausses offres assorties d'incitations, afin de favoriser leur diffusion. Si certains courriers électroniques non sollicités sont envoyés par des serveurs de messagerie légitimes utilisant des identifiants d'utilisateur volés, la plupart des courriers non sollicités sont envoyés par des réseaux de bots utilisant des systèmes d'utilisateurs compromis.

Une part importante du contenu des courriers électroniques non sollicités n'est que de la publicité naïve. Elle vise à convaincre le destinataire d'acheter un produit en ligne, comme par exemple, des produits pharmaceutiques, les escroqueries boursières, romantiques ou de faux commerçants, ou les offres d'emploi de transport illicite d'argent (Money mules). Mais le spam est également un vecteur important de logiciels malveillants. Le courrier électronique peut contenir un document joint qui, s'il est ouvert, peut exploiter une vulnérabilité logicielle pour installer un logiciel malveillant sur le système de l'utilisateur. Il peut également contenir un cheval de Troie ou un code de script qui, s'il est exécuté, installe également un logiciel malveillant sur le système de l'utilisateur.

Certains chevaux de Troie évitent de solliciter l'accord de l'utilisateur en exploitant une vulnérabilité du logiciel pour s'installer. Enfin, le spam peut être utilisé dans une attaque de phishing, qui dirige généralement l'utilisateur soit vers un faux site web qui reflète un service légitime, tel qu'un site de banque en ligne, où il tente de saisir les détails du login et du mot de passe de l'utilisateur ; soit pour remplir un formulaire avec suffisamment de détails personnels pour permettre à l'attaquant de se faire passer pour l'utilisateur dans une usurpation d'identité.

Ces dernières années, l'évolution du marché criminel a facilité les campagnes d'hameçonnage en vendant des forfaits aux escrocs qui automatisent largement le processus d'exécution de l'escroquerie [SYMA16]. Toutes ces utilisations font des courriers électroniques non sollicités un problème de sécurité important. Cependant,

dans de nombreux cas, il nécessite un choix actif de l'utilisateur pour consulter le courrier électronique et tout document joint, ou pour permettre l'installation d'un programme, afin que le compromission se produise. D'où l'importance de fournir aux utilisateurs une formation appropriée de sensibilisation à la sécurité, afin qu'ils soient mieux à même de reconnaître ces courriels et d'y répondre de manière appropriée.

2. Chevaux de Troie

Un cheval de Troie (Trojan) est un programme ou un utilitaire nécessaire, ou apparemment utile, contenant un code caché qui, lorsqu'il est invoqué, remplit une fonction non désirée ou nuisible.

Les programmes de cheval de Troie peuvent être utilisés pour accomplir indirectement des fonctions que l'attaquant ne pourrait pas accomplir directement. Par exemple, pour accéder à des informations sensibles et personnelles stockées dans les fichiers d'un utilisateur, un attaquant peut créer un programme de cheval de Troie qui, lorsqu'il est exécuté, analyse les fichiers de l'utilisateur à la recherche des informations sensibles souhaitées et en envoie une copie à l'attaquant par le biais d'un formulaire Web, d'un courriel ou d'un message texte. L'auteur pourrait ensuite inciter les utilisateurs à exécuter le programme en l'incorporant dans un jeu ou un programme utilitaire important. Le programme factice est ensuite rendu disponible via un site de distribution de logiciels ou un magasin d'applications connus. Cette approche a été utilisée avec des utilitaires qui "prétendent" être le dernier scanner anti-virus, ou la dernière mise à jour de sécurité, pour les systèmes, mais qui sont en fait des chevaux de Troie malveillants. Ces derniers sont souvent porteurs de charges virales telles que des logiciels espions qui recherchent des références bancaires. Les utilisateurs doivent donc prendre des précautions pour valider la source de tout logiciel qu'ils installent. Les chevaux de Troie entrent dans l'un des trois modèles :

- Continuer à remplir la fonction du programme d'origine et effectuer en plus une activité malveillante distincte.
- Continuer à exécuter la fonction du programme original mais modifier la fonction pour exécuter une activité malveillante (par exemple, une version cheval de Troie d'un programme de connexion qui collecte des mots de passe) ou pour dissimuler une autre activité malveillante (par exemple, une version cheval de Troie d'un programme de listage de processus qui n'affiche pas certains processus malveillants)
- Exécution d'une fonction malveillante qui remplace complètement la fonction du programme original

Certains chevaux de Troie évitent la nécessité d'une assistance à l'utilisateur en exploitant une certaine vulnérabilité des logiciels pour permettre leur installation et leur exécution automatiques. Pour cela, ils partagent certaines caractéristiques d'un ver, mais contrairement à ce dernier, ils ne se répliquent pas.

Le cheval de Troie Hydraq, utilisé dans le cadre de l'opération Aurora en 2009 et au début de 2010, est un exemple frappant d'une telle attaque. Il exploitait une vulnérabilité d'Internet Explorer pour s'installer et visait plusieurs entreprises de premier plan. Il était généralement diffusé soit par des courriers électroniques non sollicités, soit par un site web compromis au moyen d'une attaque de type "Watering-hole". Les escroqueries à l'assistance technique sont une préoccupation croissante en matière d'ingénierie sociale. Elles consistent en des centres d'appel appelant les utilisateurs au sujet de problèmes inexistants sur leurs systèmes informatiques. Si les utilisateurs répondent, les attaquants tentent de leur vendre une assistance technique inutile ou de leur demander d'installer des logiciels malveillants de type cheval de Troie ou d'autres applications indésirables sur leurs systèmes, tout en prétendant que cela résoudra leur problème [SYMA16].

Chevaux de Troie pour téléphones portables

Les chevaux de Troie des téléphones portables ont également fait leur apparition en 2004 avec la découverte de Skuller. Comme pour les vers mobiles, la cible est le smartphone, et les premiers chevaux de Troie mobiles visaient les téléphones Symbian. Plus récemment, un nombre important de chevaux de Troie ont été détectés qui ciblent les téléphones Android et les iPhones d'Apple. Ces chevaux de Troie sont généralement distribués via une ou plusieurs places de marché d'applications pour l'O/S du téléphone cible.

La croissance rapide des ventes et de l'utilisation des smartphones, qui contiennent de plus en plus d'informations personnelles précieuses, en fait une cible attrayante pour les criminels et autres agresseurs. Étant donné que cinq nouveaux téléphones sur six fonctionnent sous Android, ils constituent une cible clé [SYMA16].

Le nombre de vulnérabilités découvertes dans ces téléphones et les familles de logiciels malveillants qui les ciblent ont augmenté régulièrement ces dernières années. Parmi les exemples récents, citons un cheval de Troie de phishing qui incite l'utilisateur à saisir ses coordonnées bancaires, et un logiciel de rançon qui imite le style de conception de Google pour paraître plus légitime et intimidant.

Les contrôles plus stricts qu'Apple impose à son App store (App Store **:** endroit idéal pour découverte et téléchargement sur iPhones**)** signifient que de nombreux chevaux de Troie pour iPhone ciblent les téléphones et sont distribués via des sites non officiels. Cependant, un certain nombre de versions de l'iPhone O/S contenaient une forme de vulnérabilité graphique ou PDF. En effet, ces vulnérabilités étaient les principaux moyens utilisés pour "jailbreaker" les téléphones. Mais elles fournissaient également un chemin que les logiciels malveillants pouvaient utiliser pour cibler les téléphones. Alors qu'Apple a corrigé un certain nombre de ces vulnérabilités, de nouvelles variantes ont continué à être découvertes. Cela illustre une fois de plus combien il est difficile, même pour les organisations disposant de ressources importantes, d'écrire un logiciel sécurisé dans un système complexe, tel qu'un système d'exploitation.

En 2015, le malware XcodeGhost a été découvert dans un certain nombre d'applications légitimes de l'Apple Store. Ces applications n'ont pas été conçues intentionnellement pour être malveillantes, mais leurs développeurs ont utilisé un système de développement Xcode compromis qui installait secrètement le logiciel malveillant au moment de la création des applications [SYMA16]. C'est l'un des nombreux exemples d'attaquants exploitant l'infrastructure de développement ou d'approvisionnement des entreprises pour faciliter la distribution de logiciels malveillants.

Charge Virale et Corruption du Système Infecté

Une fois qu'un logiciel malveillant est actif sur le système cible, il faut ensuite se préoccuper des mesures qu'il va prendre sur ce système. Autrement dit, quelle est la charge virale qu'il transporte. Certains logiciels malveillants ont une charge virale inexistante ou non fonctionnelle. Leur seul but, qu'il soit délibéré ou dû à une diffusion précoce accidentelle, est de se propager. Le plus souvent, il transporte une ou plusieurs charges virales qui accomplissent des actions secrètes au profit de l'attaquant.

Une charge virale précoce observée dans un certain nombre de virus et de vers a entraîné la destruction de données sur le système infecté lorsque certaines conditions de déclenchement étaient remplies [WEAV03]. Une charge virale connexe est une charge qui affiche des messages ou du contenu indésirables sur le système de l'utilisateur lorsqu'elle est déclenchée. Plus sérieusement, une autre variante tente d'infliger des dommages réels au système. Toutes ces actions visent à nuire à l'intégrité du logiciel ou du matériel du système informatique, ou des données de l'utilisateur. Ces changements ne peuvent se produire immédiatement, mais seulement lorsque les conditions de déclenchement spécifiques programmées dans le code de la bombe logique soient satisfaites.

Le virus de Tchernobyl est un exemple précoce de virus parasite destructeur de la mémoire Windows 95 et 98 qui a été observé pour la première fois en 1998. Il infecte les fichiers exécutables lorsqu'ils sont ouverts. Lorsqu'une date de déclenchement est atteinte, Tchernobyl supprime les données du système infecté en écrasant le premier mégaoctet du disque dur par des zéros, ce qui entraîne une corruption massive de l'ensemble du système de fichiers. Cela s'est produit pour la première fois le 26 avril 1999, alors que les estimations indiquent que plus d'un million d'ordinateurs ont été ainsi contaminés.

De même, le ver Klez, qui envoie des messages en masse, est un exemple précoce de ver destructeur qui infecte les systèmes Windows 95 à XP ; il n’a été observé pour la première fois qu’en octobre 2001. Il se propage en envoyant par courrier électronique des copies de lui-même aux adresses figurant dans le carnet d'adresses et dans les fichiers du système. Il peut arrêter et supprimer certains programmes anti-virus fonctionnant sur le système. Aux dates de déclenchement, c'est-à-dire, le 13 de certains mois de chaque année, il provoque le vidage des fichiers du disque dur local.

Au lieu de simplement détruire les données, certains logiciels malveillants cryptent les données de l'utilisateur et exigent un paiement pour accéder à la clé de décryptage nécessaire à la récupération de ces informations. C'est ce que l'on appelle un logiciel de rançonnement.

Le cheval de Troie PC Cyborg apparu en 1989 en est un exemple de ver précoce. Cependant, vers la mi-2006, un certain nombre de vers et de chevaux de Troie sont apparus, comme le cheval de Troie Gpcode, qui utilisait la cryptographie à clé publique avec des tailles de clé de plus en plus grandes pour chiffrer les données. L'utilisateur devait payer une rançon, ou effectuer un achat sur certains sites, afin de recevoir la clé pour décrypter ces données.
Alors que dans les cas précédents, on utilisait une cryptographie plus faible qui pouvait être craquée sans payer la rançon, les versions ultérieures utilisant la cryptographie à clé publique avec des clés de grande taille ne pouvaient être brisées de cette façon. Les auteurs de SYMA16 et VERI16] notent que les logiciels de rançonnement constituent un défi croissant, comprenant l'un des types de logiciels malveillants les plus courants installés sur les systèmes. Ils sont souvent diffusés lors des téléchargements "drive-by-downloads" ou par des courriels de spams.

Malware de Rançonnement (Ransonware)
En mai 2017, est apparu le logiciel malveillant de rançonnement WannaCry, mentionné dans la discussion précédente sur les vers. WannaCry a infecté un grand nombre de systèmes dans de nombreux pays. Lorsqu'il a été installé sur les systèmes infectés, ce malware a crypté un grand nombre de fichiers correspondant à une liste de types de fichiers particuliers, puis a exigé le paiement d'une rançon en Bitcoins pour les récupérer. Une fois que cela s'est produit, la récupération de ces informations n'était généralement possible que si l'organisation disposait de bonnes sauvegardes, et d'un plan approprié de réponse aux incidents et de reprise après sinistre.

L'attaque du malware de rançonnement WannaCry a suscité une grande attention des médias, en partie en raison du grand nombre d'organisations touchées, et des coûts importants qu'elles ont dû supporter pour s'en remettre. Les cibles de ces attaques se sont étendues au-delà des systèmes informatiques personnels pour inclure les appareils mobiles et les serveurs Linux. De plus, des tactiques telles que la menace de publier des informations personnelles sensibles, ou de détruire définitivement la clé de cryptage après une courte période, sont parfois utilisées pour augmenter la pression sur la victime afin qu'elle paie ce qui a été exigé.

Charge Virale et Corruption du Système

Une autre variante des charges virales de corruption d'un système vise à causer des dommages aux équipements physiques. Le système infecté est clairement le dispositif le plus facilement ciblé. Le virus de Tchernobyl mentionné ci-dessus non seulement corrompt les données, mais tente de réécrire le code du BIOS utilisé pour démarrer l'ordinateur. S'il réussit, le processus de démarrage échoue et le système est inutilisable jusqu'à ce que la puce du BIOS soit reprogrammée ou remplacée.

Plus récemment, le ver Stuxnet cité précédemment cible certains logiciels de systèmes de contrôle industriels spécifiques comme sa charge virale principale [CHEN11, KUSH13]. Si des systèmes de contrôle utilisant certains logiciels de contrôle industriel de Siemens avec une configuration spécifique d'appareils sont infectés, le ver remplace alors le code de contrôle original par un code qui entraîne délibérément l'équipement contrôlé en dehors de sa plage de fonctionnement normale, ce qui entraîne la défaillance de l'équipement connecté.

Les centrifugeuses utilisées dans le programme iranien d'enrichissement de l'uranium étaient fortement suspectées d'être la cible, des rapports faisant état de taux de défaillance bien plus élevés que la normale ont été observés dans ces centrifugeuses pendant la période où ce ver était actif. Comme nous l'avons indiqué dans notre discussion précédente, cela a suscité des inquiétudes quant à l'utilisation de logiciels malveillants ciblés sophistiqués pour le sabotage industriel.

Dans son rapport sur la sécurité et la défense pour 2015, le gouvernement britannique a fait part de ses préoccupations croissantes concernant l'utilisation de cyber attaques contre les infrastructures critiques par des acteurs étatiques et non étatiques. L'attaque de décembre 2015 qui a perturbé les systèmes électriques ukrainiens montre que ces préoccupations sont fondées, étant donné que de nombreuses infrastructures critiques ne sont pas suffisamment protégées pour résister à de telles attaques [SYMA16].

La bombe logique est un élément clé des logiciels malveillants qui corrompent les données. La bombe logique est un code intégré dans le logiciel malveillant qui est réglé pour "exploser" lorsque certaines conditions sont remplies. Des exemples de conditions qui peuvent être utilisées comme déclencheurs d'une bombe logique sont la présence ou l'absence de certains fichiers ou dispositifs sur le système, un jour de la semaine ou une date particulière, une version ou une configuration particulière de certains logiciels, ou un utilisateur particulier exécutant l'application. Une fois déclenchée, une bombe logique peut altérer ou supprimer des données ou des fichiers entiers, provoquer un arrêt de la machine ou causer d'autres dommages.

Un exemple frappant de la manière dont les bombes logiques peuvent être utilisées est le cas de Tim Lloyd, qui a été condamné pour avoir posé une bombe logique qui a coûté à

son employeur, Omega Engineering, plus de 10 millions de dollars. Cela a fait saboter la stratégie de croissance de l'entreprise et a finalement conduit au licenciement de 80 travailleurs [GAUD00]. Enfin de compte, Lloyd a été condamné à 41 mois de prison et à payer 2 millions de dollars de dédommagement.

Attaque des agents Bots

La catégorie suivante de charge virale transportée par les codes malveillants est celle où le malware perturbé les ressources informatiques et réseau du système infecté pour être utilisé par l'attaquant. Un tel programme malveillant est connu sous le nom de bot (robot), zombie ou drone. Le bot s'empare secrètement d'un autre ordinateur connecté à Internet, puis utilise cet ordinateur pour lancer ou gérer des attaques dont il est difficile de remonter jusqu'au créateur du bot. Le bot est généralement implanté sur des centaines ou des milliers d'ordinateurs appartenant à des tiers sans méfiance.

Le regroupement de bots est souvent capable d'agir de manière coordonnée ; une telle collection est appelée "botnet". Ce type de malware attaque l'intégrité et la disponibilité des systèmes infectés.

Les bots au sein de botnets peuvent générer les attaques importantes de types suivants :

- Distributed Denial of Service : DDoS ou attaques distribuées par déni de service: Une attaque DDoS est une attaque sur un système ou un réseau informatique qui provoque une perte de service pour les utilisateurs par bocage ou saturation du système cible.
- Spamming : À l'aide d'un botnet et de milliers de bots, un attaquant est capable d'envoyer des quantités massives de courrier électronique en vrac (spam).
- Sniffer : Il s'agit de renifler le trafic : Les bots peuvent également utiliser un renifleur de paquets pour surveiller les données intéressantes en texte clair qui passent par une machine compromise. Les renifleurs sont surtout utilisés pour récupérer des informations sensibles comme les noms d'utilisateur et les mots de passe.
- Keylogger ou enregistrement des touches : Si la machine compromise utilise des canaux de communication cryptés (par exemple HTTPS ou POP3S), il est inutile de renifler les paquets réseau sur l'ordinateur de la victime, car la clé appropriée pour décrypter les paquets est manquante. Mais en utilisant un keylogger, qui capture les frappes de clavier sur la machine infectée, un attaquant peut récupérer des informations sensibles.
- Propagation de nouveaux logiciels malveillants : Les réseaux de zombies sont utilisés pour propager de nouveaux bots. Ce mécanisme est très facile, car tous les robots mettent en œuvre des mécanismes permettant de télécharger et d'exécuter un fichier via HTTP ou FTP. Un botnet de 10 000 hôtes qui sert de base de départ à un ver ou à un virus de messagerie permet une propagation très rapide et cause donc plus de dégâts.
- Installation de modules publicitaires et d'objets d'aide au navigateur (BHO) : Les botnets peuvent également être utilisés pour obtenir des avantages financiers. Cela

fonctionne en créant un faux site web avec quelques publicités : L'opérateur de ce site Web négocie un accord avec certaines sociétés d'hébergement qui paient pour les clics sur les publicités. À l'aide d'un botnet, ces clics peuvent être "automatisés", de sorte que des milliers de bots cliquent instantanément sur les pop-ups. Ce processus peut être encore amélioré si le bot détourne la page de démarrage d'une machine compromise de sorte que les "clics" soient exécutés chaque fois que la victime utilise le navigateur.

- Attaque de réseaux de chat IRC : Les botnets sont également utilisés pour des attaques contre les réseaux de chat Internet Relay Chat (IRC). Les attaques les plus populaires parmi les attaquants sont celles dites "clones" : Dans ce type d'attaque, le contrôleur ordonne à chaque bot de connecter un grand nombre de clones au réseau IRC de la victime. La victime est inondée par les demandes de service de milliers de bots ou par les milliers de canaux rejoints par ces bots clonés. De cette façon, le réseau IRC victime est détruit, comme dans le cas d'une attaque par déni de service (DDoS).
- Manipulation de sondages/jeux en ligne : Les sondages/jeux en ligne attirent de plus en plus l'attention et il est assez facile de les manipuler avec des botnets. Comme chaque bot a une adresse IP distincte, chaque vote aura la même crédibilité qu'un vote émis par une personne réelle. Les jeux en ligne peuvent être manipulés de la même manière.

Disposition de contrôle à distance

La facilité de contrôle distant est ce qui distingue un bot d'un ver. Un ver se propage et s'active lui-même, alors qu'un bot est contrôlé par une forme de commande réseau de serveurs de contrôle (C&C). Ce contact n'a pas besoin d'être continu, mais peut être lancé périodiquement lorsque le bot observe qu'il a accès au réseau.

L'un des premiers moyens utilisés pour mettre en œuvre le dispositif de contrôle à distance a été l'utilisation d'un IRC serveur. Tous les bots rejoignent un canal spécifique sur ce serveur et traitent les messages entrants comme des commandes. Les botnets plus récents ont tendance à éviter les mécanismes IRC et à utiliser les canaux de communication via des protocoles de communication tels que HTTP. Les mécanismes de contrôle distribués, utilisant des protocoles peer-to-peer, sont également utilisés, afin d'éviter un point central de défaillance unique.

À l'origine, ces serveurs C&C utilisaient des adresses fixes, ce qui signifiait qu'ils pouvaient être localisés et éventuellement repris ou retirés par les services répressifs.
Certaines familles de logiciels malveillants plus récentes ont utilisé des techniques telles que la génération d'un très grand nombre de noms de domaine de serveurs (DNS) que le logiciel malveillant essaierait à contacter. Si un nom de serveur est compromis, les attaquants peuvent installer un nouveau serveur attaché à un autre nom qu'ils vont testé. Pour vaincre cette attaque, les analystes de la sécurité doivent inverser l'algorithme de

génération de noms, pour ensuite tenter de prendre le contrôle sur l'ensemble du grand nombre de domaines possibles. Une autre technique utilisé pour masquer les serveurs est un DNS à flux rapide, où l'adresse associée à un
nom du serveur est changé fréquemment, toutes les quelques minutes, pour tourner sur un grand nombre de serveurs proxy, généralement d'autres membres du réseau de zombies. Ces approches permettent d'empêcher de répondre à la menace du botnet.

Une fois qu'une voie de communication est établie entre un module de contrôle et les bots, le module de contrôle peut gérer maintenant ces bots. Dans sa forme la plus simple, le module de contrôle envoie simplement une commande au bot pour lui faire exécuter des routines déjà mises en œuvre dans le bot. Pour une plus grande flexibilité, le module de contrôle peut émettre des commandes de mise à jour demandant aux bots de télécharger un fichier à partir de certains l'emplacement sur Internet et l'exécuter. Dans ce dernier cas, le bot devient un outil plus général pouvant être utilisé pour des attaques multiples. Le module de contrôle peut également collecter des informations recueillies par les robots que l'agresseur peut ensuite exploiter. Une contre-mesure efficace contre un botnet consiste à reprendre en charge le réseau C&C ou à l'arrêter. Le renforcement de la coopération et la coordination entre les services répressifs dans un certain nombre de pays s'est traduit par un nombre croissant de saisies C&C réussies ces dernières années [SYMA16], et la suppression consécutive des réseaux de zombies associés. Ces actions ont également donné lieu à des poursuites pénales à l'encontre d'un certain nombre de personnes qui y sont impliquées.

Enregistreurs de frappe, logiciels espions de vol d'informations

Considérons maintenant les charges virales où les malwares rassemblent les données stockées sur les systèmes infectés pour être utilisées par l'attaquant. Une cible commune résident dans : le nom d'utilisateur, son mot de passe, ses informations bancaires, etc., pour accéder à des sites bancaires, de jeux et autres, que l'attaquant utilise ensuite pour se faire passer pour le véritable utilisateur afin d'accéder à ces sites dans un but lucratif. Moins fréquemment, l'attaque peut cibler des documents ou des détails de configuration du système à des fins de reconnaissance ou d'espionnage. Ces attaques visent la confidentialité de ces informations.

Généralement, les utilisateurs envoient leurs identifiants de connexion et leurs mots de passe à des sites bancaires, de jeux et autres sites connexes par des canaux de communication cryptés (par exemple, HTTPS ou POP3S), ce qui les protège contre la capture en surveillant les paquets du réseau. Pour contourner ce problème, un attaquant peut installer un enregistreur de frappe, qui capture les frappes de clavier sur la machine infectée pour permettre à l'attaquant de surveiller ces informations sensibles. Étant donné que l'attaquant recevrait ainsi une copie de tout le texte saisi sur la machine compromise, les enregistreurs de frappe mettent généralement en œuvre une forme de mécanisme de filtrage qui ne renvoie que les informations proches des mots clés souhaités (par exemple, "login" ou "mot de passe" ou "paypal.com" etc.).

En réponse à l'utilisation de keyloggers, certains sites bancaires et autres sont passés à l'utilisation d'une applet graphique pour entrer des informations critiques, comme les mots de passe. Comme on n'utilise pas le texte saisi au clavier, les keyloggers traditionnels échouent à la capture de ces informations sensibles. Pour contourner ce mécanisme de sécurité, les attaquants ont développé des logiciels espions plus généraux, qui détournent la machine compromise pour permettre la surveillance d'un large éventail d'activités sur le système. Il peut s'agir de surveiller l'historique et le contenu de l'activité de navigation, de rediriger certaines requêtes de pages web vers de faux sites contrôlés par l'attaquant, et de modifier dynamiquement les données échangées entre le navigateur et certains sites web d'intérêt. Toutes ces pratiques peuvent entraîner une compromission importante des informations personnelles de l'utilisateur.

Le cheval de Troie bancaire Zeus, créé à partir de sa boîte à outils de logiciels criminels, est un exemple éminent de ce type de logiciel espion qui a été largement déployé ces dernières années [BINS10]. Il vole des informations bancaires et financières en utilisant un enregistreur de frappe et en capturant et éventuellement en modifiant les données des formulaires de certains sites web. Il est généralement déployé à l'aide de courriers électroniques non sollicités ou via un site web compromis dans le cadre d'un téléchargement "drive-by download".

Vol d'informations par hameçonnage

Une autre approche utilisée pour saisir les identifiants de connexion et de mot de passe d'un utilisateur consiste à inclure une URL dans un courrier électronique de spam qui renvoie à un faux site web contrôlé par l'attaquant. Ce site web factice imite la page de connexion d'un site bancaire, de jeu ou autre site similaire. Cette URL est normalement incluse dans un message suggérant qu'une action urgente est requise par l'utilisateur pour authentifier son compte, afin d'éviter qu'il ne soit verrouillé. Si l'utilisateur est négligent et ne se rend pas compte qu'il est connecté, le fait de suivre le lien et de fournir les détails demandés aura certainement pour conséquence l'exploitation par les attaquants, de son compte, en utilisant les informations d'identification saisies.

Plus généralement, un tel courrier électronique non sollicité peut diriger un utilisateur vers un faux site web contrôlé par l'attaquant, ou l'inviter à remplir un formulaire joint et à retourner à une adresse de courrier électronique accessible à l'attaquant. Ce courrier électronique est alors utilisé pour recueillir une série d'informations privées et personnelles sur l'utilisateur. S'il dispose de suffisamment de détails, l'attaquant peut alors usurper l'identité de l'utilisateur dans le but d'obtenir un crédit ou un accès sensible à d'autres ressources. Il s'agit d'une attaque de phishing qui exploite l'ingénierie sociale pour gagner la confiance de l'utilisateur en lui présentant des communications provenant de sources fiables [GOLD10]. Ces courriels de spam généraux sont généralement distribués à un très grand nombre d'utilisateurs, souvent par l'intermédiaire d'un réseau de zombies. Même si le contenu des communications ne correspond pas aux sources de confiance appropriées pour une fraction importante des destinataires, les attaquants s'attendent à ce qu'il atteigne un nombre suffisant d'utilisateurs, de la source de confiance nommée, dont une partie crédule répondra, pour qu'il soit rentable.

Une variante plus dangereuse de cette méthode est l'attaque par hameçonnage. Il s'agit là encore d'un courrier électronique prétendant provenir d'une source fiable. Cependant, les destinataires sont soigneusement recherchés par l'attaquant, et chaque e-mail est soigneusement conçu pour convenir spécifiquement à son destinataire, en citant souvent une série d'informations pour le convaincre de son authenticité. Cela augmente considérablement la probabilité que le destinataire réponde comme le souhaite l'attaquant. Ce type d'attaque est particulièrement utilisé dans l'espionnage industriel et d'autres formes d'espionnage, ou dans la fraude financière telle que les fausses autorisations de transfert électronique, par des organisations disposant de ressources importantes. Que ce soit par hameçonnage, par téléchargement ou par attaque directe de pirates informatiques, le nombre d'incidents et la quantité de dossiers personnels exposés ne cessent d'augmenter. Par exemple, la violation des données médicales en janvier 2015 a exposé plus de 78 millions de dossiers d'informations personnelles qui pourraient potentiellement être utilisées pour le vol d'identité. On pense que le groupe de cyber-espionnage Black Vine, qui dispose de ressources importantes, est responsable de cette attaque [SYMA16].

Le vol et l'usurpation d'identité sont des cas particuliers d'une exploration plus générale de charge virale, visant à permettre à un attaquant d'obtenir certains types d'informations souhaitées. Ces cas spéciaux sont certainement les plus courants, mais d'autres cibles sont connues. L'opération Aurora en 2009 a utilisé un cheval de Troie pour accéder et éventuellement modifier les dépôts de code source d'une série d'entreprises de haute technologie, de sécurité et de défense [SYMA16].

Le ver Stuxnet, découvert en 2010, s'intéressait la capture de détails de configurations matérielles et logicielles afin de déterminer s'il avait compromis les systèmes cibles spécifiques souhaités. Les premières versions de ce ver ont renvoyé ces mêmes informations, qui ont ensuite été utilisées pour développer les attaques déployées dans les versions ultérieures [CHEN11, KUSH13]. Il existe un certain nombre d'autres exemples très médiatisés d'exposition massive d'informations. Il s'agit notamment de la fuite de documents militaires et diplomatiques sensibles sur Wikileaks par Chelsea Manning en 2010, et la publication d'informations sur les programmes de surveillance de la NSA par Edward Snowden en 2013. Ces deux exemples montrent que les initiés exploitent leurs droits d'accès légitimes pour divulguer des informations pour des raisons idéologiques. Ces deux événements ont donné lieu à des discussions et débats mondiaux importants sur les conséquences de ces actions. En revanche, la fuite en 2016 de millions de documents relatifs à des entités offshore utilisées comme paradis fiscaux, dans certains cas du moins, seraient le fait de pirates informatiques extérieurs attaquant des systèmes mal sécurisés. De graves conséquences en résultent pour certaines des personnes citées dans des fuites scabreuses.

Les attaques APT peuvent entraîner la perte de volumes importants d'informations sensibles, qui sont exfiltrées des organisations cibles et envoyées aux attaquants. Pour détecter et bloquer ces exfiltrations de données, il faut des contre-mesures techniques

appropriées capables de gérer soit l'accès à ces informations, soit leur transmission à travers le périmètre du réseau des organisations.

Charge virale – Porte dérobée

La dernière catégorie d'attaques virales concerne les techniques utilisées par les logiciels malveillants pour dissimuler leur présence sur le système infecté, et fournir un accès secret à ce système. Ce type d'attaque compromet également l'intégrité du système infecté.

Une porte dérobée, également appelée trappe, est un point d'entrée secret dans un programme qui permet à une personne, qui en a connaissance, d'y accéder sans passer par les procédures d'accès sécurisés habituelles.

Les programmeurs utilisent légitimement des portes dérobées depuis de nombreuses années pour déboguer et tester des programmes ; une telle porte dérobée est appelée "maintenance hook". Cela se fait généralement lorsque le programmeur développe une application qui comporte une procédure d'authentification, ou une longue configuration, nécessitant à l'utilisateur de saisir de nombreuses valeurs différentes afin d'exécuter l'application. Pour déboguer le programme, le développeur peut souhaiter obtenir des privilèges spéciaux ou éviter toute la configuration et l'authentification nécessaires. Le programmeur peut également vouloir s'assurer qu'il existe une méthode d'activation du programme si quelque chose ne va pas avec la procédure d'authentification intégrée à l'application.

La porte dérobée est un code qui reconnaît une séquence spéciale d'entrée ou déclenchée par l'exécution d'un certain identifiant utilisateur ou par une séquence d'événements peu probable.

Les portes dérobées deviennent des menaces lorsque des programmeurs peu scrupuleux les utilisent pour obtenir un accès non autorisé. La porte dérobée était l'idée de base de la vulnérabilité décrite dans le film War Games. Un autre exemple est que lors du développement de Multics, des tests de pénétration ont été effectués par une équipe de l'armée de l'air (simulant des adversaires). L'une des tactiques employées consistait à envoyer un bug de mise à jour à un système d'exploitation fonctionnant sous Multics. La mise à jour contenait un cheval de Troie qui pouvait être activé par une porte dérobée et qui permettait à l'équipe d'y accéder. La menace était si bien mise en œuvre que les développeurs de Multics ne pouvaient pas la trouver, même après avoir été informés de sa présence [ENGE80].

Plus récemment, une porte dérobée est généralement mise en place comme un service réseau à l'écoute sur un port non standard auquel l'attaquant peut se connecter et émettre des commandes pour être exécuté sur le système compromis. Le logiciel de rançonnement WannaCry décrit précédemment comportait une telle porte dérobée.

Il est difficile de mettre en œuvre des contrôles de système d'exploitation pour les portes dérobées dans les applications. Les mesures de sécurité doivent se concentrer sur les

activités de développement de programmes et de mise à jour de logiciels, et sur les programmes qui souhaitent offrir un service réseau.

Charge virale - Rootkits

Un rootkit est un ensemble de programmes installés sur un système pour maintenir un accès secret à ce système avec des privilèges d'administrateur (ou Root), tout en cachant le plus possible les preuves de sa présence. Cela permet d'accéder à toutes les fonctions et à tous les services du système d'exploitation. Le rootkit altère la fonctionnalité standard de l'hôte de manière malveillante et furtive. Avec l'accès root, un attaquant a le contrôle total du système et peut ajouter ou modifier des programmes et des fichiers, surveiller les processus, envoyer et recevoir du trafic réseau et obtenir un accès à la demande par une porte dérobée.

Un rootkit peut apporter de nombreuses modifications à un système pour en dissimuler l'existence, ce qui rend difficile pour l'utilisateur de déterminer la présence du rootkit et d'identifier les modifications qui ont été apportées. En substance, un rootkit se cache en subvertissant les mécanismes qui surveillent et signalent les processus, les fichiers et les registres d'un ordinateur.

Caractéristiques de la classification des rootkits

Un rootkit peut être classé selon les caractéristiques suivantes :

- Persistant : S'active à chaque fois que le système démarre. Le rootkit doit stocker le code dans un emplacement persistant, tel que le registre ou le système de fichiers, et configurer une méthode par laquelle le code s'exécute sans intervention de l'utilisateur. Cela signifie qu'il est plus facile à détecter, car la copie dans le support persistant peut potentiellement être analysée.
- Basé sur la mémoire : N'a pas de code persistant et ne peut donc pas survivre à un redémarrage. Cependant, comme il n'est présent qu'en mémoire, il peut être plus difficile à détecter.
- Mode utilisateur : Intercepte les appels aux API (interfaces de programmes d'application) et modifie les résultats renvoyés. Par exemple, lorsqu'une application effectue un listage de répertoire, les résultats retournés ne comprennent pas les entrées identifiant les fichiers associés au rootkit.
- Mode noyau : Peut intercepter les appels aux API natives en mode noyau. Le rootkit peut également masquer la présence d'un processus malveillant en le supprimant de la liste des processus actifs du noyau.
- Basé sur une machine virtuelle : Ce type de rootkit installe un moniteur de machine virtuelle léger, puis exécute le système d'exploitation dans une machine virtuelle située au-dessus. Le rootkit peut alors intercepter et modifier de manière transparente les états et les événements se produisant dans le système virtualisé.

- Mode externe : Le malware se trouve en dehors du mode de fonctionnement normal du système visé, en mode BIOS ou en mode de gestion du système, où il peut accéder directement au matériel. Cette classification révèle un challenge permanent entre les auteurs de rootkits, qui exploitent des mécanismes toujours plus furtifs pour cacher leur code, et ceux qui développent des mécanismes pour durcir les systèmes contre une telle subversion, ou pour détecter quand elle s'est produite. Une grande partie de cette avancée est associée à la découverte de formes d'attaques "en couches". Les premiers rootkits fonctionnaient en mode utilisateur, modifiant les programmes utilitaires et les bibliothèques afin de dissimuler leur présence. Les modifications qu'ils apportaient pouvaient être détectées par le code dans le noyau, car celui-ci fonctionne dans la couche située sous l'utilisateur. Les rootkits de la génération suivante utilisaient des techniques plus furtives.

Infection du noyau du système

La génération suivante de rootkits est descendue d'une couche, apportant des modifications à l'intérieur du noyau et coexistant avec le code des systèmes d'exploitation, afin de rendre leur détection beaucoup plus difficile. Tout programme "anti-virus" serait désormais soumis aux mêmes modifications de "bas niveau" que le rootkit utilise pour masquer sa présence. Cependant, des méthodes ont été développées pour détecter ces modifications.

Les programmes fonctionnant au niveau de l'utilisateur interagissent avec le noyau par le biais d'appels système. Ainsi, les appels système sont une cible principale des rootkits au niveau du noyau pour parvenir à la dissimulation. Pour illustrer le fonctionnement des rootkits, nous examinons la mise en œuvre des appels système dans Linux. Sous Linux, chaque appel système se voit attribuer un numéro d'appel système unique.

Lorsqu'un processus en mode utilisateur exécute un appel système, le processus fait référence à l'appel système par ce numéro. Le noyau maintient une table d'appels système avec une entrée par routine d'appel système ; chaque entrée contient un pointeur vers la routine correspondante. Le numéro syscall sert d'index dans la table d'appel système. [LEVI06] énumère trois techniques qui peuvent être utilisées pour modifier les appels système :

- Modifier la table d'appel système : L'attaquant modifie les adresses syscall sélectionnées et stockées dans la table d'appel système. Cela permet au rootkit de diriger un appel système loin de la routine légitime vers le remplacement du rootkit. La figure 6.5 montre comment le rootkit knark y parvient.
- Modifier les cibles de la table d'appel système : L'attaquant écrase les routines d'appel système légitimes sélectionnées avec un code malveillant. La table d'appel système n'est pas modifiée.
- Rediriger la table d'appel système : L'attaquant redirige les références à l'ensemble de la table d'appel système vers une nouvelle table dans un nouvel emplacement de la mémoire du noyau.

Approche de contre-mesures des logiciels malveillants

La solution idéale à la menace des logiciels malveillants est la prévention : Donc empêcher les logiciels malveillants de s'introduire dans le système en premier lieu, ou bloquez leur capacité à modifier le système. Cet objectif est, en général, presque impossible à atteindre, bien que la mise en place de contre-mesures appropriées pour durcir les systèmes et les utilisateurs dans la prévention de l'infection puisse réduire considérablement le nombre d'attaques réussies de logiciels malveillants. Le document NIST SP 800-83 suggère qu'il existe quatre éléments principaux de prévention : la politique, la sensibilisation, l'atténuation de la vulnérabilité et l'atténuation des menaces. Une politique appropriée de prévention des logiciels malveillants constitue une base pour la mise en œuvre de contre-mesures préventives appropriées.

L'une des premières contre-mesures à employer est de s'assurer que tous les systèmes sont aussi à jour que possible, avec tous les correctifs appliqués, afin de réduire le nombre de vulnérabilités qui pourraient y être exploitées. La deuxième consiste à mettre en place des contrôles d'accès appropriés aux applications et aux données stockées sur le système, afin de réduire le nombre de fichiers auxquels tout utilisateur peut accéder, et donc potentiellement infectés ou corrompus, du fait de l'exécution d'un code malveillant par ces derniers. Ces mesures visent directement les principaux mécanismes de propagation utilisés par les vers, les virus et certains chevaux de Troie.

Le troisième mécanisme de propagation commun, qui cible les utilisateurs lors d'une attaque d'ingénierie sociale, peut être contré grâce à une sensibilisation et une formation appropriées des utilisateurs. L'objectif est de permettre à ces derniers d'être plus conscients de ces attaques et moins susceptibles de prendre des mesures qui pourraient les compromettre. Le document NIST SP 800-83 fournit des exemples de problèmes de sensibilisation appropriés.

Si la prévention échoue, des mécanismes techniques peuvent être utilisés pour soutenir les options suivantes de réduction de la menace :

- Détection : Une fois que l'infection s'est produite, déterminer qu'elle s'est produite et localiser le malware.
- Identification : Une fois la détection effectuée, identifiez le logiciel malveillant spécifique qui a infecté le système.
- Suppression : Une fois le malware spécifique identifié, éliminez toute trace de virus malveillant de tous les systèmes infectés afin qu'il ne puisse pas se propager davantage.

Si la détection réussit mais que l'identification ou la suppression n'est pas possible, l'alternative consiste à éliminer tout fichier infecté ou malveillant et à recharger une version de sauvegarde propre. Dans le cas de certaines infections particulièrement graves, cela peut nécessiter un effacement complet de tout le stockage et la reconstruction du système infecté à partir de supports propres connus. Pour commencer,

examinons quelques exigences pour des contre-mesures efficaces contre les logiciels malveillants :

- Généralité : L'approche adoptée doit permettre de traiter une grande variété d'attaques.
- Rapidité : L'approche doit permettre de réagir rapidement afin de limiter le nombre de programmes ou de systèmes infectés et l'activité qui en résulte.
- Résilience : L'approche doit être résistante aux techniques d'évasion employées par les attaquants pour dissimuler la présence de leurs logiciels malveillants.
- Coûts minimaux de déni de service : L'approche devrait entraîner une réduction minimale de la capacité ou du service en raison des actions du logiciel de contre-mesure, et ne devrait pas perturber de manière significative le fonctionnement normal.
- Transparence : Les logiciels et dispositifs de contre-mesure ne doivent pas nécessiter de modification des systèmes d'exploitation, des logiciels d'application et du matériel existants (anciens).
- Couverture mondiale et locale : L'approche doit permettre de traiter les sources d'attaque à la fois de l'extérieur et de l'intérieur du réseau de l'entreprise.

La réalisation de toutes ces exigences nécessite souvent l'utilisation de plusieurs approches. La détection de la présence de logiciels malveillants peut se faire à plusieurs endroits. Elle peut se produire sur le système infecté, où un programme "anti-virus" basé sur l'hôte est en cours d'exécution, surveillant les données importées dans le système, ainsi que l'exécution et le comportement des programmes s'exécutant sur le système. Il peut également se produire dans le cadre des mécanismes de sécurité du périmètre utilisés dans le pare-feu et la détection des intrusions (IDS) d'une organisation.

Enfin, la détection peut utiliser des mécanismes distribués qui recueillent des données à partir de capteurs basés sur l'hôte et de capteurs périmétriques, potentiellement installés sur un grand nombre de réseaux et d'organisations, afin d'obtenir une vue à grande échelle du mouvement des logiciels malveillants. Nous allons maintenant examiner chacune de ces approches plus en détail.

Générations de logiciels anti-virus

Le premier endroit où un logiciel anti-virus est utilisé se trouve sur chaque système d'extrémité, par exemple : les ordinateurs personnels généralement connectés à Internet. Cela permet au logiciel d'avoir un accès maximal aux informations sur le comportement du logiciel malveillant lorsqu'il interagit avec le système ciblé, mais aussi avoir une vue d'ensemble minimale de l'activité du logiciel malveillant. L'utilisation de logiciels anti-virus sur les ordinateurs personnels est aujourd'hui très répandue, en partie à cause de la croissance explosive du volume et de l'activité des logiciels malveillants. Ces logiciels peuvent être considérés comme une forme de système de détection d'intrusion basé sur l'hôte.

Les progrès de la technologie des virus et autres logiciels malveillants, ainsi que de la technologie des antivirus et autres contre-mesures, vont de pair. Les premiers logiciels malveillants utilisaient un code relativement simple et facile à détecter, et pouvaient donc être identifiés et purgés avec des logiciels antivirus relativement simples. Avec l'évolution de la course aux logiciels malveillants, tant le code des logiciels malveillants que, nécessairement, les logiciels anti-virus sont devenus plus complexes et plus sophistiqués. [STEP93] identifie quatre générations de logiciels anti-virus :

- Première génération : Simples scanners
- Deuxième génération : Les scanners heuristiques
- Troisième génération : Les pièges à activité
- Quatrième génération : Une protection complète

Un scanner de première génération nécessite une signature de logiciel malveillant pour identifier ce dernier. La signature peut contenir des "jokers", mais elle correspond essentiellement à la même structure et au même modèle de bits dans toutes les copies du malware. Ces scanners spécifiques à la signature sont limités à la détection des logiciels malveillants connus. Un autre type de scanner de première génération enregistre la durée des programmes et recherche les modifications de durée dues à l'infection par un virus.

Un scanner de deuxième génération ne repose pas sur une signature spécifique. Il utilise plutôt des règles heuristiques pour rechercher des instances probables de logiciels malveillants. Une catégorie de ces scanners recherche des fragments de code souvent associés à des logiciels malveillants. Par exemple, un scanner peut rechercher le début d'une boucle de chiffrement utilisée dans un virus polymorphe et découvrir la clé de chiffrement. Une fois la clé découverte, le scanner peut décrypter le logiciel malveillant pour l'identifier, puis supprimer l'infection et remettre le programme en service.
Une autre approche de deuxième génération est la vérification de l'intégrité. Une somme de contrôle peut être annexée à chaque programme. Si un logiciel malveillant modifie ou remplace un programme sans modifier la somme de contrôle, un contrôle d'intégrité détectera cette modification.

Pour contrer les logiciels malveillants qui sont suffisamment sophistiqués pour modifier la somme de contrôle lorsqu'ils modifient un programme, une fonction de hachage cryptée peut être utilisée. La clé de cryptage est stockée séparément du programme, de sorte que le malware ne puisse générer un nouveau code de hachage et le crypter. En utilisant une fonction de hachage plutôt qu'une somme de contrôle plus simple, le malware est empêché d'ajuster le programme pour produire le même code de hachage qu'auparavant. Si une liste protégée de programmes se trouvant dans des lieux sûrs est conservée, cette approche peut également détecter les tentatives de remplacement ou d'installation de code ou de programmes malveillants dans ces endroits.

Les programmes de troisième génération sont des programmes résidant en mémoire qui identifient les logiciels malveillants par leurs actions plutôt que par leur structure dans un programme infecté. Ces programmes présentent l'avantage de ne pas nécessiter le développement de signatures et d'heuristiques pour un large éventail de logiciels

malveillants. Il suffit d'identifier le petit ensemble d'actions qui indiquent qu'une activité malveillante est tentée, puis d'intervenir.

Les produits de quatrième génération sont des ensembles composés de diverses techniques anti-virus utilisées conjointement. Ils comprennent des composants de balayage et de piège d'activité. En outre, un tel progiciel est doté d'une capacité de contrôle d'accès, qui limite la capacité d'un logiciel malveillant à pénétrer dans un système, puis à mettre à jour des fichiers afin de les propager.

Le challenge établi entre hackers et développeurs de logiciels anti malwares se cesse de prendre de l'ampleur et se poursuit inéluctablement. Avec les progiciels de quatrième génération, une stratégie de défense plus complète est employée, élargissant le champ de la défense à des mesures de sécurité informatique plus générales. Celles-ci comprennent des approches anti-virus plus sophistiquées.

Analyse des bacs à sable

Une des méthodes de détection et d'analyse des logiciels malveillants consiste à exécuter un code potentiellement malveillant dans un bac à sable émulé ou sur une machine virtuelle. Celles-ci permettent au code de s'exécuter dans un environnement contrôlé, où son comportement peut être étroitement surveillé sans menacer la sécurité d'un système réel. Ces environnements vont des émulateurs de bac à sable qui simulent la mémoire et l'unité centrale d'un système cible, jusqu'aux machines virtuelles complètes, qui reproduisent toutes les fonctionnalités des systèmes cibles, mais qui peuvent facilement être restaurées dans un état connu. L'exécution de logiciels potentiellement malveillants dans de tels environnements permet de détecter des logiciels malveillants complexes cryptés, polymorphes ou métamorphiques.

Le code doit se transformer en instructions machine, qu'il exécute ensuite pour réaliser les actions malveillantes prévues. Le code dépaqueté, transformé ou décrypté qui en résulte peut ensuite être analysé à la recherche de signatures de logiciels malveillants connues, ou son comportement peut être surveillé pendant l'exécution afin de détecter d'éventuelles activités malveillantes [EGEL12, KERA16]. Cette analyse étendue peut être utilisée pour développer des signatures anti-virus pour de nouveaux logiciels malveillants inconnus.

Le problème de conception le plus difficile avec l'analyse de bac à sable est de déterminer combien de temps il faut pour exécuter chaque interprétation. En général, les éléments des logiciels malveillants sont activés peu après le début de l'exécution d'un programme, mais les logiciels malveillants récents utilisent de plus en plus des méthodes d'évasion telles que la mise en veille prolongée pour échapper à la détection pendant le temps d'analyse utilisé par les systèmes de bac à sable [KERA16]. Plus l'analyseur émule longtemps un programme particulier, plus il a de chances de détecter les logiciels malveillants cachés. Cependant, l'analyse des bacs à sable ne dispose que d'un temps et de ressources limités, étant donné la nécessité d'analyser de grandes quantités de logiciels malveillants potentiels.

Avec l'amélioration des techniques d'analyse, une course acharnée s'est développée entre les auteurs et les défenseurs de logiciels malveillants. Certains logiciels malveillants vérifient s'ils fonctionnent dans un bac à sable ou un environnement virtualisé, et suppriment les comportements malveillants si c'est le cas. D'autres logiciels malveillants incluent des périodes de sommeil prolongées avant de s'engager dans une activité malveillante, afin de tenter d'échapper à la détection avant la fin de l'analyse. Ou encore, le malware peut inclure une bombe logique qui recherche une date spécifique, un type de système ou un emplacement réseau spécifique avant de s'engager dans une activité malveillante, ce à quoi l'environnement de bac à sable ne correspond pas. En réponse, les analystes adaptent leurs environnements de bac à sable pour tenter d'échapper à ces tests. Malheureusement, cette course néfaste se poursuit inlassablement car elle implique des enjeux mercantiles considérables et stratégiques.

Logiciel de blocage du comportement basé sur l'hôte

Contrairement à l'heuristique ou aux scanners d'empreintes digitales, l'analyse dynamique des logiciels malveillants ou les logiciels de blocage du comportement s'intègrent au système d'exploitation d'un ordinateur hôte et surveillent le comportement du programme en temps réel pour détecter les actions malveillantes [CONR02, EGEL12]. Il s'agit d'un type de prévention des intrusions basé sur l'hôte dont nous parlerons plus en détail à la section 9.6. Ce logiciel surveille le comportement d'un code potentiellement malveillant, à la recherche d'actions potentiellement malveillantes, comme les systèmes de bac à sable dont nous avons parlé dans la section précédente. Cependant, il a ensuite la capacité de bloquer les actions malveillantes avant qu'elles n'affectent le système cible. Les comportements surveillés peuvent inclure les éléments suivants :

- Tentatives d'ouverture, d'affichage, de suppression et/ou de modification de fichiers ;
- Tentatives de formatage des disques et autres opérations irrécupérables sur le disque;
- Modifications de la logique des fichiers exécutables ou des macros ;
- Modification des paramètres critiques du système, tels que les paramètres de démarrage ;
- la création de scripts pour les clients de courrier électronique et de messagerie instantanée afin d'envoyer du contenu exécutable ; et Initiation des communications en réseau.

Comme les logiciels d'analyse dynamique peuvent bloquer les logiciels suspects en temps réel, ils présentent un avantage par rapport aux techniques de détection anti-virus établies telles que les empreintes digitales ou l'heuristique. Il existe littéralement des trillions de façons différentes de brouiller et de réorganiser les instructions d'un virus ou d'un ver, dont beaucoup échappent à la détection par un scanner d'empreintes digitales ou une heuristique. Mais en fin de compte, le code malveillant doit faire une demande bien définie au système d'exploitation. Étant donné que le bloqueur de comportement peut intercepter toutes ces demandes, il peut identifier et bloquer les actions malveillantes, même si la logique du programme semble être obscurcie.

L'analyse dynamique seule a ses limites. Comme le code malveillant doit s'exécuter sur la machine cible avant que tous ses comportements puissent être identifiés, il peut causer

des dommages avant d'avoir été détecté et bloqué. Par exemple, un nouveau logiciel malveillant peut mélanger un certain nombre de fichiers apparemment sans importance sur le disque dur avant de modifier un seul fichier et d'être bloqué. Même si la modification proprement dite a été bloquée, l'utilisateur peut être incapable de localiser ses fichiers, ce qui entraîne une perte de productivité ou pire encore.

Approches de balayage du périmètre

Le prochain endroit où un logiciel anti-virus est utilisé est le pare-feu et l'IDS d'une organisation. Il est généralement inclus dans les services de courrier électronique et de proxy Web fonctionnant sur ces systèmes. Il peut également être inclus dans la composante d'analyse du trafic d'un IDS. Cela permet au logiciel anti-virus d'accéder aux logiciels malveillants en transit sur une connexion réseau vers n'importe quel système de l'organisation, ce qui donne une vue à plus grande échelle de l'activité des logiciels malveillants.

Ce logiciel peut également inclure des mesures de prévention des intrusions, bloquant le flux de tout trafic suspect, l'empêchant ainsi d'atteindre et de compromettre un système cible, qu'il soit à l'intérieur ou à l'extérieur de l'organisation. Toutefois, cette approche se limite à l'analyse du contenu du logiciel malveillant, car il n'a accès à aucun comportement observé lorsqu'il s'exécute sur un système infecté. Deux types de logiciels de surveillance peuvent être utilisés :

- Les moniteurs d'intrusion : Ils sont situés à la frontière entre le réseau de l'entreprise et l'internet. Ils peuvent faire partie du logiciel de filtrage d'entrée d'un routeur de frontière ou d'un pare-feu externe ou d'un moniteur passif distinct. Un pot de miel peut également capturer le trafic entrant de logiciels malveillants. Un exemple de technique de détection pour un moniteur d'entrée consiste à rechercher le trafic entrant vers des adresses IP locales inutilisées.
- Moniteurs d'entrée : Ils peuvent être situés au point de sortie des différents réseaux locaux du réseau d'entreprise ainsi qu'à la frontière entre le réseau d'entreprise et l'internet. Dans le premier cas, le moniteur de sortie peut faire partie du logiciel de filtrage de sortie d'un routeur ou d'un commutateur de réseau local. Comme pour les moniteurs d'entrée, le pare-feu externe ou un pot de miel peut abriter le logiciel de surveillance.

En effet, les deux types de moniteurs peuvent être placés au même endroit. Le moniteur de sortie est conçu pour détecter la source d'une attaque de logiciel malveillant en surveillant le trafic sortant pour détecter des signes d'analyse ou d'autres comportements suspects.

La surveillance du périmètre peut également aider à détecter et à répondre à l'activité du botnet en détectant les modèles de trafic anormaux associés à cette activité. Une fois que les robots sont activés et qu'une attaque est en cours, cette surveillance peut être utilisée pour détecter l'attaque. Toutefois, l'objectif premier est d'essayer de détecter et de désactiver le botnet pendant sa phase de construction, en utilisant les différentes

techniques de balayage, en identifiant et en bloquant les logiciels malveillants qui servent à propager ce type de .

Références

1. Computer Security : Principles and Practice , fourth edition by: William Stallings and Lawrie Brown, global edition 2018.