



# L'origine de tous les maux... Les Rootkit dévoilés

Les rootkits sont-ils à l'origine de tous les maux ?  
Ou n'est-ce qu'une branche de l'arbre des menaces?

*Ce qu'il faut savoir sur la menace  
que représente les rootkit*

**David Harley**

Consultant et écrivain sur la sécurité

et

**Andrew Lee**

Directeur du bureau de recherche de Eset NOD32

---

## A PROPOS DES AUTEURS

### **David Harley - auteur et consultant indépendant**

David Harley fait des recherches et publie sur les logiciels nuisibles et d'autres problèmes de sécurité depuis la fin des années 80. A partir de 2001, il a travaillé au Service de Santé National de Grande Bretagne (National Health Service) en qualité de manager de la sécurité des infrastructures au niveau national, où il s'est spécialisé dans le traitement des logiciels malveillants et de toutes les formes d'utilisation abusive des emails, tout en dirigeant le Centre de traitement des menaces (Threat Assessment Centre). Depuis Avril 2006, il travaille comme auteur et consultant indépendant.

Il est le co-auteur de « Virus revealed », et a collaboré à beaucoup d'autres livres sur la sécurité et l'éducation chez de nombreux éditeurs, sans compter d'innombrables articles et conférences sur le sujet.

### **Andrew Lee – Directeur de recherche CISSP Eset NOD32**

Andrew Lee, CISSP, est directeur de recherche chez Eset LLC éditeur de NOD32. Il est l'un des fondateurs du réseau d'échange d'informations anti-virus (Anti-Virus Information Exchange Network -AVIEN) et de son groupe frère l'AVIEWS, (AVIEN Information Early Warning System – Système d'Information Rapide d'AVIEN) ; il est membre de l'AVAR et correspondant de l'organisation WildList. Il était précédemment au cœur de la défense contre les logiciels malveillants comme administrateur senior de la sécurité d'une importante organisation gouvernementale.

Andrew a écrit de nombreux articles sur les divers aspects des logiciels malveillants, et intervient fréquemment dans des conférences ou des congrès comme l'AVAR, le Virus Bulletin ou l'EICAR.

## Introduction

Les connaissances du public sur les rootkits ont grandi ces dernières années, mais, comme pour les vers, les virus et les autres formes de logiciels malveillants (les malware), le terme rootkit est employé indifféremment pour une grande variété de technologies, et est affublé d'un grand nombre de définitions pas toujours compatibles.

Bien que plusieurs de ces technologies et définitions soient explorées dans cet article, notre intention est de clarifier les usages habituels et non de proposer une définition « qui fait autorité ». On trouvera cependant quelques définitions dans le glossaire.

Les rootkits sont en passe de devenir les derniers d'une longue liste de menaces mal comprises et présentées comme « la fin de l'informatique telle que nous la connaissons ». On en a parlé [1] comme de « la forme d'attaque la plus sophistiquée et la plus pernicieuse qui puisse être portée contre un système Windows », faisant naître les mêmes appréhensions superstitieuses que les termes « furtif » ou « polymorphe » avaient inspiré en leur temps dans l'histoire des logiciels malveillants. En fait, les concepts de rootkits et de programmes furtifs (souvent appelés stealthware) sont assez proches et se recouvrent, s'ils ne sont pas synonymes. Ce livre blanc cherche à estimer les réalités de la menace des rootkits et à examiner les solutions aujourd'hui disponibles.

Il est facile de comprendre pourquoi le concept de rootkit est si inquiétant. Les programmes utilisant les techniques furtives sont conçus pour être invisible aux logiciels anti-software, aux autres logiciels de sécurité, au système d'exploitation et au système de gestion de fichiers. Bien que, d'une certaine manière, la technique des rootkits pose une série de défis à l'industrie de la sécurité, les technologies évoluent des deux cotés et si les rootkits ont été dans un passé récent une préoccupation des spécialistes de la sécurité, en particulier dans le milieu UNIX/Linux, les techniques furtives ne sont pas des nouveautés pour l'industrie anti-virus.

## Rooting - Racine atteinte ?

Dans un système Unix, on appelle « racine », l'utilisateur le plus privilégié. Cet utilisateur a les pouvoirs les plus étendus sur le système et est par conséquent, celui que les agresseurs vont chercher à compromettre en priorité.

« La racine » ou « le répertoire racine » désigne également la principale partie du répertoire du système de fichiers UNIX, c'est-à-dire le plus haut niveau dans la représentation classique des répertoires en arborescence (Pourquoi les branches d'un répertoire poussent vers le bas, voilà une bonne question !).

Le répertoire racine, généralement désigné par un simple slash « / » est le répertoire par lequel on peut atteindre toutes les autres branches (c'est à peu près l'équivalent du « C:\ » de Windows). En général, les utilisateurs ordinaires ne peuvent pas modifier les fichiers de ce répertoire ou de ceux dont ils ne sont pas propriétaires. Ainsi, « rooting » (atteindre la racine) un système consiste à « compromettre » l'utilisateur racine en accédant au répertoire racine et à tous les répertoires et fichiers qui en découlent.

*« On comprend facilement pourquoi le concept de rootkit est inquiétant. Les programmes qui utilisent les techniques furtives (Stealth techniques) sont conçus pour être invisibles »*

## Rootkits et programmes furtifs\*

Dans les premiers temps de la détection des malware, on définissait souvent la technologie des programmes furtifs de la manière suivante : [2, 3]

- Stealth négatif (niveau -1) : l'infection provoquait un dysfonctionnement de l'objet infecté qui rendait la détection de l'infection inévitable.
- Non-Stealth (niveau 0) : aucune mesure spécifique n'est prise pour cacher la présence de l'infection.
- Stealth élémentaire (niveau 1) : il n'y a pas de manifestation caractéristique qui attire l'attention. On effectue les étapes habituelles anti-détection, comme conserver les dates et les « Time Stamp ».
- Stealth intermédiaire (niveau 2) : on conserve une image complète ou partielle de l'objet avant son infection pour la « montrer » au système, ce qui aide à cacher la signature du virus.
- Stealth avancé (niveau 3) : on utilise des méthodes de masquage spécifiquement conçues pour cacher le virus aux logiciels de sécurité.

Bien que cette classification soit peu utilisée en dehors de la communauté anti-virus, elle reste valide non seulement dans le domaine des virus, mais aussi dans celui de tous les malware furtifs, en particulier pour les rootkits. Si l'on considère qu'il y a plusieurs années que l'on connaît et que l'on sait traiter de tels mécanismes, on ne doit pas s'inquiéter par les rumeurs disant que la technologie rootkit a pris une telle avance. La seule manière de la combattre est d'utiliser des outils spécifiques, de reformater le disque infecté et de réinstaller le système.

Potentiellement, les logiciels anti-virus doivent détecter les rootkits aussi facilement que les autres virus avant qu'ils ne s'installent ou même une fois installés. Cela ne veut pas dire qu'il n'existe ni de problème de rootkit, ni de problème de virus. Cela ne veut pas dire non plus que les tous programmes anti-virus savent traiter le problème avec le même succès, ni qu'ils réussissent aussi bien avec tous les types de rootkits. Néanmoins, le problème reste gérable.

## Définitions des rootkits

D'après Hogdlung [4], un rootkit est « un ensemble de programmes et de codes qui permet une présence régulière et indétectable dans un ordinateur ». C'est une définition très générale, admissible, mais elle ne fait pas de distinction entre les rootkits et les différents programmes furtifs\*.

Prenons alors une définition basique [5] d'une boîte à outils « malveillante » : « logiciel qui contient des scripts, des programmes, des agents autonomes qui exploitent les vulnérabilités ».

Un rootkit peut-être comparé à une boîte à outils, généralement associée à une tentative de prise de contrôle d'accès privilégiés et de maintien de ces accès en cachant le fait que le système a été corrompu. On peut alors le définir comme un ensemble de programmes malveillants qui permet à un intrus de cacher la corruption du système et de continuer à utiliser cette corruption. En réalité, la situation est plus complexe mais avant de l'étudier plus en détails, nous devons comprendre certaines choses à propos des comptes administrateur et de leurs privilèges. Après cela, nous pourrions définir des boîtes à outils étudiées dans des environnements spécifiques.

*“Potentiellement, les logiciels anti-virus doivent détecter les rootkits aussi facilement que les autres virus »*

\*Programmes furtifs : Stealth

---

Pour l'instant, cependant, une définition valide du travail d'un rootkit pourrait-être une boîte à outils spécifique, installée dans un système corrompu dans le but de :

- Maintenir des accès et contrôles privilégiés
- Permettre à un individu ou à un programme d'utiliser ces accès comme bon lui semble
- Cacher ou restreindre l'accès à des objets ou process tels que : process, traçage, fichier, dossier (répertoire/sous-répertoire), entrée de registres, port.

Notez que ces définitions ne présupposent pas :

- Une intrusion – c'est-à-dire un accès non autorisé
- Une action ou une intention malveillante
- Du « rooting » (la recherche d'une autorisation d'accès et de privilèges interdits). Nous explorerons ce concept en détail plus loin)

L'abandon de la présomption de piratage malveillant peut très bien se défendre. La plupart des OS multi utilisateurs utilisent une forme de dissimulation ou de restriction d'accès pour les fichiers systèmes sensibles aux utilisateurs non privilégiés. Ceci est fait pour des raisons de sécurité tout à fait légitimes : il faut empêcher les utilisateurs finaux d'accéder à des données auxquelles ils n'ont pas droit ou les empêcher de modifier ou supprimer des données ou des programmes, ce qui endommagerait le système.

Vue sous cet angle, la sécurité ne présuppose pas de la malveillance de la part des utilisateurs finaux et peut aisément s'intégrer dans une version légèrement modifiée de la classification des modèles de programmes furtifs décrits plus haut. Cependant, cette définition couvre l'utilisation de techniques plus avancées de stealth/rootkits pour des raisons tout aussi légitimes (voir le cas du Rootkit Sony, plus loin dans l'article).

## Un guide du compte administrateur en 60 secondes

Le plus petit des PC actuels a plus de ressources et de puissance que la plupart des ordinateurs et mainframes de jadis. L'évolution des PC, allant d'une utilisation mono utilisateur, avec des possibilités de réseaux équivalentes à celles d'un terminal passif, vers une utilisation de machines serveurs est bien moins évidente. Généralement, les PC et les portables sont utilisés par une seule personne. Néanmoins, les PC modernes et les OS peuvent supporter non seulement des process et des zones d'utilisateurs multiples, mais aussi de multiples process où plusieurs utilisateurs sont connectés. Et ce, tout en étant utilisés pour offrir un service à des utilisateurs à distance (c'est-à-dire de travailler à la fois comme un serveur et comme un poste de travail).

Un compte utilisateur ne permet pas seulement d'accéder au système, mais définit également les droits de l'utilisateur. La plupart des utilisateurs ont des droits d'accès et des privilèges limités. On peut donner ou retirer certains droits ou privilèges aux utilisateurs en fonction de leur statut ou de leur rôle, mais également en fonction de leur appartenance à certains groupes. Les groupes peuvent être définis sur la base d'intérêts, de fonctions, d'emplacements géographiques équivalents, d'appartenance à un même réseau/sous-réseau, etc. Les administrateurs peuvent généralement installer ou modifier des programmes, avoir accès aux comptes des autres utilisateurs, ce qui n'est pas le cas des autres utilisateurs. Leurs droits peuvent aussi être restreints en fonction d'une hiérarchie, par exemple, un administrateur peut avoir tous les droits sur un ensemble de serveurs ou de domaines, mais pas sur les autres.

---

Le compte « racine » a généralement tous les privilèges des administrateurs (super-utilisateur) sur les systèmes UNIX/Linux et non-UNIX. Le compte administrateur par défaut de Windows est à peu près équivalent au compte « racine » d'UNIX. Bien que les systèmes OS X MacIntosh sont basés sur BSD UNIX, le compte administrateur accessible depuis l'interface graphique est assez différent mais le principe reste le même.

## Privilèges utilisateurs et "Rooting"

"Rooting" est le terme employé pour désigner le droit d'accès à la racine et donc le contrôle de tout le système UNIX/Linux. On peut atteindre ce point par escalade directe dans les droits, c'est-à-dire en exploitant les vulnérabilités du système pour acquérir des droits plus élevés. Il peut aussi être atteint par l'action d'un virus, comme l'explique Cohen [6] aux débuts des recherches sur le malware. L'utilisation du terme « rooting » paraît curieuse dans l'environnement PC/Windows où le compte racine n'a pas de signification particulière. Cependant, on peut l'utiliser de manière générique pour décrire l'accès à une position privilégiée dans les systèmes UNIX/Linux et tous les autres.

## Les objectifs des rootkits

Le principal objectif d'un rootkit n'est pas nécessairement de devenir la "racine" du système, c'est-à-dire d'y entrer par effraction, bien qu'il puisse très bien inclure des programmes qui permettent d'obtenir ces accès administratifs. Plus souvent, son objectif est d'autoriser un intrus à prendre pied dans le système, à s'y maintenir et à y travailler d'une manière indétectable.

Cependant, certains auteurs distinguent rootkits et stealthkits. Avec cette distinction, un rootkit peut être défini comme une boîte à outils qui inclut un outil permettant de prendre le contrôle de la racine du système. Certains ont argumenté dans ce sens [7].

Néanmoins, cet article prend en compte un grand nombre de types et de composants de rootkits, plutôt que de s'attacher à une vue puriste de ce qu'est ou n'est pas un rootkit. Non seulement cette approche crée une clarification, mais elle s'attache aux réalités et à la multiplicité des attaques dans le monde réel.

Les objectifs secondaires d'un rootkit peuvent être de :

- Cacher les traces d'un intrus dans un système corrompu
- Cacher la présence d'applications ou de processus malveillants
- Cacher les activités de programmes déguisés en fichiers légitimes
- Cacher la présence de patches vers des versions antérieures (downgrade), de backdoors, de trappes d'entrée
- Recueillir des informations auxquelles l'intrus n'aurait pas accès ou pas totalement accès. Cela peut comprendre des données sur le système corrompu, des informations sur le trafic du réseau, etc.
- Utiliser le système corrompu comme intermédiaire pour accomplir d'autres intrusions ou attaques malveillantes.
- Stocker d'autres applications malveillantes et agir comme un serveur pour des mises à jour de programmes amorces, etc.

*"Un rootkit permet à un intrus de prendre pied dans le système, de s'y maintenir et d'y travailler d'une manière indétectable »*

## Composants traditionnels d'un rootkit UNIX

Des programmes utilitaires infestés par un cheval de Troie sont substitués aux utilitaires légitimes pour masquer la présence ou l'empreinte d'un intrus. Ainsi, un intrus peut accéder là où il veut et/ou recueillir des informations. Dans un environnement UNIX, des utilitaires comme « top », « ps », « login » et « password » sont des cibles habituelles pour une substitution [8]. Néanmoins, tout programme qui peut être utilisé pour pénétrer dans la racine est une cible naturelle pour un rootkit parce qu'il aide l'intrus à acquérir ou maintenir des privilèges. Ce ne sont pas là nécessairement les moyens utilisés pour accéder au système corrompu mais il peut y avoir beaucoup d'autres utilisations possibles. Par exemple, corrompre d'autres comptes ou systèmes, ou réobtenir des accès privilégiés malgré un effort pour réparer une intrusion précédente.

Les programmes qui peuvent révéler la présence d'un intrus à un administrateur comme « last », « ls », « netstat » et « ifconfig » [9] sont également des cibles pour une substitution. Les daemons\* peuvent aussi être des cibles, aussi bien pour se cacher que pour recueillir des informations.

## Rootkits Windows

Le terme Rootkit Windows est utilisé généralement pour décrire des programmes qui cachent des process, des fichiers ou des clés de registre à l'operating system. En fait, les rootkits Windows peuvent avoir des fonctionnalités très semblables à celles des traditionnels rootkits UNIX, bien que les mécanismes exacts varient suivant les plateformes.

*« Le terme rootkit Windows est utilisé généralement pour décrire des programmes qui cachent des process, des fichiers ou des clés de registre à l'operating system »*

Les versions dérivées de Windows-NT comme XP utilisent un modèle de sécurité multi-comptes très semblable aux anciens systèmes comme VMS ou UNIX, bien que la terminologie et les mécanismes puissent être très différents. Les anciennes versions de Windows sont très dépendantes d'applications externes pour leur sécurité, si bien que de très nombreux rootkits ont très peu à faire pour s'introduire comme fichier de sécurité système légitime.

Comme les systèmes UNIX, les versions dérivées de Windows-NT ont des degrés d'accès privilégiés. Ceci n'est pas uniquement un problème d'accès des utilisateurs aux données, mais d'accès au noyau du système. Les versions dérivées de NT acceptent deux modes d'exécution (ou deux niveaux de privilèges) : le mode utilisateur et le mode noyau (kernel). Les processeurs modernes x86 supportent en fait quatre niveaux de privilèges mais deux seulement sont supportés par NT qui était sensé être portable sur des processeurs non Intel.

Les utilitaires de nettoyage de logs et de même type sont aussi sur le chemin des intrus qui peuvent par exemple supprimer des fichiers log (soit les entrées relatant leur intrusion). D'autres formes de chevaux de Troie peuvent aussi être trouvées dans les rootkits habituels, comme ceux utilisés pour recueillir des informations (tels que les « keyloggers » et les « packet sniffers ») ou ceux permettant un accès futur (les backdoors). Un rootkit typique pour UNIX ou Linux contient vraisemblablement des substitutions d'utilitaires tels que les « port/shell daemons », des utilitaires pour gravir la hiérarchie des privilèges, des utilitaires pour surveiller l'utilisation des ressources (pour pouvoir cacher des fichiers), des « renifleurs » pour surveiller le trafic réseau et des vers pour cacher process, logs et connexions.

*\*daemons : programmes comme « inetd », « rshd » et « syslogd » qui travaillent en background ou en process système plutôt que d'être appelés par l'utilisateur.*

---

Les niveaux de privilèges sont faits pour protéger le niveau kernel (niveau 0) de telle manière que les données système ne puissent pas être modifiées ou remplacées par un process non privilégié. Les applications ordinaires dans un système de type NT tournent au point de vue des privilèges dans le niveau 3, le plus bas des niveaux.

### Mode utilisateur versus mode Kernel

Bien que la distinction entre utilisateurs privilégiés et administrateurs ne recouvre pas exactement la distinction entre mode kernel (niveau 0) et mode utilisateur, il y a une relation étroite entre les deux. Le kernel peut être défini comme le véritable cœur de l'operating system [10] : les services système travaillent en mode kernel de telle manière qu'un utilisateur sans privilège ne peut pas faire de modification comme supprimer ou ajouter des drivers, des périphériques, des programmes sans autorisation. Les applications utilisateurs sont généralement disponibles à tous et travaillent en mode utilisateur, limitant la capacité pour une application d'endommager les process système par des modifications inappropriées.

Un rootkit en user mode travaille à la place ou à l'intérieur d'une application utilisateur en se collant à l'API (Application Programming Interface) Windows dans chaque process applicatif. Chaque utilisateur travaille dans son propre espace mémoire, donc un rootkit en mode utilisateur doit modifier l'espace mémoire de chaque application qui tourne pour pouvoir « filtrer » ce que le système voit de l'application. Le rootkit doit donc contrôler le système pour pouvoir modifier l'espace mémoire avant qu'une nouvelle application ne soit pleinement lancée. Une manière classique d'atteindre ce but est de modifier les DLL (Dynamic Link Libraries) système au moment du lancement.

*“Un rootkit en user mode travaille à la place ou à l'intérieur d'une application utilisateur en se collant à l'API Windows dans chaque process applicatif.”*

Le mode kernel permet des accès privilégiés à la mémoire système, au système complet d'instructions CPU. Un rootkit peut alors intercepter les API kernel. Ses process, fichiers, clés de registre seront ainsi cachés. Quand une requête est lancée par un programme utilisateur, l'information retournée est filtrée pour en masquer la preuve. Un rootkit en mode kernel a sensiblement tous les pouvoirs pour détruire ou manipuler le système. Comparé à un rootkit en mode application, il est beaucoup plus difficile à installer et à maintenir à cause de sa complexité.

*“Un rootkit en mode kernel a pratiquement tous les pouvoirs pour détruire ou manipuler le système”*

S'intégrer dans le kernel est plus fiable et plus facile pour un process, en ce sens que tous les process partagent la même adresse mémoire, mais doit être fait par un utilisateur privilégié (parfois sans qu'il en ait conscience).

Notez que s'intégrer (« Hooking ») n'est pas la seule façon qu'un rootkit peut utiliser pour cacher un objet. Le process DKOM (Direct Kernel Object Manipulation), au lieu de filtrer les informations rendues par le kernel, modifie directement les objets de service que l'operating system créé pour des raisons d'audit, afin de cacher process et drivers. Un rootkit qui utilise cette méthode peut cacher un process en rendant invisible un objet associé à ce process [11]: il est alors beaucoup plus difficile à la sécurité ou à tout programme de détecter sa présence.

## Rootkit permanents versus rootkits non permanents

Beaucoup de rootkits sont permanents. Ils sont installés sur disque et s'accrochent à la séquence de démarrage du système, de manière à survivre à un redémarrage. Les rootkits non permanents ou rootkits en mémoire ne font pas ça. Ils installent leur code en mémoire vive et ne survivent pas à un redémarrage. On les détecte plus difficilement puisqu'il n'y a pas « d'empreintes » à découvrir dans le système par une analyse. D'un autre côté, leur efficacité est limitée au temps où le système corrompu reste en ligne puisqu'ils disparaissent au premier redémarrage, ce qui ne touche pas les systèmes qui sont rarement redémarrés comme les serveurs.

*“Les rootkits non permanents installent leur code directement en mémoire et ne survivent pas à un redémarrage.”*

## Rootkits Macintosh

Quelques rootkits PoC existent sur Apple OS X. Leur impact a été faible et leur importance tend à être minimisée par la communauté des utilisateurs de Mac car il leur faut accéder à la racine pour s'installer – ce qui veut dire qu'ils ne peuvent pas obtenir de privilèges. La même attitude se rencontre en face des quelques exemples connus de malware Mac. En fait, les utilisateurs Mac nient qu'il y ait le moindre virus OS.X car ceux qui existent nécessitent une action utilisateur pour infecter le système. Le fait est cependant aussi vrai pour la plupart des virus Windows. Il est plus compliqué pour un utilisateur Mac de travailler en utilisateur privilégié à moins de le vouloir vraiment. Il est cependant naïf de croire qu'un utilisateur Mac ne va jamais faire la folie de vouloir travailler sur la racine et permettre ainsi à un rootkit de s'installer.

Alors que les utilisateurs Mac font la gestion de la plupart des privilèges, il n'est pas particulièrement courant pour les rootkits Windows de pouvoir atteindre la racine ou plus précisément atteindre par eux-mêmes les privilèges d'administrateur sans l'assistance de la victime (psychologiquement manipulée). Cependant, on peut arguer que Windows est plus coulant pour laisser un utilisateur travailler avec des privilèges d'administrateur.

## Les bonnes intentions et l'expérience du rootkit de Sony

L'utilisation de la technologie des programmes furtifs (stealth) ou des rootkits n'est pas limitée à ceux qui veulent entrer dans les systèmes comme des « hackers » habituels. Elle peut être utilisée pour d'autres formes de malware comme les virus, les vers, les chevaux de Troie, pour leur permettre de cacher leur présence.

Les « greyware » (programmes qui se situent entre les logiciels autorisés et le véritable malware) comme les publicités non sollicitées, quelques programmes espions, les « trackware », etc. Ils sont définis avec précaution par certains éditeurs de solutions antivirus comme des « programmes potentiellement indésirables ». Ceci permet d'éviter les complications légales quand le fabricant peut soutenir que le programme est légitime. Dans certains cas, il peut même s'agir de programmes légitimes qui ont été corrompus pour des raisons malveillantes. Ceux-là dépendent habituellement d'un degré de dissimulation. Bien que leur présence soit parfois très évidente par l'apparition de pop-ups, de réacheminement vers une URL inattendue, etc, il faut prêter une attention considérable pour prévenir la détection et la suppression de fichiers programme réels.

*“La plupart des gens ne savent même pas ce qu'est un rootkit, alors pourquoi devraient-ils s'en méfier” Thomas Hesse, President, Global Digital Business, Sony BMG.*

---

Quelques commentateurs sur le terrain [4] veulent faire remarquer que la technologie rootkit (ou assimilée) peut être utilisée pour des raisons légitimes. Celle-ci est nécessaire pour pallier les insuffisances du système lorsqu'il s'agit de cacher des données. Quelques-uns des domaines parfois cités sont :

- La gestion des droits de propriété intellectuelle
- La protection des programmes contre le désassemblage (reverse engineering)
- L'évaluation de la menace d'intrus
- La surveillance des employés
- La gestion des droits
- La protection contre les malware ou les erreurs d'utilisateurs des programmes de sécurité
- La sauvegarde des programmes et données
- Les logiciels de restauration
- L'encryptage et la dissimulation des données dans les systèmes multi utilisateurs

Beaucoup n'approuvent pas l'utilisation d'un vocabulaire lié aux rootkits dans ces domaines, rendant parfois confuse les définitions habituelles. Après tout, Windows lui-même utilise de telles techniques pour cacher quelques fichiers système importants. En Octobre 2005, Mark Russinovitch note dans son blog « Sysinternals » [12], la découverte de ce qui semblait être un rootkit dans son système. Il s'avéra qu'il s'agissait du fameux rootKit Sony, qui résultait d'une confusion sérieuse entre le très légitime DRM (Digital Rights Management) et la technologie stealth/rootkit. On peut définir le DRM comme la technologie utilisée pour contrôler l'accès aux données et aux matériels pour protéger les droits des propriétaires des publications ou des copyrights.

Sony utilisait XCP (Extended Copy Protection) de First 4 Internet Ltd pour contrôler l'accès à certains CD audio. Les disques protégés par XCP restreignaient le nombre de copies que l'on pouvait faire et contrôlaient aussi la conversion de la musique dans un format numérique permettant de la stocker et la rejouer sur un ordinateur ou un lecteur audio.

On ne pouvait pas lire le CD dans l'ordinateur sans installer le logiciel qui cachait fichiers, process et valeurs de clés de registres en modifiant l'exécution des fonctions API. Il réalisait cela en utilisant la technique classique des rootkits de « patcher » le Service System table (SST).

Le droit de Sony d'éviter une distribution illicite de ses produits est généralement bien admis et il ne semble pas normal de considérer cela comme un rootkit malveillant. Cependant, beaucoup reprochent le fait que Sony ait modifié le système final sans dire clairement ce qu'il l'avait fait ou la manière de le désinstaller. Cela pouvait les mettre en infraction vis-à-vis de la loi qui punit les modifications ou accès illégitimes. Pire, la solution n'était pas bien conçue, codée et créait une vulnérabilité immédiatement reconnue avec gratitude par la communauté des hackers. En fait, ce « rootkit » était exploité pour cacher un cheval de Troie qui n'avait rien à voir avec Sony et son programme de protection des droits. Cet exemple montre la différence très subtile entre l'intention malveillante et la vulnérabilité involontaire. Il est clair que ni Sony, ni First 4 ne voulaient ou ne s'attendaient à ce que leur projet DRM ouvre un espace exploité par des malware. Il est aussi clair que Sony n'a pas géré la publicité qui en a découlée d'une manière appropriée. La désinstallation était rendue possible à ceux qui remplissaient un formulaire. Elle offrait le choix strict entre désinstaller le logiciel (donc ne plus pouvoir utiliser les CD protégés par XCP) ou simplement désactiver la fonction de dissimulation. Il apparut que le patch initial avait été très peu testé et comportait une fonctionnalité « téléphone » qui n'était pas mentionnée [13] dans l'EULA (End User License Agreement).

La technologie XCP a peu impressionné la communauté rootkit bien qu'elle ait attiré l'attention de quelques auteurs de malware sur la manière de profiter de la vulnérabilité créée. Elle n'a pas eu de toutes manières une grande influence sur l'industrie en général.

Il n'est pas invraisemblable qu'il y aura des tentatives de résolution du problème des rootkits par des moyens légaux [10]. Cela peut prendre la forme de législation spéciale contre les rootkit de type malveillant mais peut aussi conduire à la mise hors la loi des technologies rootkits même pour des motifs légitimes. Si cela arrivait, toute application qui utiliserait une forme quelconque de code furtif pour protéger du code ou des données serait mise en danger vis-à-vis de la loi. Cela aurait de graves conséquences sur les applications DRM qui sont généralement dépendantes dans une certaine mesure de dissimulations et d'accès limités.

*"Il n'est pas invraisemblable qu'il y aura des tentatives de résolution du problème des rootkits par des moyens légaux.."*

Même si de telles mesures ne sont pas prises, le regard des utilisateurs sur ces produits pose problème. Prenons l'exemple des produits Symantec (une corbeille sécurisée qui permet une restauration complète des fichiers supprimés qui pourraient ne pas être restaurés par la corbeille de Windows) facilitant l'utilisation de manière légitime, documentée et utile. Pourtant, après quelques gros titres sur le « rootkit de Symantec », on a supprimé son aspect caché [14] pour diminuer le risque d'exploitation par des logiciels malveillants. Ceci a eu des conséquences sur d'autres logiciels de sécurité, certains devant agir en premier. Par exemple, l'analyse de comportement et le blocage des logiciels utilisent les mêmes techniques d'accrochage aux API que les rootkits. Ceci pour des raisons de détection et non de dissimulation.

D'autres fonctions de dissimulation sont utilisées par la sécurité ou d'autres logiciels. Par exemple pour empêcher le désassemblage, l'utilisation personnelle par des utilisateurs de code malveillant (virus mis en quarantaine, etc.), pour éviter l'ingérence dans les réglages du système, etc.

## Méthodologie de détection des rootkits

Trop souvent quelqu'un « découvre » qu'une détection de malware « basée sur les signatures » est défectueuse car elle ne peut pas détecter de nouveaux virus. Heureusement, il y a plusieurs années que les antivirus ne se contentent pas de détecter uniquement les virus connus. Une large panoplie de techniques (l'analyse heuristique, les drivers génériques, la gestion de comportement, etc.) sont d'ores et déjà utilisés pour améliorer la détection des nouvelles menaces et leurs variantes. Comme pour les virus furtifs, le problème des rootkits repose sur qui agira le premier.

Il est difficile en scannant de détecter ce qui est déjà installé et qui cache la preuve d'une infection. Cependant, les éditeurs d'antivirus ont des années d'expérience pour trouver des moyens de contourner les nouveaux mécanismes de dissimulation, une fois le malware analysé. En général, les rootkits sont détectés en analysant le système de fichier et la mémoire avec une analyse basée sur les signatures.

*« L'augmentation des rootkits non permanents renforce le besoin d'analyse mémoire et de détection de process cachés »*

Une autre application pour détecter l'activité d'un rootkit est l'utilisation de méthodes heuristiques. Elles consiste essentiellement à surveiller les différences entre l'observation d'un système sain (raisonnablement vraisemblable) et d'un système pollué par la technologie rootkit [15].

Classiquement, les systèmes UNIX et similaires ont été bien servis par ce que l'on peut appeler le « lien de détente » ou l'approche de réconciliation par objets [9], plus communément appelé le contrôle d'intégrité par l'industrie antivirus. Cela peut encore être utile dans le contexte Windows mais tend à créer un taux élevé « d'overhead » car il y a de multiples endroits où les modifications à l'environnement (exécutables, valeurs des registres, configurations de fichiers) sont monnaie courante. L'augmentation des rootkits non permanents renforce le besoin d'analyse mémoire et de détection de process cachés, plutôt que de s'en remettre aux modifications dans le système de fichiers comme indicateur d'infection. Il importe aussi d'avoir une approche plus proactive de détection des variantes. Plus les produits sont intelligents lors de la distinction entre les piratages malveillants, les paramètres de registre et leur contreparties légitimes, plus l'approche heuristique deviendra efficace [10]. Les antivirus sont devenus ces dernières années des adeptes de la suppression heuristique des virus [16]. Cependant, la difficulté de la suppression heuristique (où l'identification exacte du virus n'est pas possible) demeure, et plus encore, en cas de malware non-viraux.

Même s'il est exagéré de dire que la seule solution pour sauver un système infecté par un rootkit est de formater, il peut être plus efficace de récupérer une image système que de supprimer simplement le rootkit (même pour les malware bien connus mais très imbriqués, en particulier quand d'autres exécutables ont été touchés ou altérés). Réaliser cela sans dommages excessifs aux données et à la productivité exige cependant de porter une attention particulière aux techniques de backup (de données comme d'applications ou de système) et de restauration.

## Mesures préventives

Il y a des contre mesures qui s'appliquent à toutes les plateformes. De bonnes pratiques de sauvegarde sont une défense vitale contre les menaces et les désastres inopinés et doivent être une pierre angulaire de toute politique de sécurité.

Les administrateurs ne devraient pas travailler comme administrateur racine sauf s'ils sont dans une session qui exige d'abord des droits privilégiés pour travailler. Toutes les plateformes modernes permettent de permuter entre les comptes sans rebooter quand des accès privilégiés sont nécessaires.

Il peut être dangereux de se reposer sur un logiciel de sécurité en open source. Beaucoup de communautés de projet ont produit d'excellents travaux mais il est parfois difficile de juger la compétence (et quelquefois les bonnes intentions) de tous les intervenants dans le projet. Il n'est pas normal pour une compagnie commerciale ou pour un organisme public de confier sa sécurité à un produit sans garantie et sans responsabilité financière. Les particuliers peuvent être moins concernés par ces considérations mais n'ont pas envie d'être mis en difficulté par un produit qui n'apparaît pas conforme à leur attente. Il est tout aussi important de mener des tests de vulnérabilité, de maintenir régulièrement des processus de surveillance des patch et d'éviter les manipulations hasardeuses.

*« De bonnes pratiques de sauvegarde sont une défense vitale contre les menaces et les désastres inopinés »*

## Conclusion

La technique expérimentale « Blue Pill » de Joanna Rutkowska prétend créer « un malware indétectable à 100% qui n'est pas basé sur *l'obscurité du concept* » [17], en exploitant la technologie de virtualisation AMD de SVP/Pacifica. Jusqu'à présent, trop peu de détails ont été donnés sur cette approche pour évaluer ses déclarations. Elle semble être basée sur un rootkit non permanent travaillant dans une machine virtuelle. Le rootkit SubVirt [18] utilise aussi la virtualisation mais est permanent (c'est-à-dire qu'il survit à un redémarrage). C'est une preuve intéressante du concept mais il n'est en aucun cas indétectable. Il serait intéressant de voir sur un cas approprié si Blue Pill est réellement supérieur de ce point de vue.

De toutes les manières, il est préférable de penser que la course aux malware déclenchée va perdurer. Ni la panique, ni l'assurance excessive ne sont appropriées, seule la vigilance l'est. Beaucoup d'auteurs de malware sont en train d'utiliser la technique de rootkit pour cacher leurs créations mais la fréquence de tels objets est encore très faible. De plus, la nature même du malware tend à le faire détecter facilement par ses actions.

Les annonces sur la mort de l'antivirus sont très exagérées. En fait, le terme « antivirus » est trompeur pour parler des solutions de sécurité actuelles. Le temps des programmes antivirus ne détectant que les virus est déjà loin. Beaucoup de produits sont maintenant capables de détecter une très grande variété de menaces de malware.

Quand les menaces ont évolué, l'industrie de l'antivirus n'est pas restée inactive et quelques produits antivirus ont eu des succès significatifs dans la détection des rootkits. Cependant, il est dangereux de fermer les yeux et d'attendre que d'autres prennent tout en charge.

Les antivirus « basés sur la signature » ne peuvent pas protéger contre les nouvelles menaces qui sont très différentes des menaces passées même s'ils sont régulièrement mis à jour. Les utilisateurs doivent utiliser des produits qui ont fait leurs preuves dans l'utilisation d'analyses heuristiques avancées comme dans d'autres méthodes de détection dites proactives. La vigilance, l'évaluation des techniques anti-menaces et des capacités réelles des produits reste le premier pas vers une sécurité aboutie.

## Références

1. University of Minnesota ResNet FAQ:  
[http://www.resnet.umn.edu/html/rn\\_security.html](http://www.resnet.umn.edu/html/rn_security.html)
2. "Viruses Revealed". David Harley, Robert Slade, and Urs Gattiker (Osborne).
3. "Dr. Solomon's Virus Encyclopaedia". Dr. Alan Solomon and Dmitry Gryaznov. (S&S International).
4. "Rootkits are not Malware". Greg Hoglund.  
<http://www.rootkit.com/newsread.php?newsid=504;>  
[http://www.sysinternals.com/Forum/forum\\_posts.asp?TID=5798](http://www.sysinternals.com/Forum/forum_posts.asp?TID=5798)
5. "Using a 'common language' for computer security incident information". By John D. Howard & Pascal Meunier, in Computer Security Handbook (4th Edition) ed. Seymour Bosworth & M.E. Kabay (Wiley).
6. "A Short Course on Computer Viruses" 2nd Edition. Dr. Frederick B. Cohen, Wiley; "Models of Practical Defenses Against Computer Viruses". Dr. Frederick B. Cohen: <http://all.net/books/integ/vmodels.html>
7. <http://blogs.securiteam.com/index.php/archives/382>
8. Chey Cobb, Stephen Cobb, M.E. Kabay: "Penetrating Computer Systems and Networks". In "Computer Security Handbook 4th Edition", ed. Bosworth & Kabay (Wiley).
9. "Trojans" David Harley. In "Maximum Security" (SAMS).
10. "Rootkit Threats Explained". Andrew Lee. Eset, 2006.  
[http://www.eset.com/joomla/index.php?option=com\\_content&task=view&id=1401&Itemid=5](http://www.eset.com/joomla/index.php?option=com_content&task=view&id=1401&Itemid=5)
11. "Windows Rootkits of 2005" Parts 1-3. James Butler and Sherri Sparks. <http://www.securityfocus.com/infocus/>
12. "Sony, Rootkits and Digital Rights Management Gone Too Far". Mark Russinovich.  
<http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html>; "More on Sony: Dangerous Decloaking Patch, EULAs and Phoning Home". Mark Russinovich.  
<http://www.sysinternals.com/blog/2005/11/more-on-sony-dangerousdecloaking.html>
13. <http://cp.sonybmg.com/xcp/english/updates.html>;  
<http://cp.sonybmg.com/xcp/english/form14.html>
14. <http://securityresponse.symantec.com/avcenter/security/Content/2006.01.10.html>
15. "Hide 'n Seek Revisited – Full Stealth is Back". Kimmo Kasslin, Mika Stahlberg, Samuli Larvala and Antti Tikkanen. In Proceedings of the 15th Virus Bulletin International Conference, 2005.
16. "The Art of Computer Virus Research and Defense". Peter Szor (Addison-Wesley)
17. "Subverting Vista Kernel for Fun and Profit" Joanna Rutkowska. <http://theinvisiblethings.blogspot.com/>
18. "SubVirt: Implementing malware with virtual machines". Samuel T. King, Peter M. Chen, Yi-Min Wang, Chad Verbowski, Helen J. Wang, Jacob R. Lorch. <http://www.eecs.umich.edu/virtual/papers/king06.pdf>

### CONTACT

ESET NOD32

TEL (619) 319-3000

[www.eset.com](http://www.eset.com)

Distributeur exclusif pour la France

Athena Global Services

TEL 01 55 89 08 85

[www.nod32.fr](http://www.nod32.fr)