Die Wurzel allen Übels?
- Rootkits enttarnt.

Sind Rootkits die Wurzel allen Übels oder nur ein weiterer Baustein im Bedrohungsszenario? Was Sie über die Gefährdung durch Rootkits wissen sollten.

David Harley
Author und Sicherheitsberater
und
Andrew Lee
Leiter der Forschungsabteilung
ESET LLC

Michael Dankert
deutsche Übersetzung

Inhaltsverzeichnis

Über die Autoren

David Harley

David Harley forscht und schreibt bereits seit dem Ende der 80er Jahre über Schadsoftware und andere Sicherheitsprobleme. Seit 2001 arbeitete er im britischen staatlichen Gesundheitsdienst als 'National Infrastructure Security Manager', wo er sich auf die Abwehr von Bedrohungen und aller Formen von EMail-Missbrauch spezialisiert hatte sowie das 'Threat Assessment Centre' leitete. Seit April 2006 arbeitet er als unabhängiger Autor und Berater.

Er war Mitverfasser von 'Viruses Revealed' und hat nicht nur zahlreiche Kapitel zu vielen anderen Büchern von namhaften Verlagen über Computersicherheit beigesteuert, sondern auch eine Fülle von Artikeln und Beiträgen zu Konferenzen veröffentlicht.

SMALL BLUE-GREEN WORLD
8 Clay Hill House, Wey Hill, Haslemere, SURREY GU27 1DA
Telephone: +44 7813 346129
http://smallblue-greenworld.co.uk

Andrew Lee

Andrew Lee, CISSP (Certified Information Systems Security Professional), ist der Leiter der Forschungsabteilung der Firma ESET LLC. Er ist eines der Gründungsmitglieder des Anti-Virus Information Exchange Network (AVIEN) sowie seiner Schwestervereinigung AVIEWS (AVIEN Information & Early Warning System), ist Mitglied von AVAR (Association of Anti Virus Asia Researchers) und Berichterstatter für die WildList-Organisation. Bis vor kurzem war er an vorderster Front bei der Abwehr von Schadsoftware als hochrangiger Sicherheitsadministrator in einer großen Regierungsbehörde tätig. Andrew ist Autor von zahllosen Artikeln zu Sicherheitsproblemen und tritt häufig auf Konferenzen und anderen Veranstaltungen, wie AVAR, Virus Bulletin und EICAR auf.

ESET, LLC
610 West Ash Street, Suite 1900, San Diego, California 92101, U.S.A.
Telephone: +1.619.876.5400
Fax: +1.619.876.5845
http://www.eset.com

Einleitung

Die öffentliche Aufmerksamkeit in Bezug auf Rootkits ist in den letzten Jahren gestiegen, aber wie das auch mit anderen Bezeichnungen, wie Würmer, Viren oder anderen Formen von Schadsoftware (Malware) geschehen ist, wurde der Ausdruck Rootkit ziemlich unspezifisch auf eine ganze Reihe von Technologien angewendet und hat eine Vielzahl von nicht besonders kompatiblen Definitionen hervorgebracht. Während mehrere der Technologien und Definitionen in diesem Artikel untersucht werden, ist es nicht unsere Absicht, eine einzige und allgemeingültige Definition zu liefern, sondern klarzustellen, was man im Allgemeinen Sprachgebrauch unter Rootkit versteht. Im Anhang finden Sie aber auch einige kurze Definitionen.

Heutzutage besteht die Gefahr, dass Rootkits die neuesten in einer langen Reihe von ungenügend verstandenen Bedrohungen sind, die von den Medien als "Das Ende der Computer, wie wir sie kennen" aufgebauscht werden. Da man sie mittlerweile schon mit Bezeichnungen [1] wie "die schädlichste und fortgeschrittenste Angriffsform, die derzeit gegen ein Windows-System eingesetzt werden kann " versieht, werden Rootkits schon mit der gleichen abergläubischen Furcht betrachtet, wie sie die Ausdrücke "stealth" und "polymorphisch" in den frühen Jahren der Geschichte von Computerschädlingen hervorgebracht haben. Dabei sind die Konzepte von Rootkits und Programmen mit Stealth-Techniken eng verwandt und werden teilweise synonym verwendet.

> Es ist leicht zu verstehen, warum das Rootkit-Konzept so beunruhigend ist: Software, die Stealth-Techniken benutzt, ist dafür ausgelegt, unsichtbar zu sein …

Dieser Artikel zielt darauf ab, die reale Bedrohung durch Rootkits abzuschätzen und Lösungsmöglichkeiten zu ihrer Abwehr zu untersuchen.

Es ist leicht zu verstehen, warum das Rootkit-Konzept so beunruhigend ist: Software, die Stealth-Techniken benutzt, ist dafür ausgelegt, unsichtbar zu sein und sich gegenüber Anti-Virus-Software, anderen Sicherheitsprogrammen, dem Betriebssystem und dem Dateisystem zu verbergen. Obwohl Rootkits deshalb in gewisser Weise eine besondere Herausforderung für die Sicherheitsindustrie darstellen, werden die Technologien doch auf beiden Seiten des Kriegsschauplatzes weiterentwickelt und obwohl die Beschäftigung mit ihnen bis vor kurzem noch ein ganz spezielles Betätigungsfeld, besonders in der UNIX/Linux-Gemeinschaft war, so ist Stealth für die Antivirus-Industrie doch nichts Neues.

Root geworden?

Auf einem UNIX-basierten System hat der privilegierteste Nutzer den Namen "root": dieser Nutzer hat die meisten Rechte auf dem System und ist das lohnendste Ziel für jeden Angreifer. "root" oder "root-Verzeichnis" bezieht sich außerdem auf das erste Verzeichnis in der Struktur eines Dateisystems unter UNIX: es ist das oberste oder auch Wurzelverzeichnis in einem herkömmlichen Verzeichnisbaum. (Ich frage mich allerdings, warum Verzeichnisbäume nach unten wachsen?) Das Wurzelverzeichnis, das gewöhnlich mit einem einzelnen Schrägstrich "/" bezeichnet wird (annähernd vergleichbar mit "C:\" auf einem DOS- oder Windows-PC) ist das Verzeichnis, über das alle anderen erreicht werden können. Normale, unprivilegierte Anwender können in der Regel in diesem Verzeichnis keine Dateien ändern, ebenso wenig außerhalb ihrer eigenen Home-Verzeichnisse. Deshalb bedeutet auf einem System "root" zu werden, das root-Konto zu kompromittieren und dadurch auf dem System root-Rechte für alle Dateien und Verzeichnisse zu bekommen.

Rootkits und Stealth

Die Stealth-Technologie wurde in den Anfangstagen der Erkennung von Computerschädlingen etwa folgendermaßen definiert [2; 3]

- Negative Stealth (Stufe -1): die Infektion verursacht Beeinträchtigungen an der Funktionalität des infizierten Objekts, wodurch die Entdeckung unvermeidbar ist.
- Non-Stealth (Stufe 0): es werden keine besonderen Maßnahmen ergriffen, um die Infektion zu verbergen.
- Elementary Stealth (Stufe 1): es gibt keine charakteristischen Bildschirmausgaben, um sich bemerkbar zu machen. Grundlegende Maßnahmen gegen die Erkennung, wie die Wiederherstellung von Datums- und Zeitstempeln, werden ergriffen.
- Intermediate Stealth (Stufe 2): eine Kopie oder ein Teilabbild des Objekts in seinem Zustand vor der Infektion wird gesichert, um es dem System zu „zeigen" und den "Fußabdruck" des Virus zu verbergen.
- Advanced Stealth (Stufe 3): es werden Methoden zur Verschleierung verwendet, die speziell darauf ausgerichtet sind, die Infektion vor Sicherheitssoftware zu verstecken.

Während dieses Klassifizierungsschema heutzutage außerhalb der Anti-Viren-Gemeinschaft kaum verwendet wird, bleibt es doch gültig, und das nicht nur in bezug auf Viren sondern auch auf andere versteckte Schädlinge einschließlich Rootkits. Wenn wir dann noch die Tatsache in Betracht ziehen, dass Anti-Viren-Software solche Mechanismen schon seit vielen Jahren kennt und adäquat damit umgehen kann, dann sollten wir uns von Behauptungen weniger beunruhigen lassen, die besagen, dass die Rootkit-Technologie jetzt weit fortgeschritten ist und nur noch von Spezialwerkzeugen erkannt werden kann und dass der einzige Weg, mit ihr fertig zu werden, bedeutet, eine infizierte Festplatte zu formatieren und das Betriebssystem neu zu installieren.

Antiviren-Software ist demnach potentiell dazu in der Lage, Rootkits genau so leicht wie Viren zu erkennen, ganz bestimmt jedenfalls, bevor das Rootkit die Chance hatte, sich zu installieren, aber in vielen Situationen auch nach der Installation. Das heißt natürlich nicht, dass es kein Rootkit-Problem gäbe, genau so wenig, wie es kein Virenproblem gibt. Auch heißt es nicht, dass alle Antiviren-Programme mit dem Problem gleichermaßen gut umgehen können, oder dass sie alle Arten von Rootkits mit dem gleichen Erfolg bekämpfen. Das Problem ist jedoch zu bewältigen: davon geht die Welt nicht unter.

Antiviren-Software ist potentiell dazu fähig, Rootkits genau so leicht zu erkennen wie Viren

Definition von Rootkits

Nach Hoglund [4] ist ein Rootkit „eine Gruppe von Programmen und Code, die eine dauerhafte oder bleibende, unentdeckbare Anwesenheit auf einem Computer erlauben". Das ist eine sinnvolle übergeordnete Definition, sie wirft aber kein Licht auf das Durcheinander bei der Anwendung des althergebrachten Begriffs „stealth", neueren „Stealthkits" oder Rootkits. Betrachten wir deshalb eine grundlegende Definition [5] eines Werkzeugkastens zur Erzeugung von Schadsoftware: „Eine Software-Sammlung, die Skripte, Programme oder selbständige Agenten zum Ausnutzen von Schwachstellen enthält". Ein Rootkit ist so eine Art Werkzeugkasten, der dazu dienen soll, privilegierten Zugang zu erlangen, oder diesen Zugang aufrecht zu erhalten, indem die Tatsache verschleiert wird, dass das System kompromittiert ist. Es könnte deshalb als eine Sammlung schädlicher Programmierwerkzeuge definiert werden, die es einem Eindringling erlauben, die Kompromittierung eines Systems zu kaschieren und daraus weiteren Nutzen zu ziehen. In der Realität ist die Situation allerdings etwas komplizierter, aber bevor wir uns mit den Details befassen, müssen wir zuerst unser Verständnis von Nutzerrechten und der Verwaltung von Nutzerkonten auffrischen. Anschließend können wir uns typische Werkzeugen in spezifischen Umgebungen widmen.

Für den Augenblick lassen Sie uns als sinnvolle Arbeitsdefinition für ein Rootkit eine Werkzeugsammlung annehmen, die auf einem kompromittierten System installiert ist um:

- privilegierten Zugang und die Kontrolle über das System zu erhalten
- dem Eindringling und /oder der Software zu erlauben, jeden möglichen Nutzen aus diesem Zugang zu ziehen
- den Zugriff auf Objekte oder Prozesse zu verschleiern bzw. zu verhindern, wie z.B.
    - Prozesse
    - Threads
    - Dateien
    - Ordner/Verzeichnisse/Unterverzeichnisse
    - Registry-Einträge
    - Handles
    - Offene Ports

Nebenbei bemerkt, setzen diese Definitionen Folgendes nicht voraus:

- Eindringen - d.h. unbefugter Zugang
- bösartige Aktionen oder Absichten
- "root" zu werden - d.h. unangemessene und nicht autorisierte Rechte zu erlangen - damit befassen wir uns im Detail weiter unten

Die Annahme, dass Rootkits nicht unbedingt etwas mit bösartigem „Hacken" zu tun haben, ist aktuell durchaus vertretbar. Die meisten Multi-User-Betriebssysteme benutzen die eine oder andere Art der Verschleierung und/oder des eingeschränkten Zugangs zu sensitiven Daten oder kritischen Systemdateien gegenüber unprivilegierten Anwendern. Dies wird aus völlig legitimen Sicherheitsgründen so gemacht: z.B. um zu unterbinden, dass Endanwender auf Daten zugreifen, für die sie keine Berechtigung besitzen oder um zu verhindern, dass Dateien und Programme gelöscht oder verändert werden, die für das Betriebssystem und seine Datenintegrität wichtig sind. Sicherheit in diesem Sinne geht nicht von bösartigen Absichten auf Seiten des Anwenders aus und könnte durch eine leicht modifizierte Form der Klassifizierung von Stealth-Technologien von weiter oben beschrieben werden. Die Definition deckt aber auch die Anwendung von weiter fortgeschrittenen stealth/rootkit-artigen Techniken ab, ebenfalls aus Sicherheitsgründen (man betrachte dazu die Geschichte des Sony-Rootkits weiter unten in diesem Artikel).

60-Sekunden-Anleitung zur Nutzerkontenverwaltung

Selbst der einfachste zeitgemäße PC hat Resourcen und Prozessorleistung, die vielen Großrechnern oder Mini-Computern der Vergangenheit gleichwertig oder sogar überlegen sind. Weniger offensichtlich ist die Tatsache, dass die heutigen Personalcomputer sich aus autonomen Einzelplatzsystemen mit Netzwerkfähigkeiten, die kaum das Niveau von "dummen" Terminals hatten, zu Maschinen mit Leistungsmerkmalen für den Serverbereich entwickelt haben. Obwohl moderne PCs und ihre Betriebssysteme nicht nur mehrere Prozesse und Anwenderbereiche unterstützen, sondern auch viele parallel laufende Anwenderprozesse, werden doch die meisten Desktoprechner und Notebooks im Allgemeinen nur von einer Person benutzt. Damit ist nur ein Nutzer gleichzeitig angemeldet, es sei denn der Rechner stellt einen Dienst zur Verfügung, der von anderen über das Netzwerk genutzt wird. (Das heißt, der Rechner wird gleichzeitig sowohl als Server als auch als Workstation benutzt.)

Ein Rootkit erlaubt es einem Eindringling, ein unerkanntes Standbein im System einzurichten und auszunutzen.

Ein Nutzerkonto erlaubt nicht nur den Zugang zum System, sondern legt auch fest, was der Nutzer im Einzelnen auf dem System machen kann. Die meisten Anwender haben nur eingeschränkte Administrations- und Zugangsrechte. Diese Privilegien können ihnen in Abhängigkeit von ihrer individuellen Stellung und ihren Aufgaben, aber auch entsprechend der Zugehörigkeit zu bestimmten Gruppen, entzogen oder zugeteilt werden. Solche Gruppen können auf der Basis gemeinsamer Interessen oder Aufgabenbereiche, entsprechend der geographischen Lage, rechnerbezogen, abhängig von der Position im Netzwerk/Subnetz usw. definiert werden. Administratoren haben in der Regel die Rechte, Software zu installieren oder zu verändern, auf die Konten anderer Nutzer zuzugreifen und vieles mehr, was dem Standardbenutzer nicht erlaubt ist. Ihre Rechte können ebenso hierarchisch abgestuft sein: so kann ein Administrator beispielsweise volle Privilegien auf bestimmten Servern oder Domänen haben, aber nicht auf anderen. Das "root"-Konto hat normalerweise alle Administrator(Superuser)-Rechte sowohl auf UNIX/Linux als auch auf nicht-UNIX-Systemen.

Das standardmäßige Administratorkonto auf Windows-Systemen ist mehr oder weniger gleichwertig zum UNIX "root"-Konto. Während das OS X-System auf dem Macintosh auf einem BSD UNIX basiert, unterscheidet sich die Kontenverwaltung, wie sie von der graphischen Oberfläche präsentiert wird, doch ziemlich vom Standard-UNIX, das Prinzip ist aber trotzdem das Gleiche.

Nutzerrechte und Rooting

"Rooting" ist ein Ausdruck dafür, „root"-Rechte auf einem UNIX- (oder Linux-)System zu erlangen und damit die volle Kontrolle über das System auszuüben. Dies kann zum einen über die direkte Rechte-Steigerung erreicht werden: d.h. durch Ausnutzen von Schwachstellen im System um höhere Privilegien zu erhalten. Andererseits kann es auch durch Virusaktivitäten erfolgen, wie durch Cohen [6] in den frühen Tagen der Schadsoftwareforschung berichtet wurde. Der Begriff "Rooting" klingt im Kontext eines Windows-PC, wo der Nutzername "root" keine besondere Bedeutung hat, etwas abwegig. Er kann aber trotzdem als Oberbegriff für das Erlangen eines privilegierten Zugangs sowohl auf UNIX/Linux- als auch auf nicht-UNIX-Systemen dienen.

Zielsetzungen von Rootkits

Das Hauptziel eines Rootkits ist nicht notwendigerweise, auf dem Gastsystem "root" zu werden, d.h. in den Rechner einzudringen, obwohl es durchaus Programme enthalten kann, die dabei helfen, einen Administratorzugang zu erhalten. Vielmehr könnte man sagen, dass es einem Eindringling erlaubt, ein unerkanntes Standbein im System einzurichten und auszunutzen.

Einige Quellen unterscheiden allerdings zwischen "rootkits" und "stealthkits". Durch diese Unterscheidung könnte ein Rootkit doch eher als ein Werkzeugkasten definiert werden, der Hilfsmittel zum "rooting" eines Systems enthält. Zumindest haben manche Beiträge [7] so argumentiert. Dieser Artikel befasst sich stattdessen mit den verschiedenen Arten von Rootkits und ihren Bestandteilen, als der eher philosophischen Frage anzuhängen, was denn nun ein Rootkit ist und was nicht. Diese Herangehensweise ist nicht nur der besseren Übersicht geschuldet, sondern auch der Vielzahl an Bedrohungen, die in der Realität beobachtet werden.

Zweitrangige Ziele eines Rootkits können sein:

- die Spuren eines Eindringlings auf dem kompromittierten System zu verbergen
- die Anwesenheit von Schadprogrammen oder -prozessen zu verschleiern
- die Aktivitäten von Binärdateien zu kaschieren, die sich als seriöse Dateien ausgeben
- die Existenz von Exploits zu verstecken - rückgängig gemachte Patches/Rückkehr zu älteren Versionen/ Hintertüren
- Informationen zu sammeln, auf die der Eindringling anderenfalls keinen Zugriff bzw. keinen Vollzugriff hätte. Dies kann Daten des kompromittierten Systems, aber auch Netzwerkverkehr usw. einschließen.
- das kompromittierte System als Zwischenstation für weitere Einbruchsversuche und/oder bösartige Attacken zu nutzen
- andere Schadapplikationen zu lagern und als Server für Bot-Updates zu dienen

Bestandteile traditioneller UNIX-Rootkits

Als Trojaner benutzte Hilfsprogramme ersetzen die gleichnamigen, rechtmäßigen Systemprogramme, um die Anwesenheit eines Eindringlings oder seine Spuren zu verbergen, um ihm Zugang ganz nach Wunsch zu erlauben oder um Informationen zu sammeln, für die er keine Berechtigung besitzt. In einer UNIX-Umgebung sind Dienstprogramme wie "top", "ps", "login" oder "passwd" beliebte Ziele für solchen Austausch [8], aber im Prinzip ist jedes Programm, mit dem sich eine "root shell" (siehe auch "shell" im Glossar) erzeugen lässt, ein natürliches Ziel für ein Rootkit, da sich der Eindringling mit seiner Hilfe privilegierten Zugang verschaffen oder diesen beibehalten kann. Diese Programme sind nicht unbedingt das Mittel, um sich Zugang zu dem kompromittierten System zu verschaffen, sie haben aber eine Reihe weiterer Anwendungsmöglichkeiten wie die Kompromittierung anderer Konten und Systeme oder die Sicherung erhöhter Benutzerrechte gegen Versuche, einen erkannten Einbruch in das System zu reparieren.

Programme, welche die Erkennung eines Eindringlings durch den Administrator verhindern z.B. "last", "ls", "netstat" und "ifconfig") [9] sind ebenso Ziel für einen Austausch. Dämonen (Programme wie "inetd", "rshd" und "syslogd", die im Hintergrund als Server- oder Systemprozesse (siehe Glossar) laufen und nicht direkt vom Anwender gestartet werden) bilden ebenfalls Angriffsziele, sowohl zum Zweck der Verschleierung als auch zum Sammeln von Informationen.

Der Ausdruck „Windows rootkit" wird allgemein für Programme genutzt, die Prozesse, Dateien oder Registry-Einträge vor dem Betriebssystem verbergen.

Spezialisierte Hilfsprogramme verwischen die Spuren eines Eindringlings, indem sie zum Beispiel Einträge in den Protokolldateien des Systems, die mit dem Eindringen in Verbindung gebracht werden können, aus den Protokollen löschen.

In klassischen Rootkits findet man außerdem andere Formen von Trojanern, z.B. solche zum Sammeln von Informationen (wie keylogger oder packet sniffer) und zur Sicherung des zukünftigen freien Zugangs (backdoors).

Ein typisches Rootkit für UNIX oder Linux enthält üblicherweise ausgetauschte Dienstprogramme wie Port- oder Shell-Dämonen, Programme zur Erweiterung der Nutzerrechte, Dienstprogramme zur Überwachung der Resourcenauslastung (um Dateien zu verbergen), Paket sniffer zur Überwachung des Netzwerkverkehrs sowie weitere veränderte Programme um Prozesse, Protokolle und Verbindungen zu verstecken.

Rootkits unter Windows

Der Ausdruck "Windows rootkit" wird ganz allgemein für Programme verwendet, die Prozesse, Dateien oder Registry-Einträge vor dem Betriebssystem verbergen. Tatsächlich können Rootkits unter Windows ganz ähnliche Funktionalitäten aufweisen wie ihre Pendants unter UNIX, wobei die exakte Wirkungsweise von Plattform zu Plattform unterschiedlich ist. Windows-Versionen wie XP, die direkte Abkömmlinge von Windows NT sind, besitzen ein Sicherheitsmodell mit mehreren Nutzerkonten, das dem von älteren Systemen wie VMS oder UNIX sehr ähnlich ist, obwohl die Bezeichnungen und die Technik sich im Einzelnen stark unterscheiden. Ältere Windows-Versionen sind, was die Sicherheit angeht, in hohem Maße auf Programme von Drittanbietern angewiesen, so dass ausgewachsene Rootkits auf diesen Systemen sehr wenig Sicherheitssysteme vorfinden, die es zu überwinden gilt.

Die von Windows NT abstammenden Windows-Umgebungen haben wie UNIX-Systeme eine abgestufte Rechtevergabe. Dies bezieht sich nicht nur auf den Zugriff der Anwender auf bestimmte Datenbereiche sondern auch auf Kernel-Prozesse. NT-artige Plattformen unterstützen zwei Ausführungsarten (oder Privilegstufen): Anwendermodus und Kernelmodus. (Moderne x86-Prozessoren unterstützen tatsächlich vier sogenannte Ringe, aber nur zwei davon werden von Windows benutzt, da NT so portabel entwickelt wurde, dass es auch auf nicht-Intel-Prozessoren laufen konnte.) Diese Ringe sind dafür vorgesehen, den Betriebssystemkern (Ring 0) zu schützen, so dass Systemdaten nicht von unprivilegierten Prozessen verändert oder überschrieben werden können. Normale Anwendungen laufen auf einem solchen System im Ring 3, der Ebene mit den geringsten Rechten.

Anwendermodus gegenüber Kernelmodus

Während sich die Unterscheidung von unprivilegierten Anwendern und Administratoren nicht ganz genau an den Differenzen zwischen Kernelmodus (Ring 0) und Anwendermodus festmachen lässt, so gibt es doch einen engen Zusammenhang. Der Kernel kann buchstäblich als der Kern des Betriebsystems beschrieben werden [10]: Systemdienste laufen im Kernelmodus, so dass unprivilegierte Nutzer keine unsachgemäßen Veränderungen wie das Entfernen oder Hinzufügen von Treibern, Geräten oder Programmen ohne Autorisierung vornehmen können. Anwenderprogramme dagegen, die im Allgemeinen allen Nutzern zugänglich sind, laufen im Anwendermodus, wo die Möglichkeit durch unangebrachte oder unbeabsichtigte Veränderungen Schäden an Systemprozessen zu verursachen, stark eingeschränkt ist.

Die Bestandteile eines Rootkits für den Anwendermodus laufen als oder innerhalb eines Anwenderprogramms, indem sie Aufrufe der Windows-API (Application Programming Interface) in jedem Nutzerprozess verbiegen. Da jede Anwendung in ihrem eigenen Speicherbereich läuft, muss ein derartiges Rootkit den Speicher jeder laufenden Applikation verändern, um die "Sicht" dieser Applikationen auf das, was im Betriebssystem vor sich geht, zu filtern. Dazu muss das Rootkit das System ständig überwachen und den Speicher eines gerade geöffneten Programms verändern, bevor es vollständig gestartet ist. Eine

Die Bestandteile eines User-Mode-Rootkits laufen als oder innerhalb eines Anwenderprogramms, indem sie Aufrufe der Windows-API in jedem Nutzerprozess verbiegen.

häufig verwendete Methode diese Ziele zu erreichen, ist die Modifikation von System-DLLs (Dynamic Link Libraries) zur Laufzeit.

Der Kernelmodus erlaubt den privilegierten Zugang zum Speicherbereich des Systems und zum vollen Befehlssatz der CPU und ein Rootkit für den Kernelmodus fängt die Aufrufe des Kernel-API ab. Prozesse, Dateien und Registryschlüssel usw. werden auf diese Art einfacher versteckt. Wenn ein Anwenderprogramm eine bestimmte Information anfordert, wird diese gefiltert, um Beweise für die Anwesenheit eines Eindringlings zu unterdrücken. Ein Rootkit im Kernelmodus hat, verglichen mit einem Rootkit im Anwendermodus, nahezu unbegrenzte Möglichkeiten, Schaden anzurichten oder Veränderungen am System vorzunehmen, ist aber wegen seiner inhärenten Komplexität schwieriger zu installieren und zu pflegen. Das Einnisten in den Systemspeicher macht die Arbeit für das Rootkit leichter und zuverlässiger, da alle Systemprozesse sich den gleichen Adressbereich teilen müssen, setzt aber voraus, das das Rootkit von einem privilegierten Anwender gestartet wird (nicht notwendigerweise mit seinem oder ihrem Wissen).

> Ein Kernel-Mode-Rootkit hat nahezu unbegrenzte Möglichkeiten, Schaden anzurichten oder Veränderungen am System vorzunehmen.

Man beachte, dass „Hooking" (siehe Glossar) nicht die einzige Methode ist, mit der ein Rootkit ein Objekt verstecken kann. Anstatt zum Verbergen von Prozessen und Treibern die Informationen zu filtern, die der Kernel zurückliefert, werden mit Direct Kernel Object Manipulation (DKOM) die vom Betriebssystem zur Überwachung angelegten Objekte selbst verändert. Ein Rootkit, welches diese Verschleierungstechnik verwendet, kann einen Prozess einfach dadurch verstecken, dass die Verbindung zwischen einem Prozess und dem zugehörigen Kernel-Objekt aufgetrennt wird [11], was die Erkennung für Sicherheits- und andere Software ungleich schwieriger macht.

Dauerhaft kontra flüchtig

Viele Rootkits sind dauerhaft (auch persistent genannt): d.h. sie sind auf der Festplatte gespeichert und hängen sich in die Boot-Reihenfolge ein, so dass sie den Neustart des Systems überleben. Nicht dauerhafte, auch Speicher-Rootkits genannt, tun das nicht: sie installieren ihren Code in den flüchtigen Speicher und sind nach einem Neustart verschwunden. Das macht ihren Nachweis schwieriger, da es keinen Fußabdruck im System gibt, der von einem Virenscanner auf dem frisch gebooteten System gefunden werden könnte. Allerdings schränkt das ihre Möglichkeiten auf die Zeit ein, die das infizierte System eingeschaltet bleibt (da die Infektion nach dem Neustart verschwunden ist): dies stellt auf Systemen, die normalerweise nicht ausgeschaltet oder neu gestartet werden (beispielsweise Server) ein geringeres Problem dar. Gegenwärtig scheint sich, besonders unter den Forschern im Bereich der Computersicherheit und bei denjenigen, die Code für Machbarkeitsstudien (PoC - Proof of Concept) entwickeln, ein Trend zu nicht dauerhaften Rootkits abzuzeichnen, möglicherweise hervorgerufen durch die Vorteile beim Verstecken von Malware, die keine permanenten Änderungen am Dateisystem vornimmt. Einige Trojaner haben bereits solche nicht-dauerhaften Rootkits (z.B. das FU Rootkit) benutzt, die sie in den Speicher laden, sobald sie selbst gestartet sind. Das Rootkit wird dann dazu verwendet, die Dateien und Prozesse des Trojaners zu verbergen.

> Nicht-persistente Rootkits installieren ihren Code direkt in den Hauptspeicher und überleben keinen Neustart.

Rootkits für den Mac

Für das Betriebssystem OS X von Apple existieren einige Rootkits als Nachweis der Machbarkeit (PoC). Ihre Wirkung ist aber gering und ihre Bedeutung wird von der Mac-Nutzergemeinschaft heruntergespielt, da sie Root-Rechte zur Installation brauchen - was bedeutet, dass mit ihnen keine Rechte-Erweiterung möglich ist. Die selbe Haltung findet man gegenüber den wenigen gegenwärtig bekannten Beispielen von Schadsoftware für den Mac - tatsächlich bestreiten Mac-Nutzer oft sogar die Existenz von Viren für OS X, da die vorhandenen Exemplare für Infektion und Vermehrung die aktive Mithilfe des Nutzers benötigen. (Allerdings kann man das Gleiche auch von vielen Schädlingen für Windows sagen.) Für einen Nutzer auf dem Mac ist es schwieriger, privilegierte Rechte zu bekommen (es sei denn, er verlangt es ausdrücklich): es ist aber naiv zu glauben, dass es niemals gelingen würde, einen Mac-Anwender zu täuschen und dazu zu bringen, temporär als Root zu arbeiten und damit der Installation eines Rootkits Vorschub zu leisten.

Während Mac-Anwender viel auf das "überlegene" Rechtemanagement des Mac halten, ist es auch bei Windows-Rootkits gar nicht so verbreitet, dass sie "Root" werden können - besser gesagt, dass sie selbst ihre Rechte bis auf Administratorniveau ausweiten können - zumindest nicht ohne die Hilfe des psychologisch manipulierten Opfers. Allerdings ist Windows wohl deutlich nachsichtiger, wenn es darum geht, den Anwender standardmäßig mit Administratorrechten arbeiten zu lassen.

Gute Absichten oder das Erlebnis mit dem Sony-Rootkit

Der Einsatz von Rootkit- oder Stealth-Techniken ist nicht auf diejenigen beschränkt, die mit den üblichen "Hacker"-Absichten in Systeme einbrechen, sondern sie können auch durch andere Arten von Schadsoftware, einschließlich Viren, Würmer und Trojaner zur Verschleierung ihrer Anwesenheit verwendet werden.

Sogenannte "Greyware" (Programme, die irgendwo zwischen seriöser Software und ausgemachter Schadsoftware liegen), wie beispielsweise Adware, bestimmte Spyware, Trackware usw. werden von einigen Herstellern von Antiviren-Lösungen ganz vorsichtig als "potentiell unerwünschte Programme" bezeichnet. Dies geschieht, um rechtliche Komplikationen in den Fällen zu vermeiden, wo (vom Hersteller) behauptet wird, dass die Software seriös sei. In einigen Fällen kann es sich sogar um rechtmäßige Software handeln, die aber für Schadzwecke missbraucht werden kann oder schon wurde. Normalerweise hängt das auch vom Grad der Verschleierung ab. Während die Anwesenheit solcher Programme manchmal ganz offensichtlich ist, da sie Popup-Fenster öffnen, den Anwender auf unerwartete URLs weiterleiten oder ähnliches, wird aber ein erhebliches Augenmerk darauf gelegt, die Erkennung oder Entfernung der jeweiligen Programmdateien zu verhindern.

> „Die meisten Leute wissen überhaupt nicht, was ein Rootkit ist, weshalb sollten sie sich also darum kümmern?"
> Thomas Hesse, President, Global Digital Business, Sony BMG.

Einige Kommentatoren auf diesem Gebiet [4] legen Wert auf die Feststellung, dass Rootkits (oder Rootkit-artige Techniken) nicht nur für rechtmäßige Zwecke benutzt werden können sondern sogar müssen, um Unzulänglichkeiten der Betriebssysteme in Bezug auf das Verstecken von Daten auszugleichen. Einige der angeführte Bereiche sind:

- Schutz von Urheberrechten
- Schutz von Programmen vor Reverse Engineering

- Aufdecken von Bedrohungen von Innen
- Verfolgen von Eindringlingen
- Mitarbeiterüberwachung
- Digitales Rechtemanagement
- Schutz der Sicherheitsprogramme vor Schadsoftware oder unerwünschter Nutzereingriffe
- Software zur Datensicherung
- Software zur Systemwiederherstellung
- Verschlüsselung und Verstecken von Daten auf Mehrbenutzer-Systemen

Nicht jeder fühlt sich dabei wohl, eine Rootkit-bezogene Terminologie auf derartige Problemstellungen anzuwenden, da es etwas Verwirrung in die Definitionen bringt. Schließlich benutzt selbst Windows standardmäßig solche Techniken, um wichtige Systemdateien zu schützen. Einige Anbieter sind jedoch bis jetzt mit den Begriffen und Konzepten ganz zufrieden.

Im Oktober 2005 berichtete Mark Russinovich im Blog [12] seiner Firma Sysinternals darüber, dass er scheinbar ein Rootkit auf seinem System entdeckt hätte. Es stellte sich heraus, dass es sich dabei um das berüchtigte Sony-"Rootkit" handelte, was in einem folgenschweren Durcheinander bei der öffentlichen Meinung über das Verhältnis von Digitalem Rechtemenagement (DRM) und Stealth-/Rootkit-Technologien mündete. Der Begriff DRM bezeichnet eine Technologie, die den Zugang zu Daten und Hardware in einer Weise einschränkt, dass die Rechte der Verleger bzw. Urheber gewahrt werden.

Sony benutzte die XCP(Extended Copy Protection)-Technologie der Firma First 4 Internet Ltd. um den Zugang zu einigen Musik-CDs zu überwachen. Mit XCP geschützte Medien beschränkten die Anzahl an Kopien, die von einer CD oder DVD hergestellt werden konnten und kontrollierten das "Ripping" der Musik in eine digitale Form, die für das Speichern und die Wiedergabe auf einem Computer oder einem tragbaren Abspielgerät, wie z.B. einem MPEG-Player, geeignet ist. Es war nicht möglich, diese CD auf einem PC abzuspielen, ohne zuerst die Software zu installieren, die in der Folge Dateien, Prozesse und Registry-Schlüssel/Werte versteckte, indem sie sich in die Ausführung von Systemfunktionen einhängte. Dabei kam die verbreitete Rootkit-Technik des Patchens der System Service Table (SST) zum Einsatz.

> Es ist nicht unwahrscheinlich, dass es Versuche geben wird, das Rootkit-Problem mit rechtlichen Mitteln zu lösen.

Das Recht der Firma Sony, die unerlaubte Verbreitung ihres Produkts zu verhindern, wird im Allgemeinen gar nicht in Frage gestellt und es scheint auch nicht angebracht, ihre Herangehensweise als Rootkit im negativen Sinn zu bezeichnen. Andererseits fühlten sich viele Kunden unbehaglich bei dem Gedanken, dass ihr System durch Sony modifiziert wurde, ohne klar zu stellen, was eigentlich getan wurde oder eine Möglichkeit zur Deinstallation vorzusehen. Diese hätte die Anwender sogar in Gefahr gebracht, Gesetze zu brechen, die unerlaubte Zugriffe oder Modifikationen von Programmen verbieten. Schlimmer noch, diese Lösung war insgesamt weder besonders durchdacht noch programmiert und öffnete ein Einfallstor, das von der kriminellen Szene mit Dankbarkeit angenommen wurde. Tatsächlich war das eigentliche Problem, dass das Rootkit benutzt wurde, um einen Trojaner zu verstecken, der weder mit Sony noch mit der Kopierschutz-Software irgendetwas zu tun hatte. Dies demonstriert den feinen Unterschied zwischen bösartiger Absicht und ungewollter Verwundbarkeit: sicherlich haben Sony und First 4 Internet weder beabsichtigt noch erwartet, dass ihre DRM-Maßnahme eine Lücke aufreißen würde, die von Schadprogrammen ausgenutzt werden kann. Außerdem wurde Sony nur schlecht mit der nachfolgenden Aufmerksamkeit fertig und ihr erster Versuch, das Problem zu beheben, war mangelhaft. Die De-Installationsroutine wurde erst nach dem Ausfüllen eines Webformulars zur Verfügung gestellt und ließ nur die Wahl, entweder

die Software zu deinstallieren und damit keine XCP-geschützten CDs mehr hören zu können oder einfach die Verschleierungsfunktion auszuschalten. Berichten zufolge war der erste Patch aufgebläht und schlecht getestet und besaß obendrein die Eigenschaft, nach Hause zu telefonieren, was in den EULA (End User License Agreement) nicht ausdrücklich erwähnt wurde [13].

Die XCP-Technologie hat die Rootkit-Gemeinde - damit meine ich diejenigen, die ein tiefergehendes technisches Interesse an der (nicht notwendigerweise ungesetzlichen) Anwendung von Stealth- und Rootkit-Techniken haben - nicht sonderlich beeindruckt, allerdings hat sie die von ihr selbst geschaffenen Schwachstellen ins Blickfeld mancher Autoren von Schadsoftware gerückt. Dennoch gibt es eine Reihe von Auswirkungen auf die Industrie im Allgemeinen.

Es ist nicht unwahrscheinlich, dass es Versuche geben wird, das Rootkit-Problem mit rechtlichen Mitteln zu lösen [10]. Das kann in Form spezieller Gesetze gegen bösartige Typen von Rootkits geschehen, andererseits könnte es auch auf Bestrebungen hinauslaufen, Rootkit-artige Techniken selbst zu rechtmäßigen Zwecken für ungesetzlich zu erklären. Wenn das passiert, könnte jede beliebige Anwendung, die zum Schutz ihres eigenen oder fremden Codes bzw. von Daten irgendeine Form von Stealth benutzt, Gefahr laufen, juristische Probleme zu bekommen. Dies hätte ernsthafte Auswirkungen für das Digitale Rechtemanagement (DRM), das generell auf ein bestimmtes Maß an Verschleierung und Zugriffsschutz angewiesen ist.

Selbst wenn solche Maßnahmen nicht ergriffen werden, gibt es doch bei manchen Produkten Akzeptanzprobleme durch potentielle Nutzer. Man betrachte das Beispiel des geschützten Papierkorbs von Symantec (ein sicherer Papierkorb, der die volle Wiederherstellung selbst solcher gelöschter Dateien ermöglicht, die über den Windows-Papierkorb nicht wiederherstellbar sind) - ein rechtmäßiges, dokumentiertes und potentiell nützliches Werkzeug. Trotzdem wurden, nachdem Schlagzeilen über das "Symantec rootkit" auftauchten, die versteckten Programmteile wieder sichtbar gemacht [14], um das Risiko des Ausnutzens durch Schadsoftware zu verringern. Dies hat auch Folgen für andere Sicherheitsprogramme, von denen einige Funktionszeiger des Systems verbiegen -müssen- um als erste an der Reihe zu sein (beispielsweise nutzt Software zur Überwachung des Programmverhaltens und zum Blockieren schädlicher Aufrufe ähnliche API-Hooking-Techniken wie Rootkits, allerdings zum Zweck der Erkennung und nicht zum Verstecken). Weitere versteckte Funktionen werden von Sicherheits- und anderer Software verwendet - zum Beispiel, um das Reverse Engineering zu verhindern, um den Anwender davon abzuhalten, sich selbst mit bösartigem Code zu gefährden (Virencode in Quarantäne usw.) oder an Systemeinstellungen "herumzupfuschen" und ähnliches.

Methodik zum Erkennen von Rookits

Hin und wieder "entdeckt" jemand, dass die sogenannte signaturbasierte Erkennung von Schadprogrammen fehlerhaft ist, da sie "keine neuen Viren aufspüren kann". Glücklicherweise beruht Antiviren-Software schon seit vielen Jahren nicht mehr ausschließlich auf der Erkennung bekannter Viren. Es wird bereits eine Vielzahl von ergänzenden Technologien (heuristische Analyse, generische Treiber, Überwachung des Programmverhaltens usw.) verwendet, um die Erkennung von neuen Bedrohungen und Varianten zu verbessern. Wie bei den Stealth-Viren ist die Rootkit-Problematik eine Frage von "Wer kommt zuerst?". Für einen Scanner ist es schwierig, etwas zu erkennen, das bereits installiert ist und die Beweise für eine Infektion versteckt. Die Hersteller von Antiviren-Software haben aber langjährige Erfahrung darin, Möglichkeiten zur Umgehung auch von neuartigen Verschleierungstechniken zu finden, sobald das Schadprogramm analysiert wurde. Üblicherweise können Rootkits schon bei der Überprüfung von Dateisystem und Hauptspeicher mittels des sogenannten Signatur-Scannens entdeckt werden.

> Das Wachstum von nicht-persistenten Rootkits verstärkt die Notwendigkeit, den Speicher zu scannen und versteckte Prozesse aufzuspüren.

Die typische Anwendung ergänzender Techniken um mögliche Rootkit-Aktivitäten heuristisch zu erkennen, beruht hauptsächlich darauf, dass Unterschiede zwischen einer zuverlässigen (mit einigermaßen

Genauigkeit) Sicht auf das System und der verfälschten Sicht, wie sie gefiltert durch die Rootkit-Technik präsentiert wird, gefunden werden [15].

Früher waren UNIX und UNIX-artige Betriebssysteme mit einer Herangehensweise gut bedient, die man auch als Stolperdraht oder Objektabgleich bezeichnen könnte [9], obwohl sie in der etablierten Antivirus-Industrie am häufigsten Integritätsprüfung genannt wird. In einer Windows-Umgebung kann sie immer noch nützlich sein, führt aber häufig zu einem hohen Wartungsaufwand, da es viele Fälle gibt, wo im laufenden Betrieb Änderungen an der Umgebung (Programme, Registry-Einstellungen, Konfigurationsdateien) vorgenommen werden. Die Zunahme an nicht-persistenten Rootkits verstärkt die Notwendigkeit, den Hauptspeicher zu scannen und versteckte Prozesse aufzuspüren, statt sich auf Änderungen am Dateisystem als Hinweis auf eine Infektion zu verlassen. Ein eher proaktiver Ansatz zur Erkennung von Varianten wäre ebenso wünschenswert. Je intelligenter ein Produkt ist, wenn es darauf ankommt, zwischen möglicherweise schädlichen Funktionszeigern, Registry-Einstellungen u.ä. und ihren rechtmäßigen Gegenstücken zu unterscheiden, desto effektiver kann so eine heuristische Methode sein [10].

Antiviren-Lösungen sind in den letzten Jahren bei der heuristischen Entfernung von Viren ziemlich geschickt geworden [16]. Trotzdem verbleiben bei der heuristischen Beseitigung noch Schwierigkeiten, dort wo die exakte Identifizierung des Virus nicht gelingt und noch mehr in Bezug auf Malware ohne Verbreitungsroutinen. Während Behauptungen, dass eine Infektion mit einem Rootkit nur durch eine komplette Neuinstallation des Systems beseitigt werden könne, übertrieben sind, kann es selbst bei bekannter aber tief eingebetteter Malware effektiver sein, das Image eines Systems neu aufzuspielen als das Rootkit "einfach" zu entfernen, besonders aber in den Fällen, bei denen auch andere Programme gepatcht oder verändert wurden. So etwas ohne übermäßige Schäden an Daten oder Produktivität zu bewältigen, erfordert das genaue Einhalten einer guten Sicherungs- und Wiederherstellungsstrategie, sowohl der Daten als auch von System- und Anwendungsprogrammen.

Vorbeugende Maßnahmen

Es gibt Gegenmaßnahmen, die auf jeder Plattform Sinn machen. Das Anlegen von Sicherungskopien ist ein unerlässlicher Schutz vor Bedrohungen und ungeplanten Katastrophen, es sollte der Grundstein für jede Richtlinie zur Datensicherheit sein.

Administratoren sollten nicht mit einem Root-/Administratorkonto arbeiten, es sei denn, ohne privilegierten Zugang ist die Aufgabe nicht zu bewerkstelligen: für Routineaufgaben sollte man, wenn möglich, ein unprivilegiertes Konto benutzen. Alle verbreiteten modernen Plattformen erlauben es, ohne Neustart zwischen verschiedenen Konten umzuschalten, wenn der privilegierte Zugang benötigt wird.

> Regelmäßige Backups sind ein unerlässlicher Schutz vor Bedrohungen und ungeplanten Katastrophen.

Es kann gefährlich sein, sich auf quelloffene oder auf freiwilliger Basis produzierte Sicherheits-Software zu verlassen. Viele Gemeinschaftsprojekte haben exzellente Ergebnisse hervorgebracht, aber es ist mitunter schwierig, die Kompetenz (und manchmal auch die guten Absichten) von jedem an so einem Projekt beschäftigten einzuschätzen. Es ist sicher unangebracht, dass eine kommerzielle Organisation oder die öffentliche Hand ihre Sicherheit einem Produkt anvertraut, für das weder Garantien noch eine Produkthaftung erkennbar sind. Heimanwender und Klein-/Heimbüros sind von Problemen der Unternehmensführung sicher weniger beunruhigt, wollen aber auch nicht, dass sie von Software, die ihren Zweck nicht wie erwartet erfüllt, in der Patsche sitzen gelassen werden. Wichtig ist auch, Tests auf Verwundbarkeit durchzuführen, eine geordnete Patchverwaltung zu organisieren und sich vor Tricks durch soziale Manipulation (sogenanntes social engineering) zu hüten.

Schlussfolgerung

Keine Panik, die Welt geht nicht unter, oder etwa doch?

Von Joanna Rutkowskas experimenteller "Blue Pill"-Technik wird behauptet, dass sie die Erzeugung von "100% unentdeckbarer Malware, die nicht auf Verheimlichung des Konzepts beruht" erlaubt, und zwar durch Ausnutzen der Virtualisierungstechnik SVP/Pacifica von AMD. Bis jetzt wurden nur ungenügende Details zu ihrem Ansatz veröffentlicht, so dass die Behauptungen noch nicht bestätigt werden konnten: sie scheint auf einem nicht-persistenten Rootkit zu beruhen, das innerhalb einer virtuellen Machine läuft.

Das SubVirt Rootkit [18] benutzt ebenfalls Virtualisierung, ist aber persistent (d.h. überlebt einen Neustart). Es ist eine interessante Machbarkeitsstudie, aber bei weitem nicht unentdeckbar. Es wird zu gegebener Zeit interessant sein, festzustellen, ob "Blue Pill" in dieser Beziehung wirklich überlegen ist.

Nichtsdestotrotz kann man sicher davon ausgehen, dass das Wettrüsten bei Schaden verursachender Software, wo der Vorteil regelmäßig von den Bösen zu den Guten und wieder zurück wechselt, noch eine ganze Weile andauern wird. Dabei sind weder Panik noch Selbstzufriedenheit angebracht, sondern Wachsamkeit. Viele Hersteller von aktueller Malware verwenden jetzt Rootkit-Techniken, um ihre Schöpfungen zu verstecken, aber die Verbreitung solcher Objekte ist immer noch ziemlich gering und die grundlegende Beschaffenheit von Schadsoftware führt dazu, dass sie leichter an ihrer Wirkung erkannt wird.

Berichte vom Tod der Antivirus-Industrie sind stark übertrieben; tatsächlich ist der Ausdruck "Antivirus" etwas irreführend, wenn man über heutige Sicherheitslösungen spricht. Die Tage von Antivirus-Programmen, die nur Viren entdeckten und alles andere ignorierten, sind lange vorbei. Viele Produkte können jetzt eine sehr breite Palette an modernen Bedrohungen aufdecken. Genauso wie sich die Gefährdungen entwickelt haben, hat auch die Antivirus-Industrie nicht still gestanden und einige AV-Hersteller haben bedeutende Erfolge bei der Erkennung von Rootkits zu verzeichnen. Trotzdem sollte man nicht die Augen verschließen und erwarten, dass sich schon jemand um alles kümmern werde. Reine "signatur-basierte" Lösungen können nicht vor neuen Gefahren schützen, die sich wesentlich von bereits bekannten unterscheiden, auch nicht, wenn sie regelmäßig aktualisiert werden. Endanwender müssen Produkte benutzen, die eine nachgewiesene Erfolgsgeschichte haben und dabei fortgeschrittene Heuristik und andere generische, proaktive Erkennungsmethoden verwenden und die umsichtig bei der Auswahl ihrer Abwehrtechnologien und der Einsatzmöglichkeiten sind.

Literaturnachweise

1. University of Minnesota ResNet FAQ:
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
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
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-dangerous-decloaking.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

Glossar

API Hooking	Im Zusammenhang mit Rootkits beschreibt Hooking folgenden Vorgang: Wenn bestimmte System-Informationen abgefragt werden sollen, dann wird eine API-Funktion aufgerufen, die diese Anfrage an den Kernel weiterreicht und die angeforderte Information über den gleichen „Ausführungspfad" an die anfragende Anwendung zurückgibt. Hooking bezeichnet die Methode, Daten zu verbergen, indem der Ausführungspfad über einen Filter umgeleitet wird, der die zurückgelieferten Daten modifiziert.
AV	Antivirus
Backdoor-Ali (auch als ierk oder slanret bekannt)	Ein Trojaner mit Hintertür, der oft als Rootkit bezeichnet wird.
Daemon oder demon (Dämon)	Ein Server- oder Systemprozess unter UNIX, der im Hintergrund läuft, im Gegensatz zu Programmen die vom Anwender gestartet werden. Das Programm ftpd beispielsweise ist ein Dämon, ein Systemdienst, der ständig läuft: das Programm ftp ist dagegen ein Client-Programm, das vom Endanwender aufgerufen wird und den Systemdienst ftpd benutzt.
Deepdoor	Rootkit als Machbarkeitsstudie von Joanne Rutkowska
Generischer Treiber oder generische Erkennung	In der Antivirus-Technik ein bequemer, aber nicht allgemein verwendeter Ausdruck zur Beschreibung einer Virus-Signatur bzw. Definition, die so weit verallgemeinert wurde, dass sie in der Lage ist, statt einer einzelnen Virusvariante eine ganze Virenfamilie oder mehrere Varianten zu erkennen. Das hört sich ähnlich an, ist aber nicht genau das gleiche wie die Unterscheidung zwischen genauer und fast genauer Erkennung, da die fast genaue Erkennung für gewöhnlich mit generischer Desinfektion verbunden wird. Eine weitergehende Diskussion dieser Begriffe würde den Rahmen dieser Arbeit sprengen. Eine gute Quelle für weitere Informationen dazu ist der Artikel „The Art of Computer Virus Research and Defense" von Peter Szor.
Governance (Information Governance)	Unternehmensführung im IT-Bereich - Rahmenbedingungen für den Umgang mit Informationen, besonders im Hinblick auf Richtlinien und die Einhaltung von Vorschriften
Grayware, Greyware	Eine etwas unscharf definierte Software-Gruppe, die Adware, Spyware, Spaßprogramme, Programme zum Fernzugriff usw. umfassen kann. Kann auch die Vermutung von versteckten Funktionen einschließen.
Hacker Defender	Ein weit verbreitetes Rootkit.
Hacktool	Ein Rootkit für Windows.
Heuristics	Ein Pauschalbegriff, der auf eine Reihe von Techniken zur Erkennung gegenwärtig unbekannter Malware oder Varianten angewandt wird.
Keylogger	Eine Software, die Tastaturanschläge aufzeichnet, oft (aber nicht notwendigerweise) für finstere Zwecke (z.B. Stehlen von Passwörtern).
Lrk	Ein bekanntes und gängiges Rootkit für Linux. Eine Beschreibung findet sich bei http://staff.washington.edu/dittrich/misc/faqs/lrk4.faq
Opener (auch Renepo)	Eine Malware für den Macintosh, die manchmal als Rootkit bezeichnet wird.

OS X	Das aktuelle Betriebssystem für den Macintosh, das auf einem BSD UNIX basiert, mit einer Anwenderschnittstelle, die einerseits das anwenderfreundliche Aussehen des Mac beibehält, zum anderen aber auch den Zugang zur herkömmlichen Kommandozeile von UNIX bietet.
OSXrk	Ein Rootkit für den Macintosh mit OS X
Shadow Walker	Eine Rootkit-Studie von James Butler und Sherri Sparks, teilweise basierend auf dem FU rootkit von Butler.
Shell	Der Kommandoprozessor, der die am Terminal eingetippten Kommandos interpretiert und in die Aufrufe des Betriebssystems umsetzt. Der Ausdruck wird jedoch auch benutzt, um einen Prozess oder eine Reihe von Prozessen zu beschreiben, die vom Kommandoprozessor gestartet werden. Diesen Vorgang kann man auch als „eine Shell starten" oder „eine Shell öffnen" bezeichnen. Einen Prozess, der mit Root-Rechten arbeiten kann, nennt man auch Root-Shell. Dazu ist es nicht immer notwendig, dass der Nutzer selbst Root-Rechte besitzt: manche Programme können als Root laufen ohne die Rechte des Nutzers zu erweitern. Solche Mechanismen können aber dort, wo Schwachstellen wie Pufferüberläufe existieren, von einem bösartigen Eindringling oder Programm missbraucht werden, um „Root" zu werden. (Während die hier benutzte Terminologie relativ UNIX-spezifisch ist, existieren analoge Verfahren in anderen Betriebssystemen.)
Signature scanning	Genau genommen, die Suche nach einem Virus durch Vergleich mit einer mehr oder weniger statischen Bytefolge. Tatsächlich benutzen selbst einfache Virenscanner heutzutage komplexere und effizientere Techniken, einschließlich algorithmischer Herangehensweisen, Platzhalter usw. Der Ausdruck Signatur ist wegen dieser Mehrdeutigkeit in der Antiviren-Forschung verpönt, obwohl es möglicherweise viel zu spät ist, den Gebrauch oder auch Missbrauch dieses Ausdrucks in der Öffentlichkeit und den Medien auszurotten. Er wird jedoch routinemäßig bei der Erkennung von Eindringlingen verwendet.
SOHO	"Small Office, Home Office": Das Klein- und Heimbüro
Stealth	Adjektiv: wird bei Sicherheitsthemen als Alternative zu „stealthy" benutzt, analog zu „stealth aircraft" und anderen militärischen Ausdrücken, die Bezug zu Verschleierungsstrategien haben.
Stealthware	Eine Software (üblicherweise Malware), die Stealth-Techniken benutzt, um sich zu verstecken.
Trackware	Eine Software, die das Anwenderverhalten und Systeminformationen aufzeichnet. Manchmal (aber durchaus nicht immer) wird es zur Unterscheidung von Software benutzt, die mit dem Wissen des Anwenders aufzeichnet.
WeaponX	Ein kernel-basiertes Rootkit für OS X.

www.eset.de

Exklusiv-Distribution Deutschland

DATSEC® Data Security e. K.

Talstr. 84, 07743 Jena, Germany

Tel.: +49 (0) 3641 / 63 78 - 3 Fax: +49 (0) 3641 / 63 78 - 59

E-Mail: info@datsec.de Web: http://www.datsec.de

Übersetzung aus dem Englischen: Michael Dankert