Rootkits – Die Tarnkappen der Angreifer

Das eigene System macht eigentlich gerade nichts. Das sagen zumindest der Task Manager und der fast leere Desktop. Aber die Festplatte scheint nie zur Ruhe zu kommen und auch die Lichter der Netzwerkkarte blinken ohne Unterlass.

Bald treffen Beschwerden von Kollegen und Bekannten ein: Man soll doch bitte keine Mails mit Werbung verschicken. Ein Kollege meint, dass das System bestimmt mit einem Virus infiziert sei. Die Anti-Viren Software beharrt aber auch nach Aktualisierung der Signaturen darauf, dass das System frei von Viren ist.

Schließlich wird das System von einem Experten genauer untersucht und es stellt sich heraus, dass jemand über das Netzwerk eingebrochen ist und ein „Rootkit" installiert hat. Mit Hilfe des Rootkits wurden die Programme versteckt, die für das Verschicken der unerwünschten Werbe-Mail verantwortlich sind.

Was ist ein Rootkit?

Ein Angreifer bricht in ein System ein, indem er verschiedene Sicherheitslücken ausnutzt. Um nun auf dem System unentdeckt zu bleiben, werden spezielle Programme eingesetzt: sogenannte Rootkits. Ein Rootkit dient nicht der Ausnutzung einer Sicherheitslücke, durch die der Angreifer Zugriff zum System erlangen kann. Es ist auch nicht die Hintertür, die erneuten Zugriff über das Netzwerk für den Angreifer ermöglicht. Das Rootkit dient ausschließlich der Verschleierung der Aktivitäten des Angreifers auf dem System und soll in der Regel alle Dateien, Verzeichnisse, Prozesse und Netzwerk-Verbindungen des Angreifers unsichtbar machen.

Die Geschichte der Rootkits begann in den 80'er Jahren auf UNIX-Systemen [1]. Es wurden Logfiles verändert und die Ausgaben zum Status des Systems manipuliert, um die Accounts der Angreifer zu verheimlichen. Das Ziel war, auch später noch Zugriff als „root" zu haben, dem Administratoren-Account auf UNIX und Linux Systemen. Hieraus entstand der Name „Rootkit". Bald tauschten die Angreifer auch diverse Systemprogramme aus, um bestimmte Verbindungen, Dateien und Prozesse aus deren Ausgabe verschwinden zu lassen. Nachdem Rootkits sich auch auf Linux Systemen verbreitet hatten, kamen Mitte der 90'er Jahre erstmals neuartige Rootkits auf, die den Kern des Betriebssystems manipulierten. Da das Betriebssystem die Schnittstelle aller regulären Programme zur Hardware ist, kann ein solches Rootkit an einer zentralen Stelle jegliche Manipulation vornehmen und ist sehr schwer zu entdecken. Diese Entwicklung wurde bald von Linux wieder zurück getragen auf UNIX-Betriebssysteme wie Solaris und verschiedene BSD-Varianten. Ende der 90'er Jahre verbreiteten sich Rootkits schließlich auch auf Windows Systemen. Mittlerweile gibt es kaum ein Betriebssystem, auf dem es nicht irgendeine Version eines Rootkits gibt. Die bei weitem aktivste Entwicklung findet im Umfeld von Windows und Linux statt.

Wer installiert Rootkits?

Das Ausnutzen von Sicherheitslücken wurde in den letzten Jahren dadurch erleichtert, dass die dafür notwendigen Programme und Werkzeuge frei verfügbar und teilweise auch leicht zu bedienen sind. Die gleiche Entwicklung findet bei Rootkits statt. Vor einigen Jahren war noch erhebliches Expertenwissen notwendig, um ein Rootkit erfolgreich einsetzen zu können. Aktuelle Rootkits helfen bei der Installation, besitzen integrierte Hintertüren, die automatisch versteckt werden und sind insgesamt auch deutlich schwerer zu finden. Hat ein Angreifer Zugriff zum System erlangt, muss daher immer mit der Installation eines Rootkits gerechnet werden.

Andreas Bunten

DFN-CERT
Heidenkampsweg 41
20097 Hamburg

bunten@dfn-cert.de
http://www.dfn-cert.de/

Die in Rootkits entwickelten Techniken werden mittlerweile auch durch Würmer, Viren und Spyware eingeschleust, um Anti-Viren Software effektiver auszuweichen. Zuweilen kommen Rookits aus ganz unerwarteten Quellen: Manche Audio CDs und Video DVDs verwenden einen Kopierschutz, der versucht, Software zu installieren, sobald die CDs bzw. DVDs in ein Laufwerk eines Windows Systems eingelegt werden. Diese Software verhält sich wie ein Rootkit und dient leider auch „echter" Schadsoftware als Versteck [2][3].

Wie können Rootkits aufgespürt werden?

Ein aktuelles Rootkit, das den Kern des Betriebssystems manipuliert, kann den Angreifer sehr effektiv verstecken. Durch Manipulation einer geeigneten Funktion im Kern können beispielsweise bestimmte Dateien unsichtbar gemacht werden. Da alle regulären Programme in der Regel die gleichen Schnittstellen zum Betriebssystem verwenden, werden diese Dateien damit auch für alle Programme versteckt. Obwohl die Entdeckung eines Rootkits dadurch sehr schwer ist, gibt es mittlerweile eine Reihe von Werkzeugen dafür. Diese verwenden in der Regel eine oder mehrere der folgenden Vorgehensweisen.

Signatur-basierte Suche

Nach bereits bekannten Rootkits kann im Speicher und auf der Festplatte mit Hilfe von Signaturen gesucht werden. Eine Signatur eines Rootkits kann dabei eine typische Folge von Bytes sein. Das Werkzeug „determine" sucht z.B. im Kernelspeicher eines Linux Systems auf diese Weise nach dem Rootkit Adore-NG. Das Programm „chkrootkit" sucht nach Signaturen verschiedener Rootkits auf UNIX und Linux Systemen.

Dies ist auch die typische Vorgehensweise von Anti-Viren Software und kann auf verschiedene Weisen von einem Rootkit unterlaufen werden. Der Zugriff auf die Festplatte kann vom Rootkit manipuliert werden und prinzipiell auch der Zugriff auf den Speicher des Systems. Schreibt das Rootkit keine Daten auf die Festplatte, kann es so nicht gefunden werden. Mit Hilfe einer Signatur können nur bereits bekannte Rootkits gefunden werden. Wurde das Rootkit verändert oder modifiziert es sogar selbstständig den eigenen Programmcode, ist die Suche nach einer festen Signatur erfolglos.

Suche mit generischen Signaturen

Anstatt ein festes Muster zu beschreiben, kann eine Signatur auch generisch ein Rootkit-typisches Verhalten angeben. Damit sollen auch bisher unbekannte und leicht modifizierte Rootkits entdeckt werden. Typisch für ein Rootkit ist beispielsweise der Trick, Programme in Alternate Data Streams auf Windows Systemen zu verstecken. Das NTFS-Dateisystem ermöglicht es, Daten in Alternate Data Streams für den Benutzer unsichtbar zu speichern. Da in der Regel nur Meta-Informationen auf diese Weise

Werkzeuge zum Entdecken und Entfernen von Rootkits

Linux / UNIX	
determine	http://stealth.openwall.net/rootkits/removal/
chkrootkit	http://www.chkrootkit.com/
patchfinder	http://www.phrack.org/phrack/59/p59-0x0a.txt
Windows	
RootkitRevealer	http://www.sysinternals.com/Utilities/RootkitRevealer.html
BlackLight	http://www.f-secure.com/blacklight/
Klister	http://invisiblethings.org/tools/klister-0.4.zip
Strider GhostBuster	http://research.microsoft.com/rootkit/
Patchfinder2	http://invisiblethings.org/tools/PF2/
Vice	http://www.rootkit.com/project.php?id=20
System Virginity Verifier	http://invisiblethings.org/tools/svv/

abgelegt werden, suchen z. B. die Werkzeuge „RootkitRevealer" und „BlackLight" in Alternate Data Streams nach ausführbaren Programmen.

Generische Signaturen führen zu vermehrten Falschmeldungen, da die Signaturen oft nicht nur auf Rootkits sondern auch auf nicht standardkonforme, legitime Programme passen. Wird in einem Rootkit eine neue Technik implementiert, kann es auch durch eine generische Signatur nicht gefunden werden.

Vergleichende Suche

Mit der sog. „Cross-View-Methode" wird die gleiche Information über das System auf verschiedenen Wegen eingeholt und verglichen. Ergibt sich eine Differenz, liegt wahrscheinlich eine Manipulation des Systems vor. Der eine Weg ist dabei die Standard-Methode, der andere greift direkt auf die internen Strukturen des Betriebssystems zu und verwendet nicht die normalen Schnittstellen.

Eine vergleichende Suche kann z. B. nach versteckten Dateien in einem Verzeichnis durchgeführt werden. Unter Linux / UNIX werden die Dateien in einem Verzeichnis mit dem Befehl „ls" aufgelistet, wobei u.a. der oft von Rootkits manipulierte Systemaufruf getdents() verwendet wird. Mit dem Programm „debugfs" kann das Dateisystem analysiert werden, ohne dass die üblichen Systemaufrufe wie getdents() verwendet werden. Tauchen Dateien in der Ausgabe von debugfs auf, aber nicht bei ls, wurden diese von einem Rootkit versteckt.

Ein weiteres Beispiel ist die Untersuchung der gerade aktiven Prozesse, deren Liste zuerst durch einen Standard-Befehl ermittelt wird. Diese Liste wird dann mit den Daten verglichen, die ein Programm direkt aus den entsprechenden Strukturen des Betriebssystems gewinnt. So geht z.B. das Werkzeug „klister" unter Windows vor. Weitere Werkzeuge, die zumindest teilweise vergleichend suchen, sind „RootkitRevealer", „Blacklight" und „Strider GhostBuster".

Die vergleichende Suche kann nur schwer vom Rootkit umgangen werden und ist dadurch sehr hilfreich. In der Regel kann damit aber nur die Manipulation selbst aufgedeckt werden, ohne dabei Hinweise auf das vorliegende Rootkit zu erlangen.

Suche nach Anomalien / Test der System-Konsistenz

Bei der Suche nach Anomalien wird allgemein nach Inkonsistenzen im System gesucht. Dies können z.B. Abweichungen bei Standard-Parametern oder zentralen Datenstrukturen im Kern des Betriebssystems sein. Ein Beispiel für einen zu überprüfenden Parameter ist der Zeiger auf die „Interrupt Descriptor Table", die eine zentrale Rolle bei der Ausführung von Systemaufrufen einnimmt. Der Zeiger hat typischer Weise einen bestimmten Wert, aber manche Rootkits lassen den Zeiger auf eine eigene Datenstruktur zeigen, um so den Kontrollfluss zu verändern [6]. Unter Windows werden Konsistenztests bzgl. derartiger Manipulation z. B. vom „System Verginity Verifier" durchgeführt und unter Linux durch „Kstat2".

Ein weiterer Konsistenztest kann durch die Messung der mittleren Laufzeit von Systemaufrufen erfolgen. Die Idee dabei ist, dass die Manipulationen eines Rootkits im Kern des Betriebssystems zur Laufzeit einen nicht unbeträchtlichen Mehraufwand erfordern. Es werden dafür in der Regel Referenzwerte von der gleichen Version des Betriebssystems mit den gleichen Treibern erstellt und diese mit aktuellen Messungen verglichen. Tatsächlich ist der von manchen Rootkits verursachte Mehraufwand so groß, dass auch sehr ungenaue Referenzwerte ausreichen, um die manipulierten Systemaufrufe zu entdecken. Der Konsistenztest wird unter Linux und unter Windows durch die gleichnamigen Programme „Patchfinder" realisiert. Ähnliche Suchen nach Anomalien werden durch die Werkzeuge „vice" und dem „System Virginity Verifier" durchgeführt.

Die Konsistenztests bzw. die Suche nach Anomalien sind hilfreiche Mittel auf der Suche nach Rootkits. Aber auch hier sind Falschmeldungen durch schlecht implementierte, legitime Software möglich und, wie bei der vergleichenden Suche, wird nur die Manipulation aufgedeckt und nicht das Rootkit selbst.

Weitere Beispiele zur Suche nach Rootkits auf UNIX und Linux Systemen sind im Artikel „Rootkits: Techniken und Abwehr" des 10. DFN-CERT Workshop zu finden [6]. Es gibt kein Werkzeug, dass jedes Rootkit finden kann. In der Regel müssen daher eine Reihe von Programmen verwendet werden, um einen sinnvollen Test auf Rootkits durchzuführen.

Rootkits wieder loswerden

Wurde ein Rootkit gefunden, soll es in der Regel vom System entfernt werden. Viele der oben genannten Werkzeuge ermöglichen dies auf bequeme Weise, aber in der Praxis trifft man dabei u.a. auf folgende Probleme:

1. Manche Rootkits setzen sich sehr tief im System fest und ein unbedarftes Entfernen (z.B. durch Umbenennung von Dateien) kann dazu führen, dass das System nicht mehr startfähig ist.
2. Anti-Rootkit Werkzeuge erkennen oft legitime Programme als vermeintliche Rootkits.
3. Ist ein System längere Zeit durch Sicherheitslücken verwundbar, brechen oft verschiedene Angreifer parallel ein und installieren unterschiedliche Rootkits und Hintertüren.

Vor allem aufgrund des letzten Punktes kann man nie wirklich sicher sein, dass alle Rootkits und die damit versteckten Hintertüren entdeckt wurden. Als erste Reaktion kann daher zwar das gefundene Rootkit mit einem geeigneten Werkzeug oder manuell unschädlich gemacht werden, aber die einzig konsequente Reaktion kann danach nur eine Neuinstallation sein. Diese Ansicht wird mittlerweile auch bei Microsoft vertreten [4]. Damit das möglich ist, wird vor allem ein gutes Backup sowohl der Benutzer-Daten als auch des Systems benötigt. In größeren Rechnerpools sollten einzelne Systeme schnell und ohne Aufwand automatisiert neu installiert werden können. Nur so kann effektiv auf einen Einbruch und die Installation eines Rootkits reagiert werden, damit der Schaden minimiert wird.

Der Angreifer kann aber nicht nur auf dem betroffenen System selbst aufgespürt werden. Da ein Angreifer immer etwas mit dem System vor hat, wird sich diese Aktivität auch bald im Netzwerkverkehr niederschlagen. Der Netzwerkverkehr kann mit Hilfe von Intrusion Detection Systemen oder einfach nur durch die Kontrolle der Netflows beobachtet werden, um an zentraler Stelle eine frühe Warnung zu erhalten.

Beispiel für eine vergleichende Suche nach Rootkits (Cross View) auf einem Linux System

Es besteht ein erster Verdacht, dass ein Einbruch in ein Linux-System stattgefunden hat. Der Inhalt von /tmp wird kontrolliert:

```
webhamster:~# ls -al /tmp
total 24
drwxrwxrwt   6 root root 4096 2006-02-18 01:18 .
drwxr-xr-x  21 root root 4096 2006-01-03 02:11 ..
drwxrwxrwt   2 root root 4096 2006-02-17 15:14 .ICE-unix
drwx------   2 root root 4096 2006-02-17 15:35 ssh-UZhejn9506
-r--r--r--   1 root root   11 2006-02-17 15:34 .X0-lock
drwxrwxrwt   2 root root 4096 2006-02-17 15:34 .X11-unix
```

Beim genaueren Hinsehen stellt man fest, dass der Link-Count des Verzeichnisses 5 und nicht 6 sein müsste. Für eine vergleichende Suche kann jetzt mit dem Werkzeug „debugfs" auf das Dateisystem zugegriffen werden, ohne die üblichen Schnittstellen zu verwenden:

```
webhamster:~# debugfs /dev/sda1
debugfs 1.37 (21-Mar-2005)
debugfs:  ls -l /tmp

 808001   41777 (2)      0      0    4096 18-Feb-2006 00:52 .
      2   40755 (2)      0      0    4096  3-Jan-2006 02:11 ..
 243117   41777 (2)      0      0    4096 17-Feb-2006 15:34 .X11-unix
 243118   41777 (2)      0      0    4096 17-Feb-2006 15:14 .ICE-unix
 938006   40755 (2)  21037  27421    4096 17-Feb-2006 15:30 owned
 340099   40700 (2)      0      0    4096 17-Feb-2006 15:35 ssh-UZhejn9506
 808240  100444 (1)      0      0      11 17-Feb-2006 15:34 .X0-lock
```

Es existiert tatsächlich ein weiteres Verzeichnis in /tmp. Dieses kann auch mit Hilfe von „debugfs" kopiert werden, so dass es auch wieder bei normalem Zugriff sichtbar ist:

```
debugfs:  rdump /tmp /root/test
debugfs:  quit
webhamster:/tmp# ls -al /root/test/tmp
total 28
drwxrwxrwx  6 root  root  4096 2006-02-18 01:08 .
drwxr-xr-x  3 root  root  4096 2006-02-18 01:10 ..
drwxrwxrwx  2 root  root  4096 2006-02-17 15:14 .ICE-unix
drwxr-xr-x  4 21037 27421 4096 2006-02-17 15:30 owned
drwx------  2 root  root  4096 2006-02-17 15:35 ssh-UZhejn9506
-r--r--r--  1 root  root    11 2006-02-17 15:34 .X0-lock
drwxrwxrwx  2 root  root  4096 2006-02-17 15:34 .X11-unix
```

Bei Untersuchung des so versteckten Verzeichnisses stellt sich heraus, dass die Angreifer das Kernel-basierte Rootkit Adore-NG installiert haben.

Insgesamt kann die Entwicklung der Rootkits und der Werkzeuge zu ihrer Entdeckung als eine Art Wettrennen gesehen werden. Es ist sinnvoll, sich mit der aktuellen Entwicklung zu befassen und sich z.B. frühzeitig mit den Werkzeugen vertraut zu machen. Immer auf dem aktuellsten Stand zu sein kostet aber viel Zeit und Aufwand. Ein zuverlässiges Backup, eine solide Infrastruktur und feste Kontrolle über das Netzwerk liefern eine allgemeine Grundabsicherung, mit der das Wettrennen aus einem bequemen Abstand verfolgt werden kann.

Tipps zum Umgang mit Sicherheitsvorfällen und Hinweise zum Finden von Rootkits sowie Referenzen zu den oben genanngen Werkzeugen sind auf den Incident Response Seiten des DFN-CERT zu finden [5].

Referenzen

[1] „HIDING OUT UNDER UNIX"
http://www.phrack.org/phrack/25/P25-06

[2] „DVD-Kopiersperre Alpha-DVD: Update oder Uninstaller"
http://www.heise.de/newsticker/meldung/print/71115

[3] „Sony BMGs Kopierschutz mit Rootkit-Funktionen"
http://www.heise.de/security/news/meldung/print/65602

[4] „Microsoft Says Recovery from Malware Becoming Impossible"
http://www.eweek.com/article2/0,1895,1945808,00.asp

[5] „Incident Response"
http://www.dfn-cert.de/dfncert/incident-response/

[6] „Rootkits: Techniken und Abwehr", DFN-CERT Workshop, Februar 2003
http://www.dfn-cert.de/team/bunten/rootkits_ws2003.pdf