Aktuelle Rootkits – Gefahren, Erkennung und Entfernung

Wilhelm Dolle, Edin Dizdarevic, Thomas Fritzinger
interActive Systems GmbH, Berlin

Einleitung und Motivation

Computersysteme mit einer im Internet erreichbaren IP-Adresse werden häufig täglich nach Sicherheitslücken gescannt beziehungsweise direkt angegriffen. Blickt man auf die Sicherheitswarnungen von Softwareherstellern sowie der Computer Emergency Response Teams (CERTs), so wird schnell klar, dass viele verbreitete Programmpakete von Programmierfehlern betroffen sind, beziehungsweise in der Vergangenheit betroffen waren. Sehr beliebt ist bei Einbrüchen in Rechnersysteme das Ausnutzen von Buffer-Overflow und Format-String Schwachstellen. Zum Beispiel waren in der letzten Zeit unter anderem Microsofts Internet Information Server (IIS), der Mailserver Sendmail, der FTP-Server wu-ftpd, verschiedene ssh-Server und der Nameserver BIND von dieser Art Problemen betroffen. Da die von diesen Programmen bereitgestellten Dienste normalerweise ins Internet angeboten werden und nicht immer von Firewalls abgeschirmt werden können, ist es oft nur eine Frage der Zeit, bis ein Angreifer ein ausnutzbares Sicherheitsloch entdeckt. Sobald er Zugriff auf das fremde System erlangt hat, installiert er häufig ein sogenanntes Rootkit. Diese spezielle Software erlaubt es dem Angreifer, seine eigenen Spuren zu verwischen und dauerhaften Zugang zu dem kompromittierten Server zu erhalten. Einmal aktiviert, verhindern Rootkits mehr oder weniger wirkungsvoll ihre Entdeckung und Entfernung.
In diesem Poster wird das Gefährdungspotential, dass von Rootkits ausgeht beschrieben. Gleichzeitig wird anhand von Linux-Beispielen aus dem „Real-World-Einsatz“ gezeigt werden, wie man bereits installierte Rootkits erkennen und entfernen kann. Die beschriebenen Prinzipien und Vorgehensweisen lassen sich mit wenig Aufwand auch auf andere UNIX-Systeme, MacOS sowie auf Windows übertragen und anwenden.

Grundlegende Funktion eines Rootkits

Ein Rootkit ermöglicht es einem Angreifer nicht, den erstmaligen root- Zugang zu erhalten. Es dient vielmehr dazu, einen bereits unrechtmäßig erhaltenen root-Zugang für zukünftige Aktivitäten zu sichern. Um den ungehinderten root-Zugang zum betroffenen System zu einem späteren Zeitpunkt sicher zu stellen, ist es notwendig, eigene Dienste zu installieren, die unabhängig von der ursprünglich ausgenutzten Sicherheitslücke einen direkten Zugang zum System erlauben. Dazu installiert der Einbrecher oft eine veränderte (trojanisierte) Version eines Standarddienstes. Beliebt ist unter anderem ein modifizierter sshd, der ihm etwa nach Eingabe eines bestimmten Benutzernamens und Kennwortes den direkten root-Zugang gewährt. Durch das bloße Installieren eines Rootkits selber entsteht meist noch kein direkter Schaden am System. Angreifer sind in der Regel bemüht,

ihre Anwesenheit zu verbergen und den normalen Systembetrieb möglichst wenig zu stören. Häufig benutzen erfolgreiche Einbrecher den Rechner dazu, um in weitere Systeme einzubrechen oder beispielsweise Passwörter in angeschlossenen lokalen Netzen zu sammeln.

Verschiedene Typen von Rootkits

Grundsätzlich lassen sich Rootkits in zwei Arten unterteilen, dateibasierte und kernelbasierte Typen. Erstere ersetzen im kompromittierten System Dateien und Programme durch ihre eigenen Versionen, um das System zu manipulieren und ihre Existenz zu verschleiern. Unter Linux sehr verbreitete Rootkits dieser Gattung sind zum Beispiel die Linux-Rootkits lrk3, lrk4, lrk5 beziehungsweise das t0rnkit [1]. Das t0rnkit ersetzt unter anderem die Programme du, find, ifconfig, ls, netstat, ps, top und login und lässt sich sogar über Konfigurationsdateien anpassen. Der trojanisierte login-Prozess erlaubte es bestimmten IP-Adressen, sich ohne Eingabe eines Passwortes und ohne Eintrag in ein Logfile als root-Benutzer anzumelden. Es öffnet außerdem in seiner Standardinstallation den TCP-Port 47017 zur Kommunikation von außen.
Kernelbasierte Rootkits modifizieren den Kernel entweder im Speicher und / oder auf der Festplatte. Durch den Umstand, dass sie im Kernel und nicht im normalen Benutzermodus laufen, haben sie alle nötigen Privilegien, um ohne Einschränkungen auf das komplette System zuzugreifen. Um ihre Entdeckung zu verhindern tauschen sie Systemfunktionen aus oder modifizieren die Systemsprungtabelle. Überwachungsprogramme, die ihre Informationen vom Kernel geliefert bekommen, müssen auf diese Weise nicht mehr selber verändert werden. Beliebte Systemfunktionen sind unter anderem:

- open() - lesender Zugriff Original, ausführender Zugriff trojanisierte Datei
- execve(), clone(), fork() - Ausführen von Programmen mit bestimmten Eigenschaften (u.a. verstecken in der Prozessliste), und Vererbung an Kindprozesse
- getdents(), mkdir(), chdir(), rmdir() - Verstecken von Verzeichnissen und Dateien
- stat() - Manipulation der Dateieigenschaften
- ioctl() - Device-Kontrolle

Die ersten kernelbasierten Rootkits modifizierten den Kernel über das Laden eines oder mehrerer Module. Verbreitete Vertreter unter Linux sind zum Beispiel Knark (für Kernel 2.2) [2] und Adore (für Kernel 2.2 und 2.4) [3].
Auf der DefCon 9 wurde 2001 mit dem Kernel Intrusion System (KIS) zum ersten Mal ein kernelbasiertes Rootkit vorgestellt, dass den Kernel direkt im Speicher (Kernel-Memory-Patching) über /dev/kmem verändern kann. 2002 tauchte mit SucKIT [4] ein weiterer Vertreter dieses neuen Typus auf. Das KIS bringt einen eigenen Modulloader mit und benötigt keinerlei Unterstützung des Kernels für ladbare Module. Zusätzlich zu den oben angesprochenen Eigenschaften bringt KIS noch eine versteckte Hintertür (Stealth-Backdoor) mit. Es lauscht erst auf einem

Port nachdem ein spezielles TCP-Paket auf einen beliebigen Port an den Rechner geschickt wurde, und es ist vorher nicht durch Portscans erkennbar. Außerdem wurde KIS in einer Client- / Serverstruktur realisiert und ist durch ein Interface für Plug-ins leicht erweiterbar. Dadurch wird es auch für weniger versierte Angreifer zu einem leicht bedienbarem Werkzeug.

Erkennen und Entfernen von Rootkits

Sehr oft lassen sich Rootkits durch einen Vergleich des Ergebnisses eines Portscans von außen mit der lokalen Ausgabe der offenen Ports über netstat erkennen. Normalerweise verschleiert ein Rootkit nach innen seine Anwesenheit dadurch, dass es einen nach außen geöffneten Kommunikationsport nicht anzeigt. Gibt es also Differenzen zwischen den beiden Listen, so kann dies als erster Hinweis auf ein vorhandenes Rootkit dienen. Weitere Schritte zum Aufspüren von Rootkits kann die Liste an in den Kernel geladenen Modulen oder der Vergleich der Systemsprungtabelle mit einer sauberen System.map sein. Falls eine Signatur des Rootkits bekannt ist, kann nach dieser im Speicher gesucht werden. Da viele Rootkits ihre Prozesse, und damit bestimmte PIDs, verstecken, kann mit einem Hilfsprogramm versucht werden, nacheinander alle PIDs zu belegen. Sollten hierbei Fehler auftreten, ist dies ein weiterer Hinweis auf das eventuelle Vorhandensein eines Rootkits. Zur automatisierten Erkennung von Rootkits findet man im Internet unter anderem die Projekte „chkrootkit“ [5] und „Saint Jude“.
Hat man das Rootkit erst mal gefunden und identifiziert, kann man es sehr häufig leicht entfernen oder zumindest deaktivieren. Danach sollte man das System am besten komplett neu aufsetzen. Ist dies nicht möglich, muss man alle trojanisierten Programme durch ihre Originale ersetzen und eventuell vorhandene versteckte Verzeichnisse löschen. Es bleibt dann nur noch herauszufinden, welche Sicherheitslücke sich der Einbrecher zunutze gemacht hat, und diese ebenfalls zu schließen.

Literatur

[1] Analysis of the T0rn rootkit, Toby Miller, www.sans.org/y2k/t0rn.htm
[2] Analysis of the KNARK rootkit, Toby Miller, www.securityfocus.com/guest/4871
[3] Adore Worm - security advisory; Matt Fearnow, William Stearns; www.sans.org/y2k/adore.htm
[4] Linux on-the-fly kernel patching without LKM; sd, devik; Phrack 58, December 12, 2001; phrack.org/phrack/58/p58-0x07
[5] Homepage des Projektes ChkRootKit, www.chkrootkit.org