

Core Wars: Gefahr durch Schwachstellen im Kernel

Bei Schwachstellen im Kernel und in Treibern besteht besonders hohe Angriffsgefahr, da Malware direkt auf die Hardware zugreifen kann. ZDNet nimmt mögliche Abwehrstrategien unter die Lupe.

Von [Lothar Lochmaier](#), 21. Februar 2008

"Das besonders tückische Element besteht darin, dass Schwachstellen auf Kernel-Ebene sämtliche Sicherheitsfunktionen umgehen", gab Sicherheitsberater Tobias Klein von Cirosec auf der diesjährigen [IT Defense](#) in Hamburg zu bedenken.

Der Experte hat dem Trend einen Namen gegeben und mit dem Begriff "Core Wars" bezeichnet, eine Art von schleichender und meistens unentdeckter Kriegsführung quasi mitten ins Herz der Unternehmen. Gemeint sind Angriffe, die auf den innersten Kern eines Betriebssystems abzielen. Im Gegensatz zu klassischer Malware zielen diese professionellen Varianten vor allem darauf ab, Hintertüren zu öffnen, mit denen sich das Unternehmen über eine längere Zeit gezielt ausspionieren lässt.

Schwachstellen im Kernel haben jedoch nur wenig Gemeinsamkeiten mit Rootkits, ebenfalls verdeckt operierenden Schädlingen. "Man kann Kernel-Schwachstellen für viele Dinge missbrauchen, das Laden von Rootkits ist dabei nur eine Variante", betont Klein.

Bei allen Betriebssystemen, außer bei Windows Vista 64-Bit, sei hingegen gar keine Schwachstelle im Kernel erforderlich, um ein Kernel Rootkit zu laden. Dazu seien lediglich administrative Rechte erforderlich. "Bei Windows Vista 64-Bit habe Microsoft hingegen bereits einige Sicherheitsmechanismen eingebaut, wie die erzwungene Treibersignierung, so dass Kernel-Rootkits hier nur noch über Schwachstellen im Kernel in den Kern gebracht werden könnten."

Fakt ist aber auch: Die Liste der bislang unveröffentlichten Kernel-Schwachstellen in verschiedenen Betriebssystemen wächst laut Klein generell weiter an. Von derartigen Attacken auf den Kernel betroffen sei unter anderem jetzt schon Windows Vista, aber auch Sun Solaris. Einige der Schwachstellen aus dem Jahr 2007 seien noch immer nicht behoben, die Ausbesserung dauere manchmal bis zu acht Monaten.

Der innerbetrieblichen Abwehr geht somit wertvolle Zeit verloren, insbesondere im Kampf gegen die organisierte Wirtschaftskriminalität. Klein demonstriert am Beispiel der Kernel von Mac OS X und Sun Solaris sowie diversen Treibern unter Windows Vista, welche verheerenden Auswirkungen dies haben kann.

Neben der Erweiterung der lokalen Rechte hat der Experte zahlreiche Schwachstellen enttarnt, um Rootkits in den Kernel einzuschleusen. Aber auch andere Sicherheitsmechanismen ließen sich komplett aushebeln, beispielsweise das in Solaris 10 eingeführte Zonenkonzept oder die erzwungene Treiber-Signierung unter 64-Bit-Vista. Mehr Details kann der Experte nicht verraten, da die Schwachstellen immer noch nicht behoben worden seien.



Cirosec GmbH
Edisonstraße 21
74076 Heilbronn
Tel: 07131 / 59455-0
Fax: 07131 / 59455-99
info@cirosec.de
www.cirosec.de

Nach Kleins Auffassung wird die Suche nach Schwachstellen innerhalb von Betriebssystem-Kernen durch die zeitverzögerte Reaktion der Hersteller immer unverzichtbarer. "Der Trend wird sich ganz klar in Richtung Kernel-Schwachstellen entwickeln, die entweder direkt remote ausgenutzt werden können oder mit konventionellen Userland-Schwachstellen kombiniert werden."

Hoffnungsschimmer Hypervisor

Was die strategische Abwehr angeht, so erschwert der latente Streit um die besten Methoden die sichere Orientierung. Derzeit konzentrierten sich die verfügbaren Sicherheitsmechanismen auf die Absicherung des Userland-Bereichs, auf Seiten des Kernels gebe es hingegen nur rudimentäre bis gar keine Schutzmöglichkeiten, beklagt Klein.

Als viel versprechende Lösungen sieht der Berater sowohl Mikrokern- als auch Hypervisor-Technologien an. "Diese werden jedoch noch von keinem aktuellen Mainstream-Betriebssystem umgesetzt", beklagt Klein. Derzeit seien beide Ansätze noch im Anfangsstadium. "Da es keine Hersteller oder Produkte gibt, bedeutet dies auch, dass Unternehmen kaum etwas dagegen unternehmen könnten."

"Im Moment lässt sich also der Kernel von Mainstream-Betriebssystemen nicht absichern", so die kritische Bilanz. Zwar weiß auch der Experte, dass weder ein Mikrokern- noch ein Hypervisor-basierter Ansatz ein hundertprozentiges Schutzniveau vor dem Ausnutzen von Kernel-Schwachstellen bieten. "Dies ist auch im Normalfall nie zu erreichen", relativiert Klein. Dennoch macht er zahlreiche Vorteile eines mikrokernbasierten Schutzansatzes aus.

Die Vorteile des Mikrokern-Ansatzes liegen vor allem in dem erheblich geringeren Ausmaß an Code im Kernelkontext mit nur sehr zentralen Funktionen, "was die Trusted Computing Base, also den privilegierten Kernel-Code, dem ich blind vertrauen muss, viel kleiner werden lässt als bei einem heutigen monolithischen Kernel", sagt Klein.

Sämtliche Treiber, die im User-Mode-Kontext laufen, besitzen erheblich weniger Rechte, was bedeutet, dass der Mikrokern eine extrem verkleinerte Angriffsfläche präsentiert. "Es muss aber stets Code im Kernel-Kontext betrieben werden, denn wenn dieser einen Fehler aufweist, dann läuft dies wieder auf das altbekannte Problem hinaus. Dies ist aber zumindest viel unwahrscheinlicher", bilanziert Klein.

Deshalb gilt es die Maßnahmen sinnvoll zu kombinieren. So basiert die Idee, die jeweiligen Abwehrmaßnahmen direkt beim Hypervisor anzusetzen, auf dem Konzept, eine zusätzliche Komponente einzuführen, die den Kern selbst überwachen kann. "Also eine Art Überwachungsfunktion des Überwachers", pointiert Klein.

Das Ziel bestehe darin, damit den Kernel indirekt besser abzusichern. Der Hypervisor sei dabei im Vergleich zum heutigen Betriebssystem-Kernel viel kleiner, umfasse erheblich weniger Zeilen an Code und biete somit weniger Angriffsfläche. Nichtsdestotrotz bestehe auch hier, wie bereits beim Mikrokern-Ansatz, das Risiko, dass eine Schwachstelle im Hypervisor das ganze Konzept aushebeln könne.

"Es ist aber unwahrscheinlicher", so der Experte. Im Endeffekt werde es

zwar keine perfekte und umfassende Lösung geben. Jedoch seien sowohl die mikrokernbasierte Abwehr als auch diejenige auf Ebene des Hypervisors zwei vielversprechende Ansätze, um die Risiken erheblich zu minimieren

Fazit

Ein echter Mikrokern mit Treibern im User-Kontext bietet aus sicherheitstechnischen Überlegungen natürlich Vorteile. Dem stehen allerdings erhebliche Performancenachteile gegenüber, da sich Mikrokern und Treiber nicht direkt aufrufen können.

Um die Kommunikation zu ermöglichen, muss Message-Passing implementiert werden. Zudem wird bei jedem Aufruf ein Context-Switch fällig, der die Leistung stark reduziert. Die [Tanenbaum-Torvalds-Debatte](#) hat der Anwender längst entschieden. Das auf User-Mode-Treibern basierende [Minix 3.0](#) führt ein Nischendasein im Vergleich zum Monolithen Linux.

Im Gegensatz dazu ist Virtualisierung per Hypervisor im Kommen. Hier bietet sich die Chance, eine weitere Kontrollinstanz mit sehr geringer Angriffsfläche zu etablieren. Dem Betriebssystem wird unter der Kontrolle des Hypervisors nur vorgekauelt, dass es volle Kontrolle über die Hardware besitze.

Voraussetzung ist allerdings, dass das Betriebssystem dafür konzipiert ist, mit einem Hypervisor zusammenzuarbeiten. Dies ist heute nur bei paravirtualisierten Linux-Kerneln und Windows Server 2008 der Fall. Ansonsten kann der Betriebssystemkernel zwar nicht auf die echte Hardware zugreifen, aber Schaden in der virtuellen Maschine anrichten, was im Endeffekt keinen Unterschied macht.

Bei Kerneln heutiger Bauart ist aus Expertensicht vor allem die laxen Reaktion der Hersteller problematisch. Die Lieferanten der Betriebssysteme sowie die Hersteller von Third-Party-Treibern benötigen mitunter sehr lange, um Schwachstellen innerhalb des Kernels zu beheben. Dies ist in vielen Fällen aber auch nicht weiter verwunderlich, da heutige Betriebssystem-Kernel sehr komplex sind und die darin vorgenommenen Änderungen gravierende Auswirkungen nach sich ziehen können.

Zudem halten viele Hersteller nur über verhältnismäßig geringe Ressourcen bereit, um die Schwachstellen innerhalb einer adäquaten Zeitspanne zu beheben. Um diesem latenten Missstand ein Ende zu bereiten, verfügten die Spezialisten jedoch nur über rudimentäre Möglichkeiten im Linux-Umfeld in Form der grsecurity-Kernel-Patches, sowie der neuen Kernel-Option `mmap.min.addr`, um Memory-mapped-Files nicht in Speicherbereiche des Kernels laden zu können.

Auch Werkzeuge wie der [RK Profiler](#) können nichts gegen Kernel-Schwachstellen ausrichten. Sie sind lediglich ein gutes Werkzeug, um entsprechende Kernel-Rootkits auffindig zu machen.