
Bedrohung der Systemsicherheit durch Rootkits



Wilhelm Dolle, Berlin, 22. Oktober 2005

Agenda

- **Was ist ein Rootkit?**
- **Klassifizierung und Möglichkeiten von Rootkits**
- **Rootkits im Einsatz**
- **Rootkits aufspüren**
- **Werkzeuge zum Aufspüren von Rootkits**

Typischer (gezielter) Einbruch übers Netz

- **Footprinting** (allgemeine Informationen über Ziel sammeln)
 - öffentliche Datenbanken, Google, Unternehmens-Webseiten, DNS-Einträge, Internetanbindung per traceroute, ...
- **Protokoll- und Portscans / Enumeration / Schwachstellenanalyse**
 - Betriebssystemerkennung, Bannergrabbing, Verwundbarkeitsscans
- **Ausnutzen einer Sicherheitslücke**
 - Exploits (Buffer Overflow, ...) anwenden
- **Verstecken und Festsetzen**
 - Hintertüren, Sniffer, Keylogger, ...
- **Einbruchsspuren verwischen**
- **System missbrauchen**
 - Plattform für weitere Angriffe, Botnetze, Datenspionage, ...

Werkzeuge für die Informationsbeschaffung

```
wd@T42p ~]$ traceroute www.ccc.de
traceroute to www.ccc.de (213.73.91.29), 30 hops max, 38 byte packets
 1 192.168.101.1 (192.168.101.1) 1.319 ms 0.361 ms 0.261 ms
 2 217.0.116.6 (217.0.116.6) 51.296 ms 75.974 ms 50.438 ms
 3 217.0.64.86 (217.0.64.86) 49.689 ms 50.503 ms 50.448 ms
 4 62.154.32.222 (62.154.32.222) 57.674 ms 56.867 ms 57.799 ms
 5 pos2-0.core01.ham01.atlas.cogentco.com (212.20.158.38) 55.961 ms 56.720 ms 62.724 ms
 6 p6-0.core01.sxf01.atlas.cogentco.com (130.117.1.182) 60.176 ms 62.095 ms 60.561 ms
 7 fe0-0.ca01.sxf01.atlas.cogentco.com (130.117.1.186) 61.307 ms 61.591 ms 61.063 ms
 8 130.117.22.11 (130.117.22.11) 60.829 ms 66.305 ms 66.100 ms
 9 84.23.252.21 (84.23.252.21) 62.225 ms 62.029 ms 62.553 ms
10 www.ccc.de (213.73.91.29) 61.814 ms 62.121 ms 64.555 ms
wd@T42p ~]$
```

Nessus "NG" Report

Subnet: 192.168.53

Severity: Security Warning, Security Note, Security Hole

Port: ssh (22/tcp), sometimes-rpc8 (32772/udp), sometimes-rpc17 (32777/tcp), sometimes-rpc14 (32775/udp), sometimes-rpc12 (32774/udp), sometimes-rpc10 (32773/udp), lockd (4045/udp), general/tcp, general/icmp, finger (79/tcp)

Host: 192.168.53.10, 192.168.53.100, 192.168.53.150, 192.168.53.151, 192.168.53.170, 192.168.53.200

The 'finger' service provides useful information to attackers, since they can gain usernames, check if a machine is being used, and...

Here is the output we obtained for 'root':

| Login | Name | TTY | Idle | When | Where |
|-------|------------|---------|------|-------------|-------|
| root | Super-User | console | | 2 Tue 21:46 | :0 |

Solution: comment out the 'finger' line in /etc/inetd.conf
Risk factor: Low
CVE: CVE-1999-0612

Google - Mozilla Firefox

http://www.google.de/

Google Deutschland

Web Bilder Groups Verzeichnis News Froogle Mehr »

exploit sendmail

packet storm

back to your roots

about mirrors search assessment defense advisories papers magazines

miscellaneous links forums

Archive Search Results for: exploit linux

Search Results: 1 - 25

| File Name: | sses-sshauth.txt |
|--------------|---|
| Description: | A vulnerable secure shell discovered by the popular Zeddy Consultants known as replay.com). The RHEL ssh-1.2.27-8i.src.rpm contains faulty logic allowing |

```
root@T42p ~]$ # nmap
Nmap 3.75 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
* -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
* -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
* -sV Version scan probes open ports determining service & app names/versions
* -sR RPC scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
* -p <range> ports to scan. Example range: 1-1024,1080,6666,31337
* -F Only scans ports listed in nmap-services
* -v Verbose. Its use is recommended. Use twice for greater effect.
* -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[...] Hide scan using many decoys
* -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
* -n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
* -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
* -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
* --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
root@T42p ~]$
```

Was ist ein Rootkit?

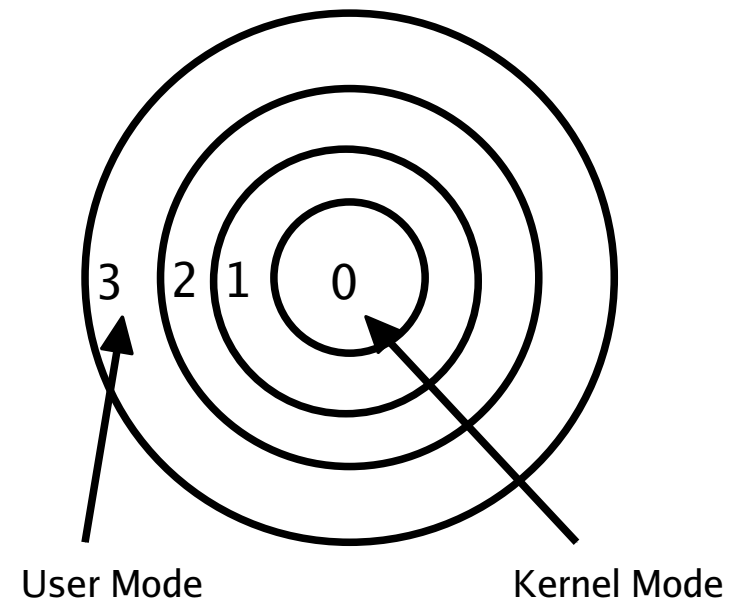
- „Ein Rootkit ist ein Satz von Programmen der dauerhafte und nicht aufzuspürende Gegenwart auf einem Computer erlaubt.“ (Hoglund, Butler)
- Der Fokus von Rootkits ist die Heimlichkeit (durchaus auch für legale Zwecke)
- Weitere Funktionen
 - Hintertüren
 - Fernsteuerung von Rechnern
 - Spyware (Netzwerk-Sniffer, Key-Logger, ...)
- Was ist ein Rootkit **nicht**?
 - Angriffswerkzeug oder Exploit (es kann aber Exploits enthalten)
 - Wurm oder Virus (Rootkit-Technik kann aber in diese integriert werden)

Klassifikation

- **Dateibasierte Rootkits** (User Mode Rootkits)
 - Manipulieren Systemprogramme oder Bibliotheken im Dateisystem
- **Kernelbasierte Rootkits** (Kernel Mode Rootkits)
 - Kommen durch Module (LKM) oder direkte Manipulation im RAM oder auf der Festplatte in den Kernel
 - Manipulieren Datenstrukturen bzw. Übergabewerte direkt im Kernel
- **Memory Based Rootkits**
 - Befinden sich nur im RAM und überstehen einen Reboot nicht
- **Persistente Rootkits**
 - Nisten sich auf der Festplatte ein und werden nach Reboot wieder aktiviert

Unterschied zwischen User und Kernel Mode

- Intel x86-Familie nutzt Ringe 0 bis 3 für Zugriffssteuerung
- Ring 0 hat die höchsten, Ring 3 die niedrigsten Privilegien zum Zugriff
- Viele Betriebssysteme (u.a. Linux, Windows) nutzen nur 0 (Kernel Mode) und 3 (User Mode)
- Zugriff auf Ringe mit kleinerer Nummer ist (bis wenige auf Ausnahmen, z.B. Laden von Treibern) verboten
- Kernel Mode (OS-Kernel) hat unbeschränkten Zugriff auf das ganze System
- Ring 3 enthält alle anderen Programme (**auch die mit Admin-Rechten!**)
- Rootkits im Kernel können beliebig Sicherheitssoftware im User-Mode manipulieren oder beenden, sie bestimmen welche Daten die Software sieht und bekommen z.B. Netzwerkpakete vor einem Paketfilter (stealth backdoor)



Für Kernel Rootkits interessante Systemaufrufe

- **open()**

lesender Zugriff ergibt Original, ausführender Zugriff ergibt modifizierte Version

- **getdents(), mkdir(), chdir(), rmdir()**

Verstecken von Dateien und Verzeichnissen

- **execve(), clone(), fork()**

Programme verstecken und Vererbung an Kindprozesse

- **stat()**

Manipulation von Dateieigenschaften

- **ioctl()**

Device-Kontrolle, z.B. kein promiscuous-Bit (Sniffer) anzeigen

Historie von Rootkits

- Ende 80er: Manipulieren von Logfiles
- 1989: Phrack-Magazin-Artikel zum Umgehen von Unix-Überwachung
- 1994: CERT-Hinweis auf Sammlung von Programmen unter SunOS („rootkits“)
- 1996: erste Linux-Rootkits
- 1997: Phrack-Magazin-Artikel zu LKM-Rootkits unter Linux (später auch UNIX)
- 1998: direktes Patchen des Kernels im RAM ohne LKM (Silvio Cesare)
- 1999: LKM-Rootkits: knark (speziell zur Täuschung von Tripwire), adore
- 1999: erstes Windows Rootkit (NT-Rootkit)
- 2000: t0rnkit v8 libproc library Rootkit
- 2001: KIS und SuckKIT manipulieren Kernel direkt im RAM (Technik aus 1998)
- 2002: erste Rootkits mit Sniffer-Backdoors

Beispiel User Mode Rootkit: t0rnkit

- syslogd anhalten
- Hash-Wert des Passwortes der Hintertür in /etc/ttyhash ablegen
- Trojanisierter SSH-Dienst unter /usr/sbin/nscd ablegen
- Start als „# Name Server Cache Daemon ...“ in der Startdatei /etc/rc.d/sysinit (Defaultport 47017)
- Konfigurationsdateien unter /usr/info/.t0rn ablegen
- Austauschen der folgenden Systemprogramme (Zeitstempel und Größen werden zurückgesetzt): login, ls, netstat, ps, ifconfig, top, du, find
- in.fingerd öffnet eine Shell am Defaultport 2555
- Verschiedene Binarys (z.B. Sniffer) unter /usr/src/.puta und /dev/.lib ablegen
- Aktivieren von telnet, rsh und finger in /etc/initd.conf
- syslogd neustarten

Beispiel Kernel Mode Rootkit: adore-ng

- Entwickelt von Team Teso für Linux Kernel 2.2, inzwischen auf 2.4 und als adore-ng auch auf 2.6 portiert
- Linux Kernelmodul, Cleaner-Modul, Kommandozeilenwerkzeug
- Features
 - Verstecken von Dateien und Verzeichnissen
 - Verstecken von Prozessen
 - Verstecken von Netzwerkverbindungen
 - Hintertür um die Rechte auf dem System zu erweitern
 - Filtern von Logfile-Einträgen
 - ...

adore-ng - Konfiguration

```
root@rh73:/home/wd/adore-ng
```

```
Starting adore configuration ...
```

```
Checking 4 ELITE_UID + ELITE_GID ... found 2229832308, 1831430087
```

```
Checking 4 SMP ... NO
```

```
Checking 4 MODVERSIONS ... YES
```

```
Checking for kgcc ... found cc
```

```
Checking 4 insmod ... found /sbin/insmod -- OK
```

```
Loaded modules:
```

| | | | |
|----------|--------|---|-------------------|
| vmhgfs | 42624 | 4 | |
| vmxnet | 8288 | 1 | |
| usbcore | 73152 | 1 | |
| BusLogic | 94592 | 3 | |
| sd_mod | 12864 | 6 | |
| scsi_mod | 108576 | 2 | [BusLogic sd_mod] |

Since version 0.33 Adore requires 'authentication' for its services. You will be prompted for a password now and this password will be compiled into 'adore' and 'ava' so no further actions by you are required.

This procedure will save adore from scanners.

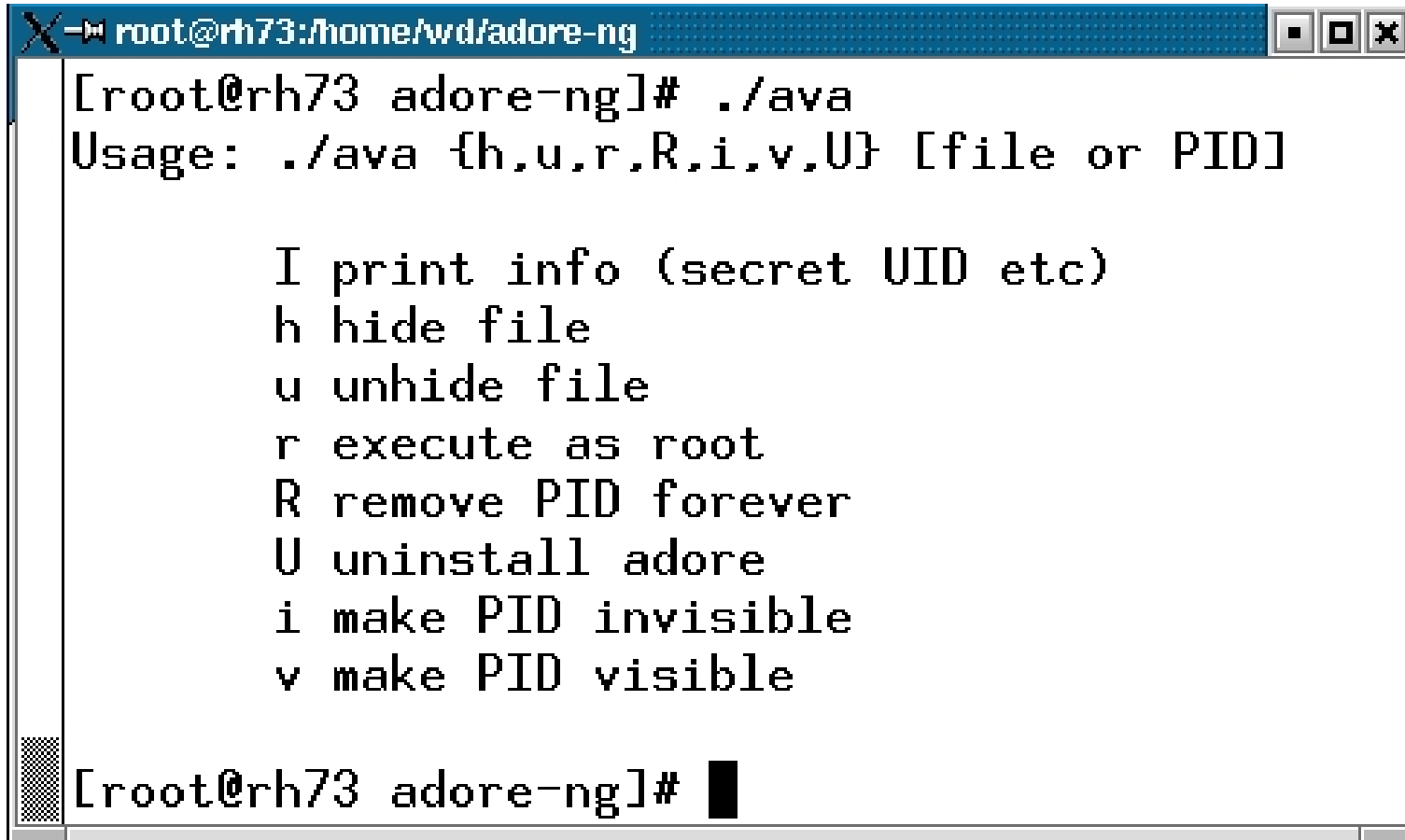
Try to choose a unique name that is won't clash with filenames in /proc.

```
Password (echoed):ADORE_KEY█
```

adore-ng – Starten und Verstecken

```
root@rh73:/home/wd/adore-ng
[Root@rh73 adore-ng]# insmod ./adore-ng.o
[Root@rh73 adore-ng]# lsmod
Module                Size  Used by    Tainted: PF
adore-ng              18624  0  (unused)
vmhgfs                42624  4
vmxnet                8288   1
usbcore               73152  1
BusLogic              94592  3
sd_mod                12864  6
scsi_mod              108576  2  [BusLogic sd_mod]
[Root@rh73 adore-ng]# insmod cleaner.o
[Root@rh73 adore-ng]# lsmod
Module                Size  Used by    Tainted: PF
cleaner                624   0  (unused)
vmhgfs                42624  4
vmxnet                8288   1
usbcore               73152  1
BusLogic              94592  3
sd_mod                12864  6
scsi_mod              108576  2  [BusLogic sd_mod]
[Root@rh73 adore-ng]# rmmod cleaner
[Root@rh73 adore-ng]# lsmod
Module                Size  Used by    Tainted: PF
vmhgfs                42624  4
vmxnet                8288   1
usbcore               73152  1
BusLogic              94592  3
sd_mod                12864  6
scsi_mod              108576  2  [BusLogic sd_mod]
[Root@rh73 adore-ng]# █
```

adore-ng – Kommandozeilentool ava



```
root@rh73:/home/wd/adore-ng
[ root@rh73 adore-ng ]# ./ava
Usage: ./ava {h,u,r,R,i,v,U} [file or PID]

    I print info (secret UID etc)
    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

[ root@rh73 adore-ng ]#
```

adore-ng – Dateien / Verzeichnisse verstecken

```
root@rh73:~/home/wd/Rootkits
[Root@rh73 Rootkits]# ls -ld ../adore-ng
drwxr-xr-x    3 2229832308 1831430087    4096 Oct 21 22:40 ../adore-ng
[Root@rh73 Rootkits]# insmod ../adore-ng/adore-ng.o
[Root@rh73 Rootkits]# ../adore-ng/ava I
Checking for adore 0.12 or higher ...
Adore 1.41 installed. Good luck.

ELITE_UID: 2229832308, ELITE_GID=1831430087, ADORE_KEY=ADORE_KEY CURRENT_ADORE=41
[Root@rh73 Rootkits]# touch geheime_datei
[Root@rh73 Rootkits]# ls -l geheime_datei
-rw-r--r--    1 root    root          0 Oct 21 22:43 geheime_datei
[Root@rh73 Rootkits]# ../adore-ng/ava h geheime_datei
Checking for adore 0.12 or higher ...
Adore 1.41 installed. Good luck.
File 'geheime_datei' hided.
[Root@rh73 Rootkits]# ls
[Root@rh73 Rootkits]# rmmmod adore-ng
[Root@rh73 Rootkits]# ls -l
total 0
-rw-r--r--    1 2229832308 1831430087    0 Oct 21 22:43 geheime_datei
[Root@rh73 Rootkits]# █
```

adore-ng – Prozesse verstecken

```
root@rh73:/home/wd/Rootkits
[1] 1298
[Root@rh73 Rootkits]# ps
  PID TTY          TIME CMD
 1257 pts/2        00:00:00 bash
  1298 pts/2        00:00:00 sleep
  1299 pts/2        00:00:00 ps
[Root@rh73 Rootkits]#
[Root@rh73 Rootkits]# insmod ../adore-ng/adore-ng.o
[Root@rh73 Rootkits]# ../adore-ng/ava i 1298
Checking for adore 0.12 or higher ...
Adore 1.41 installed. Good luck.
Made PID 1298 invisible.
[Root@rh73 Rootkits]# ps
  PID TTY          TIME CMD
 1257 pts/2        00:00:00 bash
  1302 pts/2        00:00:00 ps
[Root@rh73 Rootkits]# ../adore-ng/ava v 1298
Checking for adore 0.12 or higher ...
Adore 1.41 installed. Good luck.
Made PID 1298 visible.
[Root@rh73 Rootkits]# ps
  PID TTY          TIME CMD
 1257 pts/2        00:00:00 bash
  1298 pts/2        00:00:00 sleep
  1304 pts/2        00:00:00 ps
[Root@rh73 Rootkits]#
```


Rootkits aufspüren I

- Startscripte prüfen
- Oft trojanisierte Programme prüfen: ps, ls, find, ifconfig, netstat, du, df, sshd, httpd, login, passwd, ...
- Verdächtige Rechte oder MAC-Times in /sbin oder /usr/sbin? (im Zweifel Binarys mit „strings“ ansehen)
- Dateien mit Link Count kleiner 1 (ls -l +L1)
- History der Shell ansehen (z.B. .bash_history)
- Normale Dateien im Device-Dateisystem (find /dev -type f)?
- Geladene Module prüfen
- Nach Interfaces im promiscuous-Mode suchen (Sniffer)
- Analyse der installierten Pakete mit „rpm --verify --all“ (Anzeige: S,5,T, ...)

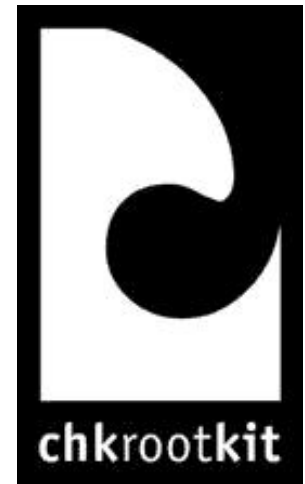
Rootkits aufspüren II

- Logfiles auf verdächtige Einträge durchsuchen (Neustart von Diensten war evtl. Buffer Overflow?)
- Portscan von außen mit einem internen Portscan oder netstat vergleichen (versteckte Port, findet aber nicht Sniffer/hidden Backdoors)
- Signaturen von bekannten Rootkits im Speicher und in Dateien suchen
- Durch PID-Test versuchen alle freien PID durch einen Testprozess zu belegen (nicht belegbare PIDs können ein Hinweis auf versteckte Prozesse sein)

- Cross-View-Based Rootkit-Detection: Vergleich der Sicht durch einen High-Level-Zugriff über die APIs mit einem Low-Level-Zugriff auf „Rohdaten“ (z.B. Lesen von einzelnen Sektoren von der Platte statt über das Dateisystem)

Werkzeuge zum Aufspüren: ChkRootKit

- www.chkrootkit.org, lokal ausführbares Script
- Erkennt sehr viele bekannte Rootkits
(Stand Februar 2005, Version 0.45, ca. 60)
- Benutzt lokale Befehle, darum mit statisch gelinkten Binaries von CD benutzen



```
root@rh73:/home/wd/chkrootkit-0.45
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... Warning: Adore LKM installed
ps: error: Thread display not implemented.
usage: ps -[Unix98 options]
       ps [BSD-style options]
       ps --[GNU-style long options]
       ps --help for a command summary

OooPS!
chkproc: Warning: Possible LKM Trojan installed
Checking `rexedcs'... not found
Checking `sniffer'... eth0: not promisc and no PF_PACKET sockets
Checking `w55808'... not infected
Checking `wted'... chkwtmp: nothing deleted
Checking `scalper'... not infected
Checking `slapper'... not infected
Checking `z2'... chklastlog: nothing deleted
Checking `chkutmp'... chkutmp: nothing deleted
[root@rh73 chkrootkit-0.45]#
```

Werkzeuge zum Aufspüren: Rootkit Hunter

- www.rootkit.nl, aktuelle Version 1.2.7, Mai 2005
- Benutzt unter anderem die folgenden Methoden
 - MD5 Hash-Vergleich mit bekannten Werten
 - Suche nach typischen Dateien, Verzeichnissen und Ports die von Rootkits genutzt werden
 - Falsche Berechtigungen von Binärdateien
 - Suche nach verdächtigen Zeichenketten im LKM
 - Suche nach versteckten Dateien (in den falschen Verzeichnissen)
 - Betriebssystem abhängige Tests (bei Linux z.B. Vergleich der Prozesse über die Ausgabe von „ps“ und das parsen von /proc)
 - Optional scannen innerhalb von Plaintext und Binärdateien nach verdächtigen Zeichenketten



Rootkit Hunter im Einsatz I

```
root@fc3-server:/home/wd/Rootkits/rkhunter
[root@fc3-server rkhunter]# rkhunter -c --createlogfile

Rootkit Hunter 1.2.7 is running
Determining OS... Ready

Checking binaries
* Selftests
  Strings (command)          [ OK ]

* System tools
Info: prelinked files found

/bin/cat
/bin/chmod
/bin/chown
/bin/dmesg
/bin/egrep
/bin/env
/bin/fgrep
```

```
root@fc3-server:/home/wd/Rootkits/rkhunter
----- Scan results -----

MD5
MD5 compared: 72
Incorrect MD5 checksums: 0

File scan
Scanned files: 342
Possible infected files: 0

Application scan
Vulnerable applications: 2

Scanning took 2093 seconds
Scan results written to logfile (/var/log/rkhunter.log)

-----

Do you have some problems, undetected rootkits, false positives, ideas
or suggestions?
Please e-mail me by filling in the contact form (@http://www.rootkit.nl)

-----

[root@fc3-server rkhunter]#
```

Rootkit Hunter im Einsatz II

```
root@fc3-server:/var/log
[05:25:25] *** Start scan T0rn Rootkit ***
[05:25:25] - File /dev/.lib/lib/lib/t0rns... OK. Not found.
[05:25:25] - File /dev/.lib/lib/lib/du... OK. Not found.
[05:25:26] - File /dev/.lib/lib/lib/ls... OK. Not found.
[05:25:26] - File /dev/.lib/lib/lib/t0rnsb... OK. Not found.
[05:25:26] - File /dev/.lib/lib/lib/ps... OK. Not found.
[05:25:26] - File /dev/.lib/lib/lib/t0rnp... OK. Not found.
[05:25:27] - File /dev/.lib/lib/lib/find... OK. Not found.
[05:25:27] - File /dev/.lib/lib/lib/ifconfig... OK. Not found.
[05:25:27] - File /dev/.lib/lib/lib/pg... OK. Not found.
[05:25:28] - File /dev/.lib/lib/lib/ssh.tgz... OK. Not found.
[05:25:28] - File /dev/.lib/lib/lib/top... OK. Not found.
[05:25:28] - File /dev/.lib/lib/lib/sz... OK. Not found.
[05:25:28] - File /dev/.lib/lib/lib/login... OK. Not found.
[05:25:29] - File /dev/.lib/lib/lib/in.fingerd... OK. Not found.
[05:25:29] - File /dev/.lib/lib/lib/li0n.sh... OK. Not found.
[05:25:29] - File /dev/.lib/lib/lib/pstree... OK. Not found.
[05:25:30] - File /dev/.lib/lib/lib/in.telnetd... OK. Not found.
[05:25:30] - File /dev/.lib/lib/lib/mjy... OK. Not found.
[05:25:30] - File /dev/.lib/lib/lib/sush... OK. Not found.
[05:25:30] - File /dev/.lib/lib/lib/tfn... OK. Not found.
[05:25:31] - File /dev/.lib/lib/lib/name... OK. Not found.
[05:25:31] - File /dev/.lib/lib/lib/getip.sh... OK. Not found.
[05:25:31] - File /usr/info/.torn/sh*... OK. Not found.
[05:25:32] - File /usr/src/.puta/... OK. Not found.
[05:25:32] - File /usr/src/.puta/.laddr... OK. Not found.
[05:25:32] - File /usr/src/.puta/.lfile... OK. Not found.
[05:25:33] - File /usr/src/.puta/.lproc... OK. Not found.
[05:25:33] - File /usr/src/.puta/.llogz... OK. Not found.
[05:25:34] - File /usr/info/.t0rn/... OK. Not found.
[05:25:34] - Directory /dev/.lib/... OK. Not found.
[05:25:34] - Directory /dev/.lib/lib/... OK. Not found.
[05:25:34] - Directory /dev/.lib/lib/lib/... OK. Not found.
:
```

Entfernen von Rootkits

- Anhand der gefundenen Spuren im Internet nach Informationen und Hinweisen zur Entfernung des Rootkits suchen
 - Module entfernen
 - Manipulierte Dateien durch Originale ersetzen
 - Startscripte säubern

- **Wann immer möglich System komplett neu aufsetzen!**

Q & A

- **Vielen Dank für die Aufmerksamkeit!**
- **Folien unter www.dolle.net**
- **„Rootkits“ (Windows);
Hoglund, Butler; Addison-Wesley 2005**

