



TECHNISCHE
UNIVERSITÄT
WIEN
VIENNA
UNIVERSITY OF
TECHNOLOGY

Security VU

183.124

WS2003/2004

Lab 3 Arbeitsprotokoll

Nachvollziehen eines Angriffs

Haider Gerald (0125638)
Radl Christoph (0102799)
Schachinger Josef (0125692)

Inhaltsverzeichnis

1	LAB 3A – MITSCHNITT ANALYSIEREN	3
2	LAB3B	6
2.1	Was hat der Angreifer gemacht und welche Schwachstellen in den Diensten hat er ausgenützt?	6
2.1.1	Tripwire Scan	6
2.1.2	Suche nach Exploits	8
2.2	Falls rootkits installiert wurden, welche Dateien wurden infiziert bzw. verändert?	9
2.2.1	Chkrootkit-Scan	9
2.3	Wurden zusätzlich noch weitere Dateien in letzter Zeit auf diesem System verändert? Sind diese Veränderungen gutartiger (Update/Patch) oder bösartiger Natur?	11
2.4	Falls rootkits installiert wurden, funktionieren diese überhaupt? (script kiddies :))	11
3	REFERENZEN	11

1 Lab 3a – Mitschnitt analysieren

```

1 valley% sh
2 $ who
   Der Angreifer informiert sich über die derzeit angemeldeten Benutzer
3 ingres      ttyp0      Jan 18 23:02
4 root        ttyp2      Jan 15 18:38      (canyon)
   „ingress“ und „root“ sind derzeit im System aktiv
5 $ cp /home2/jeff/bin/kermit.org Kermit
   „Installation“ des Netzwerktools C-Kermit
6 $ Kermit
   Aufruf von C-Kermit1
7 C-Kermit 5A(178) ALPHA, 29 Jan 92, SUNOS 4.1 (BSD)
8 Type ? or HELP for help
9 C-Kermit>rece fi
   „rece fi“ ist ein Kommando in C-Kermit, welches einen listening socket für Dateitransfer öffnet, das „fi“ steht wahrscheinlich für den Dateinamen der anzulegenden Datei
10 Escape back to your local Kermit and give a SEND command...
11 # N3
12 0 Yz* @-#Y1~N1! y-
13 %!YfiO
14 #Y@
15 ##YA
16 #YB
17 #YC
18 #YD
19 C-Kermit>
20 Stopped
21 valley% sh
   der Angreifer öffnet eine neue Shell
22 Stopped (signal)
23 valley% sh
   der Angreifer öffnet eine neue Shell
24 <overflows buffer here>
   ...und schon ist es passiert
25 $ /tmp/sh
26 # rm /tmp/sh
   der Angreifer versucht nun die Datei /tmp/sh zu löschen
27 rm: override protection 755 for /tmp/sh? Y
   obwohl nur „root“ Schreibrechte auf /tmp/sh hätte gelingt es dem Angreifer diese Datei zu löschen
28 # lsll
   Aufruf eines Scripts dass vermutlich zum Anzeigen aller symbolischen Links dient (ls long listing of symbolic links); ergebnislos?
29 # ls -tal
   ls, mit folgenden Parametern: t = Auflistung nach Modifikationsdatum, a = Auflistung aller Dateien, l = Auflistung mit Permissions und Dateigrösse usw.
30 total 1049
31 drwxr-xr-x  4 ingres      512 Jan 18 23:04 .
32 -rwsrwsrwx  1 root        24576 Jan 18 23:04 suck
33 -rw-r--r--  1 root         61 Jan 18 23:04 c.c
34 -rwxr-xr-x  1 ingres    442368 Jan 18 23:03 kermit
35 -rwxrwxrwx  1 ingres   360448 Jan 16 11:02 testit
36 drwxr-xr-x 30 root        1024 Dec 18 20:27 ..
37 -rw-r--r--  1 ingres    1148 Jun  9 1992 foo

```

¹ C-Kermit is a combined network and serial communication software package offering a consistent, transport-independent, cross-platform approach to connection establishment, terminal sessions, file transfer, file management, character-set translation, numeric and alphanumeric paging, and automation of communication tasks through its built-in scripting language.

```

38 drwxrwsrwx 6 ingres      6144 Aug 23 1991 SERVICE
39 -rwxr-xr-x 1 ingres     106496 Feb 25 1991 sun4_lookup
40 -rwxr-xr-x 1 ingres     98304 Feb 25 1991 sun3_lookup
41 drwxr-xr-x 3 ingres      512 Jan 23 1991 quoter
42 -rw-r--r-- 1 ingres      306 Nov 20 1987 .cshrc
43 -rw-r--r-- 1 ingres     1159 Nov 20 1987 .install
44 -r--r--r-- 1 ingres       20 Nov 20 1987 .version
45 -rw-r--r-- 1 ingres       36 Jan 26 1987 .oemstring
46 # who
47 ingres      ttyp0      Jan 18 23:02
48 root        ttyp2      Jan 15 18:38      (canyon)
    Der Angreifer informiert sich erneut über die derzeit angemeldeten Benutzer
49 # last | grep -i est
    Aus dem last-login-file sollen alle Zeilen ausgegeben werden die „est“ enthalten i = ignore case
50 # last lorin
    Angreifer stellt fest wann hat sich lorin zuletzt eingeloggt hat..
51 wtmp begins Sat Jan 16 11:37
    Antwort: scheinbar gar nicht
52 # grep lor /etc/passwd
    Sinnloser Befehl, da vertippt
53 grep: /etc/passwd: No such file or directory
54 # grep lor /etc/passwd
55 # ypcat passwd | grep lor
    ypcat soll Informationen über Network Information Services ausgeben in diesem Fall ist passwd das Ziel, es sollen nur die Zeilen angezeigt werden die lor enthalten.
56 lori:N.4PgZ4iUS8kk:5734:50:Lori:/home/lori:/bin/csh
57 lorimo:xxYTF8y3fSgGo:21477:50:Lori:/home/lorimo:/bin/csh
    Und hier haben wir schon die gehashten Passwörter für „lori“ und „lorimo“!
58 # ed c.c
    Der Angreifer editiert die Datei c.c
59 /uid/
60 setuid(0);
61 setuid(21477);
62 # cc .cc
    fehlerhafter aufruf des C-Compilers
63 cc: Warning: File with unknown suffix (.cc) passed to ld
64 ld: .cc: No such file or directory
65 # cc `c
66 > ^C
67 # cc c.c
    Compilieren der Datei c.c
68 # mv a.out shit
    umbenennen der durch den Compiler erzeugten Datei a.out in shit
69 # chmod 6777 shit
    die datei ausführbar machen...
70 # ./suck
    fehlingabe
71 # id
72 uid=0(root) gid=0(wheel) groups=7
73 # ./shit
    die datei „shit“ ausführen
74 $ id
75 uid=21477(lorimo) gid=0(wheel) groups=7
76 $rlogin tsunami
77 Password:
78 Login incorrect
79 Login incorrect
80 login: ^D
81 Connection closed.
82 $ rlogjn suntzu
83 rlogjn: not found
84 $ rlogin suntzu
85 Password
86 Login incorrect

```

```

87 Login: ^D
88 Connection closed.
89 $ ^D
    der Angreifer versuchte sich mittels rlogin auf tsunami und suntzu einzuloggen... an-
    scheinend aber ohne erfolg
90 # who
91 ingres      ttyp0      Jan 18 23:02
92 root       ttyp2      Jan 15 18:38      (canyon)
93 # ypcat passwd | grep lorimo
94 lorimo:xxYTF8y3fSgGo:21477:50:Lori:/home/lorimo:/bin/csh
    Erneutes Ausgeben des Passwort-Hashs für "lorimo"
95 # cd /home
96 # find . -name .rhosts -print &
    suche nach der ".rhosts" Datei und anzeige, das ganze wird im Hintergrund ausge-
    führt
97 # gupr
98 # grep^C
    2 Fehlerhafte Eingaben
99 # ypcat passwd | grep jeff
100 jeff:wW/q0t03L6xO.:13147:50:jeff :/home/jeff:/bin/csh
    Ausgeben des Passworthashs für "jeff"
101 # ed c.c
102 ?c.c: No such file or directory
103 # cd
104 # ed c.c
105 /uid/
106 setuid(21477);
107 setuid(13147);
108 # cc c.c
109 # mv a.out shit
110 # chmod 6777 shit
111 # ./shit
    Funktion des Programmes „shit“ ist fraglich... event. Password-Cracker??
112 $ id
    Informieren über die Gruppen in denen der aktuelle Benutzer ist
113 uid=13147(jeff) gid=0(wheel) groups=7
114 $ rlogj tsunami
115 rlogj: not found
    Tipfehler...
116 $ rlogin tsunami
    Nun loggt sich unser Angreifer auf dem Remotesystem "tsunami" ein
117 No directory! Logging in with home=/
118 SunOS Release 4.1.2 (TSUNAMI) #3 Sat Oct 24 07:56:45 PDT
    1992
119 You have new mail.
120 tsunami% ^C
121 tsunami% sh
122 $ who
    Der Angreifer informiert sich über die derzeit angemeldeten Benutzer
123 wendy      ttyp2      Jan  6 13:55      (arawana)
124 derek      ttyp3      Jan 13 17:57      (lajolla)
125 derek      ttyp4      Jan 15 13:11      (lajolla)
126 jeff       ttyp5      Jan 18 23:09      (valley)

```

2 Lab3b

Auf einen unserer Rechner wurde ein Angriff durchgeführt. Dieser wird von uns nun, abgetrennt vom Produktivsystem, forensisch untersucht. Im Besonderen interessiert uns dabei, wie es dem Angreifer gelungen ist in das System einzudringen. Diese Erkenntnisse sind deshalb so wichtig damit die richtigen Maßnahmen getroffen werden können um Angriffe dieser Art in Zukunft unmöglich zu machen. Denn das Sicherheitskonzept muss nach einem mehr oder weniger erfolgreichen Angriff, auch mit glimpflichem Ausgang, in jedem Fall überarbeitet werden.

2.1 Was hat der Angreifer gemacht und welche Schwachstellen in den Diensten hat er ausgenützt?

Zunächst haben wir uns auf dem korrumpierten Rechner eingeloggt und uns im root-Mode auf die Spurensuche begeben. Wir hatten auch einen tcpdump-Mitschnitt zur Verfügung, der alle Netzwerkverbindungen mitprotokolliert. Leider war dieser Mitschnitt nicht ganz vollständig.

Nach der ersten durchsicht des uns zur vorliegenden tcpdump wurde die Vermutung angestellt der Angriff wäre über den „sunrpc“ (111) port gegangen. Die spätere Analyse des Systems mit checkexploit lässt uns aber eher vermuten das der Angriff über eine Schwachstelle im WU-FTPD erfolgte.

2.1.1 Tripwire Scan

Das bekannte Intrusion Detection System „tripwire“ war bereits vorinstalliert und so waren wir so frei hiermit gleich einen Systemcheck durchzuführen. Tripwire sucht wichtige Systemdateien auf verdächtige Veränderungen, z.B. durch Eindringlinge, ab und gibt diese dem User aus. Der Befehl für einen generellen Systemcheck lautet:

```
[root@linux_6 /root]# tripwire --check
```

Hierauf erhält man eine ziemlich ausführliche Liste von Änderungen an Systemdateien, wirklich wichtige Änderungen zu erkennen braucht einige Zeit deshalb werden wir uns mit dem Output hier auf die interessantesten Zeilen beschränken.

```
-----  
Rule Name: User binaries (/usr/sbin)  
Severity Level: 66  
-----
```

```
Added:  
"/usr/sbin/ttymon"
```

```
Modified:  
"/usr/sbin"  
"/usr/sbin/lsof"
```

```
-----  
Rule Name: Libraries (/usr/lib)  
Severity Level: 66  
-----
```

```
Added:  
"/usr/lib/perl5/man/whatis"
```

```
Modified:
"/usr/lib/perl5/man"
```

```
-----
Rule Name: User binaries (/usr/bin)
Severity Level: 66
-----
```

```
Modified:
"/usr/bin"
"/usr/bin/dir"
"/usr/bin/find"
"/usr/bin/md5sum"
"/usr/bin/pstree"
"/usr/bin/slocate"
"/usr/bin/top"
```

Die hier angeführten Änderungen sind mit ziemlicher Sicherheit auf die Installation eines rootkits zurückzuführen, weil genau die modifizierten Programme zum aufspüren von rootkits sehr nützlich wären, die Ausgaben dieser Programme sind also für die Untersuchung des Systems eher skeptisch zu betrachten.

```
-----
Rule Name: Networking Programs (/sbin/ifconfig)
Severity Level: 100
-----
```

```
Modified:
"/sbin/ifconfig"
```

Die Netzwerkinterface Konfiguration wurde offenbar verändert. Dies erfolgte wahrscheinlich um zu verbergen falls die Netzwerkkarte in den „Promiscuouse Mode“ geschaltetet wird.

```
-----
Rule Name: System Administration Programs (/sbin/syslogd)
Severity Level: 100
-----
```

```
Modified:
"/sbin/syslogd"
```

Hier wurde offenbar der Daemon der für die Logging Dateien zuständig is verändert.

```
-----
Rule Name: System boot changes (/var/log)
Severity Level: 100
-----
```

```
Added:
"/var/log/iptraf"
```

```
Removed:
"/var/log/lastlog"
```

```
Modified:
"/var/log"
```

Offenbar wurde die Datei mit den Last-login Einträgen entfernt. Daher wurde auch das /log Verzeichnis modifiziert.

```
-----
Rule Name: Operating System Utilities (/bin/login)
Severity Level: 100
-----
```

```
Modified:
"/bin/login"
```

Wie sich in unseren späteren Untersuchungen noch zeigen wird wurde der „login“ auch verändert

```
-----  
Rule Name: Operating System Utilities (/bin/ls)  
Severity Level: 100  
-----
```

```
Modified:  
"/bin/ls"
```

Auch der für uns wichtige ls-Befehl wurde verändert.

```
-----  
Rule Name: Operating System Utilities (/bin/netstat)  
Severity Level: 100  
-----
```

```
Modified:  
"/bin/netstat"
```

Der Output von „netstat“ ist solange wir das System untersuchen auch mit Vorsicht zu genießen. Ein Aufruf von Netstat endet aber sowieso mit einem „Segmentation Fault“ was auf eine fehlerhafte Installation des rootkits hindeutet.

```
-----  
Rule Name: Operating System Utilities (/bin/ps)  
Severity Level: 100  
-----
```

```
Modified:  
"/bin/ps"
```

ps wäre für die Auflistung von Prozessen zuständig hier wurden Änderungen vorgenommen und daher sollte auch diesen Informationen nicht blind vertraut werden.

```
-----  
Rule Name: Critical system boot files (/boot)  
Severity Level: 100  
-----
```

```
Modified:  
"/boot"
```

Weiters is /boot verändert worden. Ein Neustart wird also auch wieder mit Aktionen des Angreifers bzw. seiner Programme verbunden sein

2.1.2 Suche nach Exploits

Wir haben im root-Verzeichnis weiters ein Archiv namens security_scanner.tar.gz gefunden. Nach der Extraktion desselbigen hatten wir den Scanner chkexploit zu Verfügung der das System analysiert und speziell nach Installation und Konfiguration des aktuellen Systems potentielle Verwundbarkeiten offen legt. Für unser angegriffenes System konnten wir so folgende Verwundbarkeiten herausfinden.

```
suidperl: VULNERABLE  
  Problem: Local users can get root access.  
  Fix: Remove the suid bit from the sperl binary.
```

```
crontab: VULNERABLE  
  Problem: Local users can get root access.  
  Fix: Disable crontab or get the newest version.
```

```
ftpd: VULNERABLE
  Problem: Non-local users may gain access to the system.
  Fix: Download and install the latest version of ftpd.

at: VULNERABLE
  Problem: Local users may gain root access.
  Fix: chmod 700 /var/spool/atjobs and upgrade the 'at'
      command to version 2.7 or newer.

suid_rw_partitions: VULNERABLE
  Problem: Local users can create and exec suid binaries.
  Fix: Mount public writable partitions with nosuid option.

bind: VULNERABLE
  Problem: Vulnerable Cache.
  Fix: Upgrade to a newer 4.9.6/8.1.1 version.
```

Wir sehen an dieser Auflistung eines sofort: wenn der Angriff über ein Fremdes System erfolgt ist, so muss dies über eine Sicherheitslücke in unserem "veralteten" FTP-Daemon erfolgt sein. Wenn wir also wissen, dass der Angriff „remote“ erfolgt ist so können wir unsere Spurensuche schon sehr stark einschränken. Ist dies nicht der Fall dann müssen wir auch die anderen fünf Dienste (und die von ihnen verwendeten Ressourcen) untersuchen. Da bei „bind“ aber nur ein vulnerable cache vorliegt (somit nur das modifizieren der DNS Einträge möglich ist), kann „bind“ als Angriffsstelle eher ausgeschlossen werden.

2.2 Falls rootkits installiert wurden, welche Dateien wurden infiziert bzw. verändert?

In oben erwähntem security_scanner-Archiv fanden wir weiters den Scanner chkrootkit der uns nach der Installation folgenden Output lieferte.

2.2.1 Chkrootkit-Scan

```
[root@linux_6 chkrootkit-0.42b]# ./chkrootkit | grep INFECTED
Checking `ifconfig'... INFECTED
Checking `login'... INFECTED
Checking `pstree'... INFECTED
unable to open lastlog-file lastlog
```

Hier bestätigen sich gleich unsere obigen Vermutungen, dass login kompromittiert ist. Vom Rootkit sind auch noch die zwei anderen Dateien betroffen.

Neben den bereits oben gefundenen Änderungen sind uns noch die folgenden Unregelmäßigkeiten aufgefallen.

Es gibt etliche Dateien auf dem System mit „seltsamer“ UserID, ganz zufällig sind das die Dateien die bereits beim check mit Tripwire aufgefallen sind. Ein listing der gefundenen Dateien:

```
[root@linux_6 /sbin]# find / -uid 3287
find: /proc/6/fd: Permission denied
/usr/bin/dir
/usr/bin/md5sum
/usr/bin/find
```

```
/usr/bin/top
/usr/bin/pstree
/usr/bin/slocate
/usr/sbin/lsof
/usr/sbin/ttymon
/bin/ls
/bin/netstat
/bin/ps
/bin/login
/sbin/ifconfig
/sbin/syslogd
```

Eine nähere Untersuchung der obigen Dateien mittels strace bracht wiederum Hinweise auf die folgenden Dateien:

```
[root@linux_6 include]# find . -uid 500
./file.h
./hosts.h
./log.h
./proc.h
```

Dies Dateien scheinen so etwas die Konfigurationsdateien für das rootkit zu sein, wie man aus dem Inhalt eben dieser schließen kann:

```
[root@linux_6 include]# cat file.h hosts.h log.h proc.h
libext-2.so.7
.sh
system
tksb
tkp
lblip.tk
tk
ldd.so
srd0
ldlib.5
.config
ld.so.hash

2 212.110
2 195.26
2 194.143
3 2002
4 2002
3 6667
4 6667
2 62.220
3
4
62.220
212.110
195.26

SH-FORCE
sh-FORCE
psyBNC
eggdrop
t0rn
torn
tornkit

3 eggdrop
3 bnc
3 psyBNC
3 sh-FORCE
3 SH-FORCE
3 synscan
3 setup
```

```
3 in.inetd
3 tk
3 xntps
```

Die erste Datei definiert scheinbar welche Dateien durch das rootkit versteckt werden sollen. In der nächsten werden wahrscheinlich die zu versteckenden Netzwerkverbindungen definiert. Die nächsten beiden Dateien definieren vermutlich die die zu versteckenden Log-Einträge und Prozesse.

2.3 Wurden zusätzlich noch weitere Dateien in letzter Zeit auf diesem System verändert? Sind diese Veränderungen gutartiger (Update/Patch) oder bössartiger Natur?

Der Tripwire Check zeigt natürlich auch die von uns eben erst installierten Tools checkrootkit und checkexploit an, diese sind natürlich als gutartige Änderung zu betrachten.

2.4 Falls rootkits installiert wurden, funktionieren diese überhaupt? (script kiddies :))

Es liegt die Vermutung nahe das das rootkit nicht ordnungsgemäß läuft, hinweise dafür sind zum Beispiel der Absturz von netstat.

3 Referenzen

[1] Skoudis, Ed: Counter Hack, Prentice Hall 2002