Forensik nach Angriffen auf Linux-Systeme

Steffen Wendzel

Schwierigkeitsgrad

Das Thema Forensik wird vielen von Ihnen bereits durch diverse TV-Serien wie CSI bekannt sein. Die Forensik beinhaltet allerdings viele Untergebiete und eines davon ist die IT-Forensik. Dieser Artikel stellt Grundlagen der Forensik unter Linux dar und soll primär dazu dienen, dem Leser eine Idee der Möglichkeiten und der Vorgehensweise zu vermitteln.

In der IT-Forensik geht es grob gesagt darum, herauszufinden wie, warum um von wem ein Angriff auf einen Computer (oder mehrere) durchgeführt wurde.

Die Forensik beinhaltet Methoden, Beweismaterial zu finden, zu sichern und natürlich zu analysieren. Welche Möglichkeiten stehen einem dabei aber unter Linux-Systemen (und generell unter Betriebssystemen) zur Verfügung?

Dateisystem-Forensik

Eine Möglichkeit besteht darin, den Inhalt der Datenträger eines Systems zu sichern und später zu analysieren. Dabei kann nach verdächtigen Dateien gesucht werden. Dass können bspw. Dateien seien, die in letzter Zeit modifiziert oder erstellt wurden. Man spricht bei diesen Informationen auch von den MAC-Daten einer Datei. MAC steht für modification, access, creation und bezieht sich auf die Timestamps zur Modifikation des Inhalts, zum letzten Zugriff und zur letzten Veränderung von Dateiattributen wie dem Eigentümer, der Gruppenzugehörigkeit oder dem Dateinamen. Manche Dateisysteme speichern diese Daten noch etwas anders oder fügen noch eine Protokollierung über den Zeitpunkt hinzu, zu dem eine Datei gelöscht wurde. Bereits gelöschte Dateien, die dem Benutzer nicht mehr sichtbar sind, geben auf diese Weise unter Umständen doch noch etwas über sich preis.

Unter typischen *Linux-/Unix-Dateisystemen* werden besonders wichtige Eigenschaften von Dateien in deren Inode-Eintrag gespeichert. Inode-Einträge befinden sich in einem extra Bereich der Partition auf den der Verwaltungscode des Dateisystems zugreift. Listing 1 bietet einen Auszug aus der Inode-Struktur des ext3-

In diesem Artikel erfahren Sie:

- Welche forensischen Möglichkeiten nach einem;
- Angriff auf ein Linux-System zur Verfügung stehen und wie diese angewandt werden.

Was Sie vorher wissen/können sollten:

- Linux Grundlagen;
- Grundlegendes Verständnis für die Sicherheitskonzepte von Linux.

Dateisystems, die in `/usr/src/linux/ext3_fs.h` zu finden ist. Zu den gespeicherten Informationen zählen die Zugriffsrechte (`i_mode`), User-ID des Eigentümers, Dateigröße, die MAC-Werte, der Löschzeitpunkt, die Gruppen-ID, der Link-Counter, die Anzahl der Blöcke, die eine Datei belegt und die Flags die einer Datei im Dateisystem zugewiesen wurden.

Unbedingt verlassen kann man sich auf MAC-Werte allerdings auch nicht, denn diese können vom Userspace mit entsprechenden Zugriffsrechten und Syscalls wie `utime(2)` modifiziert werden.

Zudem müssen nicht alle Daten, die sich innerhalb einer Partition befinden, dem Dateisystem (als Datei) bekannt sein. Im Dateisystem gibt es in der Regel nicht verwendete Bereiche (beispielsweise am Ende einer Partition), die nicht durch Inode-Einträge referenziert sind. Typischer Weise werden etwa SWAP-Partitionen so gut wie nie völlig benutzt. Am Ende einer SWAP-Partition können Daten von einem Angreifer also relativ langlebig aufbewahrt werden – er muss sich nur merken, wo genau die Daten abgespeichert wurden.

Um die Daten eines Mediums genauer zu analysieren führt man am Besten einen Dump auf ein externes Medium (bspw. NFS oder eine große Festplatte) durch, was mit Tools wie `dd` kein Problem ist. Viel schwieriger wird hingegen die Untersuchung des Inhalts. Im einfachsten Fall untersucht man die Rohdaten mit einem Programm wie `strings` nach ASCII-Zeichen und versucht sich interessante Inhalte mit `grep` herauszufiltern. Genau das ist bei der riesigen Menge an Daten, die sich auf einem Dateisystem befinden allerdings nicht so einfach und erfordert viel Geduld. Besonders hilfreich sind auch Forensik-Programme, wie *icat* (Dateiinhalte via *Inode* auflisten) oder *lazarus* (Inhalte gelöschter Dateien wieder auflisten), unter Umständen zusätzliche Tools.

Hat man aber nun einmal interessante Dateien im Dateisystem gefunden, die von einem Angreifer stammen könnten (etwa Verzeichnisse mit irreführenden Namen, die Leerzeichen am Ende des Dateinamens oder ein Newline-Zeichen beinhalten), dann kann lassen sich in solchen Verzeichnissen oft interessante weitere Dateien – etwa Exploit-Code – finden.

Weiterhin interessant und wohl das Erste, was man als Administrator intuitiv überprüfen wird, sind die Inhalte der Logdateien, doch besonders diese können einfach modifiziert

Listing 1. *Auszug der Inode-Struktur des ext3-fs*

```
 /*
* Structure of an inode on the disk
*/
struct ext3_inode {
        __le16  i_mode;         /* File mode */
        __le16  i_uid;          /* Low 16 bits of Owner Uid */
        __le32  i_size;         /* Size in bytes */
        __le32  i_atime;        /* Access time */
        __le32  i_ctime;        /* Creation time */
        __le32  i_mtime;        /* Modification time */
        __le32  i_dtime;        /* Deletion Time */
        __le16  i_gid;          /* Low 16 bits of Group Id */
        __le16  i_links_count;  /* Links count */
        __le32  i_blocks;       /* Blocks count */
        __le32  i_flags;        /* File flags */
```

Listing 2. *Erstellen eines Inode-Eintrages und inkrementieren des Link-Counters*

```
$ touch a
$ ls -il a
605050 -rw-r--r-- 1 swendzel swendzel 0 2007-02-26 01:21 a
$ ln a b
$ ls -il a b
605050 -rw-r--r-- 2 swendzel swendzel 0 2007-02-26 01:21 a
605050 -rw-r--r-- 2 swendzel swendzel 0 2007-02-26 01:21 b
```

Listing 3. *Start der Backdoor*

```
$ ./m0rtix

              ,--.      |    o
        ,-.-.|  |,---.|--- ..  ,
        | | ||  ||    |    | ><
        ` ' '`--'`    `---'`'  `

PsychoPhobia Backdoor v3 by m0rtix is starting...OK, pid = 5576
Shell on: 9997      User: swendzel        UID: 1000
Name: /sbin/syslogd  (Masked in PS! )  v: = Linux amilo 2.6.17-11-386


Rootab !! use: expand_stack, Krad(if 2004), Krad2(if 2004), Krad3 !
```

Listing 4. *Speichersäuberung in einer Backdoor*

```
char *buffer;

if (!(buffer = (char *)calloc
(0xff, sizeof(char))))
          err(1, ...);

recv(buffer, ...);
system(buffer);
bzero(buffer, 0xff);
free(buffer);
```

werden, wenn kein remote-Logging aktiviert wurde, und die Timestamps der Dateien entsprechend ihrem Inhalt angepasst wurden.

Besonders hilfreich – aber auch besonders Log-Intensiv – für die Dateisystemforensik sind übrigens File System Intrusion Detection Systeme wie AIDE, die ich in der Februar-Ausgabe der *hakin9* besprochen hatte.

Speicher-Forensik

In der Speicherforensik geht es darum, möglichst viele Informationen über einen Angriff und den Angreifer aus dem Datenspeicher des Systems zu extrahieren. Das ist nicht immer ganz trivial, da die Daten unter Umständen vom Angreifer verschlüsselt wurden, oder man gar nicht genau weiß, wonach man eigentlich sucht.

Interessant sind dabei auch Informationen darüber, welche Prozesse zuletzt liefen, wer eingeloggt war und welche Daten die einzelnen Prozesse im Speicher hielten.

Es gäbe noch sehr viel mehr über Speicherforensik zu sagen, doch statt dessen soll ein Beispiel Interesse wecken und auf die erwähnten Bücher am Ende des Artikels verwiesen werden.

Im Beispiel hat ein Angreifer die auf einem Linux 2.6-System die `m0rtix` Backdoor gestartet. Diese Backdoor ist relativ simpel und nimmt TCP-Verbindungen standardmäßig auf Port 9997 entgegen. Während der Verbindung kann der Angreifer mit einer Remote-Shell arbeiten. Die Backdoor tarnt sich standardmäßig als syslog-Daemon, was natürlich der Verschleierung dient.

Die Dektektion der Backdoor ist aber allein schon dann möglich, wenn einem auffällt, dass der *syslogd*-Prozess nicht dem Benutzer syslogd bzw. *root* gehört, sondern in diesem Fall von mir ausgeführt wird und zudem doppelt vorhanden ist.

```
$ ps auwx|grep syslog
root      4156  0.0  0.2   1652   612
 ?        Ss   19:49   0:00
/sbin/syslogd
swendzel  5576  0.0  0.2   1816   612
 pts/7    S+   20:11   0:00
/sbin/syslogd
swendzel  5617  0.0  0.3   2844   776
pts/2     R+   20:13   0:00 grep syslog
```

Gehen wir nun davon aus, dass wir nicht wissen, dass die Backdoor auf dem System installiert ist und unser Netzwerk Intrusion Detection System (NIDS), das hier mit Wireshark simuliert wird, ein verdächtiges Paket abgefangen hat. Mittels dieser Informationen kann man nun herausfinden, dass auf dem TCP-Port 9997 sehr wahrscheinlich eine Backdoor läuft. Nun könnte man mit Tools wie *lsof* oder ps(tree) versuchen, dem Prozess auf die Schliche kommen, doch wählen wir einen anderen Weg: Wir erstellen ein Abbild des Userspace-Memory und speichern es in einer Datei:

```
# dd if=/dev/mem of=/mnt/sda1/
memdump bs=1M
```

Hat man das erledigt, dann besteht der einfachste Weg darin, die Datei in einen *Hexeditor* (ich empfehle *hexcurse*) zu laden und nach den Strings/Binärdaten des abgefangenen Packages zu suchen. Der String *uid=1000(hat als Hexwert 7569643d3130303028*. Sucht man unmittelbar nach dem Angriff danach, hat man eine große Chance fündig zu werden. (Mehr zu möglichen Problemen am Ende des Artikels).

Da der String nun entdeckt wurde, und es sich mit hoher Wahrscheinlichkeit um Daten der Backdoor handelt, haben wir nun die Chance die Daten vor und hinter dem Bereich zu untersuchen, in denen die gefundenen Daten liegen. Auf diese Weise können weitere Befehle gefunden werden, die der Angreifer in seiner Remote-Shell eingegeben hat.

Wie man sehen kann, hat der Angreifer auch noch weitere Befehle eingegeben und versucht in das Verzeichnis /root zu wechseln, was aber nicht funktioniert haben dürfte.

Link-Counter

Der eben genannte *Link-Counter* gibt an, wieviel Instanzen einer Datei im Dateisystem existieren, d.h. wie oft eine Inode referenziert wird. Um dies zu verstehen muss man zunächst verstehen, wie Verzeichnisse funktionieren. Verzeichnisse sind selbst nur reguläre Dateien unter unixartigen Dateisystemen. Diese Dateien beinhalten die Inode-Nummern der Dateien, die sie aus Sicht des Benutzers beinhalten. Der Dateisystem-Code im Kernel kann über diese Inode-Nummer wiederum die bekannten Attribute an ein Programm, dass den Inhalt eines Verzeichnisses auslesen möchte, zurückgeben. Jedes Mal, wenn in einem Verzeichnis die Inode-Nummer einer Datei hinzu gefügt wird, wird der *Link-Counter,* der mit der Erstellung einer Datei auf den Wert `1` gesetzt wird, inkrementiert. Erreicht der *Link-Counter* eines Tages den Wert `0`, dann wird in keinem Verzeichnis mehr auf diese Datei verwiesen, und der benutzte Speicherbereich (der ebenfalls in der Inode referenziert wird), kann freigegeben werden. Aus der Sicht des Benutzers wurde die Datei in diesem Fall ganz einfach gelöscht. Die Inode sowie der *Link-Counter* einer Datei lassen sich übrigens mit `ls -il` ausgeben, wie Listing 2 zeigt.

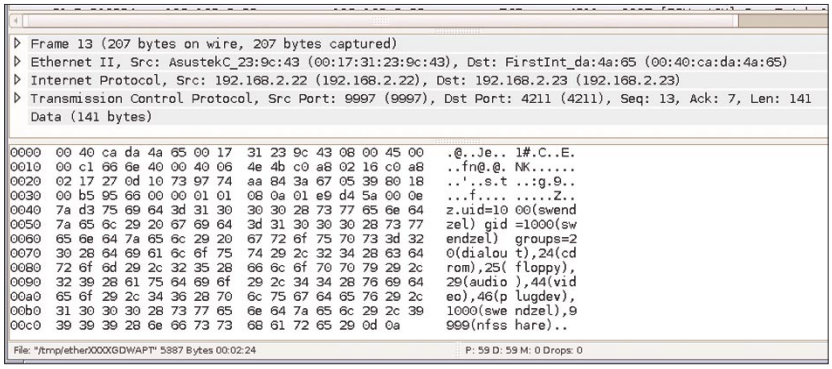

Abbildung 1. *Das gesniffte Paket in Wireshark*

Überzeugen Sie sich, wieviel wir für Sie machen können

Unsere Zeitschriften bilden die beste und billigste Zugangsplattform zu fortgeschrittenen Benutzern von IT-Technologien.

Große Auswahl der Themen der Magazine: vom Programmieren, über die Sicherheit, Webdesign, bis zum Nutzen der Linux-Systeme, garantieren Ihnen die Möglichkeit einer optimalen Selektion der Zielgruppe.

Die Veröffentlichung in 7 Sprachen und Zugänglichkeit der Magazine in ganz Europa ermöglichen die Führung präziser, lokalen Werbeaktionen und einfache Vorbereitung einer großen europäischen Werbekampagne.

Rufen Sie uns noch heute an (**+48 22 887 14 57**) oder schicken Sie uns eine E-Mail (**adv@software.com.pl**). Unser Konsultant wird für Sie ein optimales, individuelles und Ihren Forderungen entsprechendes Angebot erstellen.

Software-Wydawnictwo Sp. z o.o. ist der Herausgeber folgender Magazine: Software Developer's Journal , Linux+, PHP Solutions, hakin9, .PSD, Linux+Extra!, Software Developer's Journal Extra, Aurox Linux.

adv@software.com.pl

Eine weitere Möglichkeit, an diese Daten zu kommen ist, das Programm strings über den Memorydump laufen zu lassen und mit Tools wie grep die Ausgabe zu durchsuchen.

Netzwerk-Forensik

Um nun auch Netzwerkdaten forensisch zu analysieren gibt es verschiedene Maßnahmen. Man kann mit umfangreichen kommerziellen Softwaresystemen wie eTrust versuchen analysen durchzuführen, aber auch mit OpenSource-Software kann man alles erreichen!

Generell sind besonders Honeypots und Network-IDS wichtig um Daten zu sammeln, die im Fall der Fälle für eine forensische Analyse dienen sollen.

Praktisch jeder Angreifer wird verdächtige Spuren hinterlassen. Dazu zählen Header von eMails mit auffallendem Inhalt, DNS-Requests (besonders Version-Querys), HTTP HEAD-Requests, spezielle SMTP-Befehle wie EXPN oder VRFY, Pakete die Strings wie `uid=0(root)` beinhalten, gespoofte Pakete von Routing-Protokollen, diverse Scans usw. usf. Um die zu speichernde Datenmenge zumindest halbwegs klein zu halten, empfiehlt es sich, binäres logging zu verwenden, da viele Daten im erklärenden Plaintext-Format wesentlich mehr Platz benötigen als in ihrer tatsächlichen Übertragungsform.

Probleme

Leider gibt es bei der Forensik auch einige Probleme, die sich einem in den Weg stellen, hat man sich denn einmal vorgenommen, etwas herauszufinden.

Bei der Speicheranalyse kann man beispielsweise nur denn vernünftige Erfolge erziehlen, wenn man unmittelbar – nachdem ein Angriff statt fand – dass System untersucht, da wieder freigegebene Speicherbereiche, die die gesuchten Daten enthalten, oft schon nach kurzer Zeit vom gleichen Programm oder von anderen Programmen überschrieben werden.

Ein weiteres Problem ist, dass sich die Analyse der Speichermedien und des Speichers gegenseitig etwas ausgrenzen. Speichere ich einen Memorydump auf einer Festplatte des Systems, dann überschreibe ich eventuell im Dateisystem versteckte Daten. Während der Speicher auf die Platte geschrieben wird muss zudem Speicher für das Programm reserviert werden, dass den Speicher ausließt und schreibt, wodurch wiederum interessante Speicherbereiche überschrieben werden können.

Läd man hingegen Dateien in den Hauptspeicher, weil man sie sich mit einem Editor ansieht, so wird ebenfalls Speicher überschrieben und der Zustand des Systems wird verfällscht. Gar keine Veränderung zu Verursachen ist praktisch unmöglich. Es existieren immer irgendwelche Netzwerkverbindungen, Daemonprozesse, die etwas erledigen oder eingeloggte Benutzer, die irgendwelchen Speicher verbrauchen oder Daten auf den Speichermedien verändern.

Noch schlimmer kommt es, wenn das System nach einem erfolgreichen Angriff sicherheitshalber heruntergefahren wird, dann ist der Traum vom Memorydump dahin und auch auf der Platte werden die einen oder anderen Bytes verändert sein.

Ein weiteres Problem ist Swap-Speicher. Dieser lässt sich zwar mit etwas Glück noch nach einem Neustart auslesen, dafür überschreibt er aber eventuell versteckte Daten auf dem Speichermedium.

Zu guter Letzt könnte ein intelligenter Angreifer (oder ein Kiddie mit einem brauchbaren Programm) auch Daten im Speicher sowie die Netzwerkübertragung verschlüsseln. In diesem Fall muss zunächst ein kryptographischer Key aus dem Speicher gerettet werden oder eine Kryptoanalyse durchgeführt werden. Gute Programme können ihren eigenen Speicher, den sie für Netzwerkdaten allozieren, natürlich auch ganz einfach mit Zufallsdaten von /dev/random oder mit statischen Daten (etwa Nullen) überschreiben, wie das nachstehende Listing zeigt.

Damit nicht genug. Als verantwortlicher Administrator wird man versuchen ein wichtiges System nach einem Angriff wieder schnellstmöglich verfügbar zu machen und keine Zeit für forensische Maßnahmen investieren wollen. Wie gesagt: Das eine schließt das andere unter Umständen aus.

Zum Abschluss bleibt noch zu sagen, dass eine forensische Analyse – wenn sie ausserhalb des privaten Rechners erfolgt – immer gut dokumentiert werden sollte, um auch später alle getätigten Schritte nachvollziehen zu können. ●

Über den Autor

Steffen Wendzel beschäftigt sich seit vielen Jahren mit der Sicherheit von Unix-Systemen und TCP/IP-Netzwerken. Er entwickelte diverse OpenSource Software, ist Maintainer der Hardened Linux Distribution, Autor mehrerer Bücher zu den Themen Linux und Netzwerksicherheit, Security Consultant und Seminarleiter bei Plötner-IT sowie Student der Informatik an der FH-Kempten (Deutschland). Seine Webseite: *http://cdp.doomed-reality.org*.

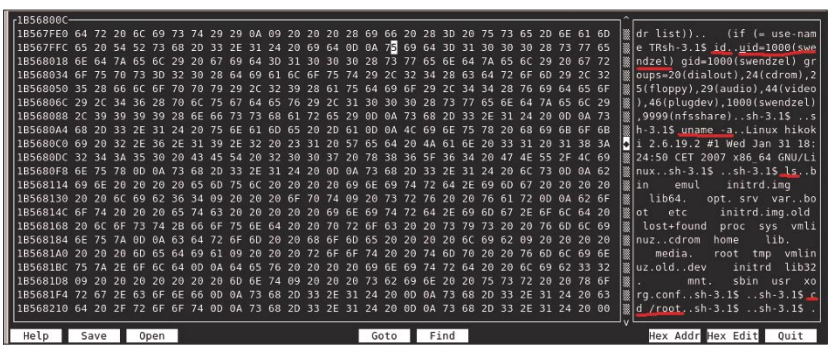

Abbildung 2. *Hexcurse mit geladenem Memorydump*