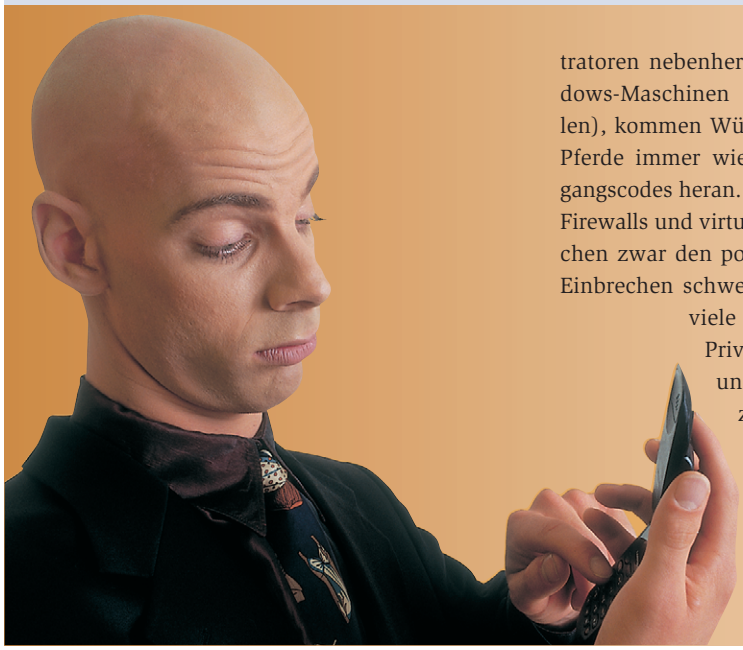
Aus dem Nähkästchen geplaudert: Benutzer zurückverfolgen

Who is who?

Admins müssen bei Bedarf schnell ermitteln, welcher User sich wann und von wo auf ihrem System angemeldet hat. Mit diesem Wissen schützen sie ihre Computer vor Missbrauch und können im Ernstfall die Eindringlinge zurückverfolgen. Die Kommandos »who«, »w« und »last« liefern die Daten. Marc André Selig



Computernetzwerke sind schon enorm praktisch. Um den Cluster an der Universität aus der Ferne zu warten, melde ich mich bequem über das Netz an und installiere ein Sendmail-Update vom heimischen Sofa aus. Mein unüberschaubares Spam-Aufkommen filtert ein Rechner in Pennsylvania, den ich ebenso selbstverständlich in meiner Wohnung programmiere.

Segen und Fluch der ständigen Erreichbarkeit

Leider gelten diese umfassenden Freiheiten auch für üble Zeitgenossen aller Art. Wer auf irgendeinem Wege die Zugangscodes eines Unix-Rechners ergattert hat – egal ob es sich dabei um einen Benutzernamen mit Passwort oder um einen kryptographischen Schlüssel handelt –, kann sich in der Regel von überall aus anmelden. Solange viele Unix-Administratoren nebenher auch noch auf Windows-Maschinen arbeiten (oder spielen), kommen Würmer und Trojanische Pferde immer wieder mal an ihre Zugangscodes heran.

Firewalls und virtuelle private Netze machen zwar den potenziellen Tätern das Einbrechen schwer, sie sind jedoch für viele kleine Firmen und Privatleute zu unhandlich und umständlich einzurichten. Umsichtige Admins untersuchen daher regelmäßig, wer sich auf ihren Rechnern alles tummelt und woher der Login kommt. So erkennen sie Unregelmäßigkeiten und können die Eindringlinge verjagen.

Wer online ist

Einen ersten Überblick der gerade angemeldeten Benutzer schafft der Befehl »who« (Abbildung 1). Er zeigt neben dem Benutzernamen und der Herkunft auch Datum und Uhrzeit des Logins sowie die (virtuelle) Konsole, an der der betreffende User arbeitet.

Auf diesem Computer sind gerade drei User angemeldet: »baier« arbeitet an der Konsole und hat einen X-Server (»:0«) sowie eine ganze Reihe zusätzlicher Terminals geöffnet (»pts/*X*«). Wahrscheinlich handelt es sich bei Letzteren um XTerm-Fenster. Neben »baier« ist noch der Account »wwwadm« zu sehen, der sich über die Maschine »zpidsu5.uni-trier.de« angemeldet hat, außerdem »mas«, von »acb6ae4b.ipt.aol.com« kommend, offenbar einem Wählzugang.

Mit dem Who-Befehl nahe verwandt ist »w« (Abbildung 2), das auf den meisten Linux-Systemen ergänzend zur Verfügung steht. Praktischerweise ist »w« kürzer zu tippen und zeigt obendrein die Uptime des Computers sowie die gerade bearbeitete Befehlszeile auf dem jeweiligen PTY (Pseudoterminal).

Wer online war

Die beiden Klassiker »who« und »w« zeigen immer die gerade angemeldeten Benutzer. Die sinnvolle Erweiterung ist »last« (Abbildung 3), es zeigt die letzten Logins. Die Liste geht zurück bis zum Beginn der jeweiligen Buchführung – da können durchaus tausende Zeilen zusammenkommen.

Gerade auf viel benutzten Rechner wäre diese Masse ohne zusätzliche Hilfe nur schwer zu beherrschen. Daher kennt »last« zwei wichtige Einschränkungen: Auf Wunsch beschränkt es die Liste auf einen einzelnen Benutzernamen oder manchmal auch auf ein Terminal (etwa »tty1« für die Linux-Konsole). Zweitens führt ein Parameter wie »-20« dazu, dass »last« nur die letzten (jüngsten) 20 Logins anzeigt. Beide Einschränkungen lassen sich kombinieren: »last -5 mas« listet die letzten fünf Logins des Users »mas« (Abbildung 3).

Ihre Informationen beziehen »who« und »last« aus speziellen Protokolldateien. Das File »utmp« verzeichnet die aktuell angemeldeten Benutzer und findet sich auf modernen Linux-Systemen im Verzeichnis »/var/run«. In »wtmp« protokolliert das System alle Logins und Logouts; diese Datei steht oft in »/var/log«. Die exakte Position beider Files ist von der Philosophie und dem Alter der Li-

nux-Distribution abhängig; relativ häufige Fundorte sind noch »/var/spool« und »/var/adm« sowie »/etc«.

In »wtmp« ist eine Liste aller An- und Abmeldungen (als Binärformat) zu finden. Beim Login an der Konsole schreibt eines der Systemprogramme »init«, »agetty« oder »login« den entsprechenden Datensatz. »init« hält auch den Logout im Protokoll fest, das sogar Reboots und ähnliche Ereignisse aufzeichnet.

Datenbasis: »wtmp« und »utmp«

Im Gegensatz dazu enthält »utmp« genau einen Datensatz für jeden Benutzer. Sie speichert nur den jeweils letzten Login. Je nach Aktualität der Libc sieht »utmp« eventuell sehr groß aus: Traditionell ist die Datei genau so lang, dass für alle möglichen numerischen Benutzerkennungen je ein Eintrag hineinpasst. Bei über 65000 erlaubten Usern ist das ziemlich viel Platz. Damit sie diesen Platz nicht sinnlos verschwendet, enthält die Datei Löcher: Die ungenutzten Abschnitte sind nur mit Nullen gefüllt, werden aber nicht auf Festplatte gespeichert. Dieser Trick beschleunigt auch die Zugriffe auf die Datei.

Programme sollten »wtmp« und »utmp« nicht direkt lesen oder beschreiben. Ordentliche Software greift nur über die Bibliotheksfunktionen »utmpname()«, »setutent()«, »getutent()« und »getutid« zu. Manche Unix-Varianten verwenden »wtmpx« und »utmpx« statt »wtmp« und »utmp«, um die Daten in einem erweiterten Format zu speichern. Das ist unter Linux unnötig, da die ursprünglichen Dateiformate bereits alle entsprechenden Anforderungen erfüllen.

Die Zugriffsrechte sind meist so gesetzt, dass jeder User die Datenbanken lesen darf, während nur ausgewählte Prozesse schreiben können. Schreibzugriff haben unter anderem »init« für Abmeldungen und Wechsel des Runlevels (einschließlich Booten und Herunterfahren), die »getty«-Prozesse und »sshd« für echte und virtuelle TTYs, »login« für eine erfolgreiche Anmeldung, »sessreg« für Logins an der grafischen Benutzeroberfläche mittels »xdm« & Co. sowie in vielen Umgebungen die grafischen Terminals wie »xterm«.

Meist ist es nicht nötig, komplizierte Benutzergruppen zu definieren, nur um die Zugriffe auf »utmp« und »wtmp« sauber zu regeln. Das für den Login benutzte Programm ergänzt die Datenbank noch bevor es seine Root-Rechte aufgibt und die Rechte des Users annimmt, der sich anmeldet.

```
Terminal <2>
[mas@sun3 mas]$ who
baier      console      Oct 13 07:36    (:0)
wwwadm     pts/1        Nov  3 07:39    (zpidsu5.uni-trier.de)
wwwadm     pts/2        Nov  3 07:40    (zpidsu5.uni-trier.de)
wwwadm     pts/3        Nov 10 10:11    (zpidsu5.uni-trier.de)
baier      pts/4        Nov  3 07:36    (:0.0)
baier      pts/5        Nov  3 07:36    (:0.0)
baier      pts/6        Nov  3 07:37    (:0.0)
baier      pts/7        Nov 10 10:08    (:0.0)
baier      pts/8        Nov 10 10:08    (:0.0)
mas        pts/9        Nov 10 20:09    (acb6ae4b.ipt.aol.com)
[mas@sun3 mas]$
```

Abbildung 1: Das »who«-Kommando listet alle derzeit angemeldeten Benutzer. Baier arbeitet am X-Server der Konsole (»:0«), während sich »wwwadm« und »mas« remote eingeloggt haben.

```
Terminal <2>
mas@ishi:/export/home/mas> w
 20:13:50 up  2:09,  3 users,  load average: 0.00, 0.01, 0.00
USER     TTY      LOGIN@   IDLE   JCPU   PCPU WHAT
mas      :0       18:07   ?xdm?   1:58   0.00s -:0
mas      pts/0    18:07    2:06m  0.00s  0.28s kdeinit: kwrited
mas      pts/1    18:07    0.00s  2.05s  0.00s w
mas@ishi:/export/home/mas>
```

Abbildung 2: Unter Linux informiert »w« noch etwas ausführlicher als »who«. Es zeigt in der ersten Zeile zusätzlich die Uptime des Rechners sowie dessen Last.

```
Terminal <2>
[mas@sun8 mas]$ last -5 mas
mas      pts/1        acb53cfc.ipt.aol Mon Nov 10 20:54   still logged in
mas      pts/3        acb0c055.ipt.aol Sun Nov  9 21:12 - 21:12  (00:00)
mas      pts/1        pd90249e0.dip.t- Tue Oct 14 20:15 - 20:38  (00:22)
mas      pts/1        pd9024626.dip.t- Tue Sep 30 18:54 - 18:54  (00:00)
mas      pts/1        pd902475d.dip.t- Tue Sep 30 13:38 - 14:50  (01:11)
[mas@sun8 mas]$
```

Abbildung 3: Wer sich für die Historie interessiert, ruft »last« auf. Das Tool listet die letzten Logins zusammen mit ihrem Ursprung und Zeitraum.

Noch mehr Protokolle

Die vorgestellten Datenbanken geben Aufschluss über zahlreiche Aktivitäten auf dem Rechner. Sie sind aber darauf angewiesen, dass jeder beteiligte Prozess sich an die Konventionen hält und die Dateien korrekt aktualisiert. Zudem müssen die gespeicherten Daten einer peinlich genau vordefinierten Struktur entsprechen. Beispielsweise ist die maximale Länge des Hostnamens, von dem aus eine Anmeldung erfolgt, meist empfindlich beschränkt. Abgekürzte Hostnamen wie »pd90249e0.dip.t-« aus Abbildung 3 sind nett, aber für forensische Zwecke beim Verfolgen eines Einbruchs unbrauchbar.

Zusätzliche Informationen sind manchmal unverzichtbar. Besonders bei komplizierten Anmeldevorgängen – etwa bei kryptographischer Authentifizierung – sind detaillierte Fehlermeldungen wichtig. Sie landen meist in einer zentralen Protokollstelle, dem so genannten Syslog. Darauf wird die nächste Folge des Workshops eingehen. *(fjl)*