Aus dem Nähkästchen geplaudert: Syslog

Computer-Logbuch

Unix-Systeme fassen Protokollmeldungen an einer zentralen Stelle zusammen. Das vereinfacht die Pflege und erleichtert die Fehlersuche – der Syslog-Dienst wird zum engen Verbündeten des Admin. Marc André Selig

Die meisten Programme produzieren zwei Sorten von Daten. Zum einen geben sie ihre normalen Ergebnisse aus, beispielsweise malt ein Spiel bewegte Bilder auf den Monitor und ein FTP-Client holt Daten vom Server. Während des Programmlaufs entstehen aber auch Meldungen über diesen Ablauf. Das Spiel könnte über die gelungene Initialisierung der Grafikkarte oder den fehlenden Joystick berichten. Der FTP-Client meldet einen erfolgreichen Verbindungsaufbau oder eine nicht gefundene Datei. Man unterscheidet also zwischen den produktiven Daten, die ein Programm ausgibt, und den diagnostischen Meldungen.

Interaktiv gestartete Software zeigt ihre Diagnosemeldungen direkt auf dem Bildschirm an, beispielsweise in Dialogboxen, mit denen moderne Office-Pakete gern um sich werfen. Die unzähligen Daemons, die im Hintergrund als Server ihre Dienste verrichten, können aber schlecht auf den Bildschirm schreiben. Diese Meldungen würden den Benutzer an der Konsole nur stören, zumal sie mit seiner eigentlichen Arbeit kaum etwas zu tun haben. Schlimmer noch: Wenn kein Mensch vor dem Monitor sitzt, gehen die Nachrichten schlicht verloren.

Für Abhilfe sorgen die Protokolldateien. Wenn ein im Hintergrund laufendes Programm irgendeine diagnostische Meldung ausgeben möchte, schreibt es sie tunlichst in ein File. Das war auf Servern schon immer so und selbst Betriebssysteme wie Windows kennen rudimentäre Logdateien.

Wenig effizient wäre es dagegen, wenn jedes kleine Tool seine eigene Protokolldatei pflegen müsste. Die zahllosen offenen Files würden unnütz Ressourcen beanspruchen. Außerdem: Wer wollte vorschreiben, wo die Dateien stehen? Es könnte leicht zum Chaos kommen, wenn jeder Programmierer sich seinen eigenen Ablageort für Logs wählen würde.

Das Syslog

Praktisch alle Unix-Systeme kennen das Syslog als effektive und praktische Lösung. Protokollmeldungen landen nicht in Dateien, sondern über eine besondere Bibliotheksfunktion bei einem zentralen Daemon. Der sortiert die Protokolleinträge und entscheidet, was mit ihnen passieren soll. Dazu benutzt er zwei Kriterien: die Priorität und die Herkunft der Meldungen.

Manche Texte sind so wichtig und kritisch, dass Syslog jeden angemeldeten Benutzer sofort informiert. Wenn beispielsweise im Laptop der Akku leer läuft, sollte der User das sofort wissen, auch oder gerade dann, wenn er im Moment mit einem Editor arbeitet. Andererseits interessieren ihn Statistiken über die Auslastung des lokalen DNS-Cache nur am Rande.

Ähnlich nützlich ist die Einteilung der Meldungen nach ihrer Herkunft. Viele Admins fassen die Meldungen über ankommende und verschickte E-Mails in einer Datei zusammen. So können sie später Statistiken erstellen oder den Verbleib vermisster Mails nachprüfen. Manche Meldungen sind zudem vertraulich: Die Fehlersuche in einem Authentifizierungsmodul oder im PPP-Daemon könnte Passwörter im Klartext hervorbringen. Derartige Logs müssen strenger geschützt werden als die Zugriffsstatistiken des Webservers. Gut, dass sie in eigenen Dateien abgelegt sind.

Regel und Ausnahme

Zu jeder Regel gehören Ausnahmen, so ist das auch mit der zentralen Protokollierung unter Unix. Abbildung 1 fasst die wichtigsten Mechanismen und Ausnahmen zusammen: Die meisten Programme schicken Meldungen an den Syslog-Daemon »syslogd«, der sie sortiert und verteilt. Meldungen vom Kernel gehen stattdessen an den Kernel-Logdaemon »klogd«. Dieser leitet sie typischerweise an das Syslog weiter. Programme mit besonders intensivem Protokollaufwand schreiben weiterhin in eigene Dateien, ein gutes Beispiel für diese Gruppe ist Apache.

Bei einer modernen Distribution finden sich die Protokolldateien des Syslog üblicherweise im Verzeichnis »/var/log«.

Das ist allerdings frei konfigurierbar. Listing 1 zeigt typische Beispiele für Syslog-Meldungen. Das Format der Meldungen ist im Prinzip immer gleich. Nach einem Zeitstempel mit Datum und Uhrzeit steht der Computername – hier »undine« – gefolgt von der eigentlichen Meldung. Diese wiederum zeigt zunächst den Namen des Programms, das die Meldung produziert hat, häufig gemeinsam mit der Prozessnummer, die in eckigen Klammern steht.

Meldungen aus mehreren Quellen

In Listing 1 stammt die erste Meldung vom Kernel. Der Computer mit WLAN-Karte war zu weit vom Access Point entfernt. Wie in Abbildung 1 skizziert ist, hat der Kernel-Logdaemon diese Mitteilung an den Syslog-Daemon weitergereicht. Das Problem, das zu dieser Meldung führte, bestand über mehrere Sekunden hinweg; in dieser Zeit hat der Kernel eine ganze Reihe identischer Warnungen generiert. Syslog erkennt deren Gleichheit und spart Platz: Es notiert nicht alle Meldungen einzeln, sondern teilt einige Sekunden später nur lapidar mit, dass sich die Meldung mehrfach wiederholt hat.

Die dritte und die vierte Zeile enthalten Mitteilungen von Programmen. Hier hat der Cron-Daemon einen Befehl selbstständig gestartet und der Nutzer »mas« hat sich über das Utility »su« Root-Rechte verschafft. Die einzelnen Programme sind für das Format der Meldung selbst verantwortlich: Cron schreit in Großbuchstaben, gibt dafür aber auch seine Prozesskennung an. »su« ist dagegen etwas dezenter.

In der untersten Zeile ist ein Lebenszeichen von »syslogd« zu sehen. Derartige Marker schreibt der Dienst in regelmäßigen Abständen, wenn sonst keine bedeutende Aktivität anfällt. Manchmal hilft das bei forensischen Untersuchungen, wenn man wissen möchte, wie lange ein System noch normal gelaufen ist, bevor es zu einem Absturz kam. Glücklicherweise sind Abstürze unter Linux extrem selten und die Mark-Funktionalität bleibt daher oft ausgeschaltet.

Konfiguration

Sehr Praktisch an Syslog ist seine freie Konfigurierbarkeit. Nicht irgendein Programmierer, sondern der Admin entscheidet, welche Protokolle wo zu stehen haben. Dazu dient die zentrale Konfigurationsdatei »/etc/syslog.conf«:

```
*.*             /dev/tty8
mail.info       /var/log/mail
*.*;mail.none   -/var/log/messages
*.crit          @loghost.zpid.de
```

Die Konfiguration ordnet einer Nachrichtenquelle (links) ein Protokollziel (rechts) zu. Die Quelle besteht aus einer so genannten Facility, also dem funktionellen Bereich des Systems, aus dem die Meldung stammt, sowie aus der Priorität (durch einen Punkt abgetrennt). Facilities sind neben dem hier erwähnten »mail« auch »news«, »ftp«, »auth« oder »kern« und andere. Die Prioritäten lauten in aufsteigender Reihenfolge »debug«, »info«, »notice«, »warning«, »err«, »crit«, »alert« und »emerg«.

Die minimalistische »syslog. conf« aus obigem Beispiel schickt alle Meldungen auf die virtuelle Konsole »/dev/tty8«. Dort kann man sie sich mit [Ctrl] + [F8| jederzeit ansehen. Die Aktivitäten des Mailsystems werden – mit Ausnahme von Debugging-Meldungen – in der separaten Datei »/var/log/mail« festgehalten; alle übrigen Meldungen gehen nach »/var/log/messages«.

Die letzte Zeile sorgt als Besonderheit dafür, dass Syslog alle kritischen Meldungen an den Syslog-Daemon auf dem Computer »loghost.zpid.de« weiterleitet. So bleiben sie der Nachwelt erhalten, selbst wenn dieser Rechner unmittelbar nach der Meldung abstürzt. Wer in einem Cluster alle Syslog-Meldungen an ein zentrales System weiterleitet, kann über die Syslog-Konfigurationsdatei auf diesem Zielrechner die gesamte Diagnostik des Clusters steuern.

Das Minuszeichen vor dem Dateinamen in der dritten Zeile ist ein kleiner Tuning-Trick. Normalerweise zwingt »syslogd« das System dazu, jede Meldung sofort auf die Festplatte zu schreiben. Fallen sehr viele Meldungen an, ist die Festplatte fast nur noch mit Syslog beschäftigt. Dateien mit dem Minuszeichen verzichten auf diesen Aufwand und setzen dafür den unter Linux üblichen Cache-Mechanismus ein. So bleiben die Meldungen noch bis zu 30 Sekunden im Hauptspeicher und das System wird spürbar entlastet.

Änderungen an »syslog.conf« muss ein Admin seinem System mitteilen: Entweder durch den Befehl »/etc/init.d/syslog reload« oder, falls die Distribution das nicht unterstützt, durch das Signal »HUP« (Hangup):

```
# ps ax | grep syslog
  442 ?     S   0:00  /sbin/syslogd
# kill -HUP 442
```

Das ist einer von vielen Wegen, wie Programme unter Unix miteinander sprechen können. Was es damit genau auf sich hat, untersucht eine spätere Folge dieser Serie. *(fjl)*

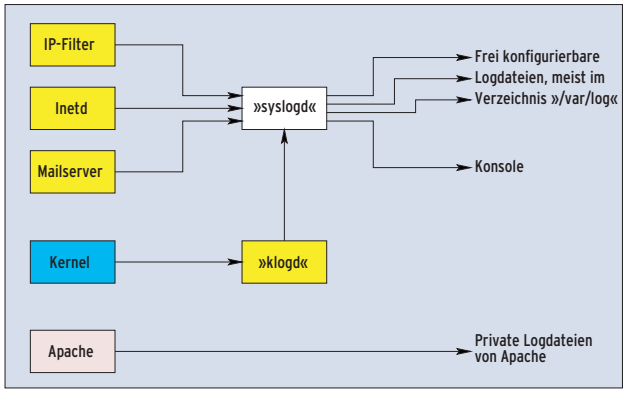

Abbildung 1: Unter Linux protokollieren die meisten Programme direkt an den zentralen Syslog-Daemon, der Kernel bedient sich dazu des »klogd«. Syslog schreibt die Protokolle dann in Files oder sendet sie an andere Rechner.

Listing 1: Syslog-Meldungen

```
01 Dec  8 21:50:21 undine kernel: Tx error occurred (error 0x10)!! (maybe
   distance too high?)
02 Dec  8 21:50:28 undine last message repeated 36 times
03 Dec  8 21:59:00 undine /USR/SBIN/CRON[1730]: (root) CMD ( rm -f
   /var/spool/cron/lastrun/cron.hourly)
04 Dec  8 22:10:06 undine su: (to root) mas on /dev/pts/0
05 Dec  8 22:29:18 undine -- MARK --
```