Systeme mit dem Handy überwachen und konfigurieren

Fernsteuerung

Mit mobilen Geräten ist der Administrator nicht länger an seinen Arbeitsplatz gefesselt. Monitoring-Software schickt ihm Alarmmeldungen aufs Handy oder nach Hause, von wo er deren Ursachen mit den passenden Programmen auch gleich nachgehen kann. Juliet Kemp

Als Systemadministrator kann man nicht immer nur an seinem Stuhl kleben. Es sollte auch möglich sein, mal von zu Hause aus zu arbeiten oder eine interessante Konferenz zu besuchen. Gleichzeitig sollen aber die betreuten Server nicht unbeaufsichtigt bleiben. Auch wer sich jeden Tag ins Büro begibt, schläft meistens nachts daheim – oder ist tagsüber auch mal in einem anderen Gebäudeteil unterwegs, um ein Kabel zu verlegen oder einen Rechner zu rebooten.

Dieser Artikel stellt Monitoring-Lösungen vor, die Alarmmeldungen an Mobilgeräte versenden, und außerdem die passende Software dafür, vom Smartphone aus die passenden Reparaturmaßnahmen am Server einzuleiten.

Zum Monitoring von Servern und der Netzwerk-Infrastruktur gibt es eine ganze Reihe an Softwarepaketen, die Alarmmeldungen verschicken, wenn etwas schiefgeht. Wer sie so konfiguriert, dass sie E-Mails oder SMS verschickt, wird auch unterwegs über alle Probleme benachrichtigt.

Monitoring

Dieser Workshop erklärt, wie Sie E-Mail-Alerts mit den beiden populären Monitoring-Paketen Nagios [1] und Hobbit [2] (jetzt unter dem neuen Namen Xymon) einrichten. Er setzt dabei voraus, dass die beiden Pakete schon installiert und konfiguriert sind. Verschiedene Artikel zu Nagios sind beispielsweise schon in früheren Ausgaben des ADMIN-Magazins erschienen.

Für Nagios richten Sie zuerst eine Kontaktgruppe namens »admins« in der Datei »contacts_nagios2.cfg« ein. Wenn die Admins hier definiert sind, können Sie sie einfach in den anderen Konfigurationsdateien verwenden. Neue Administratoren hinzufügen oder alte löschen geschieht dann ebenfalls an dieser einen Stelle. Hier muss also ein Eintrag mit Ihrer E-Mail-Adresse existieren.

Als Nächstes müssen Sie sich überlegen, für welche Dienste Sie E-Mail empfangen wollen. Die einfachste Option ist, E-Mails für alle Dienste zu aktivieren, indem Sie die allgemeine Servicekonfiguration in

»conf.d/generic-service_nagios2.cfg« bearbeiten. Fügen Sie dort die Zeilen aus Listing 1 hinzu.

Das Notification-Intervall gibt in Minuten an, wie oft Sie benachrichtigt werden, im Beispiel alle 24 Stunden. Der Wert »check_period« definiert, wann der Service laufen soll. Intervalle wie »24x7« sind in »conf.d/timeperiods_nagios2.cfg« definiert. Die Werte für »normal_check_interval« und »retry_check_interval« sind in Minuten angegeben.

Der Dienst im Beispiel soll alle 5 Minuten überprüft werden; wenn der erste Check fehlschlägt, soll Nagios es aber jede Minute wieder versuchen. Das Monitoring-System probiert es dann bis zu 10-mal wieder, danach gilt der Service als fehlerhaft. Natürlich können Sie diesen Wert auch niedriger ansetzen, aber dann häufen sich möglicherweise die Fehlarme (False Positives).

Die »notification_period« gibt den Zeitraum an, in dem Nagios Benachrichtigungen verschickt, die »notification_options« bestimmten, für welche Ereignisse es dies tut. Bei Hosts steht »d« für den Zustand »DOWN«, »u« für »UNREACHABLE«. Ein »r« zeigt an, dass sich der Host wieder erholt hat (Recovery), ein »f«, dass er sich nicht recht entscheiden kann (Flapping). Bei Diensten steht »w« für ein WARNING, »u« für unbekannten Zustand, »c« für einen kritischen Zustand (Critical), »r« bedeutet Recovery und »f« ist wiederum Flapping. In der letzten Zeile von Listing 1 gibt »contact_groups« schließlich an, welche Nagios-Gruppen die Benachrichtigungen bekommen sollen.

Wenn Sie das alles eingerichtet haben, stoppen Sie auf einem der überwachten Rechner den SSH-Dienst: Sie sollten mit dem eingerichteten E-Mail-Account eine entsprechende Benachrichtigung erhalten. Wenn Sie dann den SSH-Service wieder aktivieren, bekommen Sie wieder eine Nachricht.

Wenn verschiedene Administratoren zu unterschiedlichen Zeiten benachrichtigt werden sollen, lässt sich das am besten in deren Contact-Definitionen in »contacts_nagios2.cfg« einstellen. Listing 2 zeigt eine Definition, die eine private E-Mail-Adresse fürs Wochenende verwendet. Weil hier die Zeitangabe »weekend« vorkommt, muss diese wiederum in »conf.d/timeperiods_nagios2.cfg« definiert sein (Listing 3). Eine »timeperiod«-Definition kann sich auf andere Timeperiods beziehen und bestimmte Zeiträume ein- oder ausschließen.

Mit den passenden Zeit- und Kontaktdefinitionen kann »juliet_personal« nun in anderen Dateien als Kontakt verwendet werden, bekommt aber eben nur an Wochenenden Benachrichtigungen.

Hobbit

In Hobbit stellen Sie E-Mail-Benachrichtigungen über die Datei »/etc/hobbit/hobbit-alerts.cfg« ein. Alarmmeldungen lassen sich recht frei definieren, dabei schreiben Sie in der Konfigurationsdatei zuerst die eingetretene Bedingung auf und dann die auszuführende Aktion. Listing 4 stellt ein, dass Hobbit eine E-Mail verschickt, wenn irgendein Host (die dazu gehörige Regular Expression muss mit »%« beginnen) 30 Minuten lang (»DURATION«) nicht mehr erreichbar ist. Das wiederholt Hobbit alle 24 Stunden (»1440« Minuten) und es verschickt auch eine E-Mail, wenn sich der Host wieder erholt (»RECOVERED«).

Die meisten Parameter stecken in der Variablen »$MAILADMIN«, die meist für Einstellungen verwendet werden, die man mehr als einmal braucht. In einer spezifischen Zeile, die diese Variable verwendet, kann der Administrator die Defaults auch überschreiben.

Wie bei Nagios lassen sich für ähnliche Ereignisse unterschiedliche Aktionen konfigurieren. Zum Beispiel könnten Sie zu einer bestimmten Zeit eine Benachrichtigung an eine andere Adresse schicken lassen, etwa wie in Listing 5. Wichtig ist hierbei die »TIME«-Einstellung. Wenn sie nicht näher spezifiziert ist, bedeutet das „immer“. Der zweite Alarm in Listing 5 ist so konfiguriert, dass er nur zwischen 18 Uhr (»1800«) und 8 Uhr morgens (»800«) an einem beliebigen Tag der Woche (»*«) ausgelöst wird. Für Wochentage verwenden Sie ein »W«, für die einzelnen Tage die Zahlenwerte 0 (Sonntag) bis 6 (Samstag).

Alternativ können Sie auch zu unterschiedlichen Gelegenheiten verschiedene Verantwortliche kontaktieren lassen. Zur Differenzierung nach Diensten verwenden Sie dann den »SERVICE«-Parameter wie in Listing 6 in der »MAIL«-Zeile statt in der ersten Bedingungs-Zeile. Hier wird der Alarm jede Stunde wiederholt, wobei der Administrator eine Benachrichtigung bei Problemen mit der Festplatte und SSH bekommt. Der Webmaster dagegen wird unterrichtet, wenn der HTTP-Service sich querstellt.

Hobbit unterstützt zusätzlich zu dem »MAIL«-Schlüsselwort auch noch »SCRIPT«, mit dem sich beliebige Skripte ausführen lassen.

Listing 1: Nagios-Setup

```
01 notification_interval          1440
02 is_volatile                    0
03 check_period                   24x7
04 normal_check_interval          5
05 retry_check_interval           1
06 max_check_attempts             10
07 notification_period            24x7
08 notification_options           c,r
09 contact_groups                 admins
```

Listing 2: Benachrichtigung privat

```
01 define contact {
02     contact_name    juliet-personal
03     .....
04     host_notification_period    weekend
05     service_notification_period    weekend
06 }
```

Listing 3: Wochenende

```
01 define timeperiod {
02     name    weekend
03     timeperiod_name    weekend
04     friday          18:00-24:00
05     saturday       00:00-24:00
06     sunday          00:00-24:00
07     monday          00:00-09:00
08 }
```

Listing 4: Host nicht erreichbar

```
01 $MAILADMIN=MAIL admin@example.com REPEAT=1440
   RECOVERED
02
03 HOST=%.* SERVICE=conn
04     $MAILADMIN DURATION>30
```

Listing 5: Mehrere Kontakte für einen Dienst

```
01 $MAILADMIN=MAIL admin@example.com REPEAT=1440
   RECOVERED
02 $ONCALLADMIN=MAIL on-call@example.com REPEAT=1440
   RECOVERED TIME=*:1800:0800
03
04 HOST=%.* SERVICE=ssh
05          $MAILADMIN DURATION>30
06          $ONCALLADMIN DURATION>30
```

Alternativen zu Hobbit und Nagios gibt es viele, und die meisten bieten ähnliche, flexible Möglichkeiten an, um Benachrichtigungen über E-Mail oder andere Kanäle wie SMS zu verschicken. Nagios und Hobbit eignen sich aber besonders gut für große Netzwerke.

Mobil

Der nächste Schritt besteht darin, nicht nur zu erfahren, dass etwas kaputtgegangen ist, sondern es auch zu reparieren, auch wenn Sie gerade nicht neben dem Server sitzen. Glücklicherweise besitzen Smartphones heute eine Vielzahl an Features, mit denen sich viel anstellen lässt, solange Sie über eine funktionierende Datenverbindung verfügen.

Eines der nützlichsten Tools des umherschweifenden Sysops ist die Secure Shell (SSH). Für die meisten Smartphones gibt es einige SSH-Implementierungen, zum Beispiel die folgenden:

- Für Android-Telefone bietet sich Connectbot an, das Sie auf dessen Homepage [3] oder im Android Market finden. Es unterstützt nicht nur Passwort-Authentifizierung, sondern auch Public/Private Keys.
- iPhone-User sollten sich iSSH näher ansehen (Abbildung 1). Es kostet knapp 8 Euro und bietet einen Terminalemulator, SSH und sogar VNC-Verbindungen. Es unterstützt Schlüsselmanagement und exotische Tastenkombinationen, sodass sogar Emacs- und Vim-Anwender damit arbeiten können. Verbindungen und Konfigurationen lassen sich abspeichern. Das Scrollen des Bildschirms ist ein bisschen irritierend, aber man kann sich daran gewöhnen. Alternativen zu iSSH sind Touchterm oder P-Term, aber keines der beiden Programme unterstützt Cut & Paste oder mehrere gleichzeitige Verbindungen.

Listing 6: Kontakte für mehrere Dienste

```
01 $SSHADMIN=MAIL admin@example.com SERVICE=disk,ssh
   RECOVERED
02 $WEBADMIN=MAIL webmaster@example.com SERVICE=http
   RECOVERED
03
04 HOST=webserver.example.com
05         $SSHADMIN DURATION>30 REPEAT=1h
06         $WEBADMIN DURATION>30 REPEAT=1h
```

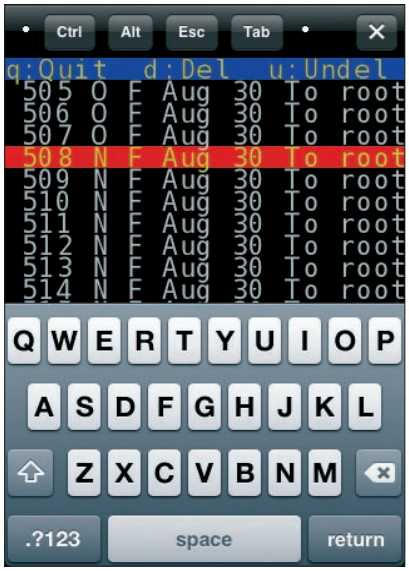

Abbildung 1: Der SSH-Client iSSH für das iPhone bietet auch einen Zoom-Modus für allzu kleine Terminal-Schriften.

Auch iPad-Besitzer können iSSH verwenden, sogar im Vollbildmodus. Es gibt noch ein paar kleinere Bugs, über die [4] mehr verrät.

- PSSH [5] unterstützt auf Palm OS 5 das SSH2-Protokoll, wer nur SSH1 braucht, kann auf Palm OS 4 oder 5 Tu-SSH verwenden [6]. PSSH warnt den Anwender davor, es für sicherheitskritische Anwendungen zu verwenden, weil es keine Hardware-unterstützten Zufallszahlen verwendet. Dafür hat es eine praktische On-Screen-Tastatur und unterstützt schlüsselbasierte Authentifizierung.
- Blackberry-Anwender können das Java-Programm Midp-SSH [7] verwenden, das auch auf anderen Java-basierten Mobilgeräten funktionieren sollte. Es bringt ein Textvorhersage-Feature mit, das vor allem auf Geräten praktisch ist, die kein vernünftiges Keyboard besitzen. Auch der eingebaute Makro-Support hilft bei der Eingabe häufig verwendeter Schlüsselwörter. Das Programm unterstützt SSH-Schlüssel, allerdings lässt sich keine Passphrase eingeben, was die Nützlichkeit des Feature ziemlich einschränkt.
- Für Symbian gibt es den von Windows bekannten SSH-Client Putty [8]. Er unterstützt zwar SSH-Schlüssel, aber nur solche, die mit dem Programm Puttygen unter Windows erzeugt wurden. Putty bringt hervorragende Dokumentation mit, die sich auch auf der Website findet.

Alternativ zu SSH können Sie sich auch per VNC verbinden, wenn Ihr Server das unterstützt. Mit weniger Bandbreite ist SSH aber die bessere Wahl. Wer sich allerdings gerade mit einem Programm herumärgern muss, das sich nur über eine grafische Oberfläche konfigurieren lässt, findet VNC vermutlich recht praktisch. Auf dem iPhone bietet iSSH einen guten VNC-Client, für Android-Telefone gibt es den Android-VNC-Viewer [9].

Fazit

Möglichkeiten, den eigenen Server von unterwegs zu überwachen und steuern, gibt es viele, egal ob man sich vom Monitoringsystem per E-Mail oder SMS benachrichtigen lässt. SSH-Clients für die wichtigsten Handy-Plattformen erlauben es dem Administrator, sich auf dem kaputten System einzuloggen und das Problem zu beheben. Wer es organisatorisch klug einrichtet, gewinnt dadurch zumindest Bewegungsfreiheit, ohne gleichzeitig sieben Tage pro Woche verfügbar sein zu müssen. *(ofr)* ■

Infos

[1] Nagios: [http://www.nagios.org]
[2] Hobbit: [http://hobbitmon.sourceforge.net]
[3] Connectbot: [http://code.google.com/p/connectbot/]
[4] iSSH für iPad: [http://www.zinger-soft.com/support_p_1.html]
[5] PSSH – SSH2 für Palm OS 5: [http://www.sealiesoftware.com/pssh]
[6] Tu-SSH – SSH1 für Palm OS 4 und 5: [http://www.tussh.com]
[7] Midp-SSH: [http://www.xk72.com/midpssh]
[8] Putty: [http://www.chiark.greenend.org.uk/~sgtatham/putty]
[9] Android-VNC-Viewer: [http://code.google.com/p/android-vnc-viewer/]

Die Autorin

Juliet Kemp beschäftigt sich mit Linux, seit sie gemerkt hat, dass es mehr Spaß macht als Abschlussprüfungen. Sie arbeitet seit über zehn Jahren als Systemadministratorin und hat das Buch „Linux Systems Administration Recipes: A Problem-Solution Approach" geschrieben.