SMB Traffic Analyzer SMBTA

Tanzlehrer

Der SMB Traffic Analyzer ist ein VFS-Modul für Samba 3.6, das eine Echtzeitanalyse des Datendurchsatzes im SMB-Netzwerk ermöglicht. Es bringt eigene Analyse-Tools mit oder bereitet die gewonnenen Daten mit RRD-Tool grafisch auf. Thomas Drilling

Der SMB Traffic Analyzer ist ein als VFS-Modul realisiertes Tool, das einen Samba-/CIFS-Server in die Lage versetzt, Traffic-Statistiken im Samba-Netzwerk aufzuzeichnen. Der SMBTA-Daemon (SMBTAD) speichert diese in einer Datenbank und macht sie unter anderem via SQL verfügbar. Die Architektur ist übersichtlich und besteht aus einem Modul im Samba-VFS (Virtual File System), einem Daemon sowie einem Set von Client-Tools (»smbtatools«) zum Auswerten und Visualisieren. SMBTA nutzt eine vorhandene SQL-Datenbank (»sqlite3«) zum Speichern der Traffic-Daten.

SMBTA-Entwickler und Erfinder Holger Hettrich von Novell arbeitet seit der SambaXP-Conference 2007 in Göttingen an seinem Samba Traffic Analyzer und konnte bisher in Fachkreisen, etwa beim Samba-Team, viel Aufmerksamkeit erregen, was ihm eine Reihe von gut besuchten Vorträgen etwa auf der Samba Xperience und bei anderen Gelegenheiten in Deutschland und den USA bescherte. Inzwischen engagiert sich sogar Arbeitgeber Novell insofern, dass Hettrich einen Teil seiner Arbeitszeit auf SMBTA verwenden kann.

Leider fehlt es in der Öffentlichkeit noch am breit angelegten Interesse, was aber bisher primär daran lag, dass jegliches Nutzer-Feedback direkt von und über Holger Hettich abgewickelt wurde und nicht über die passenden Mailinglisten oder sonstige Kanäle, was schlicht eine Frage gekonnter Öffentlichkeitsarbeit ist. Deshalb ist auch die Anzahl an Referenz-Nutzern derzeit noch überschaubar. Das dürfte sich aber mit der kommenden Samba-Version 3.6 ändern, die den SMB Traffic Analyzer als offiziellen Bestandteil von Samba enthalten wird [1].

Spezialisiert

Zwar könnten versierte Admins auch mit einem gewöhnlichen Portsniffer ausgewählte Schnittstellen etwa auf typischen Netbios-Traffic abhören, SMBTA konzentriert sich aber ausschließlich auf Samba-Traffic und ermöglicht das Erstellen umfangreicher und aussagekräftiger Statistiken, weil das Tool quasi echtes Datamining betreibt, in dessen Zentrum die Sqlite-Datenbank steht. So ist es etwa mit SMBTA möglich, Analysen gezielt auf

einzelne Benutzer, Shares oder Dateien zu beschränken. Auf Client-Seite stehen im Prinzip drei Tools zum Auswerten zur Verfügung, die Bestandteil des Pakets »smbtatools« sind. Zum einem ermöglicht ein RRD-Treiber mithilfe von RRD-Tool (Round Robin Database [2]), sich in Echtzeit an den SMBTA-Daemon zu hängen, was die grafische Aufbereitung der Traffic-Daten oder deren Weiterverarbeitung etwa in Perl-Skripten ermöglicht, wahlweise via IP oder Unix-Domain-Sockets. Das Werkzeug »smbtaquery« kann die Datenbank via XML auslesen. Wer eher auf visuelle Effekte steht, kann mit dem »smbtamonitor« auch ganz ohne SQL-Kenntnisse auf die gespeicherten Daten zugreifen.

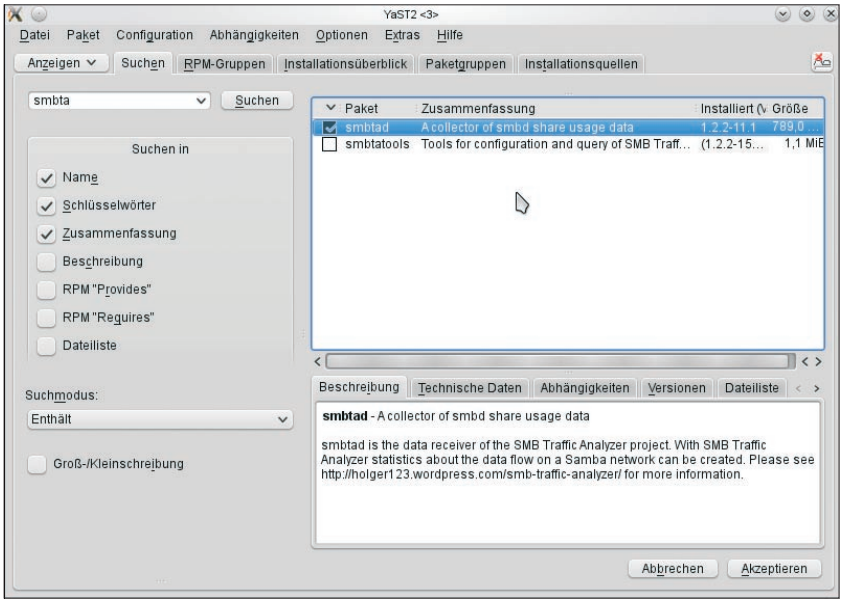

Abbildung 2: Open-Suse-Nutzer können sowohl SMBTA als auch die SMBTA-Tools direkt aus den angegebenen Repositories installieren.

Bezugsquellen

Wer SMBTA selbst einsetzen möchte, kann entweder die Sourcen der aktuellen Version 1.1.2 von [3] herunterladen oder sich von Holger Hettrichs Wordpress-Blog Binaries für Open Suse besorgen [4]. Allgemeine Informationen zu SMBTA finden sich unter [5]. Aufgrund der Tatsache, dass das Installieren von SMBTA Backports auf Samba 3.6 erfordert, fahren Anwender mit dem RPM-Binary oder dem One-Click-Installer für Open Suse 11.3 am besten. Selbstverständlich ist auch das Auschecken von SMBTA per Git möglich [6].

Last but not least können Suse-Nutzer auch die Paketquelle [7] in Yast einbinden und SMBTA mit dem Paketmanager installieren (Abbildung 1 und 2). Noch stressfreier gelingt das Ausprobieren von SMBTA mit der von Hettrich geschnürten Applicance „Stresstest“, die aktuell in der Version 0.0.2 vorliegt [8] (siehe Kasten).

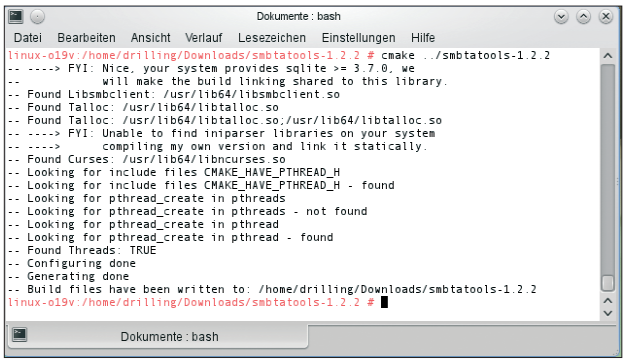

Abbildung 1: Der SMBTA-Daemon lässt sich relativ einfach aus den Quellen übersetzen oder über das angegebene Repository als Binary mit Yast installieren.

Wer SMBTA schnellstmöglich ausprobieren, dazu aber keinen dedizierten Samba-Server aufsetzen möchte, muss SMBTA und die »smbtatools« aus den Quellen bauen. Damit das klappt, müssen »cmake«, »libsmbclient-devel«, »libtalloc-devel« und »ncures-devel« installiert sein. Außerdem müssen die Datenbankumgebung Sqlite3 und die zugehörigen Devel-Pakete installiert sein. Schließlich ist zu prüfen, ob »libxslt« installiert ist, was bei Open Suse per Default der Fall sein sollte. Nun entpackt man zunächst die Sourcen »smbtatools-1.2.2.tar.bz2« und wechselt in das entstandene Verzeichnis. Der passende Aufruf von »cmake« ausgehend vom Build-Verzeichnis konfiguriert das Paket für die Übersetzung:

```
cmake ../
smbtatools-1.2.2
```

»make« und »make install« kompilieren das Paket und kopieren die Programme in den passenden Ort.

Zum Starten des Daemons genügt »smtad -u -n«, womit Daemon (»u«) und Client (»n«) über Unix Domain Sockets kommunizieren. Der SMBTA-Deamon hat primär die Aufgabe, die SQL-Datenbank mit den Daten zu füttern, die er vom VFS-Modul empfängt. Außerdem ist er für das Abhandeln sämtlicher Client-Anfragen an die Datenbank verantwortlich. Möchte der Admin nun jeglichen Traffic auf einer ausgewählten Freigabe aufzeichnen, muss er lediglich in der betreffenden Share-Definition das VFS-Modul wie folgt laden.

```
vfs objects = smb_traffic_analyzer
smb_traffic_analyzer:protocol_version = v2
smb_traffic_analyzer:mode = unix_domain_
socket
```

wobei sich der letzte Parameter vom Administrator an die eigenen Bedürfnisse anpassen lässt (siehe auch Abbildung 3). Soll die Kommunikation zum Beispiel über TCP/IP laufen, sähe eine entsprechende Share-Definition etwa so aus:

```
vfs objects = smb_traffic_analyzer
smb_traffic_analyzer:protocol_version = v2
smb_traffic_analyzer:host = localhost
smb_traffic_analyzer:port = 3490
```

Der SMBTAD-Daemon wäre in diesem Fall mit

```
smbtad -i 3490 -p 3491
```

zu starten und wartet damit auf Anfragen via Port 3490 an das VFS-Modul und behandelt Client-Anfragen auf Port 3491.

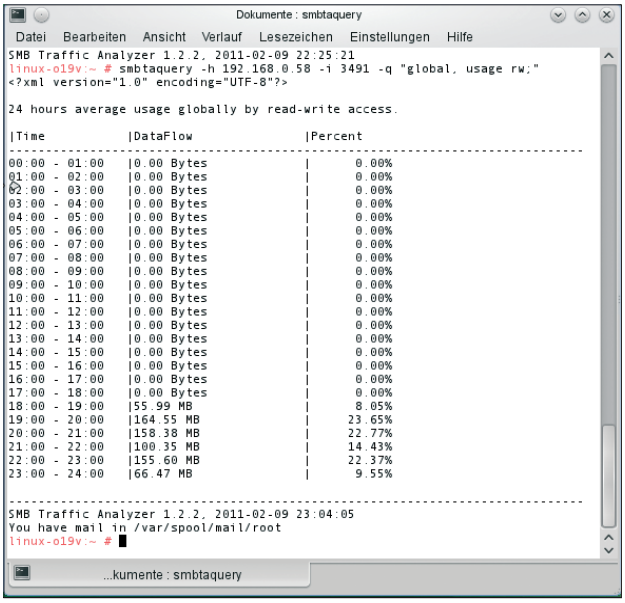

Abbildung 3: **Der Usage-Parameter ermöglicht ein zeitlich gewichtetes Auswerten der Aktivität etwa auf einer bestimmten Freigabe.**

Per Default legt »smbtad« seine Sqlite-Datenbank unter »$HOME/.smbtad/staddb« an, es sei denn, die Datenbank existiert bereits. Eine ganze Reihe weiterer Parameter lassen sich mit »smbtad -help« in Erfahrung bringen. Eine ausführliche Erläuterung sämtlicher Parameter findet sich darüber hinaus in der hervorragenden Dokumentation. Selbstverständlich ist es auch möglich, alle benötigten Konfigurationsparameter in einer Konfigurationsdatei »/etc/smbtad.conf« zusammenzufassen, im typischen Ini-Datei-Format, mit »#« als Kommentarzeichen.

Abfrage-Tool

Wie erwähnt, stehen an Client-Tools ein RRD-Treiber, das Befehlszeilenwerkzeug »smbtaquery« sowie »smbtamonitor« zur Verfügung. Selbstverständlich können versierte Admins der Traffic-Datenbank auch mit beliebigen SQL-Werkzeugen ins Innerste schauen, komfortabler ist aber »smbtaquery«, das speziell auf das Datenbank-Setup von SMBTA ausgerichtet ist und eine Reihe vorkonfigurierter Abfragen zum Erzeugen statistisch verwertbarer Daten mitbringt.

Smbtaquery erzeugt dazu XML-Output, den der Admin bei installiertem XSLT-Prozessor in sein favorisiertes Format überführen kann. Der XSLT-Prozessor erhält dazu passende Stylesheet-Informationen von »smbtaquery«. Um das Abfragen der Datenbank zu vereinfachen, enthält »smbtaquery« einen einfachen Interpreter, der auf die Zusammenarbeit mit SMBTA spezialisiert ist.

Abfragedatei

Es gibt grundsätzlich zwei Möglichkeiten, den eingebauten Interpreter zu benutzen. Die eine besteht darin, eine Datei zu übergeben, die bereits sämtliche Abfrage-Befehle enthält. Der Dateiname wird dabei durch den Parameter »-f« (File) gekennzeichnet.

```
smbtaquery -h Host -i 3491 -f ➥
befehlsdatei.txt
```

Sowohl in der Datei als auch im Interpreter-Modus ist jedes Kommando durch ein Komma zu separieren, Parameter durch ein Leerzeichen. Jede Zeile endet mit einem Semikolon. In der Konfigurationsdatei werden Kommentare wie üblich mit »#« gekennzeichnet.

Interaktive Queries

Als zweite Möglichkeit kann der Admin auch direkt den eingebauten Interpreter benutzen, was er »smbtaquery« mit dem Parameter »-q« (query) signalisiert.

```
smtaquery -h Host -i 3491 -q 'Abfrage'
```

Die Parameter »-h« und »-i« erschließen sich aus den bisherigen Erläuterungen. Hinter »-q« für „query“ folgt die eigentliche Abfragesyntax, zum Beispiel:

```
smtaquery -h Host -i➥
 3491 -q 'global,➥
usage rw;'
```

womit »smbtaquery« den globalen Traffic im kompletten Samba-Netzwerk zählt. Selbstverständlich lässt sich der zu untersuchende Traffic auch auf einen Benutzer oder einen Share beschränken:

```
smtaquery -h Host -i 3491 ➥
-q 'user drilling, total w;'
```

Smbtaquery kann sich selbstverständlich nicht nur via Hostname und TCP-Port verbinden, sondern wie erwähnt auch Unix Domain Sockets verwenden.

```
smbtaquery -u -q 'global, usage rw;'
```

Übrigens gibt »smbtaquery« per Default sämtliche Ausgaben auf dem Terminal aus, vom dem es gestartet wurde. Für das Umleiten der Ausgabe in eine Datei kann der Admin gewöhnliche Unix-Operatoren benutzen, wie »> ausgabe.txt«. Es ist aber auch möglich, mithilfe des Parameters »-o« beispielsweise die Ausgabe im HTML-Format zu erzeugen:

```
smbtaquery -u -q 'global, usage rw;'➥
 -o html > ausgabe.html
```

Abfrage-Beispiele können der hervorragenden Dokumentation entnommen werden. So ermittelt beispielsweise

```
'user drilling, total r;'
```

die absolute Anzahl (»total«) an Bytes, die vom Benutzer »user drilling« über das Samba-Netzwerk gelesen (»r«) wurden. Ein weiteres Beispiel wäre:

```
'share USB-Fritzbox, usage rw;'
```

Hierbei zeigt die Usage-Funktion die zeitliche Verteilung der Aktivität eines Objektes für einen „virtuellen“ Tag im 24-Stunden-Raster, im Beispiel für Lese-Schreib-Zugriffe auf der Freigabe »USB-Fritzbox«,

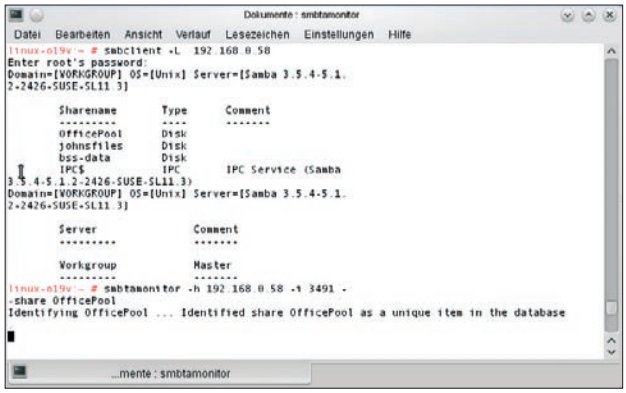

Abbildung 4: **Smbtamonitor bindet sich immer explizit an ein Objekt, also entweder einen Share, einen User oder einen Dateinamen.**

einer am Router angeschlossenen externen Festplatte (Abbildung 3).
Alternativ dazu zeigt der Ausdruck

```
'share USB-Fritzbox, total w;'
```

die absolute Anzahl an Bytes, die auf die Freigabe „USB-Fritzbox“ geschrieben wurden. Die »total«-Funktion ermittelt und zeigt stets die totale Summe an Bytes, die vom angegebenen Objekt (Share, User, gesamtes Netz) gelesen und/oder geschrieben wurden. Weitere leistungsfähige und interessante Parameter, wie »top«, »list« oder »last_activity« lassen sich ebenfalls der Dokumentation entnehmen.

Smbtamonitor

Das Tool »smbtamonitor« ermöglicht dem Administrator, den kompletten Samba-Traffic in Echtzeit zu überwachen. Der Client verbindet sich dazu direkt mit dem Daemon »smbtad«, anstatt gespeicherte Traffic-Informationen aus der Datenbank zu holen. Der Daemon sendet alle Daten-Pakete, die er vom VFS-Modul empfängt, an den »smbtamonitor«. Der wiederum horcht permanent am SMBTA-Socket und visualisiert alle empfangenen Pakete in einem Curses-Graphen. Dazu bindet sich jede Smbtamonitor-Instanz stets an ein Objekt, User, Share oder Datei (Abbildung 4). Dabei kann der Admin beliebig viele Smbtamonitor-Instanzen starten. So ist es mit »smbtamonitor« beispielsweise ebenfalls möglich, die absolute Anzahl übertragener Bytes und/oder den Datendurchsatz/Sekunde für das angegebene Objekt sichtbar zu machen. Smbtamonitor erfordert entweder die Angabe von Host (»-h«) und Portnum-

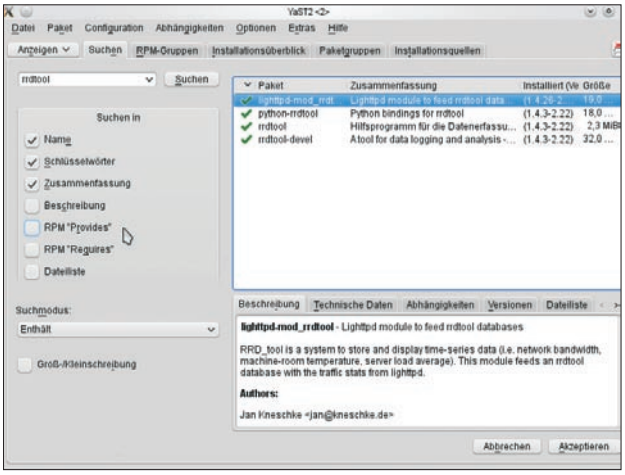

Abbildung 5: Die RRD-Tools lassen sich bei Open Suse mit Yast installieren und stellen ein leistungsstarkes Interface zur Verfügung, um grafische Messdaten zu visualisieren oder in Skripten weiterzuverarbeiten.

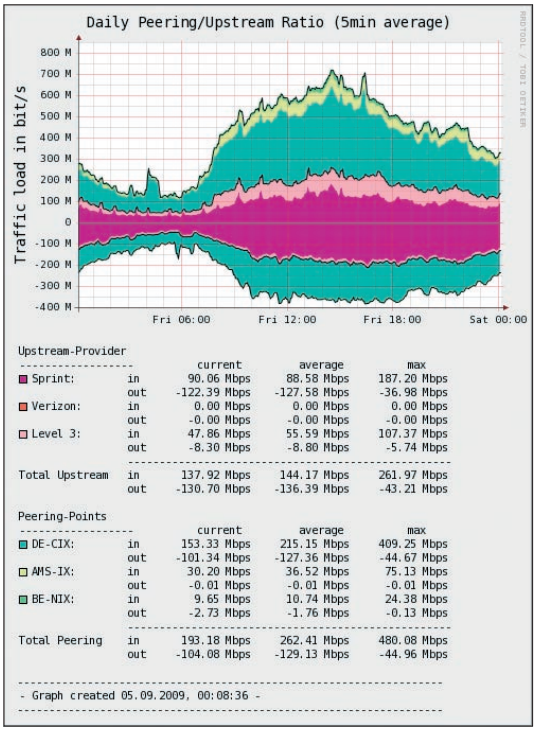

Abbildung 6: Der RRD-Treiber implementiert eine Schnittstelle zwischen SMBTA und RRD-Tool, mit deren Hilfe sich umfangreiche Graphen erstellen lassen.

mer (»-i«) oder das Iniitieren der Verbindung via Unix-Domain-Socket mithilfe des Parameters »-n«:

```
smbtamonitor -h Host -i 3491 --share ➥
Freigabe
```

Einzelne Dateien kontrollieren

Smbtamonitor kann sich auch gezielt an eine Datei binden. So ist es etwa möglich, in Echtzeit zu visualisieren, ob und in welchem Umfang die Nutzer im Netz von der Datei »RELASENOTE.TXT« Kenntnis nehmen, die auf einem Netzlaufwerk abgelegt ist:

```
smbtamonitor -h Host -i 3491➥
 --file RELEASENOTE.TXT
```

Übrigens lässt sich auch das Tool »smbtamonitor« mit einer Konfigurationsdatei »$HOME/.smbtatools/monitor-config« konfigurieren, in der Hostname und Portnummer hinterlegt sind:

```
[network]
Hostname = SMBTA-Host
Port = 3491
```

Viele weitere nützliche Parameter finden sich ebenfalls in der Dokumentation.

RRD zeichnet Graphen

Das Round Robin Database Tool (RRD-Tool) ist für seine Robustheit und die einfache Handhabung bekannt und ermöglicht eine grafische Visualisierung etwa des Durchsatzes auf einem Samba-Share. RRD-Tool steht unter [2] zur Verfügung, ist aber bei Open Suse auch in den Repositories zu finden und leicht mit dem Yast-Paketmanager zu installieren (Abbildung 5).

SMBTA selbst enthält allerdings lediglich einen Treiber als Schnittstelle zu RRD-Tool, der sich durch Aufruf von »rrddriver« gefolgt von einem oder mehreren Argumenten starten lässt (Abbildung 6). Darunter die Bekannten »-h« (Host), »-i« (TCP-Port), »-n« (Domain Sockets), »-s« (Share), »-u« (User) aber auch Neue, wie »-r«, mit dessen Hilfe der Admin einen RRD-Tool-Setup-String definiert. Default ist

```
DS:readwrite:GAUGE:10:U:U
DS:read:GAUGE:10:U:U
DS:write:GAUGE:10:U:U
```

Ein Beispiel für das Verwenden des RRD-Treibers könnte wie folgt aussehen:

```
rrddriver -b meinrrd -h Host➥
 -i 3491 -user drilling
```

Es startet den RRD-Treiber für alle Traffic-Informationen, die der Nutzer »drilling« produziert.

Per Default aktualisiert RRD-Tool seine Datenbank alle zehn Sekunden. Wer es genauer haben möchte, kann das Intervall beispielsweise mit »-S 2« auch auf zwei Sekunden senken. Jetzt kann der Administrator die RRD-Tools mit seinen zahlreichen Möglichkeiten nutzen. Mehr Informationen zu RRD-Tool gibt der gleichnamige Kasten. Auch die Projektseite des RRD-Tool ist eine ergiebige Quelle und stellt umfangreiches Dokumentationsmaterial zur Verfügung.

Ein Beispiel für den manuellen Aufruf von RRD-Tool gibt Listing 1. Er erzeugt eine PNG-Grafik »bild-smb-durchsatz.png« mit dem Titel »Datendurchsatz auf Share 'johnsfiles'«, die den Schreib- und Lese-Durchsatz auf der genannten Freigabe zeigt, getriggert im Unix-Time-Format, das der Parameter »-s« spezifiert. Hinter »DEF» folgt der virtuelle Name

RRD-Tool

Das ursprünglich von Tobias Oetiker entwickelte RRD-Tool hat sich über die Jahre zum Quasi-Standard beim Speichern von Netzwerk-Überwachungsdaten entwickelt. Es ist eine Software, mit der sich zeitbezogene Messdaten speichern und visualisieren lassen. Inzwischen steht RRD-Tool unter der GNU General Public License, sodass heute eine ganze Reihe von Autoren daran mitarbeiten. RRD-Tool ist im Sourcecode wie auch als Binary für eine Reihe von Betriebssystemen verfügbar.

Die Abkürzung RRD steht für Round Robin Database und somit die Art und Weise, wie es seine Daten speichert: Beim Anlegen seiner Datenbank reserviert das System Speicher nur für eine bestimmte Zeitspanne und erweitert die Datenbank nach deren Ablauf nicht etwa, sondern überschreibt die jeweils ältesten Daten. In der englischen Informatik heißen solche Reihum-Methoden auch Round Robin. Die Idee dabei: Stetig eintrudelnde zeitbezogene Messdaten sollen mit anwachsender Uptime nicht die Festplatte vollmüllen, denn bei älteren Daten reicht oft ein grober Über- oder Rückblick, während sich das aktuelle Geschehen in der Regel durch wichtige Details offenbart.

Das Benutzer-Interface von RRD-Tool besteht aus einem Satz von Kommandozeilen-Tools, deren Funktion auf der Projektseite ausführlich erklärt ist. Außerdem sind APIs für viele Programmiersprachen definiert, allen voran C und Perl, damit sich RRD-Tool von anderen Programmen zum Speichern nutzen lässt. RRD-Tool ruft man normalerweise nicht über die Kommandozeile auf, meistens dient es anderen Programmen als Datenquelle- und/oder Speicher, etwa Cacti [9] oder MRTG [10]. Eine umfangreiche Liste findet sich ebenfalls auf der RRD-Tool-Homepage.

Listing 1: RRD-Tool

```
01 rrdtool graph bild-smb-durchsatz.png -s 1290772099 -S 1 --title "Datendurchsatz auf Share
   'johnsfiles'" DEF:read_in=testdb:read:AVERAGE DEF:write_in=testdb:write:AVERAGE "AREA:write_
   in#AA0000:Write" "STACK:read_in#AA9999:Read"
```

für den darzustellenden Wert, dahinter die Datenquelle, der auszulesende Wert und die sogenannte Konsolidierungsmethode (hier »AVERAGE«).

Fazit

Datendurchsätze im Netzwerk messen und anzeigen an sich ist keine große Sache. Interessiert sich der Admin aber speziell für Datenverkehr, der vom CIFS-Server Samba verursacht wird, kommt vielen Administratoren ein waschechtes Datenbank basiertes Datamining-Tool wie SMBTA sicher wie gerufen. Zumindest unter Fachleuten erntete Autor Holger Hettrich bisher so viel Aufmerksamkeit, dass sich das Samba-Team entschlossen hat, den SMB Traffic Analyer zum festen Bestandteil von Samba 3.6 zu machen. Allerdings besteht bei der derzeitigen Architektur von Samba im Allgemeinen und SMBTA im Speziellen die Gefahr, dass die von SMBTA verwendete Datenbank sehr schnell anwächst. *(ofr)* ■

Der Autor

Thomas Drilling ist seit mehr als zehn Jahren hauptberuflich als freier Journalist und Redakteur für Wissenschafts- und IT-Magazine tätig. Er selbst und das Team seines Redaktionsbüros verfassen regelmäßig Beiträge zu den Themen Open Source, Linux, Server, IT-Administration und Mac OSX. Außerdem arbeitet Thomas Drilling als Buchautor und Verleger, berät als IT-Consultant kleine und mittlere Unternehmen und hält Vorträge zu Linux, Open Source und IT-Sicherheit.

Infos

[1] SMBTA wird Bestandteil von Samba 3.6 :[http://samba.org/samba/ftp/pre/WHATSNEW-3-6-0pre1.txt]

[2] RRDTool: [http://www.mrtg.org/rrdtool/]

[3] SMBTA Sourcecode: [http://morelias.org/smbta]

[4] SMBTA Binaries für openSUSE 11.3: [http://holger123.wordpress.com/smb-traffic-analyzer/smb-traffic-analyzer-download]

[5] SMBTA Homepage: [http://holger123.wordpress.com/smb-traffic-analyzer]

[6] SMBTA Git: [https://github.com/hhetter]

[7] Suse-Samba-Repository: [http://download.opensuse.org/repositories/network:/samba:/STABLE/]

[8] Stresstest Applicance 0.0.2: [http://holger123.wordpress.com/2011/01/28/smbta-stresstest-0-0-2-released-built-with-smb-traffic-analyzer-1-2-2]

[9] Cacti: [http://www.cacti.net]

[10] MRTG: [http://oss.oetiker.ch/mrtg/pub/?M=D]

Stresstest

Bei Stresstest (aktuell: 0.0.2) handelt es sich um eine fix und fertig geschnürte Suse-Appliance mit installiertem Samba-Server inklusive aktiviertem SMBTA-VFS-Modul (Abbildung 8). Stresstest basiert zwar ebenfalls auf SMBA 1.2.2, enthält aber darüber hinaus eine Reihe von Patches, die nicht in 1.2.2 enthalten sind. Die Stresstest-Appliance ist primär für das Testen von SMBTA gedacht und wird auch von den Entwicklern dazu intensiv genutzt. Wer einen schnellen Blick auf die Analysequalitäten von SMBTA werfen möchte, ohne erst ein komplexes Szenario aufzusetzen, für den ist Stresstest die beste Lösung. Die Appliance im Open Virtualization Format (OVF) enthält dazu mit »smbtaturesrv« eine kleine Server-Applikation, die über mehrere Prozess-Instanzen Dateinamen und Pfade über die Testumgebung verteilt.

Bei Stresstest 0.0.2 sind sechs User aktiv, die allesamt die Applikation »smbtatorture« benutzen, um den Serverprozess ausreichend zu beschäftigen. Smbtatorture ist quasi eine kleine Testsuite für SMBTA und wird bisher hauptsächlich von den Entwicklern selbst für Langzeittests genutzt. Das Tool simuliert ein typisches Load-Verhalten von Office-Anwendungen. Zwischen den einzelnen Verkehrsproduktions-Zyklen legt das Tool jeweils Pausen von einigen Sekunden ein. Außerdem misst es die Zeit, die es selbst läuft, und kann seine eigene Aktivität aufzeichnen und reproduzieren. Mehrere Smbtatorture-Prozesse können problemlos parallel laufen.

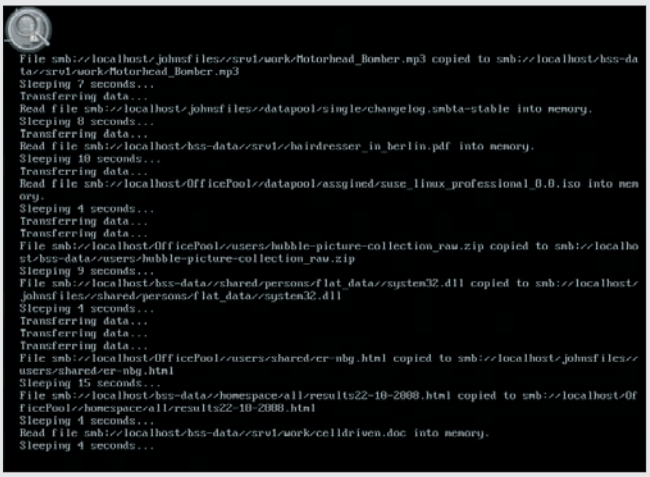

Abbildung 8: Stresstest setzt SMBTAD unter Stress, indem die Appliance laufend Samba-Traffic generiert.

Bei Stresstest 0.0.2 sind sechs User aktiv, die allesamt die Applikation »smbtatorture« benutzen, um den Serverprozess ausreichend zu beschäftigen. Smbtatorture ist quasi eine kleine Testsuite für SMBTA und wird bisher hauptsächlich von den Entwicklern selbst für Langzeittests genutzt. Das Tool simuliert ein typisches Load-Verhalten von Office-Anwendungen. Zwischen den einzelnen Verkehrsproduktions-Zyklen legt das Tool jeweils Pausen von einigen Sekunden ein. Außerdem misst es die Zeit, die es selbst läuft, und kann seine eigene Aktivität aufzeichnen und reproduzieren. Mehrere Smbtatorture-Prozesse können problemlos parallel laufen.

SMBTA-Stresstest benutzt übrigens Port 3491, über den die Werkzeuge aus smbtatools ihre Abfragen abwickeln, was bei der Firewall/Paketfilter-Konfiguration des Client-Rechners zu berücksichtigen ist. Ansonsten ist die Appliance wie folgt konfiguriert:

```
Netzwerk: DHCP
Timezone: Europe/Berlin
Language: de_DE.UTF-8
Firewall: disabled
```

Das Root-Passwort sowie die Passwörter der übrigen sechs User »holger«, nelson«, »john«, »bjoern« und »btram« sind jeweils »linux«.