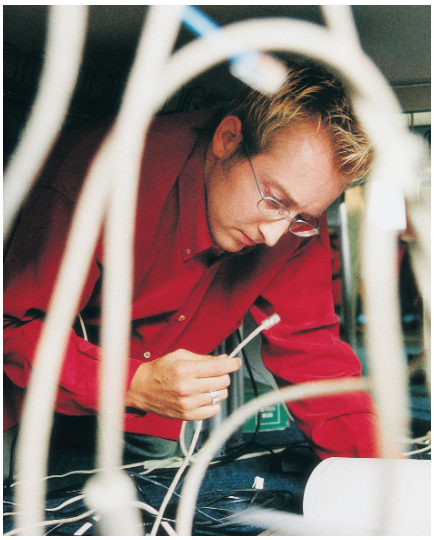
An introduction to real SysAdmin work

Heroes for hire



People who manage whole groups of Linux systems are a far cry from the root user on a stand-alone workstation. This series looks into the minutiae of Linux administration in complex environments. After a short introduction, we will be looking at a simple, but effective monitoring system in this month's column.

BY MARC ANDRÉ SELIG

Unix is user-friendly, but it does tend to decide for itself who its friends are. I am sure that most Linux users will have heard lines like this at times. Even the initial installation steps can be a fairly emotional experience, strongly influenced by the amount of good or bad luck you might have had – so much so in fact that they might be the start of a beautiful relationship, or failing that a new pet hate.

Linux users are bound to pick up some basic administrative skills at some time in their careers. Skills that were formerly the reserve of the geeks in the IT department are now part of a typical PC user's repertoire – that is, if they choose Linux. Convenient tools provided by a user's choice of distribution can help solve the more tricky issues, such as configuring network adapters, or even creating a more-or-less useful backup system.

People and Machines

Terms like admin and operator have changed their meaning over the course of time. The sheer bulk of people and machines that an administrator working on a server farm needs to handle, is what distinguishes this kind of admin from the simple root user of a stand-alone system. On a stand-alone workstation, the admin and the user are typically the same person – this is an ideal setup that avoids conflicting interests. Unfortunately, if you are responsible for a whole farm of Linux boxes, you will be forced to share these machines with other people. It is this act that provides scope for a whole range of political scenarios. The administrator knows the system, its architecture and the constraints that it imposes. Users are an entirely different case: If you do not understand your computer, you might not be prepared to accept its limitations, "Why do I need to enter a password? Why can't I run Outlook on this machine? It beeps too loud when I boot it!" We will be looking at the special requirements of Windows-biased staff in a later issue of this column.

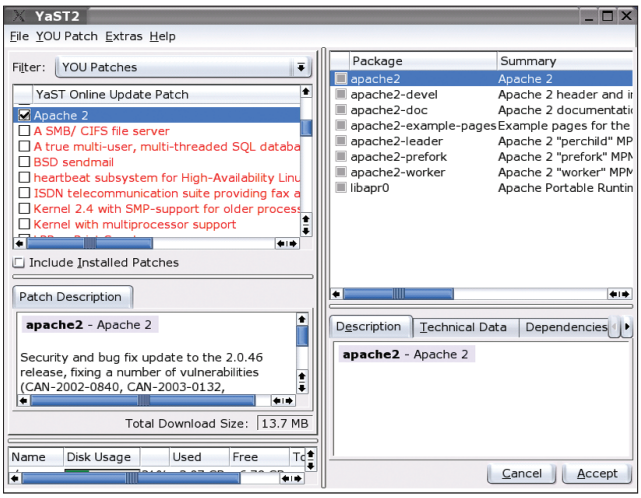


Figure 1: Manual online updates, as provided by SuSE Linux in our example, are an excellent solution for stand-alone desktop systems and allow for quick installation of security patches. However, admins in large-scale networks need different techniques

Cases where the users actually know more than the admin responsible for their machines give rise to a completely different theater of war. Say goodbye to your network design in situations like this. If you don't watch out, you end up with a completely different installation on every single machine.

A Question of Scale

The number of machines also leads to new issues. Your neighbors may not need much more than an hour to re-install their Linux system. Sadly, this kind of approach soon proves inadequate when faced with the reality of the admin's daily life. If you need to deploy 100 servers on a tight schedule, you might prefer to look for alternatives rather than working non-stop for four days.

Ensuing software installations and security patches present a similar problem. Manually installing software on a stand-alone desktop (see Figures 1 and 2) does not take much time when looked at individually – but if you multiply "not long" by a big enough number, it can easily mean losing a day's work. It is a good thing that admins have a few tricks up their sleeves. The fact that some pretty GUI tools are useless on larger networks is another challenge to real professionals.

Homogeneous networks that contain, for example, only Red Hat systems are an exception. You will typically find Windows boxes side by side with Unix machines running a variety of Unix vari-

ants. In some environments you will not even be able to select a specific distribution. Often only a small team, or a single individual is responsible for this lack of consistency. It always makes sense to standardize tasks and procedures.

A real admin gets through a heavy workload in an extremely short time – efficiency is the operative word – although admins tend to maintain that this is pure "laziness". Because they are too "lazy" to repeat the same manual steps, they write a script that saves them the effort.

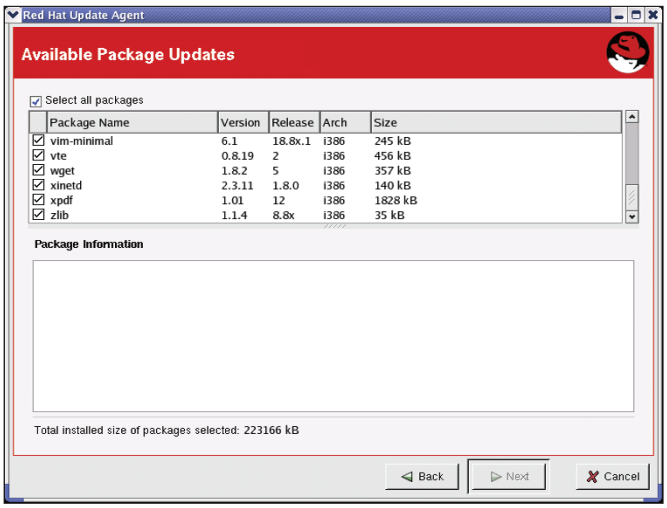


Figure 2: Red Hat also provides an automatic online update. Unfortunately, in contrast to the SuSE variant, there is no easy way of telling whether you are installing just normal bugfixes or security patches

High Availability

Sometimes the sheer number of machines is the issue, but in other cases it may be the demands made on them. If your home PC crashes, it might annoy the kids, as they would lose a round of the online game they were playing at the time. That is a problem you can get back to. However, imagine the print server in an enterprise going down; the staff would just sit twiddling their thumbs, and the admin could start looking for a new job. If a mission critical system goes down, it can mean an immense loss in terms of both revenue and data.

People say the perfect admin is one you never notice, but a lot of proactive effort at various levels is required to maintain this low profile.

Documentation is something that is often underestimated, although it is extremely valuable. Documentation requirements vary quite a bit – the more complex a network is, the more detail you need to put into your reports. A small repair shop with two part-time admins might be perfectly happy with a few sheets of paper in the server room showing the planned and existing architecture. Modifications are entered and printed out immediately (see Figure 3). Physical paper and ink documentation has the advantage of being readily available if things go wrong, especially if the file-server with the documentation files has already gone down.

Careful Planning

Admins need to carefully plan and test any changes before applying them to production machines. A modified configuration that freezes up your private Apache Web server might be acceptable, but this should never happen to a public Web server.

If something does go wrong, you need a well thought out recovery plan to help you restore things to their original status. In less critical circumstances it is quite normal for experienced admins to install a new software package on top of the previous version, or change configuration files on the fly. A more cautious approach may cause less headaches and save jobs: so, if the new module you have just compiled is missing a shared library, make sure the previous version is back up within seconds.

Caution alone will not prevent downtime. Some errors may be avoidable, but hardware and line failures are an unfortunate fact of life. A design that provides redundancy both for the services and the systems themselves is your only solution. Some enterprises may not be able to afford the necessary investment in equipment.

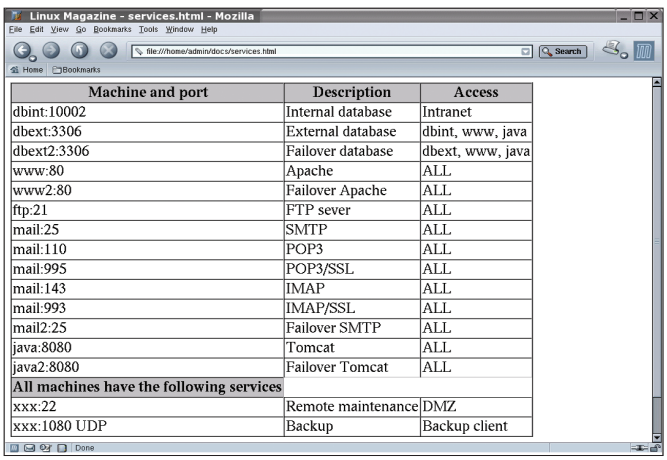
In a worst case scenario, admins should at least notice what is going on. You will not want to spend the whole night at your console checking the syslogs… Critical services need automatic monitoring, and should alert you of some problems without intervention.

Big Brother

It should not take long to set up a simple, but effective monitoring system. The well-known services shown in Figure 3 turn up again in Listing 1, where column one indicates the path to a script that monitors the service, and column two supplies the name of the host running the service.

Listing 2 includes the *test/smtp* file, a miniature test script for the mail service. If the connection to the SMTP port fails, or the welcome message does not start with *2*, the script issues an alert (line 11); it does nothing in any other case (line 8).

A control script (see Listing 3) reads the file from Listing 1 (line 6) and executes its content. This shell programming trick concatenates the output of all our test scripts to create a single string, *$RES*. If one of the tests fails, the master script alerts the address indicated in line 3. As many Mail-to-SMS gateways only forward the subject, the script adds the error messages in line 9. The script can run several times an hour, but will



Machine and port	Description	Access
dbint:10002	Internal database	Intranet
dbext:3306	External database	dbint, www, java
dbext2:3306	Failover database	dbext, www, java
www:80	Apache	ALL
www2:80	Failover Apache	ALL
ftp:21	FTP sever	ALL
mail:25	SMTP	ALL
mail:110	POP3	ALL
mail:995	POP3/SSL	ALL
mail:143	IMAP	ALL
mail:993	IMAP/SSL	ALL
mail2:25	Failover SMTP	ALL
java:8080	Tomcat	ALL
java2:8080	Failover Tomcat	ALL
All machines have the following services		
xxx:22	Remote maintenance	DMZ
xxx:1080 UDP	Backup	Backup client

Figure 3: Precise documentation is a major administrative task: this list of services running on the server might help a temporary admin

mail the specified address only in case of a failure. If you specify your cellphone as the target, you may even receive a surprise message during a visit to the opera, if your Web server crashes…

Remote

Of course, you would not want to run the monitoring system on your own Web server, as a Web server crash would also take down the monitoring system. Most professionally operated Websites use external fail-safe systems that provide a perfect platform for a monitoring system. If needed, there are a few companies that offer this service on the Internet.

It is difficult to tell whether a monitoring system is still up and running, as a crash will simply prevent it from sending messages. If you perform ten tests an hour, all those success reports might prove slightly irritating. One option would be to log the results of all your tests, and collate the results once or twice a day. This could happen at 8 a.m. and 8 p.m., or straight away if a failure occurs. If you are unlucky you might lose both your monitoring system and a critical service for 12 hours at the most.

Last, but not least the topic of backups raises its ugly head. Backup copies on a stand-alone machine are so bothersome that overly optimistic users tend to neglect them. A larger number of computers amplifies the volume of data and the logistical problems involved. It is some consolation to know that more computers mean more possibility of failure, so at least your backups will be more worthwhile.

Listing 1: Services

```
01 test/ping  www
02 test/star  dbint
03 test/mysql dbext
04 test/mysql dbext2
05 test/http  www
06 test/http  www2
07 test/http  java
08 test/http  java2
09 test/ftp   ftp
10 test/smtp  mail
11 test/smtp  mail2
12 test/pop   mail
13 test/pops  mail
14 test/imap  mail
15 test/imaps mail
```

Listing 2: SMTP test

```
01 #!/usr/bin/perl -w
02
03 use Socket;
04 die "No host name given" if not defined $ARGV[0];
05 if (socket(SOCK, PF_INET, SOCK_STREAM, getprotobyname('tcp')) and
06    connect(SOCK, sockaddr_in(25, inet_aton($ARGV[0])))) {
07    my $line = <SOCK>;
08    exit 0 if defined $line and $line =~ /^2/;
09 }
10 # Error:
11 print "smtp on $ARGV[0]\n";
12 exit 1;
```

Data Security and Integrity

In a corporate environment, security plays a far more important role than at home. The most important security strategy for a SOHO network is probably to down your servers and use packet filters to block any incoming connections. This does not scale well to a server farm as your machines' whole purpose in life is to provide network services. You need to harden the machines. This is not easily done with a firewall.

Updates and patches are far more critical. Admins need to close down security holes as soon as they are discovered and disclosed, but that does not mean losing your head and updating all your software, come what may. New versions are often buggy. Things that used to work may not in the latest version. That means some careful investigative work on your part and making informed decisions as to what, and when, to update.

Incident detection and response is a controversial topic at present. Everyone seems to have their own private recipe for success, and most people fail to understand why others are unwilling to adopt the same approach. However, there is no denying the fact that professional computer management in a professional environment does require a well thought-out solution to the issue. Manually poring over 25MBytes of logfiles every day doesn't qualify as an answer.

Things to come

Even if working with Linux is no longer a challenge, today's admins are faced with a barrage of tasks that make their jobs both interesting and appealing. No wonder capable system administrators are sought after.

This column will be looking into various system administration topics over the course of the next few issues. We will be looking at many more critical aspects of system administration. ■

Listing 3: Monitoring Script

```
01 #!/bin/sh02
02 # Target adresses for mail
03 DEST="01712345678@t-d1-sms.de, musikfan@imail.de"
04 RES=`. list`
05 if [ "$RES" != "" ]; then
06       cat <<EOF | /usr/lib/sendmail -t -U
07 From: root@localhost
08 To: $DEST
09 Subject: URGENT: Server Alert: $RES
10 Hello,
11 There is a problem with the following services on the Web cluster:
12 $RES
13 EOF
14 fi
```