ADANAC SOFTWARE

A simple remote site monitor

By Dirk Hart

One of the many things I do is run a mailserver for hire (www.mailstarusa.com). Understandably, folks like to have high availability on that machine - as do I since I get all the calls if that machine is down. It was surprising when I got a call from one customer saying the machine had been up and down a lot since I had experienced no problems with it at all. And the fact that this customer was connected through Hell South made me a bit suspicious.

I decided that the thing to do was to monitor their IP address for a while and see how many dropouts they had. I searched the net and found many monitors, some of them very nice, but what I wanted was something very simple that would record the events in a log file. There didn't seem to be much but I did come across pingchecker (http://www.pingangel.com/pingchecker.htm).

Pingchecker seemed simple enough for me, so I pasted it into a file and modified it to check my clients site. I put in my own email address. I edited crontab (/etc/crontab) to kick it off every 5 minutes and waited. This seemed like less and less like a good idea after a while. I wasn't getting much email and it seemed to take a long time. The problem was with the command does actually does the deed: ping -c 2 $pagesite. Perhaps the default behaviour of ping is different with the unix that pingchecker was written for but on RedHat Linux 8.0 it looks like ping -c2 $pagesite will happily wait until the next millennium. Consulting the manual I re-arranged things until I got ping -c2 -w10 $pagesite.

Well, that was pretty good but there were still some spurious results. Also, I don't usually spend my time waiting for email to arrive - computer guy likes to get out and about. So a couple of new features were added at this point.

First of all I wanted to be notified by phone when I was out moving and shaking, and I was still getting some false positives when connections were slow rather than actually down. Also, I wanted to be able to monitor more than 1 site. Since this was a brand new effort I renamed it pink.

```
#!/bin/bash
# reads sites from a file
# ping a site if no response then email a message
logfile="/var/log/pink.log"
smsnotify="5085551515 @vtext.com"
mailnotify="monitor@yourdomain.com"

cat /usr/lib/pink/pink.sites|while read pingsite pingname
do
      ping -c 2 -w 10 -Q 0x04 $pingsite 2>&1 > /dev/null
```

```
	#if 100% packet loss - a bad ping
	if [ $? -gt 0 ]
	then
	echo no reply from $pingsite \($pingname\) on `date` >>$logfile
	echo $pingname $pingsite "Alert" `date`| mail -s "$pingname" $mailnotify
	echo no reply from $pingname $pingsite | mail -s "$pingname" $smsnotify
	else touch "$logfile"
	fi
done
```

Verizon customers can send themselves SMS messages for $0.02 each through vtext.com, so I added

```
echo no reply from $pingname $pingsite | mail -s "$pingname" $smsnotify
```

to the script. This way you can get a message on your cell phone and call your client right away - they'll think you are possessed of supernatural powers.

Note also that if the remote site is pinged correctly the date and time on the log file are updated using touch. This is a handy way to see if your script is indeed running.

More than one site can be checked:

```
cat /usr/lib/pink/pink.sites|while read pingsite pingname
do
...
done
```

This loop reads an IP address (or domain name) and a short description for reporting purposes from a file named pink.sites.

Pink.sites contains:

```
123.123.123.123    Fake Site - never pings
146.115.8.20    Ultranet DNS
198.6.1.5       UUNet DNS
pcunix.com pcunix
```

This way I can now monitor several clients sites and get reports on failures even while I am on the road.

It seemed to work quite well, but after a week or so I knew I was getting too many results for them to be accurate. Something had to be done to improve the quality of the results. cartoon

The first thing I noticed is that sometimes all the sites I was checking were reported at once. I would get a string of messages and then things would be calm again. Clearly my own connection was having occasional dropouts rather than mailstarusa.com or my client sites. I reasoned that I would report outages only if machines I knew to be highly reliable were available. That is, if I could not ping certain DNS servers then I could safely keep quiet about the rest.

```
chk1="146.115.8.20"
chk2="198.6.1.5"
ping -c 2 -w 10 $chk1 2>&1 > /dev/null; chk1=$?
ping -c 2 -w 10 $chk2 2>&1 > /dev/null; chk2=$?

if [ $chk1 -gt 0 -a $chk2 -gt 0 ]
      then
      echo "reliable hosts are down. no sites checks performed." `date` >>$logfile
      else
{ ...
}
```

chk1 and chk2 are DNS servers belonging to Ultranet (now RCN) and UUNet and were chosen only because I knew them to be reliable and had their addresses memorized. Basically the results of each ping are recorded and if they are both non-zero then the local connection must be down and we can keep quiet about the rest.

Much to my interest something had escaped my notice the first time I read the man page for ping. ping -c2 123.123.123.123 pings a site twice and waits endlessly for results - this led me to add -w 10, thinking that I had made ping timeout after 2 pings and 10 seconds. This is not quite the case. What happens when these two parameters are combined is that the remote site is pinged 10 times and the command finishes when 2 replies are received (no error is reported) or when 10 seconds are up (an error is recorded). That means 8 pings go astray and we are still willing to say the site is up. You could certainly make the argument that the results are of poor quality as a result. On the other hand the results seem to pretty accurately match the reality of the situation.
This gave good results, but I still got messages about sites being down. I would immediately ping the sites again and they would not be down at all. I decided after some research, that things on the net were just slow and the packets were still not returning before 10 seconds (-w 10) were up. I carefully read the man pages again. I didn't want to make -w 10 much larger in case the list of sites grew large and cron kicked the script again before it had finished.

There was some reference in the man pages to QoS bits and it turns out that if we set -Q 0x04 we get a more reliable result. This is a good thing as checking SMS messages while traveling in the geek-mobile is a Bad Thing.

I also changed the script so that different people could be notified for each site. If my clients are interested in the results I can email them without delay. I edited pink.sites to match:

```
123.123.123.123    Fake site    5085551212@vtext.com    monitor@yourdomain.com
pcunix.com    pcunix.com    5085551212@vtext.com    monitor@pcunix.com
```

Finally, here is the whole script - not pretty, but functional:

```
#!/bin/bash
# ping a site if no response then email a message
logfile="/var/log/pink.log"
smsnotify="5085551515@vtext.com"
```

```
mailnotify="monitor@mailstarusa.com"
chk1="146.115.8.20"
chk2="198.6.1.5"


ping -c 2 -w 10 -Q 0x04 $chk1 2>&1 > /dev/null; chk1=$?
ping -c 2 -w 10 -Q 0x04 $chk2 2>&1 > /dev/null; chk2=$?


if [ $chk1 -gt 0 -a $chk2 -gt 0 ]
      then
      echo "reliable hosts are down. no sites checks performed." `date` >>$log
file
      else
{
cat /usr/lib/pink/pink.sites|while read pingsite pingname smsnotify mailnotify
do
      ping -c 2 -w 10 -Q 0x04 $pingsite 2>&1 > /dev/null
      #if 100% packet loss - a bad ping
      if [ $? -gt 0 ]      then
      echo no reply from $pingsite \($pingname\) on `date` >>$logfile
      echo $pingname $pingsite "Alert" `date`| mail -s "$pingname" $mailnotify
      echo no reply from $pingname $pingsite | mail -s "$pingname" $smsnotify
      else touch "$logfile"
      fi
done
}
fi
```

Here is a bit of the log file:

```
no reply from mydomain.com (mydomain) on Fri Mar 21 20:01:06 EST 2003
reliable hosts are down. no sites checks performed. Mon Mar 24 05:15:17 EST 2003
no reply from mydomain.com (mydomain) on Mon Mar 24 18:45:18 EST 2003
```

Using this script I recorded results over a period of a few weeks and noted that my customer had far more dropouts and of longer duration than did my mailserver. I was able to show the results to my client and suggest that they contact their DSL provider.