system and network monitoring

Using Big Brother

by John Sellens
<*jsellens@gnac.com*>

John Sellens is Associate Director, Technical Services, with GNAC in Toronto. He is also proud to be husband to one and father to two.

This article will outline, review, and express opinions on the Big Brother System and Network Monitor. I'll share an overview of Big Brother, relate some of my experience with it, and describe it in terms of the evaluation criteria outlined in the first article (*;login:*, June 2000) in this series on system and network monitoring.

Stealing a sentence from the Big Brother FAQ, "Big Brother is a Web-based Systems and Network monitor written by Sean MacGuire <*sean@bb4.com*> and Robert-Andre Croteau
<*robert@bb4.com*>," and it is available at <*http://www.bb4.com/*>.

Big Brother (BB) was first released in October 1996. As of this writing, I've been running Big Brother version 1.4c on an experimental basis for a little over a month. During that time, the current version of Big Brother has progressed to 1.4g, which, according to the README.CHANGES file, has added a few fixes and a few new features, including a likely to be useful "failover" ability for when your monitoring or paging host fails.

Overview

Big Brother is a collection of Bourne shell scripts and C programs. It consists of four main components or services:

BBDISPLAY: A machine running a Web server that generates and provides the Big Brother status pages.

BBNET: A machine that tests the availability of various network services on hosts of interest.

BBPAGER: A machine that accepts notification requests, interprets the notification rules, and sends email or pager messages.

Client: A machine running the Big Brother client software that performs various local tests and reports the results to the BBDISPLAY and BBPAGER servers.

Big Brother is primarily a monitoring system for UNIX hosts, but it can be used to monitor network services on just about any network-connected device. BB client implementations for Windows NT and Novell are available.

Configuration of Big Brother is done through a number of text files contained in the BB build/installation directory hierarchy. Once configured, the primary status interface to BB is through Web pages that are regularly rebuilt by the Big Brother daemon.

The Web interface provides a comfortable, understandable, and portable interface to status, history, and other reports. Through various configuration and HTML fragment files, Big Brother can be configured to report in multiple groups, pages, and hierarchies. And it's possible to report status summary information to other BBDISPLAY servers to create a server hierarchy.

The design and implementation of Big Brother make it easy to extend. Other monitors or probes can use the BB tools for reporting and logging by sending simple one-line messages to the BBDISPLAY and BBPAGER servers.

Evaluation Criteria

SIZE AND COMPLEXITY

Big Brother is intended to be a simple, relatively lightweight monitoring system that handles the most common needs in a reasonable fashion. It generally succeeds in its goals, with a few exceptions.

I found the configuration of Big Brother to be overly convoluted, with too many options in too many places. (More on this below.) Additionally, installing the BB client on a new host apparently can't be done by just copying from an existing host — the instructions require using the bbclient command to create a host-specific tarball for each new client.

The fact that Big Brother is implemented in sh and C is both good and bad. It means that Big Brother is fairly easy to port and will run on just about any (UNIX) machine with a minimal software base, but it also means that the resultant implementation is sometimes more convoluted and inefficient than it might otherwise be. For example, the `bb-network.sh` script uses about 20 lines of commands to look up network services in the `/etc/services` file, each time through, for each service, and for each host being monitored. A different implementation could allow the use of services' file-entry caching and the`getservbyname()` library routine, which will work better in an environment that relies on NIS.

SCALABILITY

Because of its implementation, its multiple configuration mechanisms, and its use of a nonfilterable Web interface for display, my expectation would be that Big Brother won't scale very well beyond perhaps 50 or 75 systems or devices. While I'm sure that there are BB installations larger than that, my impression is that Big Brother is best suited for small- to medium-sized installations.

RELIABILITY

With proper configuration and installation, Big Brother can be fairly reliable. If BBDISPLAY and BBNET are actually different hosts, the net effect is that they (mostly) watch each other — BBNET will send notification messages if BBDISPLAY is down, and BBDISPLAY will complain about stale information if BBNET is down. With careful planning, and a certain amount of replication, you can do fairly effective monitoring of all your hosts, including the monitoring hosts themselves.

Some of the Big Brother ancillary scripts seem to be a little fragile. For example, the `bbchkcfg.sh` script (which checks certain BB configuration files for errors) seems to work only if you run it from the BB etc directory and fails in curious ways if you don't. Similarly, the `install/bbclient` script needs to be run from the install directory. And, as a final example, the `runbb.sh` start/stop script fails in an ugly way if you ask it to stop Big Brother when BB isn't already running:

```
% ./runbb.sh stop Stopping Big Brother... cat:
/home/jsellens/bigbrother/bb14c/tmp/BBPID: No such file or directory usage:
                       kill [-s signal_name] pid ...
                       kill -l [exit_status]
                       kill -signal_name pid ...
                       kill -signal_number pid ...
rm: /home/jsellens/bigbrother/bb14c/tmp/BBPID: No such file or directory
```

Admittedly, these kinds of errors don't have much impact on BB's ability to operate correctly under normal circumstances, but they did make me wonder about the overall reliability of the Big Brother implementation.

COST

Unless you're reselling monitoring services or the Big Brother software itself, there is no charge for Big Brother. You can choose to buy a support contract or support the Big Brother project by purchasing a license, but otherwise the cost for Big Brother is limited to the time you take to install and configure the software.

NUMBER AND TYPE OF PROBES

Big Brother comes ready to monitor the most common system attributes and network services: network connectivity, ftp, http, https (with lynx), telnet, ssh, nntp, disk space, UNIX process existence, etc. The ease with which Big Brother can be externally extended makes adding "internal" services possible without too much trouble.

CONFIGURATION COMPLEXITY AND FLEXIBILITY

I had a certain amount of difficulty with Big Brother's configuration mechanisms — things sometimes seemed nonintuitive, and there seem to be more places to configure things than I would have expected.

My first surprise was with BB's `etc/bb-hosts` file — it uses # (a.k.a. hash, pound, octothorpe) both to indicate comment lines and to act as a separator on configuration lines. The `bb-hosts` file lists the hosts to be probed and specifies certain output formatting information for display. Some typical lines might look like this:

```
192.168.1.12 hostname         # ftp smtp telnet ssh
192.168.1.27 monitorhost      # BBNET telnet ssh ftp
192.168.1.39 webhost          # BBDISPLAY http://webhost/ telnet smtp
192.168.1.49 pagerhost        # BBPAGER telnet ssh  group

<h3>Workstations</h3>
192.168.1.51 work1            # telnet ssh
192.168.1.52 work2            # telnet ssh
192.168.1.53 work3            # telnet ssh
```

The documentation states that the bb-hosts file is "identical to the standard `/etc/hosts` file," with a few additional directives — I suspect that that is the original reason for using # as a separator and not just as a comment indicator. But with the addition of the various formatting and summary commands, this no longer seems to be a reasonable choice.

Additionally, the documentation warns against simply commenting out entries in some configuration files — I think that this is because some of the Big Brother programs search (with `grep`) for keywords in the file, blindly ignoring any comment indicators that might exist. There is no obvious way to continue lines in the `bb-hosts` file, which means that the definitions of some hosts can get a little ugly. Finally, the services for Big Brother to check are all specified as service names from the services file (ftp, telnet, etc.), except for http and https, which require a complete URL. This is an inconsistency I found somewhat problematic.

I had trouble understanding what all of the BB config files are for, and I kept finding out about new and different config files. A somewhat complete list of config files is:

```
etc/bbdef.sh
        various shared settings
etc/bbsys.local
        OS specific paths for commands and logs files
etc/bbwarnsetup.cfg
        paging and notification setup
etc/bbwarnrules.cfg
        who to notify, how, for what, and when
etc/security
        which hosts/networks can connect to the Big Brother daemon;
        no comments are allowed
etc/bb-proctab
        local override for the PROCS and PAGEPROCS variables from bbdef.sh
etc/bb-dftab
        local or shared disk warning configuration, overriding DFWARN
        and DFPANIC from bbdef.sh
www/notes/hostname.html
        additional per-host notes that can be accessed from the generated Web
        pages
```

Each of the various files seems to have a somewhat different syntax and rules about comments. Tools such as bbchkcfg.sh and bbchkhosts.sh are provided to check for certain configuration errors, but they don't notice some of the possible errors in the files and are somewhat fragile in their operation. Some of the files in the etc directory are intended to be shared, some are local, and some can be shared if desired.

EXCEPTION REPORTING STYLE

The Big Brother Web displays always provide a "full" status display — every device is always listed, and there's no way to have it display just those devices that have a nonnormal status.

This can be modified somewhat by the use of "summary" lines (which aggregate the information from another BBDISPLAY host), but that adds some level of "convolution" to your configuration, and may end up being harder to manage.

BB provides a simple mechanism to "acknowledge" a particular problem report, which suppresses additional trouble notifications for however long the user requests. A similar mechanism allows an administrator to disable notifications on a group of hosts for a specified number of minutes.

EXCEPTION-REPORTING TOOLS

Exceptions identified by Big Brother are reported in three standard ways: Web interface, email, and pager (via kermit, qpage, or sendpage). A mechanism allows the notification destinations to be set based on problem host and time of day. There's not really a useful command-line or curses interface — the lynx text browser is "usable" with BB, but painfully.

Overall the notification mechanism is more advanced and more flexible than I was expecting — it does, however, seem to require a certain amount of thought and staring at the documentation to get things set just right.

LOGGING AND DATA STORAGE

Big Brother logging, history, and status information is stored in a number of text files, most of which are separated into different files or directories by hostname and service type. BB typically logs only state changes, rather than every contact, so the logs tend to grow in a reasonable fashion. There doesn't seem to be a mechanism for pruning old information, but a simple script with a few `finds`, `rms`, and `mvs` can probably take care of most situations.

One thing that seems a little discomfiting to me: the files in BB's logs directory have modification times that are some number of minutes in the future. I think this is done in order to keep track of when state changes or renotifications should happen, but manipulating modification times makes me wonder if there isn't some better way achieve the same result.

REPORTING MECHANISMS

Big Brother's primary strength is near-realtime monitoring and reporting — some historical reporting is available, but it's somewhat limited. Trend information (such as disk-space growth over time, or load averages) is not available through Big Brother.

Other Notes

Overall, I thought that the Big Brother documentation could benefit from a little consolidation and elaboration. For example, when trying to determine what all the parameters on the `bb-hosts` file "summary" lines are for, I had to examine the source code before I really understood.

If you remove a host or service from a `bb-hosts` file line, you are required to manually remove the various log and history files related to those hosts or services, because otherwise you get various trouble indications and alerts for the now-removed service. This means that temporarily disabling the polling of a host may force you to remove history information you might otherwise wish to keep.

Similarly, if you've got various messages in your syslog files that Big Brother is warning you about, there's no way to tell Big Brother to stop telling you about particular problems unless you modify or remove the log files themselves or temporarily disable all warnings for those files, neither of which is a particularly appealing choice.

I happened to find the default Web display a little busy for my taste, but Big Brother provides ample opportunity to change and customize the displays, so I can easily make them just as plain as I want them to be.

One other thing I found surprising: I initially installed Big Brother on a machine that didn't have the lynx text browser installed. Big Brother didn't notice or report that lynx was missing; it indicated instead that the http service had failed. It took me a few tries to figure out just what was going wrong.

Opinion

In the world of monitoring software, Big Brother is fairly well known and, by all appearances, is widely used in small- to medium-sized installations. Lots of people find it very useful, and it certainly seems to fill a niche.

But even so, I've ended up in a sort of "like-hate" relationship with it. I wanted to like it and expected that I'd be able to recommend it enthusiastically for smaller sites. But I kept running into things that seemed overly complicated or more annoying than they had to be. Perhaps I was lulled into a false sense of "very simple" just because Big Brother has a Web interface. But as time went on, I kept seeing other little nagging things about it, and I kept finding out about yet another configuration file or option.

I was disappointed, but not surprised, by the lack of use of SNMP in Big Brother. To be honest, if I had started to implement a monitoring system in 1996, there's a pretty good chance that I wouldn't have used SNMP either. But with the advent of tools such as MRTG and Cricket, it's almost a necessity to have an SNMP agent running on every device and system. It seems a bit of a shame to have a separate client and network protocol and not make use of the SNMP agent that you probably want to have running anyway.

I think my conclusion is that Big Brother would benefit from a substantial rewrite, perhaps in some other more effective language, and a general consolidation and reconsideration of the configuration structure and syntax.

As the heading above indicates, this is just my opinion, and I suspect that large numbers of people will disagree with me and think I'm being overly critical. Some of my dissatisfaction with Big Brother is likely simply due to laziness or impatience on my part, or perhaps because I came to Big Brother with unreasonable expectations.

Clearly, Big Brother is effective for a lot of sites and fills a need. And, honestly, what more can one ask from a piece of software, or the kindness of software authors who make their considerable efforts available for free to the community at large?

The Big Brother home page, documentation, samples, and distribution are all available athttp://www.bb4.com/, along with information on support and commercial licensing, courtesy of the MacLawren Group Inc. The two presentations from SANS '98 http://www.bb4.com/bbsans98.pdf and SANS '99 (http://www.bb4.com/bbsans99.pdf) are worth a read. Spong "is a simple systems and network monitoring package" by Stephen L. Johnson, written in Perl, that is similar in some ways to Big Brother. See http://spong.sourceforge.net/.