The Sysadmin's Daily Grind: Sarg

PEDRO'S ANALYSIS TOOL

A busy proxy server is something that no self-respecting admin should leave to its own devices. The Squid logfile analyzer, dubbed Sarg by its author, helps you keep your Squid servers on track.

BY CHARLY KÜHNAST

I really enjoy browsing sites such as Sourceforge or Freshmeat for interesting software packages. Of course, packages with interesting sounding names are more likely to catch my eye. I couldn't help noticing a tool by the Brazilian software developer, Pedro Orso. I'm sure Pedro wasn't aware of the slightly morbid connotations of his Squid Analysis Report Generator (Sarg) for German Linux users. (Sarg is the German word for coffin.) But this didn't put me off in the least, and it's just as well it didn't, because Sarg is exactly the kind of tool that I like: lean and quick. And it gets on very efficiently with the task for which it was intended: creating reports based on Squid logs.

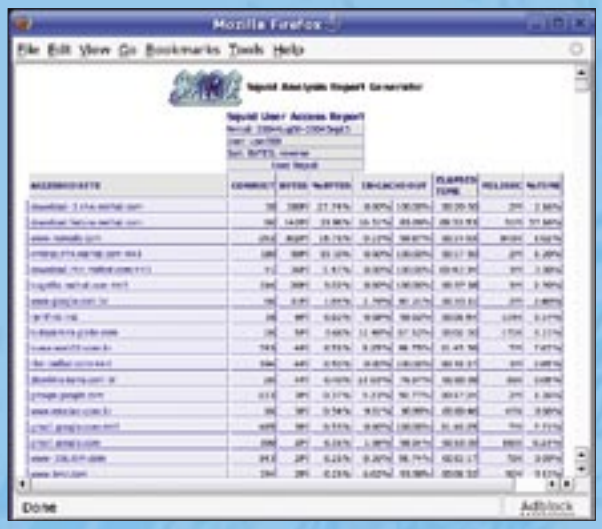

Figure 1: Sarg creates clear, user-specific reports, keeping you up to date on what Squid is doing.

Sarg source code and binary packages for various Linux distributions, *BSD, MacOS, and even OS/2, are available from [1]. Sarg takes Squid logfiles and uses the data to generate a useful statistical overview, like the overview shown in Figure 1. But in contrast to the Squid add-on Calamaris [2], which we looked at in a previous issue, Sarg generates user-specific statistics.

You can pass the most important parameters to Sarg at the command line, and the *sarg.conf* file (Sarg comes with an example) gives you more options, such as modifying the output design. Pedro obviously put a lot of thought into what most users expect from the Sarg tool and has provided meaningful defaults for most settings. This means that to generate a report, you can simply specify the source file, that is, the Squid *access.log* file, and the target directory where you would like Sarg to put the results.

```
sarg -l ➥
/var/log/squid/access.log ➥
-o /www/sarg/
```

Sarg and DNS

For more convenience, Sarg has a *-n* command line option that enables DNS resolution of addresses. This is fine for a small Squid with just a few users, but if you have a large cache that processes billions of requests a day, you will not want to enable Sarg name resolution because the analysis could take all day. Apart from this, most DNS admins would not be too pleased about the involuntary stress test this puts their servers through.

The ability to restrict analysis to a specific period of time, using the *-d TT/MM/YYYY-TT/MM/YYYY* option, is another useful feature. Time has always been an issue with Squid, which stores time in seconds past the eon with a resolution of one thousandth of a second in its *access.log* file. Although you can stop Squid from doing this by telling the tool to use the legacy common logfile format, you do lose some information in the process. Sarg is a big help here. Entering

```
sarg -convert ➥
/var/log/squid/access.log
```

will output the logfile on STDOUT – and it gives you a readable date format. .

A value such as *1126705707.537* is thus converted to *09/14/2005 15:48:27*. And losing the thousandths doesn't faze me in the least.

Thanks for the app, Pedro, I haven't had this much fun in ages, but you should rethink the acronym for the benefit of all those German Linux hackers. ■

INFO

[1] Sarg: *http://sarg.sourceforge.net/sarg.php*

[2] Charly Künast, "The Sysadmin's Daily Grind: Calamaris," Linux Magazine 12/03, p. 60.

THE AUTHOR

Charly Kühnast is a Unix System Manager at the data-center in Moers, near Germany's famous River Rhine. His tasks include ensuring firewall security and availability and taking care of the DMZ (demilitarized zone).