The sys admin's daily grind: Snoopy

Guide Dog

Sometimes sys admin Charly needs to know when exactly he did something ingenious on one of his servers. Finding an infallible memory aid is difficult, you might think. "Peanuts!" says Charly. *By Charly Kühnast*

At work, I'm sometimes plagued by annoying gaps in my memory: What exactly was the name of that neat tool that I used to flash the LEDs on a specific network adapter to help me find the NIC in the rack? Or: How exactly did I delete all files that were more than a week old in a directory? The answer to all of these questions is in the Bash history, but Murphy's Law dictates that the history is always a little bit too short. And, in my case, there's another degree of uncertainty: Which server did I do this on?

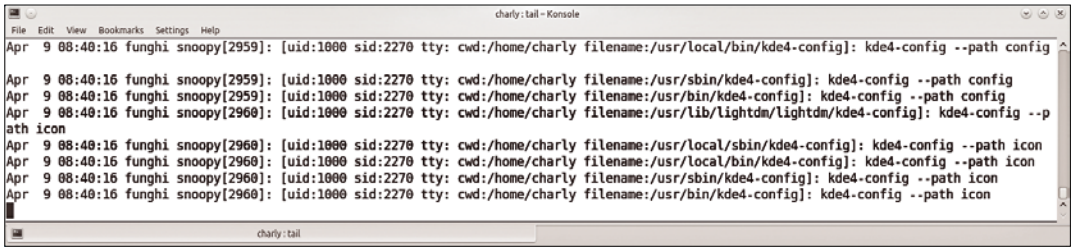


Figure 1: A neatly maintained history – thanks to Snoopy.

Snoopy potentially offers a solution. The small library with the dog's name, wraps around `execve()` and always wakes up when the computer runs a command. Many distributions have Snoopy in the pen, but if not, GitHub [1] will help you out. To enable Snoopy at boot time, you need an entry in `/etc/ld.so.preload`. I added the following line: `/<path>/snoopy.so`. The path is typically `lib`. If you are building Snoopy yourself, the library is likely to be found in `/usr/local/lib/` or something similar.

Building Snoopy yourself does offer some benefits. For example, you can edit the `snoopy.h` header file in the source up front. If you enter

```
#define ROOT_ONLY 1
```

Snoopy only logs commands that run with root privileges, but if you install the tool from the distribution repositories, this option is not set, and it logs any old command no matter who ran it.

Unless configured to do otherwise, Snoopy writes to `/var/log/auth.log`. Figure 1 shows the log for some simple commands. The structure always stays the same; each entry starts with the user ID, followed by the session ID and the TTY you use. This is then followed by the working directory, which is important because Snoopy does not log commands like `cd /etc`. Navigating the system is not the same for this dog as executing a file.

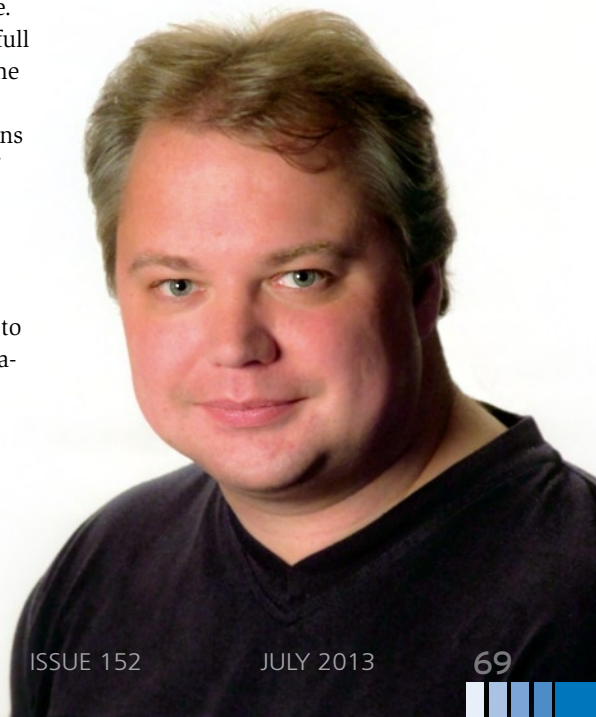
This information is followed by the full path to the executed file and, finally, the expanded command (e.g., aliases can cause an expansion). Many distributions run `ls --color=auto`, so, in this case, if you only type `ls`, Snoopy reveals all.

Collection Point

Now you just need to consolidate the logs centrally. I configured one server to accept the log messages from other machines. If the server runs rsyslog, you can just pass in the `-r` parameter at boot time to switch rsyslog to receive mode. Next, you can tell your other servers also to send entries in `/var/log/auth.log` to the newly configured syslog server. To do this, you just need to add one line to the syslog configuration:

```
auth,authpriv.*   @<192.168.2.80>
```

The auth log tends not to grow drastically, which means you can rotate on a weekly or even monthly basis. Snoopy fills a substantial log of my heroic deeds of administration day after day – including typos and similar peanuts. ■■■



INFO

[1] Snoopy: *https://github.com/a2o/snoopy*

CHARLY KÜHNAST

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.