The Sysadmin's Daily Grind: Arpalert

ARP WATCH

Corporate policies prohibit the unauthorized connection of hardware to the company network, threatening dire consequences in the case of non-compliance. Fair enough, but how do you actually go about catching somebody trying to plug an illegal laptop into your Ethernet? **BY CHARLY KÜHNAST**

My choice for a faithful watchdog is Arpalert [1]. Arpalert creator Thierry Fournier recommends the following incantation to send the beast off into the wild:

```
./configure --prefix=/usr/local
make
make install
```

This series of commands puts the C program in */usr/local/sbin* and the *arpalert.conf* configuration file in */usr/local/etc/arpalert*.

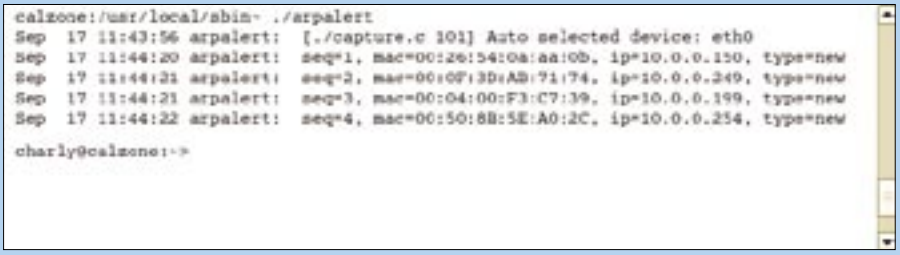
No Place Like Home

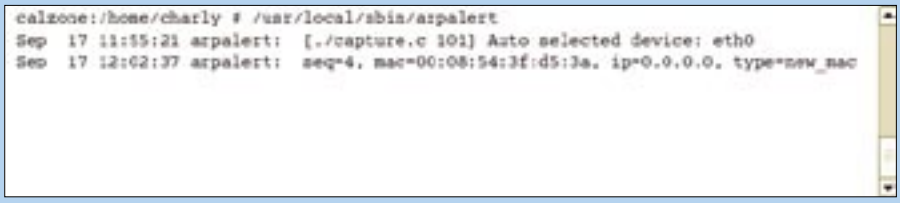
For my initial experiments, I decided to use a network that gives me excellent visibility, such as the network in my home office. It's the weekend, and my wife has gone down to the local library, so I shouldn't have more than four of five computers on the network. I did the following to launch Arpalert:

```
/usr/local/sbin/arpalert
```

I then sat back to see what would happen. The tool quickly assumed that I would want to use *eth0*; good guess, it being the only network adapter in the machine. If you have more than one network adapter, you might prefer to help Arpalert out by setting the *-i* flag and pointing to the right interface.

I left out the daemon mode parameter, *-d*, at first, but you actually need this option to monitor what Arpalert is doing on your screen. Things started to happen fairly quickly at this point: my watchdog detected the MACs of four machines in quick succession, including a printer and a WLAN access point (see Figure 1), writing the addresses in *MAC IP address* format to */usr/local/var/lib/arpalert/arpalert.leases*.





Figures 1 and 2: Arpalert detects the MACs of four devices in quick succession (top). The alarms go off at two minutes past twelve – an unknown machine has just connected (bottom).

As this is a fairly small network, I was fairly sure Arpalert had learned all the relevant addresses after a short while. I quit the program and then copied the address file, *arpalert.leases*, to */usr/local/etc/arpalert/maclist.allow* before relaunching Arpalert. From now on, Arpalert will pop up a message on the console, or create a log entry, whenever it detects an address that is not specified in *maclist.allow*.

To test this, I booted another machine, and sure enough I was alerted to the presence of the new computer (Figure 2). The IP address is 0.0.0.0 because the computer has not been serviced by the DHCP daemon at this point. I could use the *-e* option to tell Arpalert to run a script. The script could either mail me, or do something more drastic like modifying my packet filter rules.

Initial Conclusions

Arpalert performed perfectly on my miniature network, and I'm convinced that it will be useful for those of you with small, high-security networks, such as wireless LANs with just one or two dozen machines. In a larger environment, the tool would take too much manual attention, if it worked at all – as segmenting and VLANs would probably trip up Arpalert. ■

INFO

[1] Arpalert: *http://www.arpalert.org*

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).