Introduction

From time to time we'd like to be able to save details about a packet; perhaps it's malicious, or maybe we're just logging who connects to our IMAP server. The `−j LOG` target does perfectly well for this; it saves a text summary of the packet headers to syslog and they end up in /var/log/messages, or whatever file we've configured.

What if we want to log the *entire packet*? It turns out that's possible too.

Software setup

If you're using Linux kernel 2.4.18 or newer, support for ULOG is included in the core kernel. If you're using 2.4.17 or earlier, you'll need to compile your own kernel, using the patch−o−matic system to add the new modules. Make sure you include the ULOG target; we'll use this in a moment.

Unlike other firewall modules, you also need a userspace program called ulogd to actually do something with the packets. I have some rpms of the 1.02 release. Pull this program down and install it.

You'll want to edit `/etc/ulogd.conf`. For this example, comment out the `plugin /usr/lib/ulogd/ulogd_LOGEMU.so` line, and uncomment the `plugin /usr/lib/ulogd/ulogd_PCAP.so` line. Then start up the ulogd daemon with:

```
/etc/rc.d/init.d/ulogd start
```

`/var/log/ulogd.log` should hold a few lines saying the daemon started, and /var/log/ulogd.pcap should be 24 bytes long; this is a packet capture file with no packets yet. Let's fix that. :−)

Pulling packets out of the kernel

Let's do a simple test. We'll tell the kernel to log all packets destined for 99.99.99.99, and then create a few for it to log:

```
iptables −I OUTPUT −d 99.99.99.99 −j ULOG −−ulog−nlgroup 1 −−ulog−cprange 100
```

The nlgroup parameter is a kernel "netlink group". My best understanding of this is that by using different nlgroup numbers, you can start more than one copy ulogd, each listening on a different netlink group, and send packets to different files.

The cprange specified how many bytes of the packet to capture, similar to the "−s" snap length parameter for tcpdump. Rather than capturing the (default) entire packet, I only want to capture the first hundred bytes.

Now let's ping that address:

```
ping −c 5 99.99.99.99
```

If you take a directory listing, you'll notice that file as grown a little bit. Let's see what we have:

```
[root@sparrow root]# tcpdump −r /var/log/ulogd.pcap −qtnp
172.27.1.66 > 99.99.99.99: icmp: echo request (DF)
172.27.1.66 > 99.99.99.99: icmp: echo request (DF)
```

```
172.27.1.66 > 99.99.99.99: icmp: echo request (DF)
172.27.1.66 > 99.99.99.99: icmp: echo request (DF)
172.27.1.66 > 99.99.99.99: icmp: echo request (DF)
```

Bingo! We've now captured packets, directly from the kernel.

Watching packets live as they're pulled from the kernel

To see the packets live, simply run this command in a window:

```
tail −f /var/log/ulogd.pcap | tcpdump −r − −qtnp
```

Where to go next

Obviously, you can log any type of packets you'd like by simply changing the `iptables...−j ULOG...` line above. Once you've got the packets to ulogd, it's very flexible about what it will do for you. In addition to our above example of logging to a pcap (aka bpf, aka tcpdump) file, you can log packets to a text file or to a MySQL or Postgresql database. See `/etc/ulogd.conf` for information about these options.

Credits

Harald Welte, netfilter developer, wrote the ulog kernel module and the ulogd daemon − thanks Harald! Bill Stearns wrote this text.

William is an Open−Source developer, enthusiast, and advocate from New Hampshire, USA. His day job at SANS pays him to work on network security and Linux projects.

Last updated 12/4/2003.