

**Ethereal**

Re: [Ethereal-users] What is "Linux cooked capture" and why does it add 2 bytes?

Search: 

GO

[Home](#) | [Introduction](#) | [Download](#) | [Documentation](#) | [Lists](#) | [FAQ](#) | [Development](#) | [Wiki](#)[Main Index](#) | [Thread Index](#) | [Mailing Lists](#) | | [Date Prev](#) | [Date Next](#) | [Thread Prev](#) | [Thread Next](#)**Ethereal**Advanced Network Analyzer  
Solution. Start Analyzing Today.  
[www.Paessler.com/Packet-Sniffing](http://www.Paessler.com/Packet-Sniffing)**Analyze User & Flow Data**Leverage NetFlow & Active  
Directory to monitor operations &  
compliance  
[www.securify.com](http://www.securify.com)**Robust Capture Replay**Reshape, Scale, Replay Pcaps Any  
protocol including peer to peer  
[www.CaptureReplay.com](http://www.CaptureReplay.com)**Run SNORT up to 10 Gbps**Application Acceleration, DPI and  
Granular Flow Processing.  
[www.netronome.com](http://www.netronome.com)

Ads by Google

**ETHEREAL-USERS: DECEMBER 2004**

- *Subject:* Re: [Ethereal-users] What is "Linux cooked capture" and why does it add 2 bytes?
- *From:* Guy Harris <[gharris@xxxxxxxxx](mailto:gharris@xxxxxxxxx)>
- *Date:* Tue, 28 Dec 2004 11:49:45 -0800

Rutger Thomschitz wrote:

However, just recently, as I was doing an experiment with VTun (an opensource VPN solution) I noticed Ethereal reporting "Linux cooked capture", which seems to add an additional 2 bytes. What is "Linux cooked capture"?

On Linux, packet capturing is done by opening a socket. In systems with a 2.2 or later kernel, the socket is a PF\_PACKET socket, either of type SOCK\_RAW or SOCK\_DGRAM.

A SOCK\_RAW socket supplies the packet data including what the driver specified, when constructing the socket buffer (skbuff) holding the packet, to be the packet's link-layer header; a SOCK\_DGRAM packet supplies only data above what was specified by the driver to be the link-layer header.

For the purposes of libpcap, which is the library used by programs such as tcpdump, Ethereal/Tethereal, snort, etc. to capture network traffic, a SOCK\_RAW socket is usually the appropriate type of socket on which to capture, and is what's used.

Unfortunately, the purported link-layer header might be missing (as is the case for some PPP interfaces), or might contain random unpredictable amounts of data (as is the case for at least some interfaces using ISDN), or might not contain enough data to determine the type of the packet (as is the case with at least some ATM interfaces), so capturing with a SOCK\_RAW socket doesn't always work well.

For interfaces of those types - and for interfaces of a type that libpcap currently doesn't have code to support - libpcap uses a SOCK\_DGRAM socket, and constructs a fake link-layer header from the address supplied by a "recvfrom()" on that socket.

A "Linux cooked capture" is one done with libpcap using a SOCK\_DGRAM socket.

**References:**

- [\[Ethereal-users\] What is "Linux cooked capture" and why does it add 2 bytes?](#)
  - *From:* Rutger Thomschitz
- Prev by Date: [RE: \[Ethereal-users\] Ethereal 0.10.8 crashes with page fault in msvcrt.dll](#)
- Next by Date: [Re: \[Ethereal-users\] Hidden text \)](#)
- Previous by thread: [\[Ethereal-users\] What is "Linux cooked capture" and why does it add 2 bytes?](#)
- Next by thread: [\[Ethereal-users\] MSVCRT.DLL illegal operation on Windows 98](#)
- Index(es):
  - [Date](#)
  - [Thread](#)