

Ethereal Power

Roderick W. Smith

One of the most important tools in a network administrator's toolbox is a packet sniffer, a program that intercepts and displays network traffic. Using a packet sniffer, you can examine the individual bytes that make up packets sent across the network wire. Depending on your network configuration, packet sniffers can see traffic sent to or from the computer on which the sniffer runs and even monitor traffic between other computers on the network.

A packet sniffer is quite a powerful tool, but because it operates at such a low level, running one also requires greater expertise than many other network diagnostic tools. Advanced packet sniffers (often called protocol analyzers) can avoid such complications by helping you interpret data packets.

One of the more popular and flexible packet sniffers for Linux is Ethereal. Ethereal has protocol analysis features and provides a graphical user interface (GUI). Its protocol analysis features include information on a wide variety of network protocols, enabling Ethereal to parse the data in the packets, greatly simplifying data analysis.

Uses and Abuses of Packet Sniffers

Before proceeding further, you should be aware that, although packet sniffers are very useful and legitimate tools, the tools can also be abused by crackers. Crackers typically use packet sniffers to look for unencrypted passwords and other sensitive data. For this reason, many organizations forbid the use of packet sniffers except under certain limited conditions. Before deploying a packet sniffer, be sure to obtain written permission from somebody who's authorized to grant that permission, lest you be accused of wrongdoing, get fired, or even prosecuted for breaking the law.

Packet sniffers are commonly used as a troubleshooting tool. By using a packet sniffer, you can verify that packets are (or are not) reaching their destination. More significantly, by examining the packets' contents, you can discover whether particular features are active, check that clients or servers are sending the data they should be sending, and so on.

For instance, suppose you've configured Samba to use encrypted passwords. To be absolutely certain that encrypted passwords are being used, you can use a packet sniffer to monitor a password exchange that you initiate. If you can see an unencrypted password in the packets being sent by the client, then you know that something is wrong with your configuration. If you can't see such a password, then you can be reasonably sure that Samba and the client you tested are using encrypted passwords. Ethereal can help on this score by positively identifying the password string, removing all doubt that you've correctly identified it.

These very features are the ones that make packet sniffers attractive to miscreants. If a cracker sets up Ethereal on your local network and monitors the traffic to and from a Samba server (or any other server that accepts passwords), the cracker has access to any passwords that are exchanged. Passwords aren't the only problem, though: by monitoring data exchanges, the intruder may be able to reconstruct entire documents, such as confidential budget files that are emailed, stored on a file server, or even printed.

You can employ several practices to help protect yourself against malicious sniffing:

- Encryption can render intercepted data useless, so you should use encrypted protocols whenever possible. (You can use your own packet sniffer to verify that encryption is active.)
- Use switches rather than hubs on an Ethernet network. Switches direct traffic from the sending system only to the recipient, whereas hubs echo all traffic to all connected computers. Thus, it's harder to intercept data from a third computer if your network uses switches.

Ethereal Power

Roderick W. Smith

- Avoid wireless connections whenever possible, as a wireless signal can be intercepted even off your premises. If you must use wireless connections, enable the wireless hardware's encryption system and use encrypted protocols for any remotely sensitive data transfers. (The encryption built into wireless hardware is notoriously weak.)

Ethereal and Network Setup

To use Ethereal, you must first decide where to run the program. Several possibilities exist:

- On a server. Running Ethereal on a server guarantees that you can monitor traffic to and from that server. On the other hand, this placement tends to degrade the server's performance, which may be unacceptable for some servers.
- On the client. You can run Ethereal on a client system, which guarantees that you'll be able to monitor that client's traffic, but possibly not traffic between other systems. This placement also degrades the client's performance, but that may be acceptable if you just want to perform a few tests.
- On a third system. It's possible to run Ethereal on a third system. This approach only works if that system can monitor the packets sent between other computers. You can replace a switch with a hub to let Ethereal monitor data to and from all the computers connected to that hub. You might even place a hub between a client or server and its normal switch, then attach the Ethereal system to the hub. This configuration enables a dedicated Ethereal system to monitor all the traffic to and from a given client or server without burdening the targeted system or degrading overall network performance. High-end switches can often be configured to echo data to particular systems to a third computer, which can be a useful feature in such situations.

Once you've determined where to set up Ethereal, you must do install the package. Many Linux distributions ship with Ethereal, so check your distribution's package set and install from that, if possible. If you can't find Ethereal in that way, go to its web site and download it.

Ethereal requires certain kernel options to be set in order to work. In particular, you must activate the "Device Drivers, Networking Support, Networking Options, Networking Options, Packet Socket" option in the kernel configuration tool (when using `make xconfig` or a similar option when compiling the kernel). This option enables user programs, such as Ethereal, to interact with packets at a low level without the help of the kernel's usual TCP/IP stack. Most distributions enable this feature by default, but if you have problems getting Ethereal to monitor traffic, you may want to check its status.

Normally, most network interface hardware filters packets based on low-level hardware addresses, meaning that even low-level kernel routines won't "see" packets addressed to other computers. To work around this limitation, Ethereal normally reconfigures the network hardware to operate in promiscuous mode, which means that the interface delivers packets addressed to any computer. This is necessary when Ethereal runs on a computer other than the one whose traffic it's to monitor, but it can increase the work done by the kernel if you don't care about other systems' data packets, depending on the network hardware. If you don't want to put the network interface into promiscuous mode, you can pass the `-p` option to the `ethereal` command or change the promiscuous mode option in the Ethereal interface. Note that the interface might still be in promiscuous mode because of other programs' activities, though, and Ethereal won't change this.

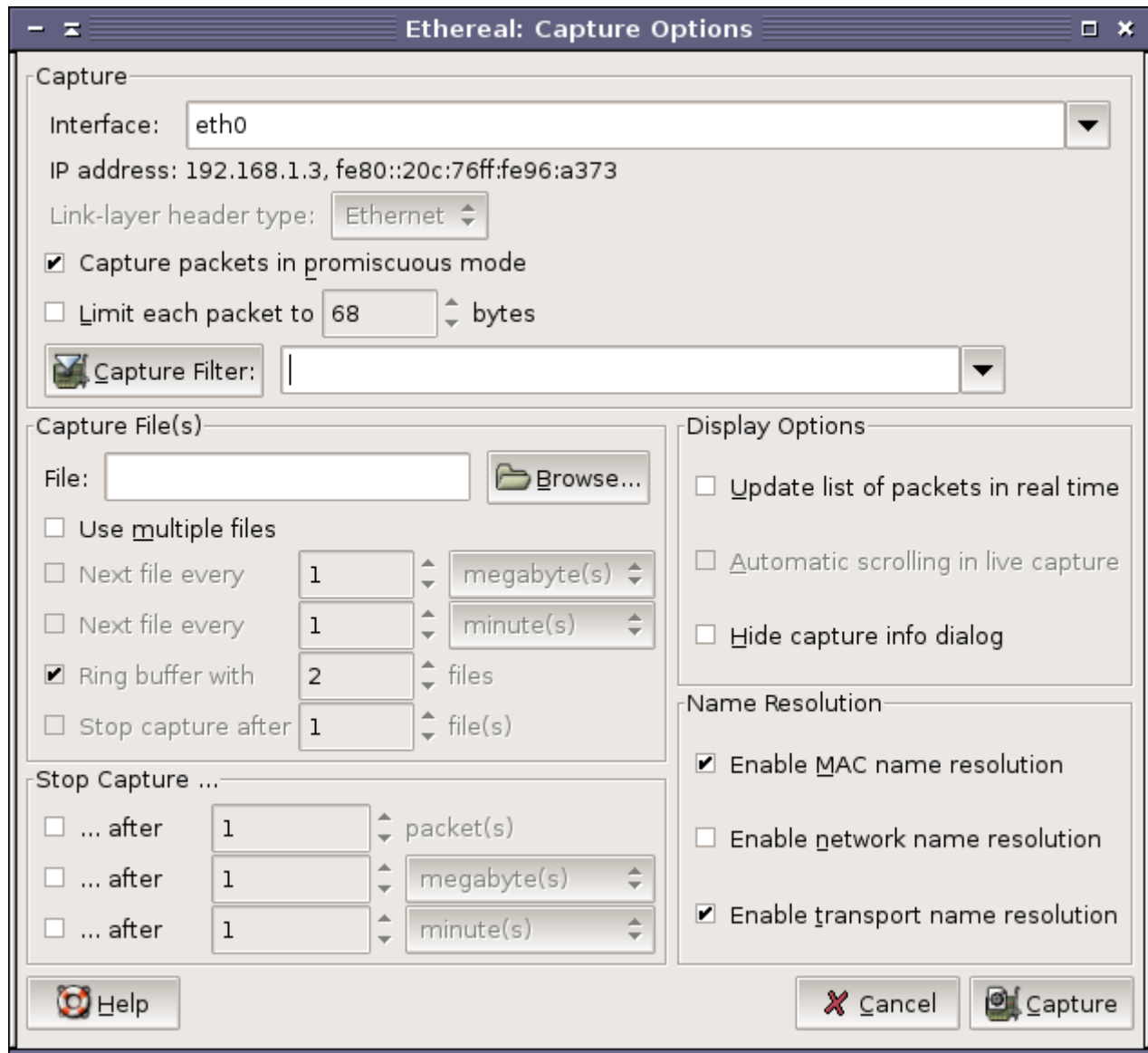
Monitoring Your Traffic without a Filter

To use Ethereal, type `ethereal` as root in an `xterm` or similar GUI command prompt window. The result is the main Ethereal window, which initially is mostly empty, consisting mainly of its menu bar and icon bar shortcuts.

Ethereal Power

Roderick W. Smith

Figure One: You can set a variety of options when you tell Ethereal to begin capturing packets.



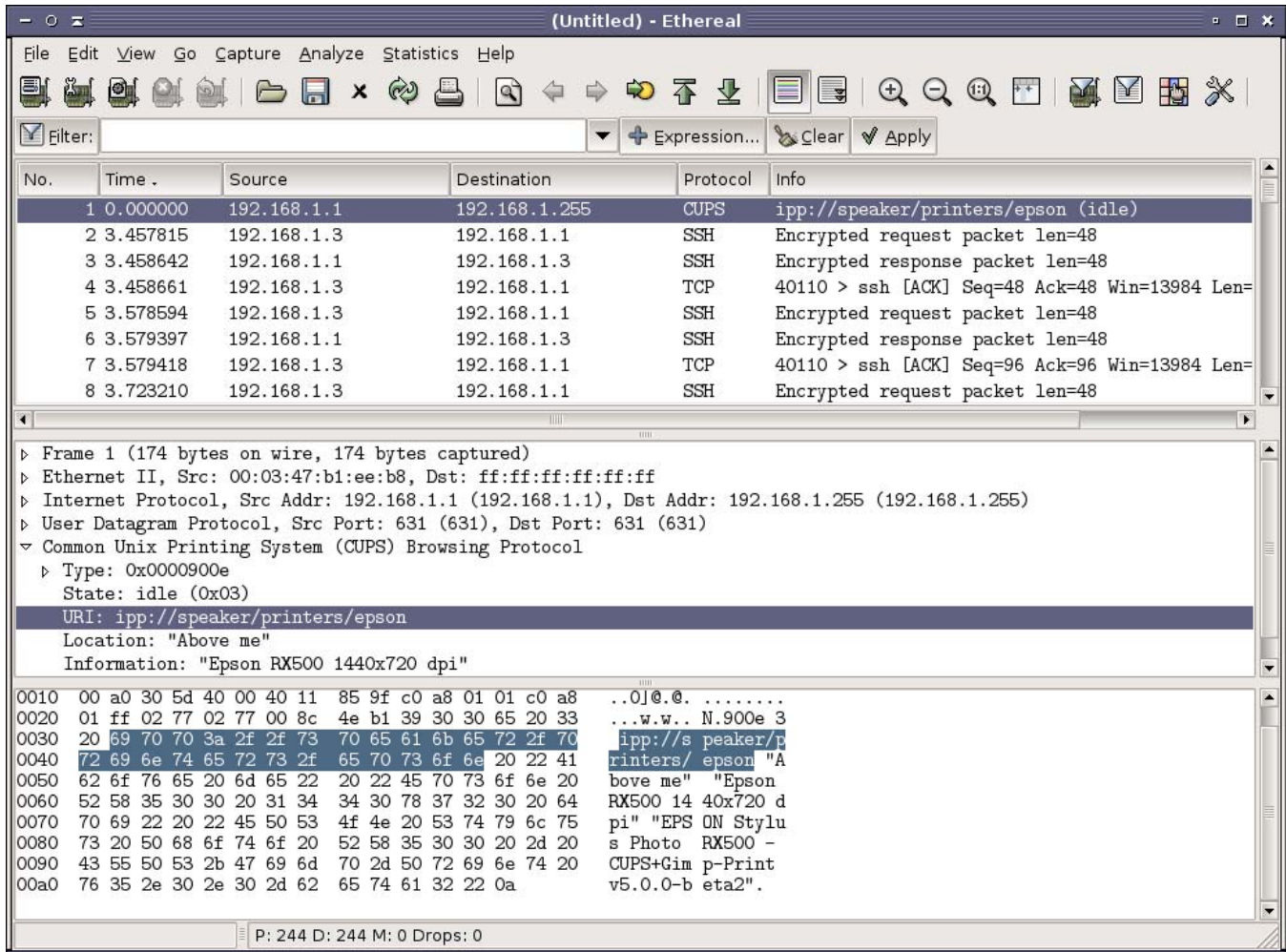
To begin capturing network packets for analysis, follow these steps:

1. Pick the "Capture, Options" menu item. The result is the "Ethereal: Capture Options" window shown in Figure One.
2. To perform a basic test, simply click Capture. Ethereal displays a window entitled "Ethereal: Capture", in which it summarizes the number and type of packets it's recording (TCP, UDP, ICMP, and so on).
3. Perform some actions that produce network traffic, such as pinging another computer or initiating an SSH connection.
4. Click the Stop button in the "Ethereal: Capture" window. Once pressed, Ethereal summarizes the packets you've captured in its main window, as shown in Figure Two.

Ethereal Power

Roderick W. Smith

Figure Two: Ethereal's main window is dominated by a display of captured packets



You can begin perusing the data you've captured, if you like. How to analyze it is explained in more detail shortly in "Analyzing the Data You Sniff."

You may have captured far more packets than you would like — Ethereal might have captured extraneous packets that were generated by automated network tools, packets generated by other users, packets sent between other computers, and so on. For instance, Figure Two shows packets associated with both Common Unix Printing System (CUPS) and SSH. If you're only interested in packets for particular protocols, computers, or otherwise restricted, a capture of everything may be overkill. Fortunately, you can configure Ethereal to restrict the packets that it captures.

Monitoring Specific Types of Traffic

The key to restricting the data Ethereal captures lies in the "Ethereal: Capture Options" dialog box (Figure One). The quickest way to begin is to enter capture criteria in the text-entry field to the right of the Capture Filter button in the "Capture" area of the dialog box.

The syntax used to describe what types of packets to capture is fairly complex and is described in the tcpdump man page. (tcpdump is another packet sniffer that shares the same capture filter syntax with Ethereal, although tcpdump refers to these as expressions, so search the man page for that word.) By

Ethereal Power

Roderick W. Smith

the way, Ethereal's capture filters are not the same as its display filters, so don't try to use an Ethereal display filter (as described in the ethereal-filter man page) as a capture filter.

Capture filters consist of one or more primitives. Each primitive consists of one or more qualifiers (keywords) followed by an identifier (a name or number). Qualifiers may specify computers, networks, or ports (host, net, or port); direction of transfer (src, dst, src or dest, or src and dest); or protocol (ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp, udp, or icmp). A few special primitives are also available, such as broadcast to capture broadcasts. Table One presents some examples that should help you understand how to construct filters.

Table One: Examples of Ethereal capture filter primitives	
Primitive	Meaning
tcp port 25	Captures all TCP traffic to or from port 25
tcp dst port 25	Captures all TCP traffic to (but not from) port 25
host tinfoil.example.com	Captures all traffic to or from tinfoil.example.com
src tinfoil.example.com	Captures all traffic originating from tinfoil.example.com
net 192.168.1	Captures all traffic to or from the 192.168.1.0/24 network

You can combine capture filter primitives using the and, or, and not keywords, as in tcp port 25 and dst 172.24.21.1 to find all traffic directed to 172.24.21.1's port 25.

Analyzing the Data You Sniff

Now that you know how to capture packets with Ethereal, it's time to begin analyzing them. Precisely how you'll proceed depends, of course, on precisely what you hope to accomplish. To begin, though, you should understand how Ethereal presents its data.

Once you've captured some packets, Ethereal presents data on those packets in three panes, as shown in Figure Two:

- The top pane shows a summary of the packets, one packet per line. Depending upon your reason for running Ethereal, this may be all you need. For instance, you can verify that packets are being transferred to and from particular computers from the top pane.
- The bottom pane shows the raw data associated with each captured packet, both in hexadecimal and ASCII form. Sometimes the raw data is useful. For instance, you might be able to spot a cleartext password or parts of an ASCII data file in the raw data. If you're intimately familiar with the protocol, you can also study the raw data using this pane.
- The middle pane shows an analysis of the packet in protocol tree format; that is, the contents are interpreted in a hierarchical manner, with low-level features (such as Ethernet frame data) in the upper lines and high-level features (such as the analysis of CUPS information in Figure Two) in the lower lines. The contents of this pane vary depending on the type of packet, but in most cases, this is the pane you'll be using to perform in-depth analyses of the packets Ethereal captures.

To analyze your data, pick an interesting packet from the top summary pane. You might want to enlarge the middle protocol tree frame so that you can see as many elements as possible. As initially presented, you'll only see one entry for each level of the analysis; however, when you click on the

Ethereal Power

Roderick W. Smith

triangle to the left of a line, it expands, showing you information on all the fields that apply to this level in the protocol tree.

For instance, Figure Two shows the CUPS Browsing Protocol item expanded. This reveals several pieces of information that Ethereal has identified as being part of the CUPS protocol, such as a state, a URI, a location string, and an information string. If you click one of these elements in the protocol tree pane, the corresponding areas are highlighted in the raw data pane, as shown in Figure Two. This can be a good way to learn how data packets are put together.

Sometimes, the expansion of elements can continue for several levels. For example, in Figure Two, look at the triangle to the left of the Type field. You'd click on it to see a breakdown of what the 0x0000900e code means.

Ethereal is an extremely powerful tool, and you can do a lot with it. For instance, you can use display filters to search for filters that meet certain criteria among those you've captured. (Display filter syntax is described by the `ethereal-filter` man page, and is quite different from capture filter syntax.) You can apply display filter rules to have Ethereal color packets that match certain criteria. You can save packet filter captures for later analysis by Ethereal or by other programs.

Consult the Ethereal man page for more information, or use the items on the "Help" menu.