Using Ethereal As A Tool For Network Security

Mentor: Mr. Christopher Edwards
Team Members: Jerome Mitchell, Anthony Anderson, and Napoleon Paxton

Abstract

The Office of Navel Research Network Team actively listened to network traffic to fingerprint transmitted data packets that could potentially affect the availability of resources within the ONR Local Area Network (LAN) segment. Network traffic was examined using ethereal graphical user interface to identify and analyze Transmission Control and User Datagram Protocol packets to and from end-user hosts and Elizabeth City State University (ECSU) campus intranet servers. Captured packet frames were decoded to see if a problem exists with a packet. Capture statements were created to find out what traffic is crossing the network, identify unauthorized protocols, and identify the top talkers.

During the 2004 – 2005 Network Research Program the ONR Network Team limited its research and discovery phase to understanding the various methods to observe, capture, identify, analyze, and decode packets within a packet switched Local Area Network. To further the analysis of packet capturing the ONR Network Research Team will expand its research and discovery during the 2005 - 2006 program to develop a network diagram to determine the best place to capture traffic for analysis campus wide monitoring during different times of the day instead of once a two times a week during ONR mentoring sessions.

The development of an active packet monitoring network team can help the ONR network-mentoring program strengthen the capabilities of the team members, helped the ECSU Mathematics and Computer Science Department develop a new course to its program, and/or turnover over the research to the ECSU IT department for them to develop an network analysis vulnerability prevention program using packet analyzers and sniffers.

Introduction

Over the last fifty years the world has seen incredible advances in the area of computers and technology. The Internet is arguably the most remarkable of them all. In a matter of seconds information for a report can be obtained in the dormitory of a student from Russia, or any other distant place. This ease of information sharing has greatly benefited the world. Unfortunately, this access to the world also provides an equally easy way to access illegal items or other material deemed inappropriate by whoever is in control of the network. Many colleges and universities have the same problem, which are students visiting web pages that are unauthorized. Pornographic web sites are the most common among the unauthorized sites. The network administrator is the person that has the responsibility to deal with all matters that pertain to the network, which includes reducing the amount of unauthorized web sites viewed in a particular network. There are many products on the market that are able to block certain types of sites from being

accessed. These products are referred to as firewalls. Most of the firewalls on the market are expensive, and therefore are not feasible solutions. Another more economically friendly solution is needed to deter students from trying to access unauthorized web sites all together.

Methods

" The basic tool for observing the messages exchanged between executing protocol entities is called a **packet Sniffer**. As the name suggests, a packet Sniffer captures ("sniffs") messages being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured messages. A packet Sniffer itself is passive. It observes messages being sent and received by applications and protocols' running on your computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet Sniffer. Instead, a packet Sniffer receives a *copy* of packets that are sent/received from/by application and protocols executing on your machine. Figure 1 shows the structure of a packet Sniffer. At the right of Figure 1 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet Sniffer, shown within the dashed rectangle in Figure 1 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. Recall from the discussion from section 1.7.2 in the text (Figure 1.181) that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable. In Figure 1, the assumed physical media is an Ethernet, and so all-upper layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you all messages sent/received from/by all protocols and applications executing in your computer.

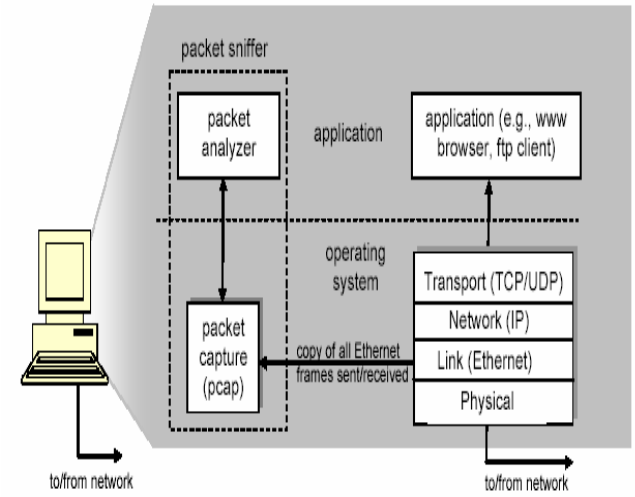


Figure 1: Packet sniffer structure

The second component of a packet Sniffer is the **packet analyzer**, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in Figure 1. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment structure, so it can extract the HTTP message

contained in the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD," as shown in Figure 2.8 in the text."

Getting Ethereal

In order to use ethereal, the following steps are necessary:
Download a copy of ethereal to a folder on your local hard drive named ethereal. After downloading ethereal, download a copy WinPcap. WinPcap is a driver that ethereal needs in order to run. WinPcap is an open source library for packet captures and network analysis for the Win32 platforms. It includes a kernel-level packet filter, a low-level dynamic link library (packet.dll), and a high-level and system-independent library (wpcap.dll, based on libpcap version 0.6.2) [1 http://winpcap.polito.it/]

Running Ethereal

To open it, click on the start menu and double click.

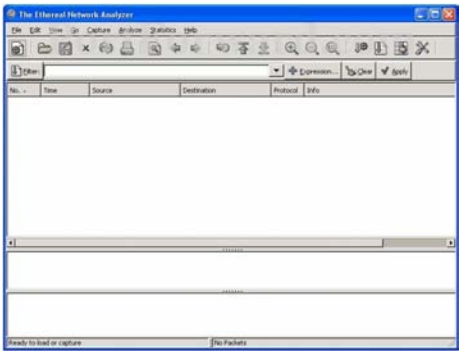


Figure 1 shows an ethereal open dialogue box.

Ethereal involves the capability of seeing exactly what data is being traveled over a network, what ports a program is using, and what comes in and out of a router; this can cause a problem of receiving abundant data. The problem can be corrected by supplying a filter to the data being captured. A filter sorts data based on the user's preferences. To capture a packet, click on the capture tab on the Ethereal open dialogue box; then click start. When the capture dialogue box appears, the user can then start capturing data. The ethereal software package allows the user to select the network card to choose data from; this is accomplished by using the Interface combo box.

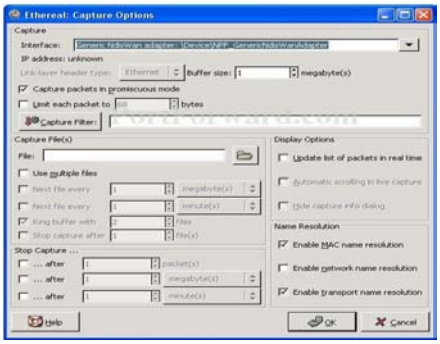


Figure 2 shows the Interface selection in green of the network card.

On the Interface combo box select the capture filter box; this allows the user to start capturing packets. When the capture filter box is selected, the capture filter dialog box appears.

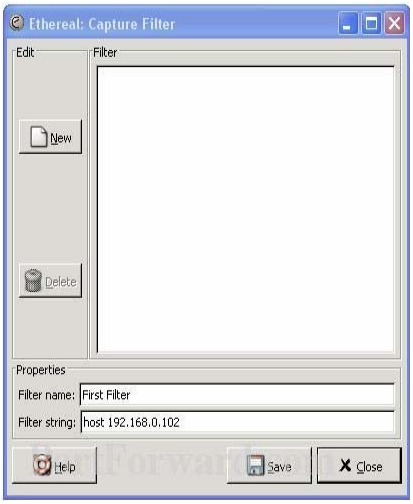
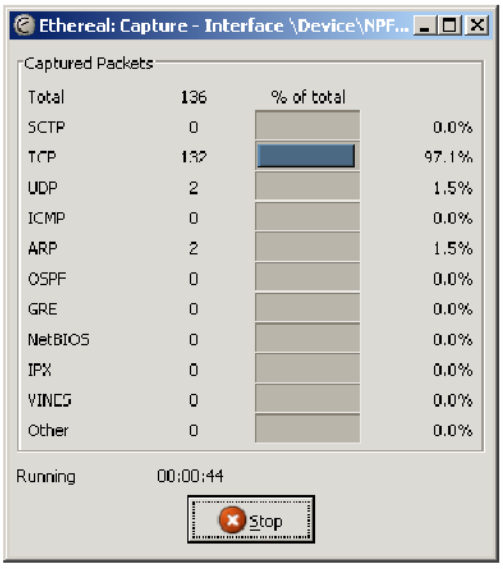


Figure 3 depicts a filter capture for a workstation with the IP address 192. 168.0.102

In this dialogue box, the user puts “First Filter” in the Filter name box under the properties selection. The IP (internet protocol) address of the workstation the user is currently using is supplied to the Filter string. By completing these two steps, click the save button. The user is telling Ethereal to capture everything coming from and going to the IP address. The user will get a log of all the traffic that is coming from or going to your computer. After saving, the Interface dialogue box reappears. Click Ok at the bottom of this screen to start the capture and a capture dialog box will appear displaying the number of packets, increasing continuously as packets are captured.



Three Packet capture files were created and limited to 1000 packets for analysis. The capture files were analyzed to identify all network protocols that are transmitted across the LAN, find any unauthorized traffic, and identify the two top talkers.

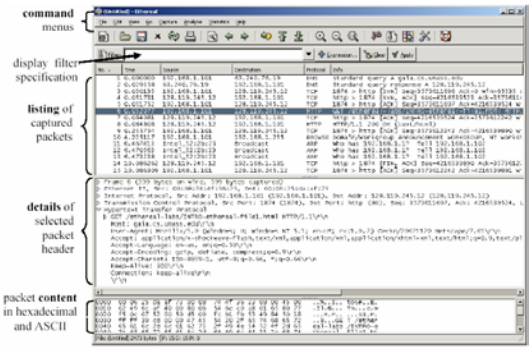


Figure 2: Ethereal Graphical User Interface

For instance, “While Ethereal is running, the ONR Network Team entered the URL: http://umfort.cs.ecsu.edu to have that page displayed in the ONR host browser. In order to display this page, the ONR host browser will contact the HTTP server at umfort.cs.ecsu.edu and exchange HTTP messages with the server in order to download this page. The Ethernet frames containing these HTTP messages will be captured by Ethereal. After the ONR host browser has displayed http://umfort.cs.ecsu.edu, stop ethereal packet capture by selecting stop in the Ethereal capture window. This will cause the Ethereal capture window to disappear and the main Ethereal window to display all packets captured since the ONR Network Team began packet capture. The main ethereal window now looks similar to Figure 5 pictured below. The ONR Network Team now have live packet data that contains all protocol messages exchanged between the ONR host computer and other hosts within the ONR LAN segment! The HTTP message exchanges with ONR’s mail server, umfort.cs.ecsu.edu appeared in the listing of packets captured along with many other types of packets displayed as well. Even though the ONR Network Team only downloaded a web page, there were many other protocols running in the background that were unseen by the ONR Network Team and visible to

the LAN switches and routers. To only have HTTP messages displayed in the packet-listing window the ONR Network Team entered in “http” without the quotes, into the display filter specification window at the top of the main Ethereal window and selected Apply. The ONR Network Team selected the first http message shown in the packet-listing window that was sent from the ONR host to the umfort.cs.ecsu.edu mail server. When the ONR Network Team selected the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information was displayed in the packet-header window. http://www.geocities.com/packetanalyzer.

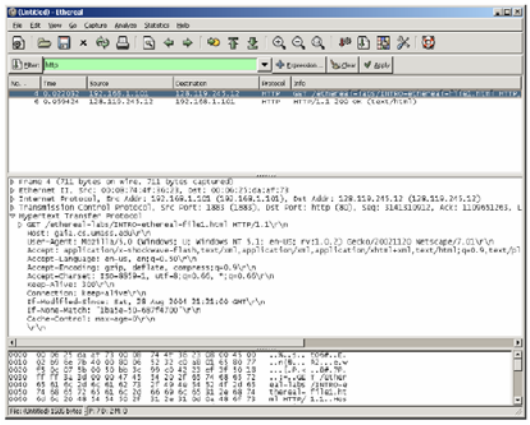


Figure 5: Ethereal display after step 9

Understanding Findings

The two areas of interest the ONR Network Team focus on were how to use filters and capture only the information of interest to you and how to read the outputs. The Team captured Network traffic on the network between two hosts by Snooping Passwords from an FTP session between UMFORT and the ONR host. To perform this step, the ONR Network Team executed the following steps:

1.On the ONR host, the ethereal command with capture filters was ran set to capture traffic between the ONR teams host and UMFORT.
2.On the ONR host, initiate a FTP session to UMFORT by typing ftp://umfort.cs.ecsu.edu/
3. Log on to UMFORT – using your assigned username and password or as anonymous.
4.Inspect the payload (data) in the packets that are sent between the ONR host and UMFORT. During this activity the ONR Network team learned that FTP sessions use TCP connections for data transfer. In ethereal, there is a simple method to view the payload sent in a TCP connection. Simply select a packet that contains a TCP segment in the main window of ethereal, and then click on "Follow TCP Stream" in the "Tools" menu of the ethereal window. This will create a new window that displays only the payload of the selected TCP connection.
5.Save the details of the packets, select “Print details” in the “”Print” window of ethereal, which transmit the login name and password. Set the display filter in ethereal to show only the desired packet(s).
From the saved output collected the ONR Network Team identified the port numbers of the ONR host (FTP client) and the UMFORT (FTP server) as well as, identify the login name and the password, shown in plain text in the payload of the packets that were captured.
During this exercise the ONR Team realize the benefit of how to use a filter properly and how a filter can be used to remove broadcast traffic. The first captured traffic reviews the ONR team performed; the teamed noticed a lot of packets with destination addresses that

were continually displayed as FFFFFFFF or always ends in .255 (IP). After investigating, the ONR Team understood if the packet capture file contains only broadcast traffic, then only broadcast packets have been captured and they are not useful.

Conclusion

Observing and examining the sequence of message exchanges between two protocol entities allowed the ONR Network Team to investigate the particulars of protocol operation, and protocols used to perform denial of service attacks that could potentially affect the availability of resources within the ONR Local Area Network (LAN) segment. The analysis of the transmitted data packets using capture statements were helpful in decoding what traffic is crossing the network, identifying unauthorized protocols to see if a problem exists with a packet, and identify the top talkers consuming a high rate of network bandwidth.

For the 2004 – 2005 ONR Academic year, the ONR Network Team focused their research this year learning why and how to become familiar with a tool that can be used for probing what's happening at the Transport, Network, and Link Level of hosts within the ONR Local Area Network (LAN) segment.

To further the analysis of watching and understand packets traveling across the network, the ONR Network Research Team will expand its research and discovery during the 2005 - 2006 program to perform campus wide monitoring during different times of the day instead of once a day. Moreover, The development of an active packet monitoring network team can help the ONR network mentoring program and ECSU IT Department strengthen its capabilities by holding "monthly lessons learned meetings to share information about the normal behaviors of networks, systems, and applications. After each meeting, all network analysis vulnerability prevention teams who use packet analyzers and sniffers to understand normal behavior will be able to recognize abnormal behavior more easily and better protect the assets of ECSU". [NIST 800-61, Computer Security Incident Handling Guide, Section 3.6 Recommendations, p.3-27 - 3-29, http://csrc.nist.gov/publications/nistpubs/800-61/sp800-61.pdf