10-2005

# A Taxonomy of Free Network Sniffers for Teaching and Research

Victor A. Clincy
*Kennesaw State University*, vclincy@kennesaw.edu

Halaweh Abu Nael
*Kennesaw State University*

## Recommended Citation

Victor A. Clincy and Nael Abu-Halaweh. 2005. A Taxonomy of free Network Sniffers for teaching and research. J. Comput. Small Coll. 21, 1 (October 2005), 64-75.

# A TAXONOMY OF FREE NETWORK SNIFFERS FOR TEACHING AND RESEARCH[*]

*Victor A Clincy and Nael Abu-Halaweh*
*Kennesaw State University*
*Kennesaw, Georgia 30144*
*vclincy@kennesaw.edu, 770-420-4440*

## ABSTRACT

Today's networking environment has become very complex. Networks have been growing in size rapidly and have come to support more complex applications. As result, troubleshooting and maintaining networks has become cumbersome and has created the need for new specialized tools such as Network Protocol Analyzers, better known as "Network Sniffers".

Network Sniffers have become critical tools in today's networking management and troubleshooting processes. They enable network managers to evaluate and examine the data running through their network by troubleshooting network performance problems and identifying certain network faults. Network Sniffers can help identify network attacks and detect security threats; they can be used in intrusion detection systems.

Besides their usage in the technical environment, network sniffers can be used for educational and research purposes. They can be used to help understand packets' architecture and traffic patterns generated by common network applications. Network Sniffers can also be used to evaluate protocol performance and assist in protocol development. Despite their usefulness, network sniffers can be harmful when used by hackers. With network sniffers, hackers can capture data and steal information from targeted networks.

This study consists of two major efforts. The first major effort entails researching and determining a set of criteria to use in evaluating and comparing network sniffers. The second major effort involves using the criteria to evaluate and compare three free network sniffers, thus building a taxonomy. The three free network sniffers used in this study were Ethereal, EtherSnoop and Packetyzer. Each of these three sniffers was evaluated and tested. Then their features and capabilities were compared.

## INTRODUCTION

The scope of the original sniffers was limited to capturing message headers of data packets on networks. This enabled network administrators the ability to view low-level information about data packets such as source and destination addresses, file size and transmissions. Textual and graphical representation of data was used to troubleshoot network performance problems and to identify their sources. This approach required the network administrator to go offline to analyze the data they captured from the network, which caused the approach to be very time consuming and slow to responding to network problems. As a result of the earlier approaches, any suspicious network activity or its impact could possibly not be detected until it is completed.

The newer sniffers are more efficient, allowing for near to real-time or real time analysis. The newer sniffers capture packets from the network and decode them into human readable format. In addition to that, the newer sniffers are able to analyze the packets and display the results. With the real-time approach of the newer sniffers, responses to network problems are much more timely. Some of the newer network sniffers can be configured by network administrators to send an alert when a specific network event occurs. Examples of network events could be slow HTTP response or too many TCP retransmissions.

Current sniffers can analyze data from all the seven layers of the OSI reference model, rather than just displaying the source and destination addresses. In fact, some of them are capable of recommending solutions to some network problems. If the application level analysis fails to identify the problem and find a solution, sniffers can dig into lower-level details.

Sniffers are also capable of providing graphical and statistical data about communication systems. Sniffers can provide graphical and statistical data about the volume of traffic passing between two communicating peers. This statistical data can provide an overall view of traffic patterns. More detailed statistics are also available. The network analyst can get exact percentages of network traffic attributed by a specific protocol or application.

Modern sniffers incorporate monitoring standards that define a standard way for systems to capture key performance data, and compare this data against historical data and trigger an alert when a threshold value is exceeded.

## OVERVIEW OF NETWORK SNIFFERS

### TYPES OF NETWORK SNIFFERS

Two broad types of network sniffers are available. The first one is a standalone product incorporated into a portable computer that can easily be connected to a network. This type of network sniffer is usually used by network consultants to plug into their customers' networks and capture data. The second general type of network sniffer comes as a part of an overall network-monitoring hardware and software package. These types of network sniffers give network administrators a centralized view of their networks and enable them to monitor high-level activities, such as running applications, users currently logged into to the network, and the sources of high traffic volumes.

## COMPONENTS OF NETWORK SNIFFERS

A typical network sniffer has the following components: hardware, capture driver, buffer, real-time analysis features, decoder and packet editor. An overview of each feature:

1. *Hardware:* most network sniffers work with standard network adapters, though some of them require specialized hardware. Those requiring specialized hardware typically have more analytical capabilities such as the capability to analyze CRC errors, jitter and others.
2. *Capture driver:* the capture driver is the most important component of a network sniffer. It captures data from the network, filters it according to criteria set by the user and stores the data into a buffer.
3. *Buffer:* the buffer is used to store the captured data. It operates in two modes. One mode is for the sniffer to capture until the buffer is filled up. Another mode is for the sniffer to use a round-robin approach where the newest data replace the oldest data. Some sniffers have the capability to store the captured data to a hard drive allowing storage of hundreds of gigabytes of data.
4. *Real-time analysis:* this feature analyzes the data as it comes off the wire. It has the capability to detect network performance issues and faults while capturing.
5. *Decoder:* The decoder component displays the captured network data using descriptive text, so that the analyst will be able to understand the captured data and to figure out network issues.
6. *Packet editor/transmission:* the editing feature allows the user the ability to edit captured packets and to retransmit them on the network.

## HOW NETWORK SNIFFERS WORK

Most network sniffers operate in the same way and display the same basic information. Typically, a network interface card (NIC) on the host computer is set to filter out traffic that is addressed to it. A network sniffer running on a host system puts the network interface card (NIC) into the promiscuous mode turning off the filter and enabling the NIC to capture all traffic passing through it. The captured traffic is passed to a packet engine decoder that identifies packets and splits them into their respective layers. The captured network data is typically displayed in a three-pane window. The top pane displays a summary of data packets captured. Typically this pane shows date and time the packet was captured, source and destination IP and port addresses, protocol type, and a summary of the packet data. The middle pane shows the logical breakout of a selected packet, and the third pane shows the packet in a hexadecimal or ASCII format.

## WHERE TO DEPLOY NETWORK SNIFFERS

Some network topologies such as Ethernet is designed so that all machines connected to a network segment will share the same transmission media; thus, machines connected to the same network segment will be able to see all traffic passing through that segment. Ethernet hardware is built to filter out traffic that either belongs to it or has a broadcast address and ignores all other traffic. This is done using the Media Access Control (MAC) address.

In earlier shared-media networks, monitoring was very simple. A host connected to the network segment could see all traffic passing through a network. Thus a network sniffer could be placed anywhere in the network and capture data.

In switched and routed networks; deployment of network sniffing tools requires careful planning. Multiple traffic paths exist and as a result, multiple network sniffers are needed unless there is a common point in the network through which all traffic is passed. Deploying a network sniffer on every point-to-point connection is too expensive and could severely affect network performance. In such an environment, a switch port analyzer can be used. The port configures the switch to copy data from a port or Virtual Private Network (VLAN) to another port called the span port. A network sniffer can be placed on the switch. Another option in capturing data in a switched network is called hub configuration. This is usually done between two switches, a switch and router or a switch and a server. The hub allows traffic to flow between the two devices and directs a copy to the network sniffer. The last option is a “tap”. A tap is hardwired into the device and allows a network sniffer to capture data.

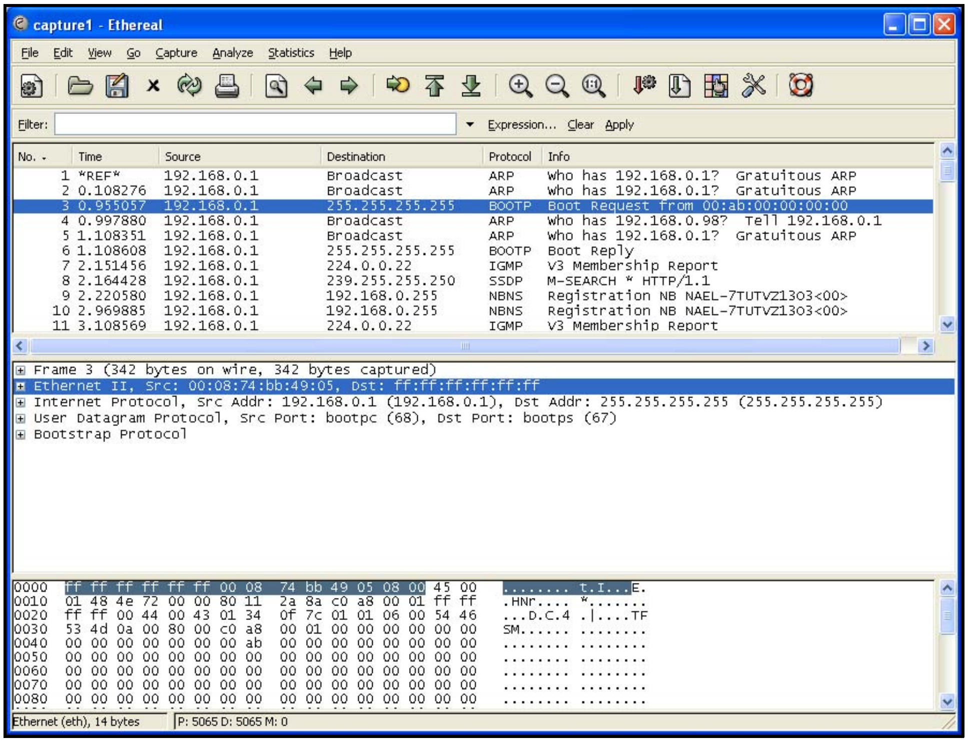
## OVERVIEW OF EACH FREE NETWORK SNIFFERS

### ETHEREAL

Ethereal is an open-source network sniffer originally created as a Unix/Linux application based on libPcap (an open-source network packet-capturing interface). Ethereal is also available for Windows. Ethereal comes with about 400-pages of documentation and has an easy to understand graphical user interface (GUI). Refer to Figure 1.

As figure 1 shows, the display window of Ethereal consists of three panes. The top pane displays a summary of captured packets; this pane usually displays the time the packet was captured, the source and destination IP addresses and protocol type and summary. The user can configure Ethereal to display address information such as IP and MAC addresses into their common names. The middle pane displays the details of a selected captured packet, and the third pane displays the data from the packet selected in the packet list pane, and highlights the field selected in the packet details pane. Data in this pane is diplayed in Hexadecimal and ASCII format.

To install Ethereal on a Windows system, WinPcap should be installed first. WinPcap is a free Windows version of libPcap that can be downloaded from http://winpcap.polito.it/install/Default.htm. Ethereal comes in a command line and GUI versions. The command line version is useful for scripting or activating Ethereal’s capturing feature based on the occurance of a certain event.


capture1 - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. -  Time      Source        Destination      Protocol Info
     1 *REF*     192.168.0.1   Broadcast        ARP      Who has 192.168.0.1?  Gratuitous ARP
     2 0.108276  192.168.0.1   Broadcast        ARP      Who has 192.168.0.1?  Gratuitous ARP
     3 0.955057  192.168.0.1   255.255.255.255  BOOTP    Boot Request from 00:ab:00:00:00:00
     4 0.997880  192.168.0.1   Broadcast        ARP      Who has 192.168.0.98?  Tell 192.168.0.1
     5 1.108351  192.168.0.1   Broadcast        ARP      Who has 192.168.0.1?  Gratuitous ARP
     6 1.108608  192.168.0.1   255.255.255.255  BOOTP    Boot Reply
     7 2.151456  192.168.0.1   224.0.0.22       IGMP     V3 Membership Report
     8 2.164428  192.168.0.1   239.255.255.250  SSDP     M-SEARCH * HTTP/1.1
     9 2.220580  192.168.0.1   192.168.0.255    NBNS     Registration NB NAEL-7TUTVZ1303<00>
    10 2.969885  192.168.0.1   192.168.0.255    NBNS     Registration NB NAEL-7TUTVZ1303<00>
    11 3.108569  192.168.0.1   224.0.0.22       IGMP     V3 Membership Report
```

```
⊞ Frame 3 (342 bytes on wire, 342 bytes captured)
⊞ Ethernet II, Src: 00:08:74:bb:49:05, Dst: ff:ff:ff:ff:ff:ff
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
⊞ Bootstrap Protocol
```

```
0000  ff ff ff ff ff ff 00 08  74 bb 49 05 08 00 45 00   ........ t.I...E.
0010  01 48 4e 72 00 00 80 11  2a 8a c0 a8 00 01 ff ff   .HNr.... *.......
0020  ff ff 00 44 00 43 01 34  0f 7c 01 01 06 00 54 46   ...D.C.4 .|....TF
0030  53 4d 0a 00 80 00 c0 a8  00 01 00 00 00 00 00 00   SM...... ........
0040  00 00 00 00 00 00 00 ab  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Ethernet (eth), 14 bytes    P: 5065 D: 5065 M: 0

Figure 1: Ethereal Network Sniffer Interface

Ethereal has all the features found in a typical nework sniffer. Ethereal can be set to capture all network traffic passing through a network segment or set to capture specific network packets that meet specific criteria set by capturing filters. With Ethereal, the netwrok interface card can be set to operate in the promiscuous or non-promiscuous modes. If the interface is not set to run in the promiscuous mode, only packets sent or received by the interface are captured. Ethereal supports capturing on only one interface, in cases where more than one network interface is found on the system, the user should specify which interface he or she wants to use for capturing packets. Ethereal does not support the Windows loopback interface.

Despite its name, Ethereal does support capturing traffic from media other than Ethernet. However, the kinds of media supported by Ethereal depends on the hardware installed, the operating system and the installed libpcap/WinPcap version. Table 1 obtained from http://www.ethereal.com/medi.html shows different supported media under different operating system. Please refer to the ethereal website for more details.

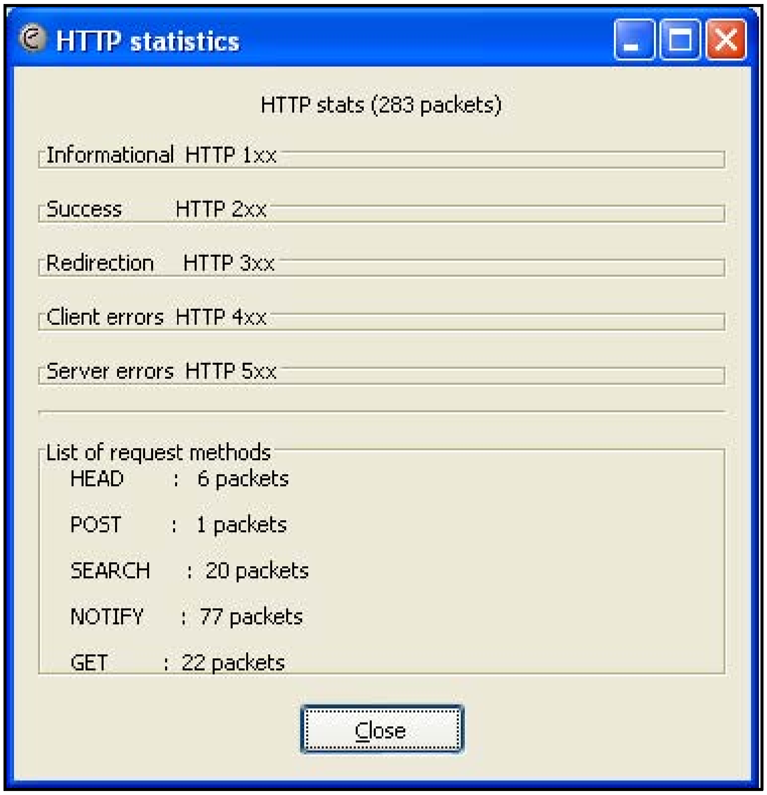
| | 802.11 | ATM | Ethernet | FDDI | Frame Relay | Loopback | Serial | Token Ring |
|---|---|---|---|---|---|---|---|---|
| AIX | Unknown | Unknown | Yes | Unknown | Unknown | Unknown | Unknown | Yes |
| FreeBSD | Yes1 | Unknown | Yes | Unknown | Unknown | Yes | Unknown | Yes |
| HP-UX | Unknown | Unknown | Yes | Unknown | No | No | Unknown | Unknown |
| Irix | Unknown | Unknown | Yes | Unknown | No | Unknown | Unknown | No |
| Linux | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Mac OS X | Yes | No | Yes | No | No | Yes | Yes | No |
| NetBSD | Yes | Unknown | Yes | Unknown | Unknown | Yes | Unknown | Yes |
| OpenBSD | Yes | Unknown | Yes | Unknown | Unknown | Yes | Unknown | Yes |
| Solaris | Unknown | Yes | Yes | Yes | No | No | No | Yes |
| Tru64 UNIX | Unknown | Unknown | Yes | Unknown | No | Yes | Unknown | Unknown |
| Windows | Yes | Unknown | Yes | Unknown | No | N/A | Yes | Yes |

Table 1: Media Supported by Ethereal

Ethereal supports three modes for saving captured packets: temporary file, single file, and continuous or ring multiple files. In continuous multiple files mode, data captured is saved in multiple files. A new file is created whenever a predefined switch condition is met. The ring multiple files mode is the same as the continuous multiple file mode. A new file will be created when one of the predefined switch criteria is met given that the specified maximum number of files to be used is not reached, otherwise the oldest capture file is replaced. Under Ethereal, the captured data packets can be saved in different file formats. They can also be exported to a plain text, postscript and/or XML file. Ethereal also supports saving part of the captured data packets that meet certain criteria, and has the ability to merge data from a captured file into a currently loaded file. In addition to what was mentioned, Ethereal also supports display filters. Ethereal enables a user to display packets that met specific criteria set by the display packet filters. It also allows the user to choose a different display color for certain packets that met specific criteria. These features make it easier to navigate through and analyze the captured data.

Ethereal can decode 603 protocols as mentioned on the Ethereal website. Information about standard Ethereal features and supported protocols can be found at http://www.ethereal.com/introduction.html#features . With Ethereal, the user can decode packets that are not recognized by Ethereal such as packets generated on non-default port.

Ethereal can display summary and statistical information about captured data. Although Ethereal does not provide graphical representation of statistical data, the statistical information displayed by it is very useful. Ethereal can display conversations that occur between two hosts, group packets that are related to the same TCP connection, and display protocol spectrum spread. Beside that, Ethereal can display conversation lists, endpoints lists, service response times and other protocol statistics. Figure 2 shows HTTP statistics.



```
HTTP stats (283 packets)

Informational  HTTP 1xx

Success        HTTP 2xx

Redirection    HTTP 3xx

Client errors  HTTP 4xx

Server errors  HTTP 5xx

List of request methods
   HEAD     :  6 packets
   POST     :  1 packets
   SEARCH   :  20 packets
   NOTIFY   :  77 packets
   GET      :  22 packets
```

Figure 2: HTTP Statistics

Finally, Ethereal is a good network snifter that can be used in small to medium size networks and for educational purposes. It can be used to capture and analyze traffic passing through a network segment for troubleshooting purposes or to study protocol architectures. It is an easy to use and easy to understand free network sniffer that has the typical standard features of a regular non-free network sniffer. It supports a variety of network protocols and provides useful statistical information. It also runs on the most popular platforms including Unix/Linux and windows. Ethereal cannot recommend network fixes and is not intended for use in intrusion detection systems. Being an open-source software, Ethereal lacks technical support.

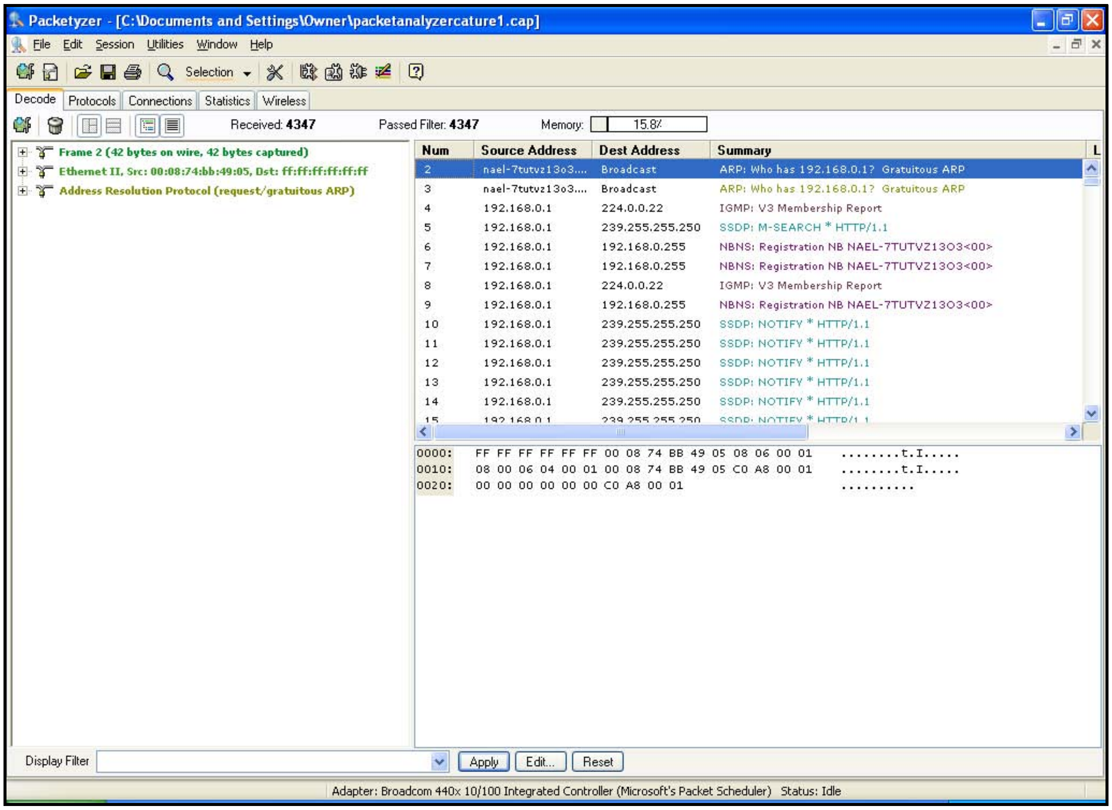
## NETWORK CHEMISTRY PACKETYZER

Packetyzer is a Windows-based free network sniffer built based around Ethereal's protocol capturing and dissection libraries. It can be installed on any Windows 32-bit platform including Windows 95, 98, ME, 2000 and XP. Basically, Packetyzer supports the same file formats and protocols supported by Ethereal. Currently, Packetyzer can decode 483 protocols including the most common ones. For a complete list of these protocols, refer to http://www.networkchemistry.com/products/packetyzer/protocols.php.

Installing Packetyzer is an easy task. The installation files can be downloaded from http://www.networkchemistry.com/products/packetyzer/#download. Packetyzer is distributed with WinPcap and Ethereal. As mentioned above, it runs only on Windows 32-bit platforms.

Packetyzer has a user-friendly graphical user interface that is easy to use and understand. The interface is well designed and clear. Packetyzer does not come with command line tool to start capturing packets. It comes with a user manual and useful help files. The main interface of Packetyzer is shown in Figure 3

As figure 3 shows, the display window of Packetyzer consists of three panes. The left pane displays a tree detailed view of a selected captured packet. The top right pane displays a summary of the captured packet. This includes the source and destination IP addresses and protocol summary. The right bottom pane displays the data from the packet selected in the packet list pane, and highlights the field selected in the packet details pane. Data in this pane is diplayed in Hexadecimal and ASCII format.



Received: **4347** Passed Filter: **4347** Memory: 15.8%

⊞ Frame 2 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:08:74:bb:49:05, Dst: ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request/gratuitous ARP)

| Num | Source Address | Dest Address | Summary | L |
|---|---|---|---|---|
| 2 | nael-7tutvz13o3.... | Broadcast | ARP: Who has 192.168.0.1? Gratuitous ARP | |
| 3 | nael-7tutvz13o3.... | Broadcast | ARP: Who has 192.168.0.1? Gratuitous ARP | |
| 4 | 192.168.0.1 | 224.0.0.22 | IGMP: V3 Membership Report | |
| 5 | 192.168.0.1 | 239.255.255.250 | SSDP: M-SEARCH * HTTP/1.1 | |
| 6 | 192.168.0.1 | 192.168.0.255 | NBNS: Registration NB NAEL-7TUTVZ13O3<00> | |
| 7 | 192.168.0.1 | 192.168.0.255 | NBNS: Registration NB NAEL-7TUTVZ13O3<00> | |
| 8 | 192.168.0.1 | 224.0.0.22 | IGMP: V3 Membership Report | |
| 9 | 192.168.0.1 | 192.168.0.255 | NBNS: Registration NB NAEL-7TUTVZ13O3<00> | |
| 10 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |
| 11 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |
| 12 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |
| 13 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |
| 14 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |
| 15 | 192.168.0.1 | 239.255.255.250 | SSDP: NOTIFY * HTTP/1.1 | |

```
0000:   FF FF FF FF FF FF 00 08 74 BB 49 05 08 06 00 01    ........t.I.....
0010:   08 00 06 04 00 01 00 08 74 BB 49 05 C0 A8 00 01    ........t.I.....
0020:   00 00 00 00 00 00 C0 A8 00 01                      ..........
```

Adapter: Broadcom 440x 10/100 Integrated Controller (Microsoft's Packet Scheduler) Status: Idle

Figure 3: Packetyzer Interface

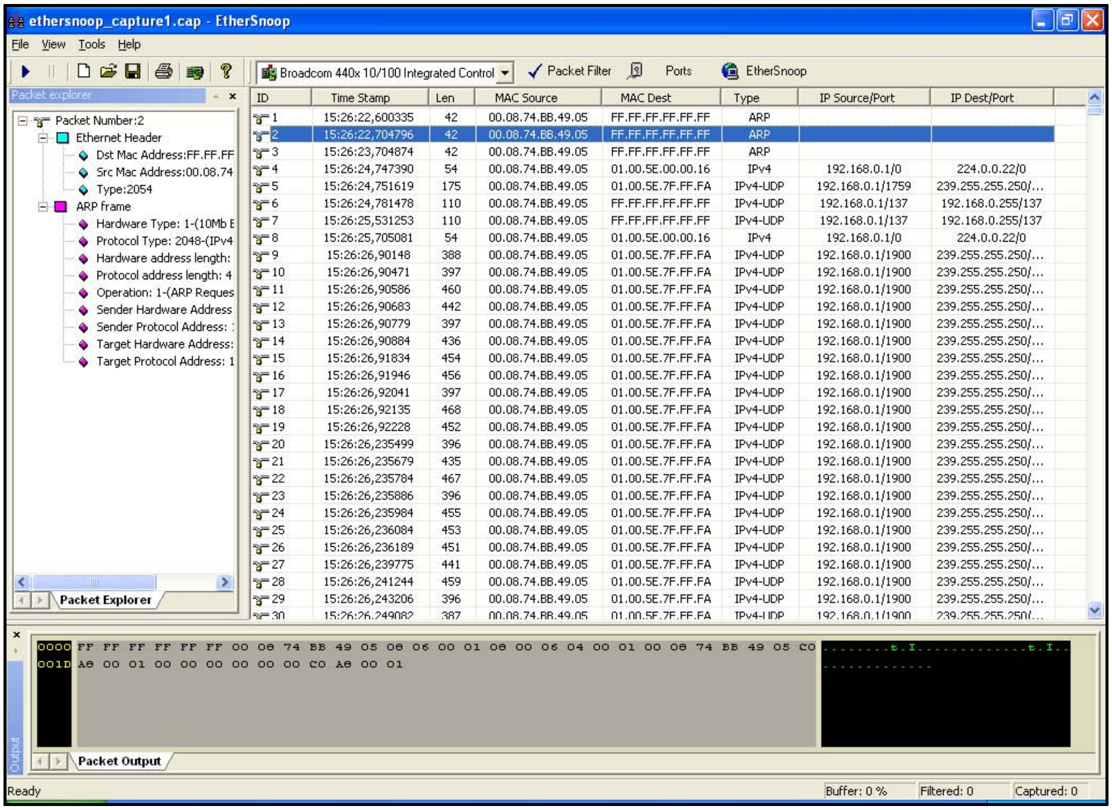
Packetyzer supports different kinds of media including Ethernet, FDDI, PPP, Token-Ring and Wireless LANs based on the 802..11. It supports the use of both capture filters and display filters to simplify and faciltate the process of displaying and analyzing captured data. Capture and display filters can be applied to the data easier than in Ethereal. As illustrated in Figure 3, data can be filtered depending on the network address or port number or any other criteria and can be dispalyed in different colors. Packetyzer enables the user to decode packets that are not recognized by its default implementation.It also provides the user with a packet editing tool. In using this tool, the user can edit, delete, duplicate and insert packets. Packetyzer also supports saving captured data in different file formats.

As in Ethereal, for Packetyzer the user can choose to set the Network Interface Card (NIC) in the promiscuous mode. Packetyzer provides useful information about protocols, connections and other statistical information such as the total number of packet and utilization.

Finally, Packetyzer is a good free network sniffer that has the typical standard features of a network sniffer. It is easy to use and has a user-friendly graphical interface. It provides capture and display filters that help the user capture and navigate captured network data packets, and display almost the same statistical information as Ethereal. Packetyzer runs only on windows 32-bit based platforms and supports a variety of network media and network protocols. It is useful for use for educational purposes and troubleshooting networks.

## ETHERSNOOP LIGHT

EtherSnoop Light is a very simple free network sniffer designed for capturing packets passing through a network segment. It analyzes data and displays it in a readable format. It runs only on a WIN32 environment and has convenient and easy to use graphical user interface. See Figure 4.



- Packet Number:2
  - Ethernet Header
    - Dst Mac Address:FF.FF.FF
    - Src Mac Address:00.08.74
    - Type:2054
  - ARP frame
    - Hardware Type: 1-(10Mb E
    - Protocol Type: 2048-(IPv4
    - Hardware address length:
    - Protocol address length: 4
    - Operation: 1-(ARP Reques
    - Sender Hardware Address
    - Sender Protocol Address: 1
    - Target Hardware Address:
    - Target Protocol Address: 1

| ID | Time Stamp | Len | MAC Source | MAC Dest | Type | IP Source/Port | IP Dest/Port |
|---|---|---|---|---|---|---|---|
| 1 | 15:26:22,600335 | 42 | 00.08.74.BB.49.05 | FF.FF.FF.FF.FF.FF | ARP | | |
| 2 | 15:26:22,704796 | 42 | 00.08.74.BB.49.05 | FF.FF.FF.FF.FF.FF | ARP | | |
| 3 | 15:26:23,704874 | 42 | 00.08.74.BB.49.05 | FF.FF.FF.FF.FF.FF | ARP | | |
| 4 | 15:26:24,747390 | 54 | 00.08.74.BB.49.05 | 01.00.5E.00.00.16 | IPv4 | 192.168.0.1/0 | 224.0.0.22/0 |
| 5 | 15:26:24,751619 | 175 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1759 | 239.255.255.250/... |
| 6 | 15:26:24,781478 | 110 | 00.08.74.BB.49.05 | FF.FF.FF.FF.FF.FF | IPv4-UDP | 192.168.0.1/137 | 192.168.0.255/137 |
| 7 | 15:26:25,531253 | 110 | 00.08.74.BB.49.05 | FF.FF.FF.FF.FF.FF | IPv4-UDP | 192.168.0.1/137 | 192.168.0.255/137 |
| 8 | 15:26:25,705081 | 54 | 00.08.74.BB.49.05 | 01.00.5E.00.00.16 | IPv4 | 192.168.0.1/0 | 224.0.0.22/0 |
| 9 | 15:26:26,90148 | 388 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 10 | 15:26:26,90471 | 397 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 11 | 15:26:26,90586 | 460 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 12 | 15:26:26,90683 | 442 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 13 | 15:26:26,90779 | 397 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 14 | 15:26:26,90884 | 436 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 15 | 15:26:26,91834 | 454 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 16 | 15:26:26,91946 | 456 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 17 | 15:26:26,92041 | 397 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 18 | 15:26:26,92135 | 468 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 19 | 15:26:26,92228 | 452 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 20 | 15:26:26,235499 | 396 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 21 | 15:26:26,235679 | 435 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 22 | 15:26:26,235784 | 467 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 23 | 15:26:26,235886 | 396 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 24 | 15:26:26,235984 | 455 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 25 | 15:26:26,236084 | 453 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 26 | 15:26:26,236189 | 451 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 27 | 15:26:26,239775 | 441 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 28 | 15:26:26,241244 | 459 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 29 | 15:26:26,243206 | 396 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |
| 30 | 15:26:26,249082 | 387 | 00.08.74.BB.49.05 | 01.00.5E.7F.FF.FA | IPv4-UDP | 192.168.0.1/1900 | 239.255.255.250/... |

```
0000 FF FF FF FF FF FF 00 08 74 BB 49 05 08 06 00 01 08 00 06 04 00 01 00 08 74 BB 49 05 C0   ........t.I.............t.I..
001D A8 00 01 00 00 00 00 00 00 C0 A8 00 01                                                     .............
```

Ready    Buffer: 0 %    Filtered: 0    Captured: 0

Figure 4: EtherSnoop Display Windows

As shown in Figure 4, EtherSnoop display window consists of three panes; the left pane displays the packet logical tree structure. The right pane displays the captured protocol summary including time stamp, packet length, source and destination MAC addresses, protocol type and source and destination IP addresses and port numbers. The bottom pane displays the packet details in hexadecimal and ASCII formats.

EtherSnoop supports a limited number of network protocols. Protocols supported are Ethernet, IPv4, ARP, ICMP, TCP and UDP. It supports saving captured data to a file. EtherSnoop also support a limited number of protocols and ports capture and display filters. The supported protocol filters are ARP, TCP, UDP, ICMP and others, selecting any of these filters will cause EtherSnoop to drop any packet that does not meet the conditions set by the filter. The supported port filters are HTTP, SMTP, POP3, FTP and Telnet. When any of these filters are checked, only packets that met the filter are captured.

EtherSnoop does not provide any statistical information about the captured data. Also, navigating through the captured data requires knowledge of the application TCP/UDP ports.

## COMPARISON CRITERIA

The following comparison criteria were used to compare the three free Network Sniffers:

1. Platforms/ Operating Systems supported.
2. User Interface.
3. Number of protocols that the network sniffer can decode.
4. Supported Interfaces and media.
5. Utilities available to enable the user to customize capturing and displaying network packets.
6. Support for Customized Protocol Decodes.
7. Readability of captured data
8. Provided Statistical information.
9. Decoding data captured on non-default port.

## COMPARISON AND CONCLUSION

Before deploying any network sniffer it is important to identify under which platforms it runs. While both EtherSnoop and Packetyzer can run only on Windows 32bit platforms, Ethereal can run on Windows and Linux/ UNIX platforms. The three network sniffers have user-friendly interfaces. However among the three network sniffers, Packetyzer has the most user-friendliness interface and is the easiest to use. In fact, Packetyzer iss a windows user interface of Ethereal. EtherSnoop comes in at second and Ethereal comes in third based on this feature.

An important feature of a network sniffer is the number of protocols it can decode. Ethereal can decode 601 protocols based on the information obtained form the Ethereal website. Packetyzer is second; it can decode 483 protocols. EtherSnoop is in third place; it can only decode a limited number of protocols. Both Ethereal and Packetyzer can be customized to decode new protocols and applications that run on non-default protocols that cannot be decoded under default settings. They also can be configured to decode protocols running on non-default ports. On the other hand, EtherSnoop does not support these features.

Both Ethereal and Packetyzer support a variety of media. Since Packetyzer is a Windows' interface of Ethereal, Packetyzer supports the same set of media that Ethereal supports in a Windows environment. Since Ethereal can be run on other platforms, the media supported by Ethereal depends on the hardware used, the operating system and the installed version of libPcap/WinPcap. On the other hand, EtherSnoop seems to support Ethernet only.

All three network sniffers include capture and display filters that enable the user to capture or display specific network packets that meet specific preset criteria. Both Ethereal and Packetyzer have large set of filters that can be used to filter packets based on source and/or destination ports, and source and/or destination addresses. On the other hand, EtherSnoop includes a limited number of filters.

Readability of collected data is an important feature of network sniffers. As mentioned previously, all three sniffers have tools to filter data packets. Applying display and/or capture filters in all three sniffers is easy. However, it is easier to apply these filters in both Packetyzer and EtherSnoop than in Ethereal. While both Ethereal and Packetyzer support colorization to display captured data packets, EtherSnoop does not include this feature. In Ethereal the captured data packets can be displayed with different colors depending on user configured colorization criteria. This enhances the readability of the captured data. Although in Packetyzer, a user can change the display color of a selected packet or group of packets, it is performed on a per packet basis and cannot be performed based on user configured criteria, a feature that can be useful for referencing purposes only. Another factor that is related to readability is the ability to decode captured data packets. Ethereal and Packetyzer can decode a larger set of data packets than EtherSnoop. This affects the readability of captured data. In EtherSnoop, it is necessary for the user to know the default assigned port numbers in order to be able to identify applications.

An important feature of a network sniffer is to include data analysis tools. Both Ethereal and Packetyzer provide the user with analysis tools and statistical reports, while EtherSnoop does not. Ethereal provides several useful statistical reports including end-to-end conversation lists based on media access control protocols Network layer protocols and transport protocols. It can also report service response time for a specific set of services. Packetyzer on the other hand provides statistical information about connections, protocols and network utilization. It also enables the user to follow a TCP flow. Packetyzer supports a feature that is not supported by either EtherSnoop or Ethereal. Packetyzer enables the user to edit, duplicate, delete, insert and send a packet.

All three sniffers support saving the captured data to a file. While in Ethereal and Packetyzer the captured data can be saved in a set of different file formats, EtherSnoop supports only one file format to save captured data. Ethereal support an extra feature compared to the other two sniffers. It supports exporting data to a text file, a postscript file and a XML file. This feature enables the user to import captured data to other applications for analysis purposes.

## REFERENCES

[1] Daniel Magers, (May 9, 2002), “Packet Sniffing: An Integral Part of Network Defense ”, Retrieved on September 10, 2004 from http://www.giac.org/practical/Daniel_Magers_GSEC.doc

[2] Roger E. Grimes, (July, 2004), “6 Network Protocol Analyzers ”, Retrieved on September 10, 2004 from http://www.winntmag.com/Windows/Article/ArticleID/42922/42922.html

[3] Alan Joch, (July 23, 2001), “Network Sniffers”, Retrieved on September 10, 2004 from http://www.computerworld.com/networkingtopics/networking/lanwan/story/0,10801,62390,00.html

[4] Ethereal Website, http://www.ethereal.com

[5] EtherSnoop Website, http://www.arechisoft.com/

[6] Packetyzer Website, http://www.packetyzer.com