PACKET ANALYZER ON TCP/IP PROTOCOL

By

SUNEET BABORIA-051025
KARAN GUPTA-051026
MANDAKINI GUPTA-051075

MAY-2009

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY-
WAKNAGHAT

CERTIFICATE

This is to certify that the work entitled, "Packet Analyzer On Tcp/Ip Protocol" submitted by Suneet Baboria, Karan Gupta and Mandakini Gupta in partial fulfillment for the award of degree of Bachelors of Technology in Electronics and Communication Engineering of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Prof. S.V. Bhooshan
H.O.D. ECE Deptt.

Dr. Vinay Kumar
Project Supervisor

ACKNOWLEDGMENT

I would most of all like to thank my supervisors Dr. S.V.Bhooshan and Dr. Vinay Kumar for the invaluable advice and positive encouragement they have provided me throughout the course of this project. Special thanks to Dr. Vinay Kumar for his help at all stages during the project. Thanks also go to all my fellow students for making the four years of B.Tech the most memorable ones. Finally, I would like to express my gratitude to all my friends and family for their continuous love and support, and for always being there for me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS USED

ADSL	Asymmetric Digital Subscriber Line
ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency Network
ATM	Asynchronous Transfer Mode
DHCP	Dynamic Host Configuration Protocol
DPC	Deep Packet Capture
DPI	Deep Packet Inspection
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute Of Electrical And Electronics Engineers
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IONL	Internal Organization of the Network Layer
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
LAN	Local Area Network
MAC	Medium Access Control
MIME	Multipurpose Internet Mail Extensions
NAC	Net Access Corporation
OSI	Open system interconnection
OSIRMMF	OSI Reference Model Management Framework
PING	Partimage Is Not Ghost
POP3	Post Office Protocol 3
PPP	Point-to-Point Protocol
RFC	Request for Comment
RPC	Remote Procedure Call
SMB	Server Message Blocks
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSH	Secure Shell Remote Protocol
TCP	Transmission Control Protocol
Telnet	Telnet Remote Protocol
TTL	Time to Live
UDP	User Datagram Protocol

Abstract

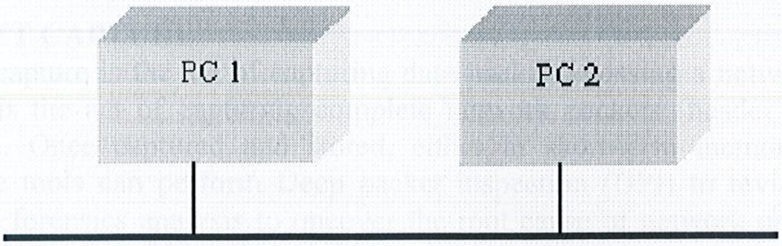
Packet is a unit of data that is routed between an origin and destination on any packet - switched network. When a file is sent over the network the tcp layer of tcp/ip protocol divides it into chunks for efficient routing. A packet analyzer is a program that monitors the network traffic by grabbing information traveling over a network, thereby saving the transmitted files. The application captures every packet on the wire to display important information such as a list of packets and network connections and other vital statistics. Packet capturing is required when we need to see what client and server are actually saying to each other or when we need to analyze the type of traffic on network. It is also required for the understanding of network protocols to be used effectively

CHAPTER1- INTRODUCTION

PACKET ANALYZER

A packet analyzer is computer software or computer hardware that can intercept and log traffic passing over a part of a network. As data streams flow across the network, the analyzer captures each packet and eventually decodes and analyzes its content according to the appropriate RFC or other specifications.

Design Modules



- Packet Capture Module
- Packet Parser Module
- User Interface Module

Packet Capture Module:

- This module will integrate with library winpcap and provide a method to investigate the packet.
- Winpcap is a tool that allows applications to capture and transmit network packets.
- Winpcap is compatible wid libpcap which means we can use it to port our existing Unix or Linux tools to Windows, making it portable to Unix

Packet Parser Module:

- This module will parse the packet header and identify the details.

User Interface Module:

- This module will have the user interface and method to trigger actions based on user request. It will use the other two modules to accomplish the triggered action.
- This Module will be implemented in java using swing and awt components

Capabilities

On wired broadcast LANs, we can capture traffic on all parts of the network from a single machine within the network; however, there are some methods to avoid traffic narrowing by switches to gain access to traffic from other systems on the network (e.g. ARP spoofing). For the purpose of network monitoring all data packets in a LAN can be monitored by using a network switch with a monitoring port, whose purpose is to mirror all packets passing through all the ports of the switch. However if the computer is connected to a switch port the analyzer will be unable to read the data owing to the intrinsic nature of switched networks. In this case a shadow port is created so that the analyzer can capture data. In order to capture traffic on wired broadcasts ,other than unicast traffic sent to the machine running the analyzer, multicast traffic sent to a multicast group to which that machine is listening, and broadcast traffic.

Notable packet analyzers

Carnivore
dSniff
Microsoft Network Monitor
NetScout Sniffer
Network Instruments Observer
tcpdump
Wireshark (formerly known as Ethereal)

PACKET CAPTURE

Packet capture is the act of capturing data packets crossing a network. Deep packet capture (DPC) is the act of capturing complete network packets (header and payload) crossing a network. Once captured and stored, either in short-term memory or long-term storage, software tools can perform Deep packet inspection (DPI) to review network packet data, perform forensics analysis to uncover the root cause of network problems, identify security threats, and ensure data communications and network usage complies with outlined policy. Some DPCs can be coupled with DPI and can as a result manage, inspect, and analyze all network traffic in real-time at wire speeds while keeping a historical archive of all network traffic for further analysis. Partial packet capture can record headers without recording the total content of datagrams. This can reduce storage requirements, and avoid legal problems, but yet have enough data to reveal the essential information required for problem diagnosis.

Filtering

Packet capture can either capture the entire data stream or capture a filtered portion.

- ***Complete capture:***

  Packet capture has the ability to capture packet data from the data link layer on up (layers 2-7) of the OSI model. This includes headers and payload. Headers include information about what is contained in the packet and could be synonymous to an address or other printed information on the outside of an envelope. The payload includes the actual content of the packet and therefore synonymous to the contents of the envelope. Complete capture encompasses every packet that crosses a network segment, regardless of source, protocol or other distinguishing bits of data in the packet. Complete capture is the unrestricted, unfiltered, raw capture of all network packets.

- ***Filtered capture:***

  DPC devices may have the ability to limit capture of packets by protocol, IP address, MAC address, etc. With the application of filters, only complete packets that meet the criteria of the filter (header and payload) are captured, diverted, or stored.

Historical capture and analysis

Once data is captured, it can be analyzed right away or stored and analyzed later.

Many deep packet inspection tools rely on real-time inspection of data as it crosses the network, using known criteria for analysis. DPI tools make real-time decisions on what to do with packet data, perform designated analysis and act on the results. If packets are not stored after capture, they may be flushed away and actual packet contents are no longer available. Short-term capture and analysis tools can typically detect threats only when the triggers are known in advance but can act in real-time.

Historical capture and analysis stores all captured packets for further analysis, after the data has already crossed the network. As DPI and analysis tools deliver alerts, the historical record can be analyzed to apply context to the alert, answering the question "what happened leading up to, and after, the alert?"

Identifying security breaches

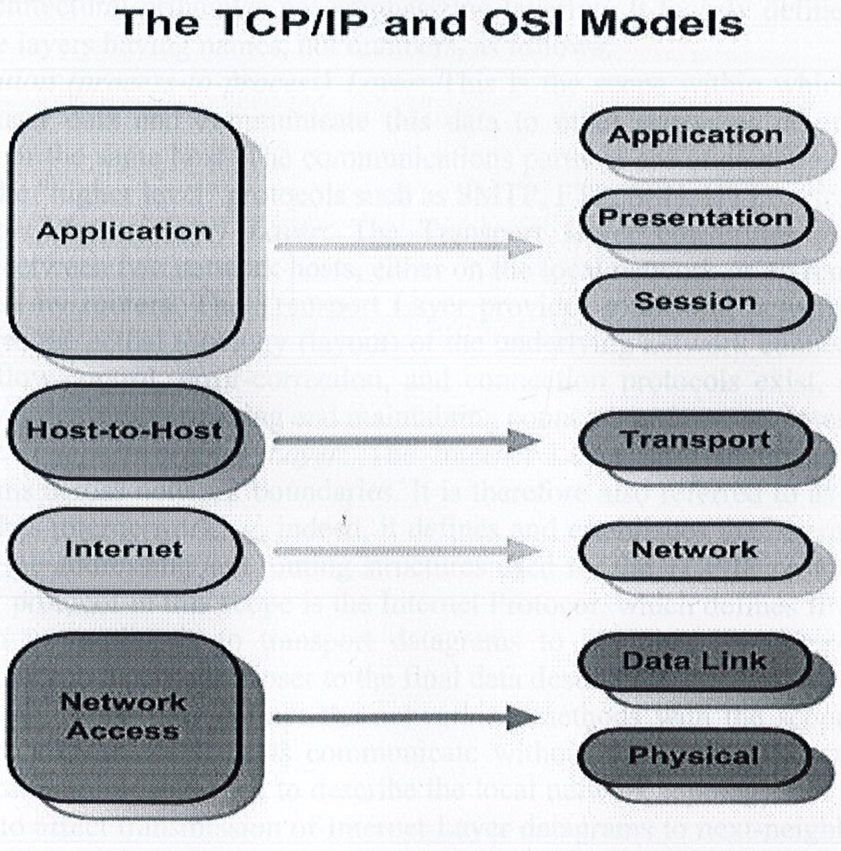
Analysis of historical data captured with DPC assists in pinpointing the source of the intrusion. DPC can capture network traffic accessing certain servers and other systems to verify that the traffic flows belong to authorized employees. However this technique cannot function as an intrusion prevention system.

Network Troubleshooting

If an adverse event is detected on a network, its cause or source can be more reliably determined if the administrator has access to complete historical data. DPC can capture all packets on important network links continuously. When an event happens, a network administrator can then assess the exact circumstances surrounding a performance event, take corrective action, and ensure that the problem will not reoccur. This helps reduce the Mean Time To Repair.

CHAPTER2- TCPIP

The TCP/IP model is a description framework for computer network protocols created in the 1970s by DARPA, an agency of the United States Department of Defense. It evolved from ARPANET, which was the world's first wide area network and a predecessor of the Internet. The TCP/IP Model is sometimes called the Internet Reference Model or the DoD Model.The TCP/IP model, or Internet Protocol Suite, describes a set of general design guidelines and implementations of specific networking protocols to enable computers to communicate over a network. TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. Protocols exist for a variety of different types of communication services between computers.TCP/IP is generally described as having four abstraction layers (RFC 1122). This layer architecture is often compared with the seven-layer OSI Reference Model formalized after the TCP/IP specifications.The TCP/IP model and related protocols are maintained by the IETF


The TCP/IP and OSI Models

Key architectural principles

An early architectural document, RFC 1122, emphasizes architectural principles over layering.

- End-to-End Principle: This principle has evolved over time. Its original expression put the maintenance of state and overall intelligence at the edges, and assumed the Internet that connected the edges retained no state and concentrated on speed and simplicity. Real-world needs for firewalls, network address translators, web content caches and the like have forced changes in this principle.
- Robustness Principle: "In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g.,

not object to technical errors where the meaning is still clear) RFC 791." "The second part of the principle is almost as important: software on other hosts may contain deficiencies that make it unwise to exploit legal but obscure protocol features. RFC 1122"

As with all other communications protocol, TCP/IP is composed of layers:

- IP - is responsible for moving packet of data from node to node. IP forwards each packet based on a four byte destination address (the IP number). The Internet authorities assign ranges of numbers to different organizations. The organizations assign groups of their numbers to departments. IP operates on gateway machines that move data from department to organization to region and then around the world.
- TCP - is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
- Sockets - is a name given to the package of subroutines that provide access to TCP/IP on most systems.

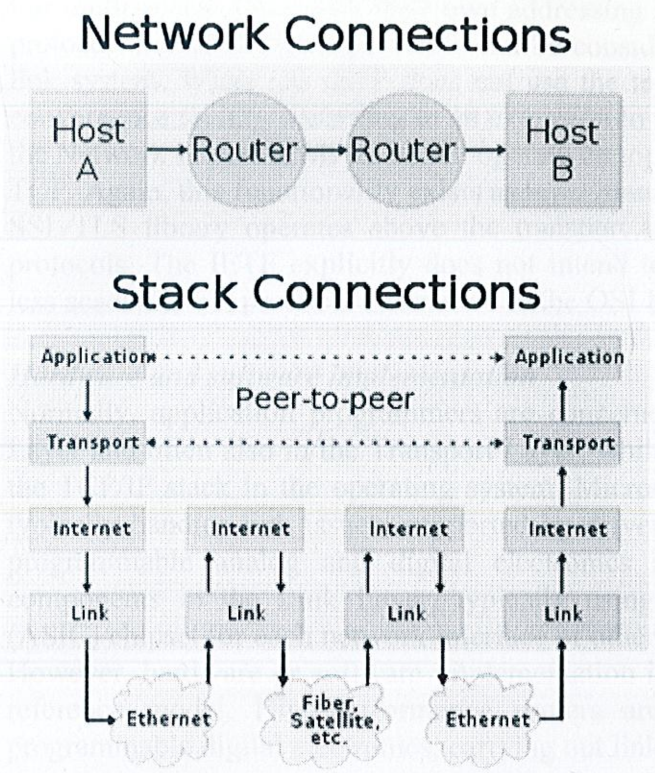
RFC 1122 on Host Requirements is structured in paragraphs referring to layers, but refers to many other architectural principles not emphasizing layering. It loosely defines a four-layer model, with the layers having names, not numbers, as follows:

- *Application (process-to-process) Layer*: This is the scope within which applications create user data and communicate this data to other processes or applications on another or the same host. The communications partners are often called peers. This is where the "higher level" protocols such as SMTP, FTP, SSH, HTTP, etc. operate.
- *Transport (host-to-host) Layer*: The Transport Layer constitutes the networking regime between two network hosts, either on the local network or on remote networks separated by routers. The Transport Layer provides a uniform networking interface that hides the actual topology (layout) of the underlying network connections. This is where flow-control, error-correction, and connection protocols exist, such as TCP. This layer deals with opening and maintaining connections between Internet hosts.
- *Internet (internetworking) Layer*: The Internet Layer has the task of exchanging datagrams across network boundaries. It is therefore also referred to as the layer that establishes internetworking, indeed, it defines and establishes the Internet. This layer defines the addressing and routing structures used for the TCP/IP protocol suite. The primary protocol in this scope is the Internet Protocol, which defines IP addresses. Its function in routing is to transport datagrams to the next IP router that has the connectivity to a network closer to the final data destination.
- *Link Layer:* This layer defines the networking methods with the scope of the local network link on which hosts communicate without intervening routers. This layer describes the protocols used to describe the local network topology and the interfaces needed to affect transmission of Internet Layer datagrams to next-neighbor hosts. (cf. the OSI Data Link Layer).

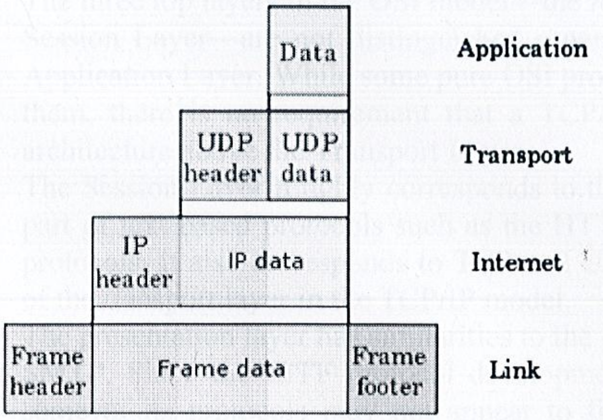
The Internet Protocol Suite and the layered protocol stack design were in use before the OSI model was established. Since then, the TCP/IP model has been compared with the OSI model in books and classrooms, which often results in confusion because the two models use different assumptions, including about the relative importance of strict layering.

Layers in the TCP/IP model

The layers near the top are logically closer to the user application, while those near the bottom are logically closer to the physical transmission of the data. Viewing layers as providing or consuming a service is a method of abstraction to isolate upper layer protocols from the nitty-gritty detail of transmitting bits over, for example, Ethernet and collision



Two Internet hosts connected via two routers and the corresponding layers used at each hop.



Encapsulation of application data descending through the TCP/IP layers

detection, while the lower layers avoid having to know the details of each and every application and its protocol.

This abstraction also allows upper layers to provide services that the lower layers cannot, or choose not to, provide. Again, the original OSI Reference Model was extended to include connectionless services. For example, IP is not designed to be reliable and is a best effort delivery protocol. This means that all transport layer implementations must choose whether or not to provide reliability and to what degree. UDP provides data integrity (via a checksum) but does not guarantee delivery; TCP provides both data integrity and delivery guarantee (by retransmitting until the receiver acknowledges the reception of the packet). This model lacks the formalism of the OSI reference model and associated documents, but the IETF does not use a formal model and does not consider this a limitation, as in the comment by David D. Clark, "We reject: kings, presidents and voting. We believe in: rough consensus and running code." Criticisms of this model, which have been made with respect to the OSI Reference Model, often do not consider ISO's later extensions to that model.

For multi-access links with their own addressing systems (e.g. Ethernet) an address mapping protocol is needed. Such protocols can be considered to be below IP but above the existing link system. While the IETF does not use the terminology, this is a subnetwork dependent convergence facility according to an extension to the OSI model, the Internal Organization of the Network Layer. ICMP & IGMP operate on top of IP but do not transport data like UDP or TCP. Again, this functionality exists as layer management extensions to the OSIRM MF. The SSL/TLS library operates above the transport layer (utilizes TCP) but below application protocols. The IETF explicitly does not intend to discuss transmission systems, which is a less academic but practical alternative to the OSI Reference Model.

Hardware and software implementation

Normally, application programmers are concerned only with interfaces in the Application Layer and often also in the Transport Layer, while the layers below are services provided by the TCP/IP stack in the operating system. Microcontroller firmware in the network adapter typically handles link issues, supported by driver software in the operational system. Non-programmable analog and digital electronics are normally in charge of the physical components in the Link Layer, typically using an application-specific integrated circuit (ASIC) chipset for each network interface or other physical standard.

However, hardware or software implementation is not stated in the protocols or the layered reference model. High-performance routers are to a large extent based on fast non-programmable digital electronics, carrying out link level switching.

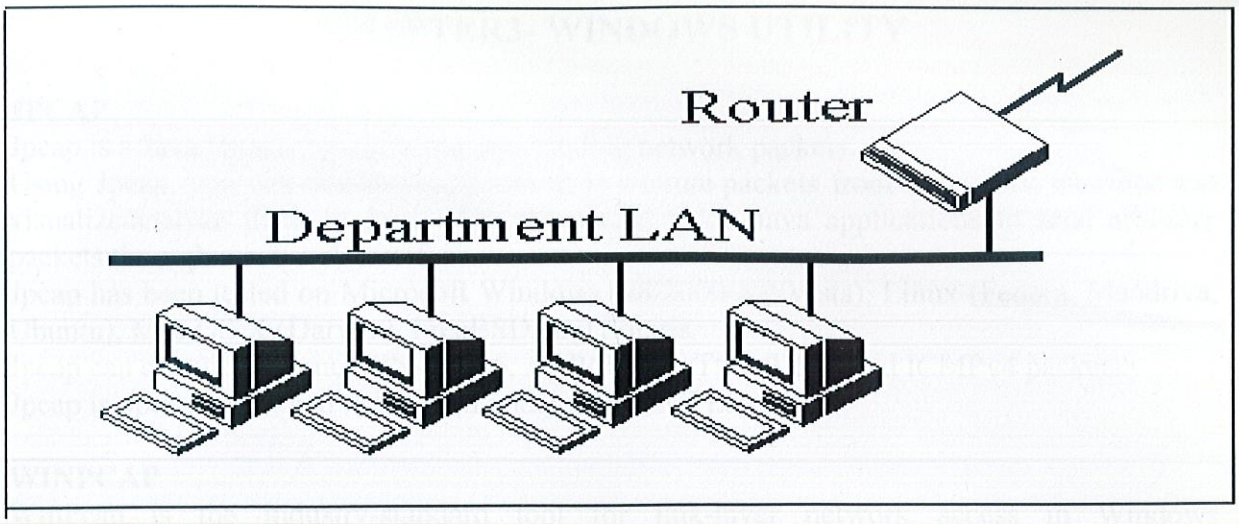
OSI and TCP/IP layering differences

The three top layers in the OSI model—the Application Layer, the Presentation Layer and the Session Layer—are not distinguished separately in the TCP/IP model where it is just the Application Layer. While some pure OSI protocol applications, such as X.400, also combined them, there is no requirement that a TCP/IP protocol stack needs to impose monolithic architecture above the Transport Layer.

The Session Layer roughly corresponds to the Telnet virtual terminal functionality, which is part of text based protocols such as the HTTP and SMTP TCP/IP model Application Layer protocols. It also corresponds to TCP and UDP port numbering, which is considered as part of the transport layer in the TCP/IP model.

The presentation layer has similarities to the MIME standard, which also is used in HTTP and SMTP. Since the IETF protocol development effort is not concerned with strict layering, some of its protocols may not appear to fit cleanly into the OSI model. These conflicts, however, are more frequent when one only looks at the original OSI model, ISO 7498, without looking at the annexes to this model (e.g., ISO 7498/4 Management Framework), or the ISO IONL. When the IONL and Management Framework documents are considered, the ICMP and IGMP are neatly defined as layer management protocols for the network layer. In like manner, the IONL provides a structure for "subnetwork dependent convergence facilities" such as ARP and RARP.

IETF protocols can be encapsulated recursively, as demonstrated by tunneling protocols such as Generic Routing Encapsulation (GRE). While basic OSI documents do not consider tunneling, there is some concept of tunneling in yet another extension to the OSI architecture, specifically the transport layer gateways within the International Standardized Profile framework [11]. The associated OSI development effort, however, has been abandoned given the overwhelming adoption of TCP/IP protocols.

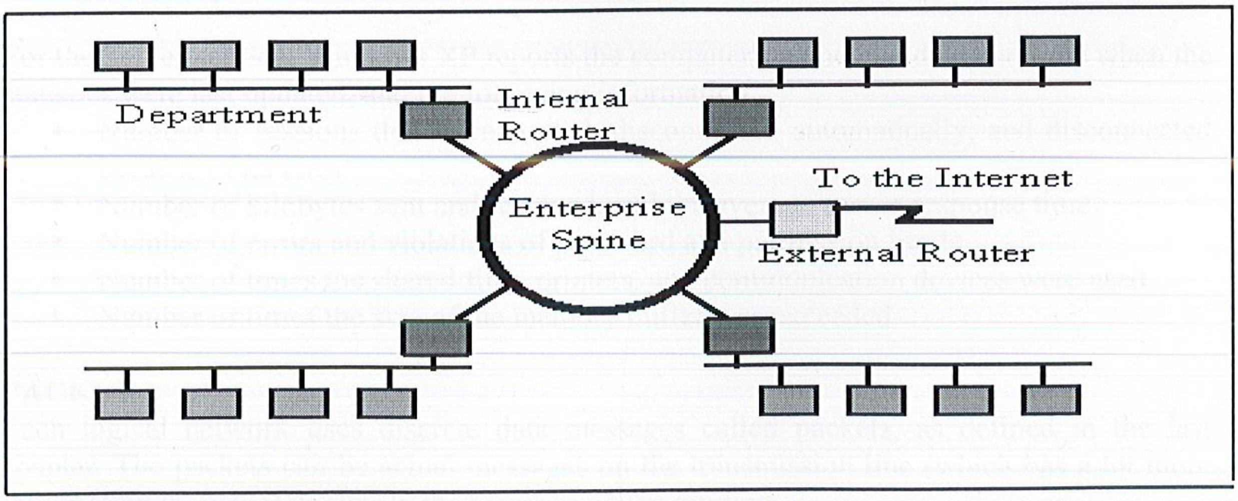


Addresses

Each technology has its own convention for transmitting messages between two machines within the same network. On a LAN, messages are sent between machines by supplying the six byte unique identifier (the "MAC" address). In an SNA network, every machine has Logical Units with their own network address. DECNET, Appletalk, and Novell IPX all have a scheme for assigning numbers to each local network and to each workstation attached to the network. On top of these local or vendor specific network addresses, TCP/IP assigns a unique number to every workstation in the world. This "IP number" is a four byte value that, by convention, is expressed by converting each byte into a decimal number (0 to 255) and separating the bytes with a period. For example, the PC Lube and Tune server is 130.132.59.234.

Subnets

Although the individual subscribers do not need to tabulate network numbers or provide explicit routing, it is convenient for most Class B networks to be internally managed as a much smaller and simpler version of the larger network organizations. It is common to subdivide the two bytes available for internal assignment into a one byte department number and a one byte workstation ID. The enterprise network is built using commercially available TCP/IP router boxes. Each router has small tables with 255 entries to translate the one byte department number into selection of a destination Ethernet connected to one of the routers. Mssages to the PC Lube and Tune server (130.132.59.234) are sent through the national and New England regional networks based on the 130.132 part of the number.



CHAPTER3- WINDOWS UTILITY

JPCAP

Jpcap is a Java library for capturing and sending network packets.

Using Jpcap, you can develop applications to capture packets from a network interface and visualize/analyze them in Java. You can also develop Java applications to send arbitrary packets through a network interface.

Jpcap has been tested on Microsoft Windows (98/2000/XP/Vista), Linux (Fedora, Mandriva, Ubuntu), Mac OS X (Darwin), FreeBSD, and Solaris.

Jpcap can capture Ethernet, IPv4, IPv6, ARP/RARP, TCP, UDP, and ICMPv4 packets.

Jpcap is open source, and is licensed under GNU LGPL.

WINPCAP

WinPcap is the industry-standard tool for link-layer network access in Windows environments. It allows applications to capture and transmit network packets by passing the protocol stack, and has additional useful features, including kernel-level packet filtering, a network statistics engine and support for remote packet capture.

WinPcap consists of a driver, that extends the operating system to provide low-level network access, and a library that is used to easily access the low-level network layers. This library also contains the Windows version of the well known libpcap Unix API.

Thanks to its set of features, WinPcap is the packet capture and filtering engine of many open source and commercial network tools, including protocol analyzers, network monitors, network intrusion detection systems, sniffers, traffic generators and network testers. Some of these tools, like Wireshark, Nmap, Snort, ntop are known and used throughout the networking community.

Winpcap.org is also the home of WinDump, the Windows version of the popular tcpdump tool. WinDump can be used to watch, diagnose and save to disk network traffic according to various complex rules.

NETSTATS

For the Workstation service, Windows XP reports the computer's name, the date and time when the statistics were last updated, and the following information:

- Number of bytes and server message blocks (SMB) received and transmitted
- Number of read and write operations that succeeded or failed
- Number of network errors
- Number of sessions that failed, disconnected, or were reconnected
- Number of connections to shared resources that succeeded or failed

For the Server service, Windows XP reports the computer's name, the date and time when the statistics were last updated, and the following information:

- Number of sessions that were started, disconnected automatically, and disconnected because of an error
- Number of kilobytes sent and received, and the average server-response time
- Number of errors and violations of password and permission limits
- Number of times the shared files, printers, and communication devices were used
- Number of times the size of the memory buffer was exceeded

PACKET

Each logical network uses discrete data messages called packets, as defined in the last chapter. The packets can be actual messages on the transmission line (which has a lot more information included) or simply the message you're sending.

The logical network packet at the generic level consists of information about the source, destination, and data payload. Each logical network offers varying degrees of features and interfaces (protocols). All packet types and protocols are available with network programming. Each type has significant strengths and weaknesses. Like shopping for tools, your choice of packet type depends on how you use it.

You can choose from four basic Internet packet protocols: raw IP, ICMP, UDP (unreliable messaging), and TCP (streaming) all layered on top of the physical network

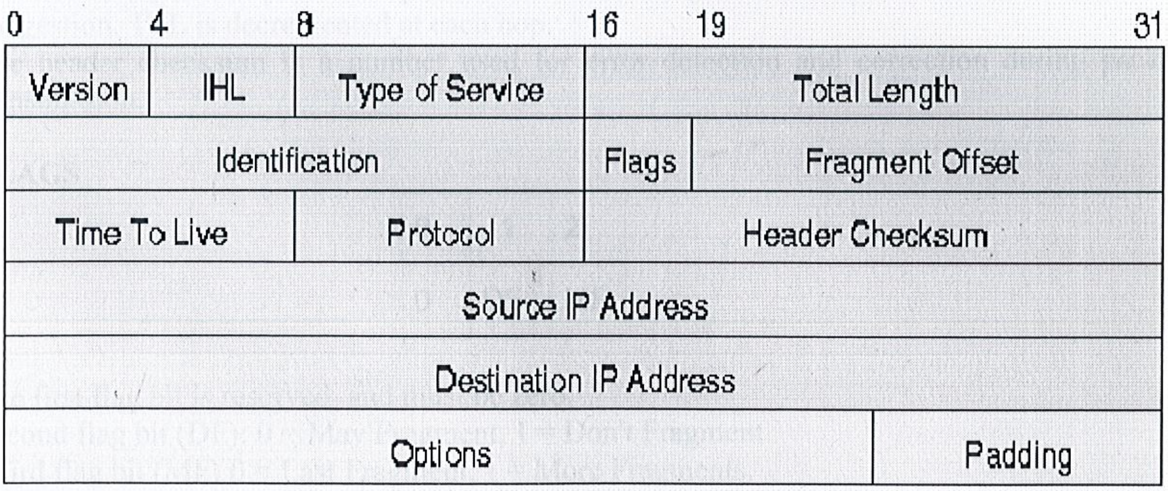
Each protocol is very different, but they all share one common necessary feature: They all carry the program's message. Some protocols include a source address, while some require a destination. You may think that not requiring a destination is a little unusual, but some protocols (like UUCP) use the connection as the address to the destination.

The Internet Protocol (IP) [RFC791] requires a packet to have three basic elements: source, destination, and data. (The data payload includes its length.) These elements provide the packet a level of autonomy. No matter where a packet is, you can identify where it came from, where it's going, and how big it is.

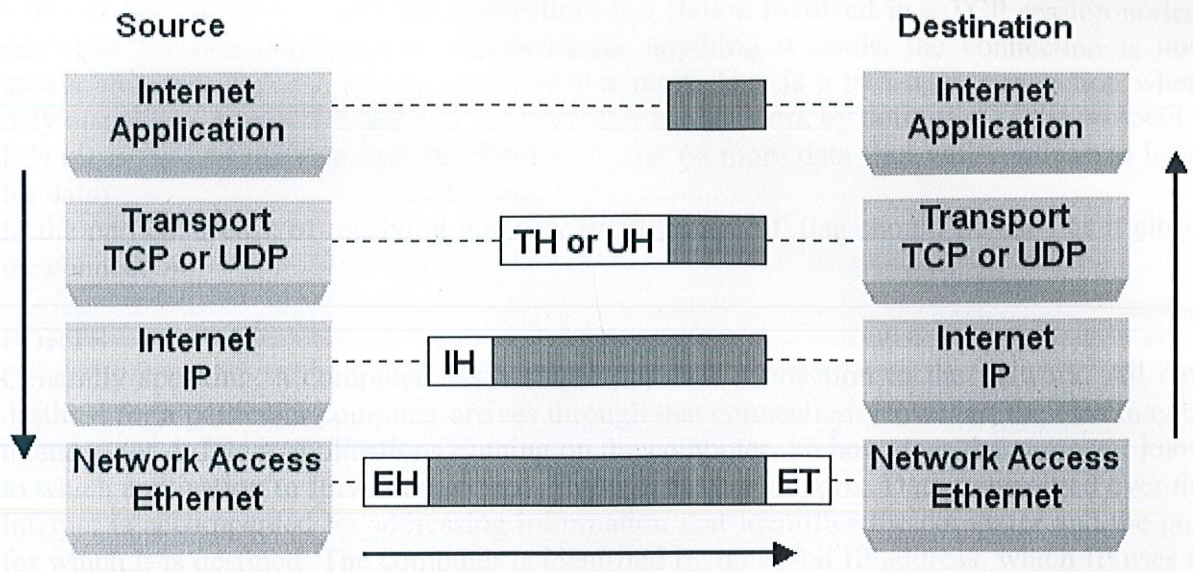
The packet's autonomy is an important feature of the Internet. As long as the packet is *alive* (the data is timely and relevant), routers can move data to its destination when the packet is launched onto the network.

IP PACKET STRUCTURE

All IP packets are structured the same way - an IP header followed by a variable-length data field. A summary of the contents of the internet header follows:



Each IP (Internet Protocol) packet consists of a header followed by a data field. The header length can vary between 20 and 60 bytes, and the total size of the packet can be up to 65535 bytes. However, many systems cannot handle packets as large as the protocol allows and a working maximum size is 576 bytes. The header must have 5 words (of 32 bits each) of defined contents, and may have up to 10 more words of optional information. Most network data transmission technologies use packets to transmit data from a source device to destination. The IP protocol is not exception. IP packets are the most important and fundamental components of the protocol. The two main functions of the IP protocol are routing and addressing. To route packets to and from machines on a network, IP uses IP addresses which are carried along in the packets. A lot of other information is carried along as well, in the packet header. The structure of an IP packet is shown in the picture here.



The identification tag is used to help reassemble the packet from several eventual fragments.
The flag states whether the packet can be fragmented or not.
The fragment offset is a field to identify which fragment this packet is attached to.
Time to Live (TTL) is a number that indicates how many hops the packet can make before it dies. This is done to prevent a packet from remaining forever on a network, thus causing congestion. TTL is decremented at each hop.
The header checksum is a number used for error detection and correction during packet transmission.

FLAGS

0	1	2
0	DF	MF

The first flag bit is reserved, and must be zero.
Second flag bit (DF): 0 = May Fragment, 1 = Don't Fragment..
Third flag bit (MF) 0 = Last Fragment, 1 = More Fragments.

There are six ‘control bits’ defined in TCP, one or more of which is defined in each packet. The control bits are ‘SYN’, ‘ACK’, ‘PSH’, ‘URG’, ‘RST’, and ‘FIN’. TCP uses these bits to define the purpose and contents of a packet.
SYN bit is used in establishing a TCP connection to synchronize the sequence numbers between both endpoints.
ACK bit is used to acknowledge the remote host’s sequence numbers, declaring that the information in the acknowledgment field is valid.
PSH flag is set on the sending side, and tells the TCP stack to flush all buffers and send any outstanding data up to and including the data that had the PSH flag set. When the receiving TCP sees the PSH flag, it too must flush its buffers and pass the information up to the application.
URG bit indicates that the urgent pointer field has a valid pointer to data that should be treated urgently and be transmitted before non- urgent data.

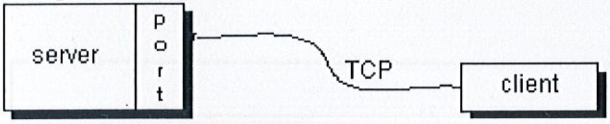
Reset or RST is used to reset the connection. If a station involved in a TCP session notices that it is not receiving acknowledgements for anything it sends, the connection is now unsynchronized, and the station should send a reset. This is a half-open connection where only one side is involved in the TCP session. This cannot work by definition of the protocol.

FIN bit is used to indicate that the client will send no more data (but will continue to listen for data).

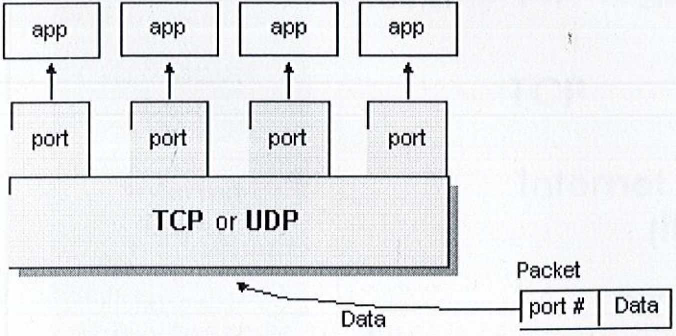
In the particular case of receiving one-way traffic, the RST flag should be used as it closes the connection.

PORTS

Generally speaking, a computer has a single physical connection to the network. All data destined for a particular computer arrives through that connection. However, the data may be intended for different applications running on the computer. So how does the computer know to which application to forward the data? Through the use of ports. Data transmitted over the Internet is accompanied by addressing information that identifies the computer and the port for which it is destined. The computer is identified by its 32-bit IP address, which IP uses to deliver data to the right computer on the network. Ports are identified by a 16-bit number, which TCP and UDP use to deliver the data to the right application. In connection-based communication such as TCP, a server application binds a socket to a specific port number. This has the effect of registering the server with the system to receive all data destined for that port. A client can then rendezvous with the server at the server's port, as illustrated here:



The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer. In datagram-based communication such as UDP, the datagram packet contains the port number of its destination and UDP routes the packet to the appropriate application, as illustrated in this figure:



Port numbers range from 0 to 65,535 because ports are represented by 16-bit numbers. The port numbers ranging from 0 - 1023 are restricted; they are reserved for use by well-known services such as HTTP and FTP and other system services. These ports are called *well-known ports*. Your applications should not attempt to bind to them.

PROTOCOL

In computing, a protocol is a convention or standard that controls or enables the connection, communication, and data transfer between computing endpoints. In its simplest form, a protocol can be defined as the rules governing the syntax, semantics, and synchronization of communication. Protocols may be implemented by hardware, software, or a combination of the two. At the lowest level, a protocol defines the behavior of a hardware connection. Typical properties. While protocols can vary greatly in purpose and sophistication, most specify one or more of the following properties:

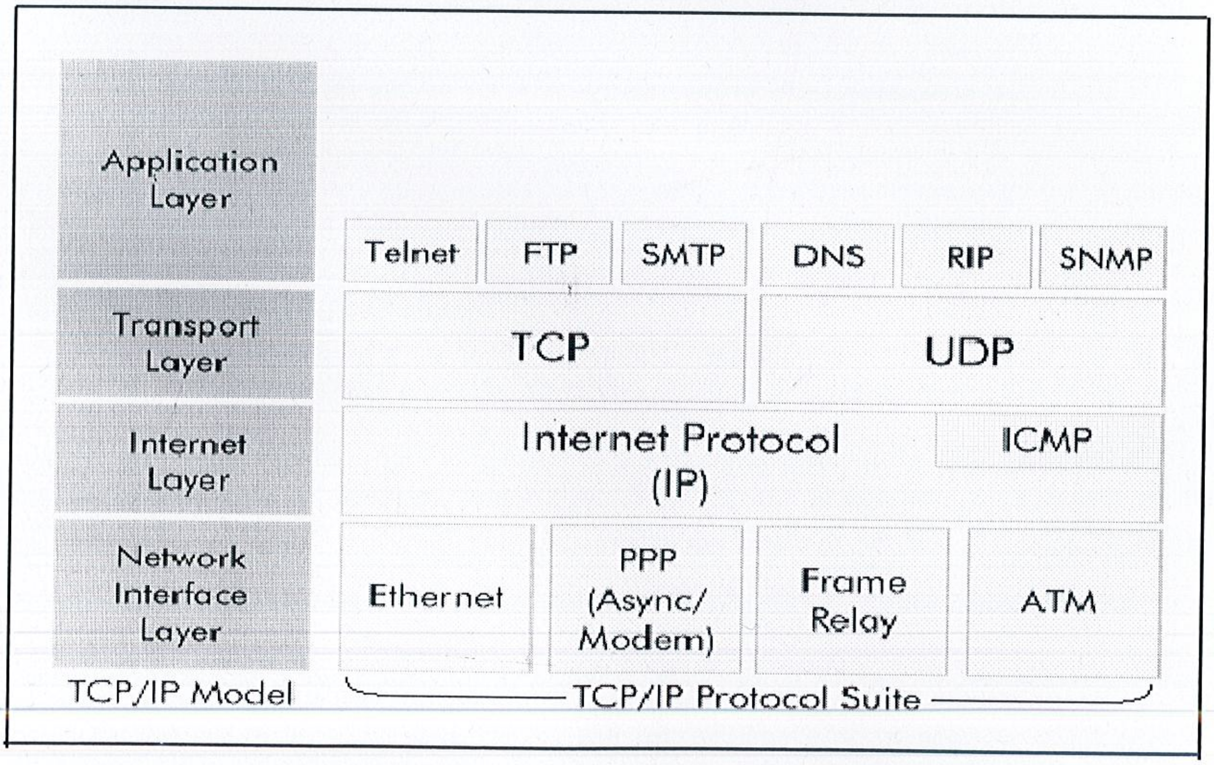
Detection of the underlying physical connection (wired or wireless), or the existence of the other endpoint or node

- Handshaking
- Negotiation of various connection characteristics
- How to start and end a message
- How to format a message
- What to do with corrupted or improperly formatted messages (error correction)
- How to detect unexpected loss of the connection, and what to do next
- Termination of the session and or connection.

Importance

The widespread use and expansion of communications protocols is both a prerequisite for the Internet, and a major contributor to its power and success. The pair of Internet Protocol (or IP) and Transmission Control Protocol (or TCP) are the most important of these, and the term TCP/IP refers to a collection (or protocol suite) of its most used protocols. Most of the Internet's communication protocols are described in the RFC documents of the Internet Engineering Task Force (or IETF).

The protocols in human communication are separate rules about appearance, speaking, listening and understanding. All these rules, also called protocols of conversation, represent different layers of communication. They work together to help people successfully communicate. The need for protocols also applies to network devices. Computers have no way of learning protocols, so network engineers have written rules for communication that must be strictly followed for successful host-to-host communication. These rules apply to different layers of sophistication such as which physical connections to use, how hosts listen, how to interrupt, how to say good-bye, and in short how to communicate, what language to use and many others. These rules, or protocols, that work together to ensure successful communication are grouped into what is known as a protocol suite.



Object-oriented programming has extended the use of the term to include the programming protocols available for connections and communication between objects.

Generally, only the simplest protocols are used alone. Most protocols, especially in the context of communications or networking, are layered together into protocol stacks where the various tasks listed above are divided among different protocols in the stack.

Whereas the protocol stack denotes a specific combination of protocols that work together, a reference model is a software architecture that lists each layer and the services each should offer. The classic seven-layer reference model is the OSI model, which is used for conceptualizing protocol stacks and peer entities. This reference model also provides an opportunity to teach more general software engineering concepts like hiding, modularity, and delegation of tasks. This model has endured in spite of the demise of many of its protocols (and protocol stacks) originally sanctioned by the ISO.

Common protocols

- IP (Internet Protocol)
- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- HTTP (Hypertext Transfer Protocol)
- FTP (File Transfer Protocol)
- Telnet (Telnet Remote Protocol)
- SSH (Secure Shell Remote Protocol)
- POP3 (Post Office Protocol 3)
- SMTP (Simple Mail Transfer Protocol)
- PPP (Point-to-Point Protocol)

CHAPTER4- SNIFFING

Packet analyzer captures network packets and provide view for full TCP conversations and UDP threads The Packet sniffing and analysis features of the packet analyzer are unique. Combining packet sniffing and analysis with network statistics gathering and presentation, it makes the job for a network administrator easier. There are three types of sniffing methods. Some methods work in non-switched networks while others work in switched networks. The sniffing methods are: IP-based sniffing, MAC-based sniffing and ARP-based sniffing.

IP BASED PACKET SNIFFING

Sniff IP information with ease. IP Sniffer provides IP sniffing utility to help you work with IP addresses. The suit includes Domain-to-IP Converter, Batch Ping, Tracert, Whois, Website Scanner and Connection Monitor as well as an IP-to-Country Converter into a single interface. With the powerful IP & Web tool you can:

1. Perform batch and continuous pings on multiple servers;
2. Obtaining all available information on a given IP address or domain name such as Organization or the ISP that owns the IP address, including the country, state, city, address, contact phone numbers and e-mails;
3. Lookup IP address for a single or list of domain names and vice versa;
4. Find out the country associated with a single or list of domains or IP addresses;
5. Trace IP addresses to their destination and investigate connection problems;
6. Determine name, date, last-modified,version and operation system of the remote web server;
7. Allow you to scan any given web site and produce a list of links (including htm cgi php asp jsp jpg gif mp3 mpeg exe zip rar swf and more file types) found in the site, using several criteria to filter the results;
8. Monitor all the TCP/IP connections from your computer to the internet automatically;
9. Get all of the information about the website currently open in the Internet Explorer.

MAC BASED PACKET SNIFFING

MAC based packet sniffing intercepts all network activity; both incoming and outgoing. This means we can to see people's IM conversations, emails sent and received, websites visited, and even pictures they are viewing from web pages. Additionally, we can also hack their usernames and passwords. As you can see from the images above, the data captured may not be exactly intuitive, but its easy enough to see from the first image the person is looking at a Google image search results page in Safari. Looking at images tab (2nd image) you can see the images that were downloaded from the images.google.com page.

ARP BASED PACKET SNIFFING

This method works a little different. It doesn't put the network card into promiscuous mode. This isn't necessary because ARP packets will be sent to us. This happens because the ARP protocol is stateless. Because of this, sniffing can be done on a switched network.

To perform this kind of sniffing, you first have to poison the ARP cache1 of the two hosts that you want to sniff, identifying yourself as the other host in the connection. Once the ARP caches are poisoned, the two hosts start their connection, but instead of sending the traffic directly to the other host it gets sent to us. We then log the traffic and forward it to the real intended host on the other side of the connection. This is called a man-in-the-middle attack. See Diagram 1 for a general idea of the way it works.

PACKET ANALYZER

Packet analyzer is a program that captures all of the packets of data that pass through a given network interface, and recognizes and decodes the packets of interest. It is sometimes referred to as a network monitor. The captured network data can be browsed via a GUI.

Working

- A computer connected to the LAN has 2 addresses.
- The Data Link Layer uses an Ethernet header with the MAC address of the destination machine
- The Network Layer maps IP network addresses to the MAC address as required by the Data Link Protocol
- It initially looks up the MAC address of the destination machine in the Address Resolution Protocol cache
- If no entry is found for the MAC address, the ARP broadcasts a request packet to all machines on the network
- The machine with that address responds to the source machine with its MAC address
- This MAC address then gets added to the source machines ARP Cache.
- This MAC address is then used by the source machine in all its communications with the destination machine

CHAPTER5- FEATURES

FLAGGED HOSTS

It is implemented for the purpose of security. The suspected hosts are marked so as to vigil their activities. The packet analyzer can then generate information that when the suspected users used the network with proper time and date.

PING

Ping is a computer network tool used to test whether a particular host is reachable across an IP network; it is also used to self test the network interface card of the computer, or as a speed test. It works by sending ICMP "echo request" packets to the target host and listening for ICMP "echo response" replies. Ping measures the round-trip time[1] and records any packet loss, and prints when finished a statistical summary of the echo response packets received, the minimum, mean, max and in some versions the standard deviation of the round trip time.The word ping is also frequently used as a verb or noun, where it is usually incorrectly used to refer to the round-trip time, or measuring the round-trip time. The tool is used in a simple denial-of-service attacks, known as a ping flood, in which the attacker overwhelms the victim with ICMP echo request packets. Several tools have been added and written to make IT the perfect choice to backup and restore whole partitions, an easy way.

TRACEROUTE

Traceroute is a computer network tool that shows us the route over the network between two systems, listing all the intermediate routers a connection must pass through to get to its destination. The traceroute tool is available on practically all Unix-like operating systems. Variants with similar functionality are also available, such as tracepath on modern Linux installations and tracert on Microsoft Windows operating systems. It helps us determine why our connections to a given server might be poor and helps us figure out where exactly the problem is. It also shows you how systems are connected to each other, letting you see how ISP connects to the Internet and how the target system is connected.

Implementation

Traceroute works by increasing the "time-to-live" value of each successive batch of packets sent. The first three packets sent have a time-to-live (TTL) value of one (implying that they are not forwarded by the next router and make only a single hop). The next three packets have a TTL value of 2, and so on. When a packet passes through a host, normally the host decrements the TTL value by one, and forwards the packet to the next host. When a packet with a TTL of one reaches a host, the host discards the packet and sends an ICMP time exceeded (type 11) packet to the sender. The traceroute utility uses these returning packets to produce a list of hosts that the packets have traversed en route to the destination. The three timestamp values returned for each host along the path are the delay values in milliseconds for each packet in the batch. If a packet does not return within the expected timeout window, a star is printed. Traceroute doesn't list the real hosts. It indicates that the first host is at one hop, the second host at two hops, etc. IP does not guarantee that all the packets take the same route. Also note that if the host at hop number N does not reply, the hop will be skipped in the output

.

Uses

Traceroute is used for network troubleshooting. By showing a list of routers traversed, it allows the user to identify the path taken to reach a particular destination on the network. It helps identifying routing problems and firewalls that may be blocking access to a site. It can be used to gather information about network infrastructure and IP ranges around a given host. It also tells us the IP addresses of the domains being used and the maximum number of hops it will take before it times out.

Security concerns

Traceroute has been frequently used by hackers to acquire sensitive information about company's network architectures and can quickly map out intermediate routers for known destinations on the architecture. So many networks block traceroute requests or de-prioritize the ICMP time exceeded messages. However, filtering traffic except at network end-points is a controversial practice.

MAC ADDRESS

A unique identifier assigned to network adapters by the manufacturer's for identification. it is intended to be permanent and globally unique; encoding manufacturer's registered identification number; but it is now possible to alter it.

IP ADDRESS

It is a unique number for identification of each system on internet. It might be permanently assigned or supplied each time a user connects to the internet, depending on the service provider, but it has only one IP address at a time.

PORT ADDRESSES

They are assigned at the beginning of the packet so as to make the monitoring of the data fast. However a machine can have more than one port addresses at a time, depending on the number of ports available.

CHAPTER 6- USES AND APPLICATIONS

USES

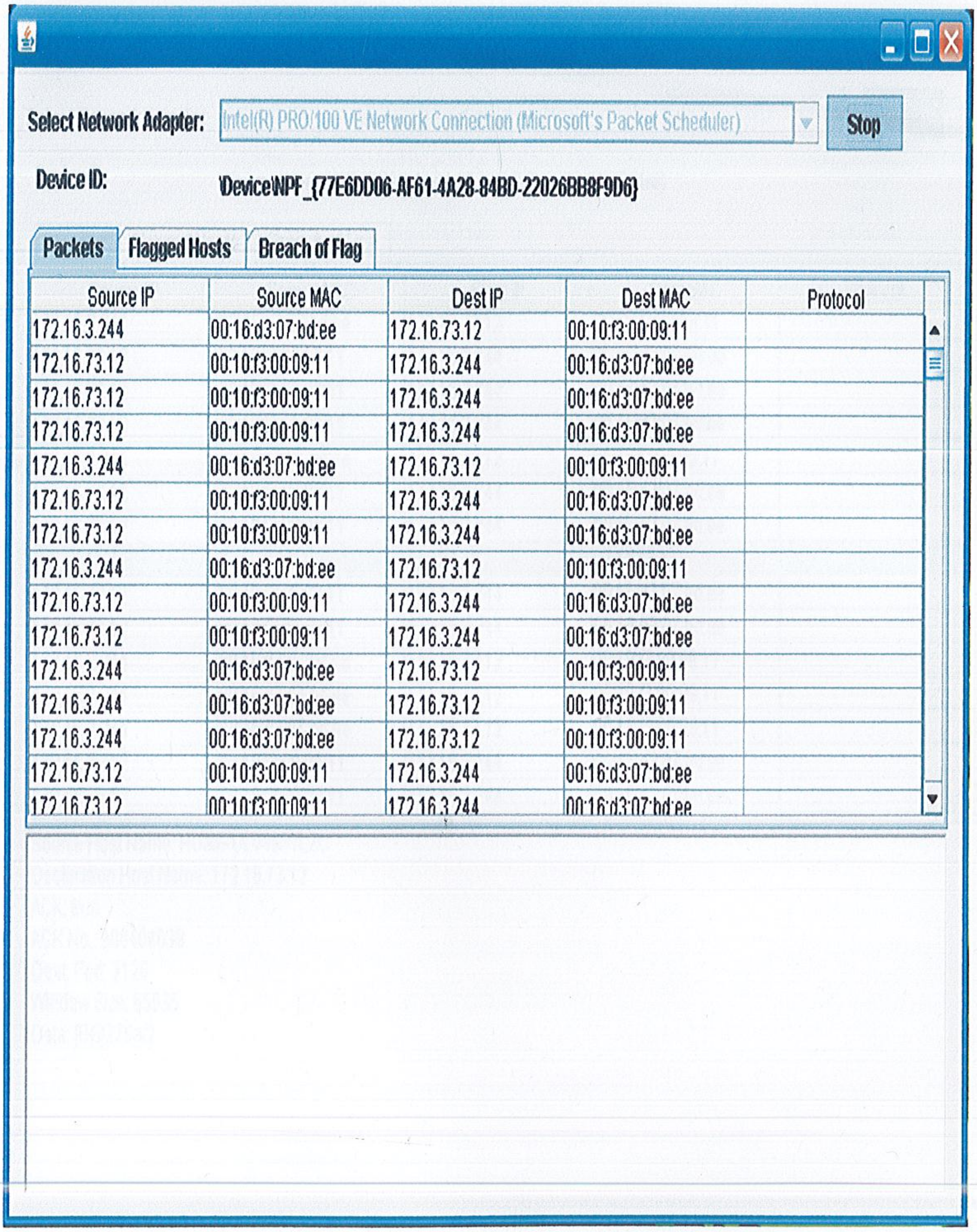
- Analyze network problems
- Detect network intrusion attempts
- Gain information for effecting a network intrusion
- Monitor network usage
- Gather and report network statistics
- Filter suspect content from network traffic
- Spy on other network users and collect sensitive information such as passwords (depending on any content encryption methods which may be in use)
- Reverse engineer proprietary protocols used over the network
- Debug client/server communications
- Debug network protocol implementations
- Can be used in education to demonstrate how network protocols work
- Often used in the development and debugging of networking software
- For a token ring network it can detect if the token has been lost or the presence of too many tokens
- Can detect if messages are being sent to a network adapter, if the network adapter did not report receiving the messages then this would localize the failure to the adapter
- Can detect excessive messages being sent by a port, detecting an error in the implementation
- Can collect statistics on the amount of traffic (number of messages) from a process detecting the need for more bandwidth or a better method
- Used to extract messages and reassemble into a complete form the traffic from a process, allowing it to be reverse engineered
- To analyze data sent to and from secure systems in order to understand and circumvent security measures, for the purposes of penetration testing or illegal activities

HOW DOES IT HELP

- Managing a LAN
- Creating network-oriented software
- Performing a security audit.
- Maintain efficient network data transmission
- Intrusion detection systems
- Identify problems with network-based applications.
- It costs a fraction of the price of information, time, software, and hardware that may potentially be lost or wasted by *not* using a network analyzer

SCREEN SHOTS

SCREEN SHOT 1: A SCREEN SHOT OF ADDRESS DETAILS BEING GENERATED OF THE TRANSFERRED PACKETS

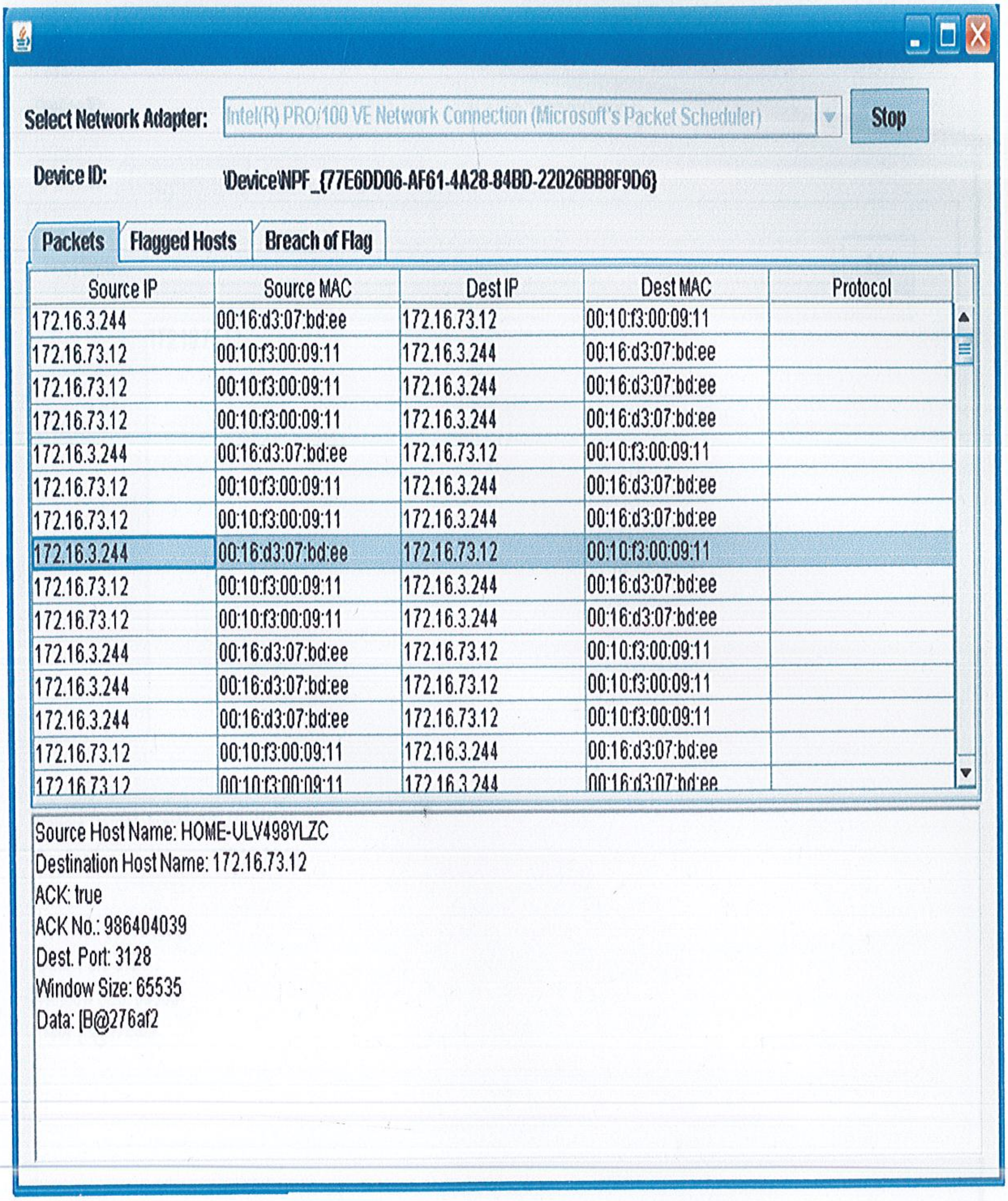


Select Network Adapter: Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler)

Device ID: \Device\NPF_{77E6DD06-AF61-4A28-84BD-22026BB8F9D6}

Source IP	Source MAC	Dest IP	Dest MAC	Protocol
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	

SCREEN SHOT 2: VITAL DETAILS OF A SELECTED PACKET



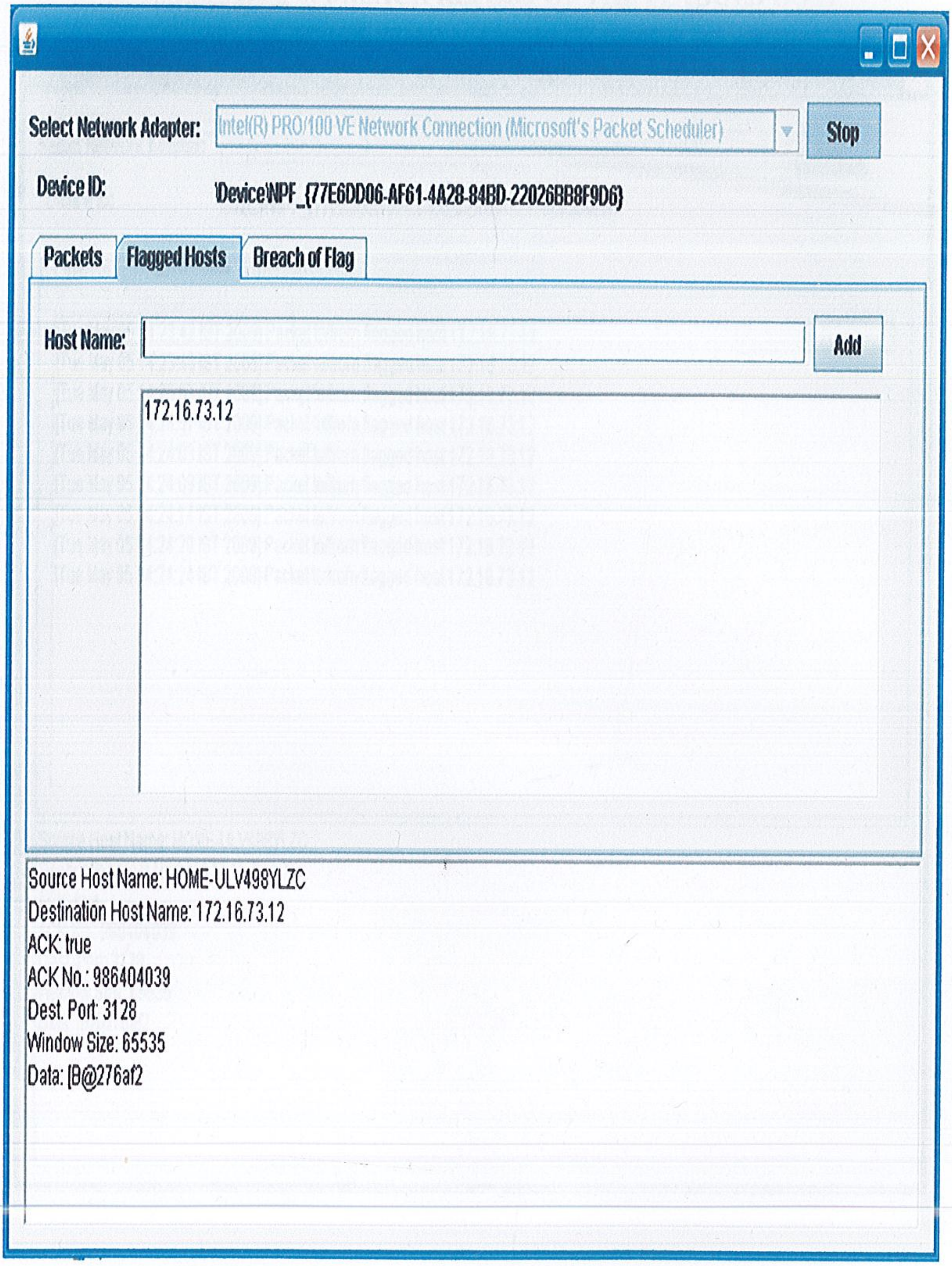
Select Network Adapter: Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler)

Device ID: \Device\NPF_{77E6DD06-AF61-4A28-84BD-22026BB8F9D6}

Source IP	Source MAC	Dest IP	Dest MAC	Protocol
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.3.244	00:16:d3:07:bd:ee	172.16.73.12	00:10:f3:00:09:11	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	
172.16.73.12	00:10:f3:00:09:11	172.16.3.244	00:16:d3:07:bd:ee	

```
Source Host Name: HOME-ULV498YLZC
Destination Host Name: 172.16.73.12
ACK: true
ACK No.: 986404039
Dest. Port: 3128
Window Size: 65535
Data: [B@276af2
```

SCREEN SHOT 3: ADDING A HOST IP ADDRESS FOR SURVEILLANCE



Select Network Adapter: Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler)

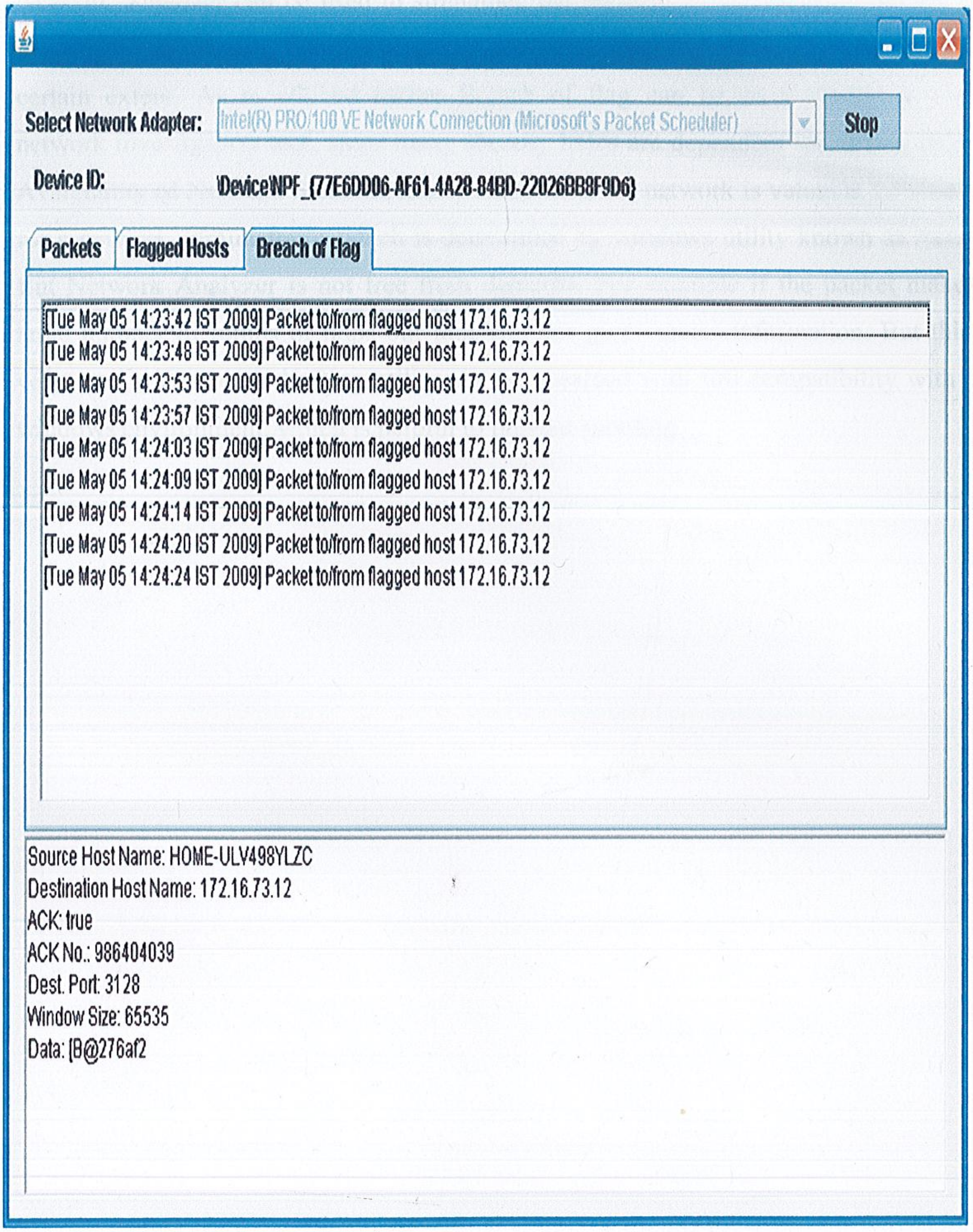
Device ID: \Device\NPF_{77E6DD06-AF61-4A28-84BD-22026BB8F9D6}

Host Name:

172.16.73.12

```
Source Host Name: HOME-ULV498YLZC
Destination Host Name: 172.16.73.12
ACK: true
ACK No.: 986404039
Dest. Port: 3128
Window Size: 65535
Data: [B@276af2
```

SCREEN SHOT 4: BREACH REPORT OF THE FLAGGED HOST



Select Network Adapter: Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler)

Device ID: \Device\NPF_{77E6DD06-AF61-4A28-84BD-22026BB8F9D6}

```
[Tue May 05 14:23:42 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:23:48 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:23:53 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:23:57 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:24:03 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:24:09 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:24:14 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:24:20 IST 2009] Packet to/from flagged host 172.16.73.12
[Tue May 05 14:24:24 IST 2009] Packet to/from flagged host 172.16.73.12
```

```
Source Host Name: HOME-ULV498YLZC
Destination Host Name: 172.16.73.12
ACK: true
ACK No.: 986404039
Dest. Port: 3128
Window Size: 65535
Data: [B@276af2
```

CONCLUSION

Network Analyzer can be used to strengthen the security of our network. Its resource identification like protocol detection is helpful to prevent security vulnerability in a certain extent. As mentioned earlier Breach of flag can be an extremely valuable network investigation tool, since many security holes are dependent on Breach of flag. Availability of Network Statistics at any moment in the network is valuable information for a network administrator which is determined by windows utility known as Netstat. But Network Analyzer is not free from demerits. For example if the packet makes a large number of jumps or hops our tool will not give correct information. But this is very rare in the network. Ping utility can be accessed with full compatibility with the windows environment, which is helpful to prevent spoofing.

BIBLIOGRAPHY

Books:

- Java- How To Program by Deitel and Deitel
- Java 2-Complete Reference by Herbert Schildt
- TCP/IP JumpStart- Internet Protocol Basics by Andrew G. Blank
- Thinking in Java by Bruce Eckel
- Computer Networks 4th Ed. by Andrew S. Tanenbaum

Web Pages:

- www.wikipedia.org
- www.winpcap.politeo.com
- www.google.com
- www.java.sun.com
- www.javabeginner.com