Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.406 – 410

RESEARCH ARTICLE

A Protocol Based Packet Sniffer

Yash Ketkar[1], Wasim Khan[2], Deep Makwana[3], Vikrant Nemade[4], Ankush Hutke[5]

Department of Information Technology, MCT's Rajiv Gandhi Institute of Technology, University of Mumbai, Mumbai, Maharashtra, India

[1] yashketkar93@gmail.com; [2] wasimkhan1992.wk@gmail.com; [3] dee2rocker@gmail.com; [4] vicky.nemade@gmail.com; [5] ankush.hutke@mctrgit.ac.in

Abstract— In the past five decades computer networks have kept up growing in size, complexity and in the number of its users as well as being in state of permanent evolution. Therefore the size of network traffic flowing over their nodes has increased dramatically. With the development and popularization of network technology, the maintenance and monitoring of network is important to keep the network smooth and improve economic efficiency. This report focuses on the basics of a protocol based packet sniffer and its working, development of the tool on Windows platform and its use for Intrusion Detection. Focus has also been laid to analyze the bottleneck scenario arising in the network by use of this protocol based packet sniffer.

Keywords— Packet, Capture, Sniffer, Network, Protocol

I. INTRODUCTION

Packet sniffing is a method of tapping each packet as it flows across the network. Packet Sniffing is a technique in which a user sniffs data belonging to other users of the same network. Packet sniffers can be used for malicious purposes or it can be operated as an administrative tool. It is dependent on the user's intentions. Network admins generally use packet sniffers for monitoring and validating network traffic. Packet sniffers are nothing but applications. They are application programs used to read packets that travel across the network layer of the Transmission Control Protocol/Internet Protocol layer. (The packets are retrieved from the network layer and the data is interpreted.) Packet sniffers are used by the network administrators as utilities for efficient network administration. However, a user can employ a number of techniques to detect sniffers on the network and protect the data from sniffers.

II. LITERATURE SURVEY

We studied the working behavior of already existing sniffer software such as Wireshark [1] (formerly known as ethereal), tcpdump, and snort [2]. Each of these programs offers different features and limitations. Studying the current system we have drawn the following conclusions

- Packet Sniffers only give the log of data, this has to be analyzed by a network administrator to find the error or an attack on the network adapter.
- Current systems are only able to show the logs of packets.
- They are not up to date with the current generation of network infrastructure and thus are deficient.
- The limitations of protocol based analysis include the fact that it is extremely time-consuming to capture every packet, examine them, disassemble every one, and manually take an action based on the interpretations from the analysis.

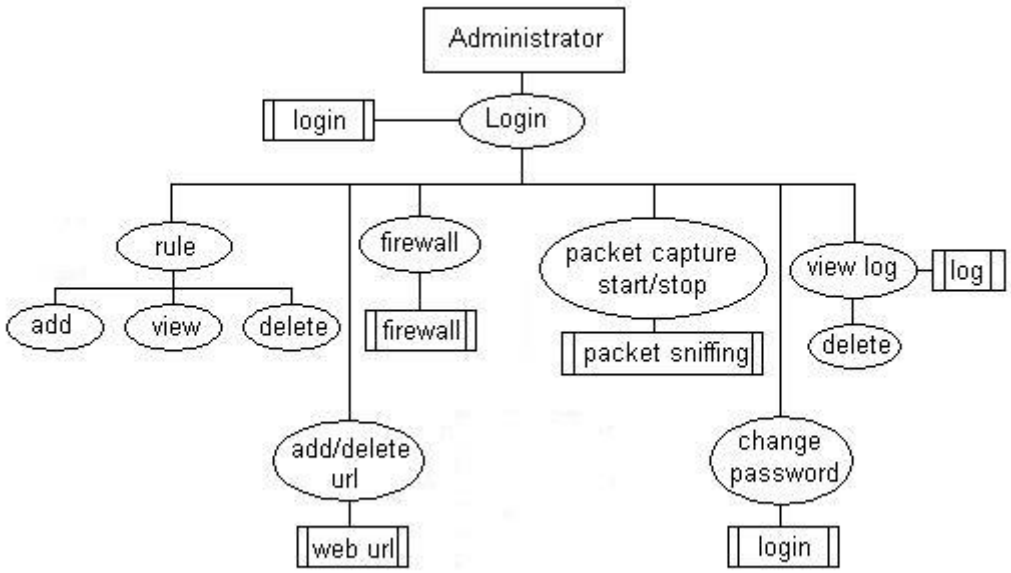

Figure 1 : System Diagram

Hence intrusion detection systems needs to be developed for transforming manual functions into automated programs to analyze and decipher data collected [3].

III. PROPOSED SYSTEM

A protocol based packet sniffer which will trace packets by attaching itself to a network adapter in promiscuous mode.

The packets will be traced according to the filtering criteria set by the network administrator.

The acceptance and rejection of packets will be done based on this criteria.

It will also store the log into a database for future analysis [4].

The built-in intrusion detection system will reduce the efforts of the network admin required to manually analyze each packet.

The automated system will reduce time required to analyze each packet and effectively decode the contents of each packet [5].

Figure 1.0 illustrates the proposed system to curb the drawbacks obtained upon studying the current system.

A. Packet Sniffing in a Network

A Sniffer is a piece of software that grabs all of the traffic flowing into and out of a computer attached to a network. Intrusion Detection Systems (IDS) use sniffers to match packets against a rule-set designed to flag any kind of malicious or strange activity. Network utilization and monitoring programs often use sniffers to gather data necessary for analysis. Cyber-Crime and Law enforcement agencies that need to monitor email during investigations, likely use a sniffer designed to capture very specific kind of traffic. Knowing that sniffers simply grab network data.

When a computer sends a data in the network it sends in the form of packets. These packets are the chunks of data are actually directed to the certain designated system. Actually every sent data has a predefined receiving point. So, all the data are directly directed to a particular computer. Normally a system in a network is designed to receive and read only those data which are intended for it, but when we install a packet sniffer on a network, it looks out for all the data traveling across the network.

The packet-sniffing process involves a cooperative effort between software and hardware. This process can be broken down into three steps such as

Packet sniffer collects raw binary data from the wire. Typically, this is done by switching the selected network interface into promiscuous mode

Analysis of the captured and converted data. The packet sniffer takes the captured network data and then verifies its protocol based on the information extracted, and begins its analysis of that protocol's specific features.

B. Intrusion Detection System

Intrusion detection system (IDS) is a type of security management system for computers and networks. An Intrusion Detection system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). Intrusion Detection uses scanning (also known as vulnerability assessment), which is a type of technology developed to assess the security of a computer system or network.

Packet trace is a software that traces all the incoming packets onto a computer from an Intranet or Internet. Here the software will read the packet header information and display:

1) Source protocol
2) Source port
3) Destination Port
4) Source IP
5) Destination IP
6) Date and Time

Based on this information the network administrator can either reject or accept the packet. Acceptance and Rejection can be done on the basis of the protocol.

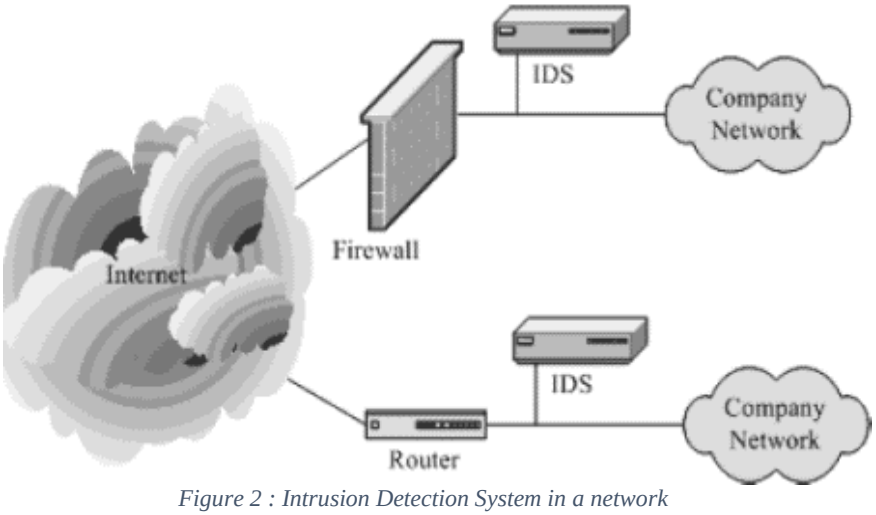

Figure 2 : Intrusion Detection System in a network

Figure 2 illustrates Intrusion Detection System in a network, An Intrusion Detection system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).

C. Methodology

The sniffer program makes the network interface card (NIC) on the machine enter into a promiscuous mode. An Ethernet NIC is built with a "filter" that ignores all traffic that does not belong to it. It will ignores all frames whose destination MAC address does not match with its own. Through the network interface card's driver, a sniffer turns off this filter and thus puts the network interface card into promiscuous mode.

The typical NICs used in workstations and PCs can be put into promiscuous mode with ease. In fact, on many network interface cards, it is possible to modify their MAC addresses. The equipment needs to observe all traffic, and thus be promiscuous.

The program will have the following components. Firstly, the packet sniffing using the Winsock API for capturing all the packets from the network traffic.

Secondly, the construction of a log of all the packet with the data from their fields into a database (a Microsoft Access database in this case), and then analyzing the fields based on the filtering criteria. We shall also detect any unusual activity or malicious attacks on a system. The network administrator would not have to manually go through each and every log as the intrusion detection system will automatically do so.

D. Windows Sockets

The Windows Sockets API (WSA), which was later shortened to Winsock, is a technical specification that defines how Windows network software should access network services, especially TCP/IP.

It defines a standard interface between a Windows TCP/IP client application (such as an FTP client or a web browser) and the underlying TCP/IP protocol stack. The naming is based on the Berkeley sockets API model used in BSD for communications between programs.

The Windows Sockets API specification defines two interfaces: the API used by application developers, and the Service Provider Interface, which provides a means for network software developers to add new protocol modules to the system. The application programming interface guarantees that a conforming application will function correctly with a conforming protocol implementation from any network software vendor. The Service Provider Interface contract guarantees that a conforming protocol module may be added to Windows and will thereby be usable by an API-compliant application.

Microsoft has shipped the TCP/IP protocol stack with all recent versions of Windows, and there are no significant independent alternatives. Nor has there been significant interest in implementing protocols other than TCP/IP.

E. Implementation

Initially the software will attach itself to the network interface controller. It will then enter a promiscuous mode. The Promiscuous mode is a mode for a network interface controller that causes it to pass all traffic it receives to the central processing unit (CPU) rather than just passing frames the NIC is intended to receive. The network administrator will then select the filtering criteria. This would include all the various protocols. The software would then filter the packets based on this criteria.

The software would save the log of all the incoming network traffic in the database. It would also detect any malicious activity by using the Intrusion Detection System. The network administrator would then be reported about such activity. Such user could later be blocked from the network by the admin.

We will achieve this using the WinSock control, Winsock is extensible by a mechanism known as a Layered Service Provider. Winsock Layered Service Providers are available for a wide range of purposes, including parental controls, web content filtering, Quality of service etc. The layering order of all providers is kept in the Winsock Catalog. In older versions of Windows, removing a buggy Layered Service Provider could result in corruption of the Winsock catalog in the registry, possibly resulting in a loss of all network connectivity.

Some issue proposed system will overcome over the current system:

Some tools only capture network traffic without analysis, therefore the researcher has to use another tools for analysis to get the traffic feature as needed in work.

Some tools have large memory requirement. So we can design a tool that captures network traffic and analyze it and allows user to filter the packets as per needs and store it in database to use it later, this will reduce the memory that is used to store the data.

The Intrusion Detection System will automatically detect any malicious activity and report it to the network administrator.

IV. CONCLUSIONS

Compared to other similar softwares this application shows the protocols involved in sniffing as well as other fields of the packet.

Protocol based Packet Sniffer has a user friendly graphical user interface as compared to other command line applications. Thus it is very easy to use. With Windows Sockets, the most considerable advantage is that it is compatible with older as well as the newer versions of Windows. The application itself is also very small in size and requires very less memory as we can easily export the captured data to a database. The application can also adapt any further changes because of its design.

ACKNOWLEDGEMENT

It gives the authors great pleasure in expressing our gratitude to all those people who have supported us and had their contributions in making this dissertation possible. First and foremost, we express our profound sense of reverence to our guide **Prof. Ankush Hutke**, for his constant guidance, support, motivation and untiring help. We are also thankful to Head of the Information Technology Department and Principal of MCT's Rajiv Gandhi Institute of Technology for their support and valuable suggestions. We are also thankful to all staff members of Information Technology Department, without whom the completion of this report would have been impossible.

REFERENCES

[1] S. Ansari, Rajeev S.G. and Chandrasekhar H.S, "Packet Sniffing: Brief Introduction", *IEEE Potentials*, Dec 2002- Jan 2003, Volume: 21 Issue: 5, pp: 17 – 19

[2] Qadeer M.A., Zahid M., Iqbal A., Siddiqui M.R "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer" *ICCSN* '10 Second International Conference, 2010, Page(s): 313 - 317

[3] G.Varghese, "Network Algorithmic: An Interdisciplinary Approach To Designing Fast Networked Devices", San Francisco, CA: Morgan Kaufmann, 2005.

[4] J. Cleary, S. Donnelly, I. Graham, "Design Principles for Accurate Passive Measurement," in Proc. PAM 2000 Passive and Active Measurement Workshop (Apr. 2000).

[5] A. Dabir, A. Matrawy, "Bottleneck Analysis of Traffic Monitoring Using Wireshark", 4th International Conference on Innovations in Information Technology, 2007, *IEEE Innovations* '07, 18-20 Nov. 2007, Page(s):158 – 162

