International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Network Sniffer Using Graphical Visualization

Tejas Nalawade[1], Ashutosh Singh[2], Prof. Vidya Bharde[3]

[1,2]Student, Dept. of Computer Engineering, M.G.M. College of Engineering and Technology, Kamothe, Maharashtra, India
[3]Assistant Professor, Dept. of Computer Engineering, M.G.M College of Engineering and Technology, Kamothe, Maharashtra, India

---

Abstract - *This system presents the total implementation of the Network sniffer application code that captures network information as well as provides ample suggests that for the decision-making method of an administrator. This system aims to rewrite the C language somebody into Java, and conjointly develop an application that consumes very little memory on the hard disc. This work illustrates the need required to style a brand-new application; it absolutely was developed in Java and it consumes very little memory on the hard disc. This application includes 5 modules that handle completely different tasks expeditiously. this technique will monitor network traffic, analyzes traffic patterns, determine and troubleshoot network issues. This application doesn't transmit any information onto the network, uses 1MB of the hard disc space, friendly user interface, and is extremely straightforward to put in.*

Key Words: Network traffic, packets, packet capture, packet Analyzer /sniffer**.**

1. INTRODUCTION

Packet sniffing is a method of tapping each packet as it flows across the network; i.e., it is a technique in which a user sniffs data belonging to other users of the network. Packet sniffers can operate as an administrative tool or for malicious purposes [2]. It depends on the user's intent. Network administrators use them for monitoring and validating network traffic. Packet sniffers are applications. This is the system used to read packets that travel across the network layer of the Transmission Control Protocol/Internet Protocol (TCP/IP) layer [4]. (Basically, the packets are retrieved from the network layer and the data is interpreted.) Packet sniffers are utilities that can be efficiently used for network administration [6]. At the same time, it can also be used for nefarious activities. However, a user can employ several techniques to detect sniffers on the network and protect the data from sniffers.

A packet sniffer can intercept and log traffic passing over a digital network or part of a network. As data streams travel back and forth over the network, the sniffer captures each packet and eventually decodes and analyzes its content according to with any specifications [4].

1.1 MOTIVATION

Wi-Fi and Bluetooth network usage is growing continuously, increasing traffic capacities in the wireless medium. Therefore the need to monitor the network also increases to protect it from other people who are trying to steal sensitive information. While wireless network monitoring does require new tools, it incorporates many of the same reasons for monitoring as described within other sections. Packet sniffing still provides significant insight into the operations of users, and may be of higher use within wireless networks; if the administrator can easily sniff and decode passwords and sensitive information transmitted within the network, it is quite likely that an attacker can and will do the same. Monitoring the application usage of users is similarly of great importance; active wireless users are at higher risk of attack as compared to wired users behind firewalls and network address translation (NAT), and consequently detecting any vulnerable systems on wireless machines is critical. The security issues with wireless networking lead to its motivations and impact its analysis..

For most organizations packet sniffer is largely an inner danger. Packet sniffers can be operated in each switched and non-switched surroundings [4]. Determination of packet sniffing in a non-switched environment is an era that may be understood with the aid of each person. In this generation, all hosts are related to a hub. There is a huge quantity of commercial and non-commercial tools are available that make possible eavesdropping of community traffic. Now trouble comes that how this network site visitor can be eavesdrop. this trouble may be solved using setting the network card right into a special promiscuous-mode.

2. LITERATURE REVIEW

Lliterature on the design and implementation of packet sniffers is fairly sparse. However, there is a wealth of information on the design of general network devices [1]. This literature is general enough to apply to any networked computer system and describes in detail the many protocols used for network communication.

Existing network management systems typically use a combination of textual displays and 2D directed graph representations of network topology. Designing a network management system that instead uses a virtual world presented through a 3D stereo display and manipulated with a 3D mouse [1]. Our goal is to allow the user to better understand and control the structure and behavior of a large, complex network. In our current prototype, the user interacts with a 3D representation of a network whose topology and behavior are specified by a separate network emulator. The user can choose from among a set of different views of the network. For example, one view shows a selected virtual path as a series of logical links contained within a physical path [3].

Most current management systems employ graphic-user-interface displays to visualize the networks being managed, this approach is rather difficult to apply to extremely large-size networks (e.g. those with hundreds of complexly connected devices) since the full picture cannot easily be presented within the limited display space available [6]. The explosive growth of malicious activities on worldwide communication networks, such as the Internet, has highlighted the need for efficient intrusion detection systems [2]. The efficiency of traditional intrusion detection systems is limited by their inability to effectively relay relevant information due to their lack of interactive/immersive technologies. In this system, we explore several network visualization techniques geared towards intrusion detection on small and large-scale networks

3. SYSTEM ANALYSIS

Packet sniffers work by intercepting and logging network traffic that they can 'see' via the wired or wireless network interface that the packet sniffing software has access to on its host computer. On a wired network, what can be captured depends on the structure of the network [6]. A packet sniffer might be able to see traffic on an entire network or only a certain segment of it, depending on how the network switches are configured, placed, etc. On wireless networks, packet sniffers can usually only capture one channel at a time unless the host computer has multiple wireless interfaces that allow for multichannel capture. Once the raw packet data is captured, the packet sniffing software must analyze it and present it in human-readable form so that the person using the packet sniffing software can make sense of it [5]. The person analyzing the data can view details of the conversation' happening between two or more nodes on the network. Network technicians can use this formation to determine where a fault lies, such as determining which device failed to respond to a network request.

3.1 PROBLEM DESCRIPTION

A packet sniffer can be a computer system or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network [1]. A sniffer system that targets packets of data transmitted over the Internet. Packet sniffing may sound like the latest street drug craze but it's far from it. Packet sniffers or protocol analyzers are tools that are commonly used by network technicians to diagnose network-related problems [2]. A packet analyzer (also known as a network analyzer.

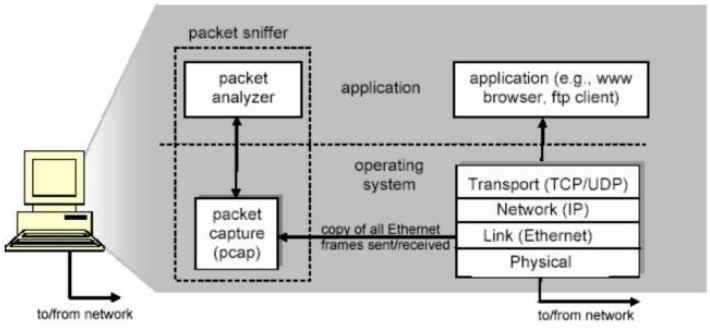

Fig -1: Flow of System

3.2 ISSUES & CHALLENGES

The first challenge was to correctly detect the packets traveling throughout the network. To detect the different layers from which it has come has the main issue as well as challenge for us. The Detection and Identification of the different layers and all surrounding information about the packet was a big challenge for that. Then the key challenge in writing such software is to collect raw packets directly from the interface cards and parsing them to reveal useful information. In a normal network stemming through sockets, a software module listens on a particular socket for packets intended for its use, hence for a module wanting to sniff for all packets, it shall have to listen on all the TCP ports so that TCP does not throw away packets on finding any module attached to the intended port number in the packet. Also, each protocol layer performs filtering of the traffic, for example, any TCP control packet will not be passed above the TCP layer, any IP control packet is consumed by the IP layer, and so on. Moreover, the hardware network interface does initial filtering of packets not intended for it. Hence, it is almost certain that the normal stemming methods will not allow for the capabilities that we seek to capture in packet sniffing-software.

3.3 PROPOSED SYSTEM

As a network analyzer (as an a. packet sniffer), this system makes it easy for us to monitor and analyze network traffic in

its intuitive and information-rich tab views. With this system network traffic monitor feature, we can quickly identify network bottlenecks and detect network abnormities [2]. This application is to discuss how we can monitor network traffic with this network traffic monitor feature

This system provides a summary view that provides general information of the entire network or the selected node in the explorer. In the Summary view, we can get a quick view of the total traffic, real-time traffic, broadcast traffic, multicast traffic and so on. When we switch among the node from the explorer, corresponding traffic information will be provided [6].

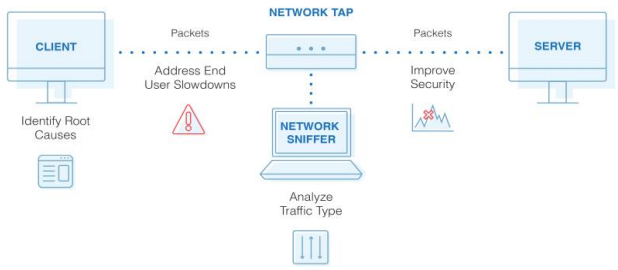

Fig-2: Basic Working of Network Sniffer

In the Endpoints view; we can monitor network traffic information of each node, both local and remote. With its easy sorting feature we can easily find out which host is generating or has generated the largest traffic [4].

The Protocols view will list all protocols applied in network transmission. In the Protocols view, we can monitor network traffic by each protocol. By analyzing network traffic by protocol, we can understand what applications are using the network bandwidth, for example, "HTTP" protocol stands for website browsing, "pop3" stands for email, etc [3].

4. SYSTEM DESIGN

4.1 Input Design

The Network sniffer is developed in Java TM. This application is designed into 5 independent modules which take care of different tasks efficiently are :

1. User Interface Module.
2. Packet Sniffing Module.
3. Analyze layers Module.
4. Free Memory Module.
5. Protocol Analysis Module.

4.2 User Interface Module

Every application has one user interface for accessing the entire application. The user interface for the Network Sniffer application is designed completely based on the end-users. It provides an easy-to-use interface to the users. This user interface has an attractive look and provides ease of navigation. Technically, the swing is used in core Java for preparing this user interface.

4.3 Packet Sniffing Module

This module takes care of capturing packets that are seen by a machine's network interface. It grabs all the packets that go in and out of the Network Interface Card (NIC) of the machine on which the sniffer is installed. This means that, if the NIC is set to the promiscuous mode, then it will receive all the packets sent to the network.

4.4 Analyze Layers Module

This module contains the code for analyzing the layers in the system. Mostly in this module, we have to discuss three layers Transport layer, the Application Layer, Network Layer. The module shows the graphical representation of the usage of different layers in packet capturing time. It can show the graph in two manners like line graph and pie graph.

4.5 Protocol Analysis Module

This module analyzes the protocols of the layers. Like Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Hypertext Transfer Protocol (HTTP), etc. It can show the source port, destination port, and packet length of the system of each protocol.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity, this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create DFD's are as follows:

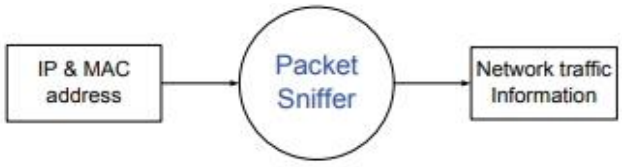

Fig -3: Level 0 Process Flow Diagram

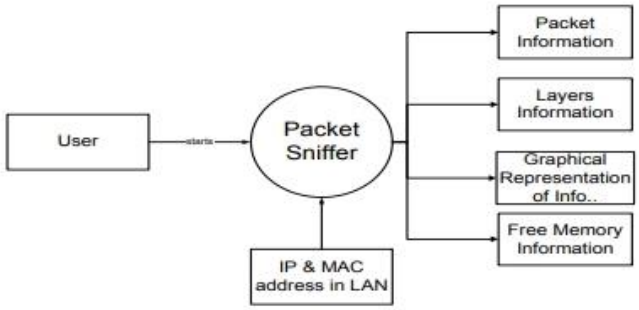

Fig -4: Level 1 Process Flow Diagram

A deployment diagram serves to model the physical deployment of artifacts on deployment targets. It shows "the allocation of Artifacts to Nodes according to the Deployments defined between them. "Deployment of an artifact to a node is indicated by placing the artifact inside the node. Instances of nodes (and devices and execution environments) are used in deployment diagrams to indicate the multiplicity of these nodes. For example, multiple instances of an application server execution environment may be deployed inside a single device node to represent application server clustering.

4. RESULTS & DISCUSSIONS

Performing packet captures using a sniffer may be a particularly powerful technique for identification advanced problems. When all else fails, it's typically useful to look at the data being sent across the wire. The packets don't lie, and analyzing your application traffic at an occasional level will reveal deeper symptoms of a haul (or reveal a haul that you simply didn't even grasp existed).

In computer networking, promiscuous mode is a mode for a wired network interface controller (NIC) or wireless network interface controller (WNIC) that causes the controller to pass all traffic it receives to the central processing unit (CPU) instead of passing solely the frames that the controller is specifically system to receive.

This mode is often used for packet sniffing that takes place on a router or a computer connected to a wired network or one being a part of a wireless LAN. Interfaces area unit placed into promiscuous mode by computer code bridges typically used with hardware virtualization.

5. CONCLUSIONS

The Application is developed in Java Language. The system is able to capture all the packets and categorizes them according to its types. The system is also able to detect and display all the information about the packet i.e IP version , Source IP , Destination IP, throughput etc and many more.

The system also successfully able to display with the help of pie charts and categorized all the things in it. Network Sniffer has a very rich and user friendly GUI developed in Java Swing Technology. Thus it is totally easy to use. With Java, the most considerable advantage is platform independence therefore Network Sniffer is also platform independent.

6. FUTURE SCOPE

It is not possible to develop a system that meets all the requirements of the user. User requirements keep changing as the system is being used.

As the technology emerges, it is possible to upgrade the system that can be adaptable to desired environment. The present application is a standalone application, i.e. only in intranet. So we have chance to extend this in internet. Based on the future security issues, security can be improved using emerging technologies

REFERENCES

[1] Awodele, O., & Otusile, O. (2012). The design and implementation of Network Sniffer model for network security. International Journal of Electronics Communication and Computer Engineering.

[2] Chan, C. Y. (2002). A network packet analyzer withDatabase support. Retrieved from http://www.cs.rpi.edu/~szymansk/theses/chan.ms.02.pdf A Packet Sniffer (Network Sniffer) Application.

[3] Tom King, "Packet sniffing in a switched environment", SANS Institute, GESC practical V1.4, option 1, Aug 4th 2002, updated june/july 2006.

[4] Ansari, S., Rajeev, S., & Chandrashekar, H. (2002).Packet sniffing: A brief introduction. IEEE Potentials,21(5), 17-19.

[5] Pallavi Asrodia, Hemlata Patel, "Network traffic analysis using packet sniffer", International Journal of Engineering Research and Application (IJERA), Vol.2, pp. 854-857, Issue 3, May-June 2012.

[6] Ryan Splanger, "Packet sniffing detection with Anti sniff", University of Wisconsin-Whitewater, May 2003.