Global Scientific JOURNALS

GSJ: Volume 10, Issue 7, July 2022, Online: ISS 2320-9186

www.globalscientificjournal.com

ONLINE NETWORK MONITORING AND PACKET TRAFFIC ANALYSIS USING SNIFFER APPLICATION.

Nwakeze Osita Miracle
Ma.nwakeze@coou.edu.ng

Prof. Okeke Ogochukwu C
ogoookeke@yahoo.com

Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli

ABSTRACT

Network analysis is the process of capturing network traffic and inspecting it closely and determine what happened on the network.

The data packets of popular protocols are decoded by a network analyzer, which then shows the network traffic in readable format. A sniffer is a program that keeps track of data as it travels across a network.

This study aims on the design and development of a virtual real time intranet networks monitoring using packet sniffing, provide accurate evidence on corporate fraud when investigation is being carried out in an organization. These network analyzers converts raw binary data into human-readable format which helps to analyze the network. The methodology adopted and implemented was the Object Oriented Analysis and Design Methodology (OOADM). C# with SQL was used as a programming language to develop this system.

Keywords: Network, Packet, Protocol, Intranet, sniffing, Software, Monitoring

1.0 Introduction

Network analysis (also known as traffic analysis, protocol analysis, sniffing, packet analysis, eavesdropping, and others) is the technique of capturing network traffic and closely studying it to figure out what's going on the network (Rane,2017).

A network analyzer decodes common protocol data packets and shows network traffic in format that is readable. A sniffer is a program that keeps track of data as it moves across a network (Rane,2017).

A network analyzer can be either a standalone hardware device with specialized software or software loaded on a desktop or laptop computer. Network analyzers are computer programs or, in certain cases, hardware devices that can listen to all traffic traveling via a network. These network analyzers can convert raw binary data into human-readable format which helps to analyze the network. Packet sniffing is a network monitoring technique that examines every packet that passes across it. Wireshark and Netcut are two tools that are often used to perform packet sniffing techniques. The packet sniffing process is broken into three steps: collecting, converting, analyzing, and data theft. In this study, the sniffing packet was only used for monitoring and analyzing network traffic.

Network monitoring entails a variety of techniques that are used to ensure the security and integrity of an internal network. The internal network is also referred to as a Local Area Network (LAN), and monitoring includes hardware, software, viruses, spyware, vulnerabilities such as backdoors and security flaws, and other factors that can jeopardize a network's integrity.

Network administrators are continuously attempting to keep their networks running smoothly. Monitoring agents must detect, isolate, and fix network faults, as well as possible recover the failure, when a network failure occurs. Furthermore, they must monitor network performance on a frequent basis when network devices are overburdened. Packet Sniffer is a software or hardware device that can be used to monitor a network. Packet sniffing is the method used to track network transport packets (Xie, 2019).

Every packet moving through it is detected by Packet Sniffer. The administrator will detect network problems and ensure secure network data transfer by means of the information gathered by packet sniffers.

Packet sniffers are installed on computers in a network, and once activated, they make copies of all network traffic packets that are sent and received by the host computer (Paessla, 2018).

They are used for a variety of reasons, including: as a problem solving tool to fix network problems, as a performance tool to identify bottlenecks in the network and areas where efficiency can be improved, and as a technique in security management. A network administrator can grab those packets that are passing through the network to trace any access to the network (Kong, 2018).

A packet sniffer is used for detecting messages being sent and received from a network interface, detecting an error implementation in network software and collecting statistics and the network traffic (Garg, 2019).

1.2 Statement of the Problem

i. Lack of intranet network monitoring packet traffic analysis.

ii. Most of the packet sniffer have less user-friendly graphical interface or sometimes use command-line tool only.

Objectives of the Study

i. To design and develop a virtual real time intranet networks monitoring using packet sniffing.

ii. To display a more user-friendly graphical interface in packet sniffer.

Literature Review

Concept of Network Sniffing

Network sniffing describes the process of monitoring, capturing and interpreting all incoming and outgoing traffic as it flows across a network, it is achieved by a packet sniffer; a tool used to capture raw network data bits going across the medium (Bhandari, 2017). Network sniffing can help to understand network characteristics, learn who is on the network, determine who and what is utilizing available bandwidth, identifying peak network usage time, identifying possible attacks or malicious activity, and find unsecured and bloated applications (Bhandari, 2017).

Information is transported across a network in the form of packets," which are smaller segments with a destination and source address attached. A Packet sniffer can reveal all sorts of things going on behind the scenes, including unknown communication between network devices, actual detailed error codes provided by layer-specific protocols (Sudibyo, 2019).

Sniffing is the process of capturing, decoding, inspecting, and interpreting the data from packets sent over a transmission channel, such as a TCP/IP network. The sniffer is an application that does the sniffing process. It is also called as network protocol analyzer. The following are the two modes of operation for a sniffer:

i. The Promiscuous mode: In this mode, the sniffer can steal information from network traffic, which includes any devices connected to the host system (Bradley, 2017).

ii. The Non- Promiscuous mode: In this mode The sniffer can steal only the information going to and from its host system (Bradley, 2017).

Sensitive information such as credentials like IDs and passwords, account data, network specifics, credit card numbers, email texts, file transfers, DNS Queries, chat sessions, online sites visited, and other sensitive information is secretly stolen by the sniffer. Sniffing can lead to dangerous attacks that are difficult to detect (Bradley, 2017).

Sniffing is thus classified as a "passive" attack because the attackers can remain silent or undetectable across the network. These sniffing attacks are vulnerable to protocols that send either passwords or data in plain text, as well as protocols that send both passwords and data in clear text. For example, Telnet, HTTP, SMTP, NNTP, POP, FTP and IMAP are some of the protocols vulnerable to sniffing (Bradley, 2017).

There are three basic ways to sniff a network: -

i. The Wireless sniffer: This is specifically designed to capture data on wireless networks. Also called as wireless packet sniffer or wireless network sniffer (Vimalesvaran, 2016).

ii. The External sniffer: Externally monitoring all inbound and outbound traffic from an external location to a web server by acquiring information about the server is possible with this type of sniffer. In simple terms, sniffing from the third -party external location or sniffing data from the external interface using the sniffer tools (Vimalesvaran, 2016).

iii. Internal sniffer: These sniffers were created to take advantage of the internal collaboration network. The hacker compromises a workstation on the internal network and uses a sniffer to steal data in order to compromise additional devices on the network. In this manner, sniffing refers to "information that can be obtained in a stealthy manner". The techniques for this are as follows:

i. The A LAN sniff: The sniffing software will be deployed on the LAN. The sniffer/attacker checks all of the hosts connected to the LAN for IP addresses. Through this, the information (like open ports, active hosts, server portfolio, etc.) can be stealth. The port specific attacks can be launched with this information (Vimalesvaran, 2016).

ii. A protocol sniff: Information about the network protocols used is sniffed by the attacker. The following stages are taken by the attacker: a large list of protocols is established from the information sniffed, and several types of attacks and sniffers are constructed to carry them out. For example, when the list contains the UDP protocol, then a special UDP sniffer will be initialized to capture and decrypt the details of associated applications like DNS, Telnet and others (Vimalesvaran, 2016).

iii. The ARP sniff: The attacker obtains the set of IP addresses as well as the MAC addresses by sniffing through this resolution procedure. This information will be sufficient to launch router, spoofing, and ARP poisoning attacks (Vimalesvaran, 2016).

Types of network packet sniffing

i. Hardware Packet Sniffing: A hardware packet sniffer is generally designed to be connected into a network and examined in this situation. When trying to see the activity of a specific network segment, a hardware packet sniffer is very useful. A hardware packet sniffer can assure that no packets are lost owing to filtering, routing, or other deliberate or incidental causes by inserting directly into the physical network at the proper location. A hardware packet sniffer either stores the collected packets or forwards them on to a collector that logs the data collected by the hardware packet sniffer for further analysis.

ii. Software Packet Sniffing: Most packet sniffing these days are done by way of specialized software. While any network interface connected to a network can receive all network traffic that passes across it, most are not. This configuration is changed by a software packet sniffer so that the network interface feeds all network traffic up the stack. This configuration is known as promiscuous mode for most network adapters. When a packet sniffer is in promiscuous mode, its functionality consists of isolating, reassembling, and logging any software packets that flow through the interface, independent of their destination addresses. All traffic that goes across the physical network interface is collected by software packet sniffers. That traffic is then logged and used in accordance with the software's packet sniffing requirements.

2.2.2.1 The Proposed System

The proposed system is out to improve the detection of inside threats and investigations (fraud) of activities done by the network users. These systems use screen capture traces to create a log for every activity (files) that is accessed by the user.

In the proposed system, while the software will be keeping track of the activities of the client systems in real time (which can be changed at any time) and keeping the logs in the server system.

The proposed system which is network based system is made up of three modules which includes: -

i. Data and Content Monitoring

The method of network monitoring is similar to intrusion detection in some ways but focuses on network performance instead of network security threats. Performance monitoring can't only detect problems within the network like slow or faulty network equipment, it also can be used to verify service level agreements for tenants. To monitor network latency, special time stamp functions such as those provided in the IEEE 1588 standard are frequently employed. This can also be done using network monitoring equipment by sending specific packets through the network and measuring the round-trip delay. To continuously monitor the network health, connectivity checks messages similar to those used in carrier networks will be issued throughout the data center network. Congestion can also be checked using these same messages, which the orchestration layer can use to reroute data to improve network utilization and functionality.

Certain nodes, however, are configured to ignore this usual practice and gather all or a defined sample of packets, regardless of their destination address in packet sniffing. These packets are used by packet sniffers to analyze a network.

ii. Network Behavior Monitoring

The collecting and analysis of network data to detect malicious activity is known as behavior monitoring. It is also known as network behavior analysis (NBA) or network behavior anomaly detection. Behavioral monitoring software analyzes data from a variety of sources and employs machine learning to spot trends that could indicate an attack is underway.

Behavior monitoring, when carried out over a long period of time, allows organizations like the Nigerian Communications Commission to benchmark usual network behavior, which aids in the detection of anomalies. Any anomalies found can be escalated for further investigation.

iii. Real Time Monitoring

The process of maintaining up-to-date values of a set of indicators, which are some resource properties whose values are used to measure or evaluate the working status of a certain aspect of network functionality. What indicators to observe and how to compute their values are the two questions we have to answer.

Most indicators that are useful for network monitoring fall into two categories: rate-oriented and ratio-oriented. Rate-oriented indicators reflect the varying speed of some underlying network attributes. A ratio-oriented indicator represents the proportional relationship between two quantities, usually in terms of percentage. Because of the inherent statistical nature of these indicators, they are sometimes further processed to generate some corresponding statistics, based on which analysis is finally carried out.

Architectural Design of the Proposed System

This is where the programs that will run the modules identified in the control centre are specified. This will enable the researcher to capture the complete working picture of the application and how each component is related to another.

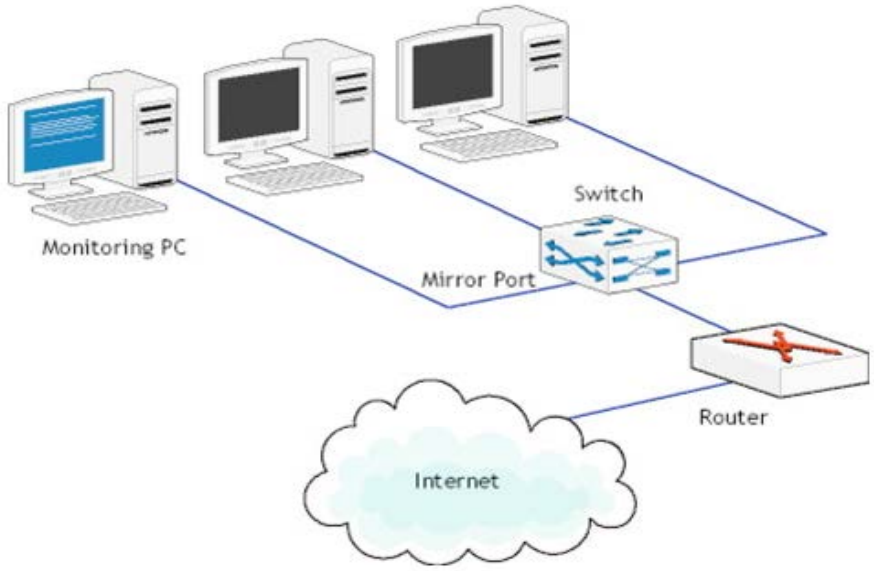

Fig. 3.1 The Typical architecture of an Online Packet Sniffing System

Functional Requirements

The User is just required to register, login and make purchases on the interface while the Admin is responsible for managing users, orders and products.

User End

The user performs the following functions:

i. Login: The user logs in with valid details as stored by the system. If login details are invalid, they would not be allowed to use the system.

ii. View results of analysis carried out by the system

3.3 Methodology Adopted

The research methodology adopted and implemented in the course of executing this research work, is the Object Oriented Analysis and Design Methodology (OOADM).

This is a software engineering method that views a system as a collection of interconnected objects. Each object is defined by its class, state (data components), and behavior, and given the system being studied, reflects some entity of interest. Different models can be made to depict the static structure, dynamic behavior, and run-time deployment of these cooperating elements.

These models can be represented using a variety of notations, such as Unified Modeling Language (UML); Object-oriented analysis (OOA) is an object-modeling technique that is used to examine a system's functional requirements.

Object-oriented Design (OOD) is the process of developing analysis models into implementation specifications. OOA focuses on what the system does, and OOD focuses on how the system does it.

In this journal for the development of online network monitoring and traffic packet analysis using sniffer application, two types of sniffing methods are used.

1. IP Based Sniffing: In this method a requirement of setting network card into promiscuous mode exist. When network card is set into promiscuous mode then host will be able to sniff all packets. A key point in the IP based sniffing is that it uses an IP based filter, and the packets matching the IP address filter is captured only.

Normally the IP address filter is not set so it can capture all the packets. This method only works in non-switched network.

2. MAC based Sniffing: This is like IP based sniffing. Same concept of IP based sniffing is also used here besides using an IP based filter. Here also, requirement of setting network card into promiscuous mode exists. Here, filter is used to sniffing all packets matching the MAC addresses.

System implementation and Results

4.1 Algorithm Used

The algorithm used to create this network sniffing system using java programming language is as follows:

Step 1: Setup a Network Session

Step 2: Define the Client and the Server

Step 4: Make the Client Send and Receive Data

Step 5: Make the Server Send and Receive Data

Step 6: Create a log of the data sent and received

Step 7: Identify packets

Step 8: Conduct analysis

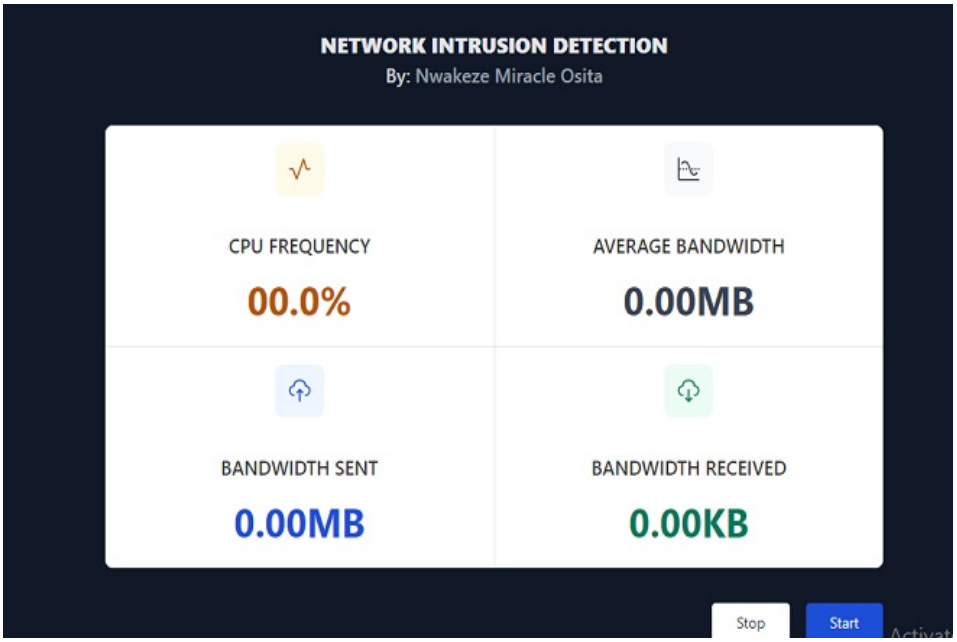

Fig. 4.1

Network Intrusion Detector before activation

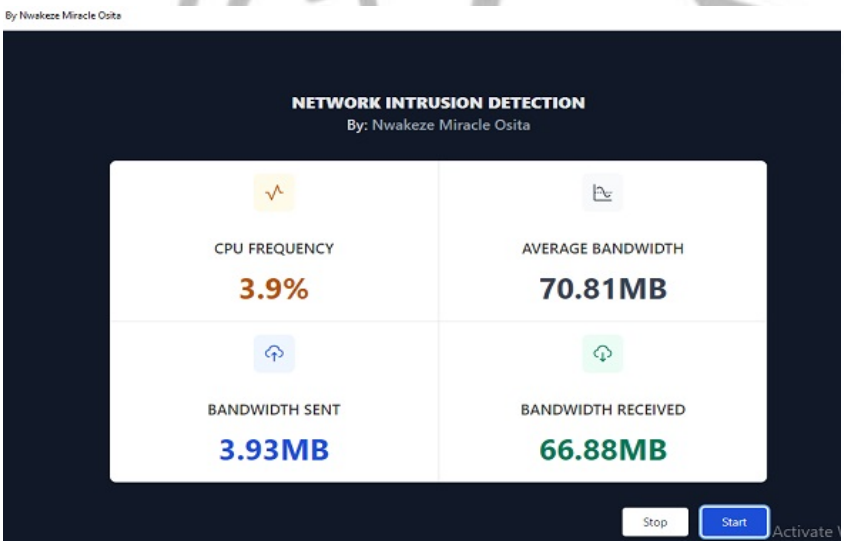

Fig.4.2

Network Intrusion Detector after activation

{"cpu_meter":"22.9","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":1643374741048}{"cpu_meter":"22.9","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":1643374741048}{"cpu_meter":"22.9","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":1643374741048}{"cpu_meter":"24.4","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":1643374741048}{"cpu_meter":"33.1","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":1643374741048}{"cpu_meter":"26.9","average_bandwidth":"681.25MB","bandwidth_received":"36.88MB","bandwidth_sent":"644.37MB","time":164337

Fig.4.3. Log of data sent and received

5.2 Conclusion

Following the completion of the online sniffing application system, behavior monitoring, also known as network behavior analysis (NBA) or network behavior anomaly detection, is the collecting and analysis of network data to detect malicious activities.

Data from a variety of sources was analyzed, and machine learning was used to discover patterns that could indicate an assault was taking place. When conducted over an extended period of time, behaviour monitoring allows organizations such as the academic institutions to benchmark typical network behaviour, which helps to identify deviations. Any anomalies identified can then be escalated for further analysis. In addition, the network monitoring equipment also can accomplish this by sending special packets through the network and measuring the round-trip delay. To keep track of the network's health, connectivity checks messages similar to those used in carrier networks

will be transmitted throughout the data center network. Congestion can also be observed using these messages, which the orchestration layer can employ to reroute data to improve network utilization and performance.

ACKNOWLEDGEMENTS

Firstly, my humble gratitude goes to the Almighty God for His mercies and protection throughout my study in this institution. I am most grateful to Him who makes all things possible.

I would like to express my sincere gratitude to everyone who has helped and supported me not only to complete this dissertation but also throughout my master's programme. This research work is the result of a collective endeavour; I received invaluable assistance and support in various forms from a good number of persons. On this note, I wish to express my heartfelt gratitude to my God given Supervisor, Prof. Okeke Ogochukwu, who painstakingly read through the manuscripts and made corrections that led to the actualization of this work. I cannot thank you enough.

I owe a million thanks to the Head of Department, Prof. Okeke Ogochukwu who devoted his time and energy in moving all PG programme of this great Department to a higher level. In fact, I lack words strong enough to unfold the exact feeling of my heart towards her contribution in this work. She is the secret behind the successful completion of this work. He did everything humanly possible to fast track this programme even against all odds. Thank you ma.

I acknowledge and am greatly indebted to all the authors whose works I have cited in this work. Moreover, my dynamic and unique lecturers: Prof. Ike Mgbemfulike, Mrs. Chekwube Nwakwo, Mrs. Chinwe Ndigwe, Mr. Tochukwu Umeasiegbu, Mr. Peter Ezeanyeji, for their tireless efforts and immense contributions to instill knowledge on my path of academic pursuit.

Most importantly, I thank my beloved wife Dcns. Nwakeze Divine Ogechukwu; for her immense support, contributions and encouragement throughout the period this work lasted. With all humility and love, I remain ever grateful to my wonderful Uncle, Chief Hon. Sir Nnamdi Mekoh for his moral effort and support.

To my good and wonderful friends who were firmly by my side throughout my studies; Ananti Henry, Akabuike Happines, Onuigbo-Chikodili Chinemelu, Joseph Odoh, Uba Chioma. I appreciate your love and support.

May God bless us all.

REFERENCES

Bhandari, A. and Ailawadhi, A. (2017). Literature Review on an Approach to Detect Packets Using Packet Sniffing.Journal of Network Communications and Emerging Technologies [online] Available from <www.jncet.org> [June 2018]

Bhandari, S. Gautam, T. K. Koirala, and M. R. Islam, "Packet Sniffing and Network Traffic Analysis Using TCP—A New Approach," in Advances in Electronics, Communication and Computing, ed: Springer, 2018, pp. 273-280.

Bradley, M. (2017). What is a Network Sniffer? Both Admins and Hackers Can Capture Network Traffic. Retrieved from https://www.lifewire.com/definition-of-sniffer-817996 on 24/10/2017

Garg and M. Dave, "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6

Kong, B. Lyu, F. Chen, and Z. Yang, "The Security Network Coding System With Physical Layer Key Generation in Two-Way Relay Networks," in IEEE Access, vol. 6, pp. 40673-40681, 2018

Paessla, (2018). PRTG Network Monitoring Software. Retrieved from https://www.paessler.com

Rane, (2017). Simple Network Management Protocol. Retrieved from http://www.rane.com

Sudibyo, N. Funabiki, M. Kuribayashi, K. I. Munene, M. M. Islam and W. Kao, "A TCP Fairness Control Method for Two-Host Concurrent Communications in Elastic WLAN System Using

Suricata Open Source IDS / IPS / NSM engine [online]. [cit. 2015-04-21]. URL, http://suricata-ids.org/.

Vimalesvaran, M. (2015). Packet Sniffing: What it's used for, its Vulnerabilities, and How to Uncover Sniffers [online] available from <http://www.cs.tufts.edu/comp/116/archive/fall2015/mvimalesvaran.pdf> [22 July 2018]

Xie, Z. Yan, Z. Yao and M. Atiquzzaman, "Data Collection for Security Measurement in Wireless Sensor Networks: A Survey," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2205-2224, April 2019