IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 1, Ser. III (Jan - Feb 2019), PP 26-34
www.iosrjournals.org

An enhanced Sniffing Tool for Network Management

Bukie, P. T.[1] Oyo-Ita, E.U.[2]Ideba, M. E.[3] Oboyi, J.[4]
Department of Computer Science, University of Calabar
Department of Computer Science, Cross River University of Technology, Calabar
Department of Computer Science, University of Calabar
Department of Mathematics, University of Calabar
Corresponding Author: Bukie, P. T.

Abstract:*The use of packet sniffers is most effective in monitoring and troubleshootingcomputer networksagainst vulnerabilities mostly threaten data security and integrity. However, an initial investigation showed that the efficiency of existing packet sniffing platforms suffered setbacks in the areas of malware detection, platform dependency and user interface. As such, an easy graphical user interface (GUI) and minimal memory requirements were established asthe most essential features required to effectively monitor network activities. This report analyses the limitations of existing packet sniffing tools, using an Object-OrientedModelling (OOM) design methodology; in a bid to develop a platform independent tool with simpler GUI, better performance and security whereby a user can perform different routines built into the packet sniffing system by selecting the File or View menus for sub menus to appear for selection, in less time.*

Date of Submission: 01-02-2019 Date of acceptance:18-02-2019

I. Background

The development of third generation computers introduced supportive capabilities for computer networks and integrated communication components into operating systems. This integration brought about some colossal vulnerability into computer which has grown significantly as operating systems keeps evolving. The innovation, though, has made computers that hitherto used to be safe, vulnerable to different kinds of attacks by cyberpunks (Esin, 2017), has also reduced the entire globe into a unique "village", where people and organisations can share data or information with ease. The innovation has brought "terror" to the globe as the world has been completely cybernated by cyberneticians.

Generally, communication takes different forms: from simple voice conversations to complexlight manipulations. The process is exponentially more complicated in computerized data communication because of the dynamic nature of the type and medium of communication as data is transmitted from one node to another(Tillers and Fish, 1999). When a computer communication is established, numerous vulnerabilities in the accessibility of the communication do exist in different forms: man, in the middle attack, spoofing, phishing, and other cybercrimes. The ability to intercept the communicationdepends on the type and medium of communication adopted. Any communication regardless of the type and medium employed (wired or wireless) can be intercepted based on available resources and environmental conditions at some point in time.

Computer network systems allow multiple computers to be interconnected together via suitable communication media: cables, microwave, etc. (Forouzan, 2007). This allows for sharing of resourcesand peripherals betweeninterconnected systems. Informationtravelson computer networks (local and internet) in the form of packets.Packets in computer communications can be defined as a quantity of data of limited sizetravelling on the network.

Unfortunately, network users may not fully be aware of all the underlying services and packetsrunning on the network because they do not have the right tools for network administration and management. One of the contemporary tools for network management is called "Packet sniffer" or "Network sniffer".A Network sniffer is a tool that can scan the Transmission Control Protocol (TCP) ports to identify which services are currently running on the computer. It can also be used to detectintrusionof unauthorized services such as malware running on the system.Packet sniffing tool are used in network management, monitoring and ethical hacking of information flow across network (Qadeeretal, 2010). Kevin (2003) defined a packet sniffer as a computer program or piece of computer hardware that can intercept and log traffic that passes over a digital network or part of a network. Sniffers function by capturing individual packet and decoding if necessary, the raw data from packets as they flow across networks, showing the values of various fields in the packet, and analysing its content according to the appropriate rules for communication (RFC) or other specifications.

Conversely, sniffers can also be deployed by network attackers as a powerful tool to obtain information from a network communication. Nevertheless, Bradley (2017) affirms that packet snifferscan be used

legitimately as network monitors oranalysers byadministrators to monitor and troubleshoot network traffic.She went further to say that sniffers can be deployed both for good and bad intentions targeted at a computer network.

Network administrators need to intercept the packet traffic on a network to identify underlying problems, detect intrusion and control content transmission for the overall network performance.By analysing various properties of the intercepted communications, an administrator can use the information collected to diagnose network related performance issues such as, poorly configured devices, system errors, and network activities to assist in the determination of network design (Tillers and Fish, 1999).

Packet sniffing can be used for network traffic monitoring, traffic analysis, troubleshooting, capturing username/password credentials, amongst other functions. Practically,Asrodia and Sharma (2013) assert that sniffing a non-switched network is easier as is difficult with a switched network due to the use of switches which narrow traffic. Vimalesvaran (2015) infers accordingly that, since all machines on an un-switched network are attached to the same hub, its network packets are sent to all connected machines, in other words, broadcasted to the entire LAN. However, switched networks tend to be faster and cheaper over time as each network connected machine is attached to a switch which keeps track of the machines using information and sends packets to the specified machine as they arrive. Hence the use of independent sniffer program platforms based on neural networks is recommended as an effective approach towards enhancing packet sniffing.

Most current network sniffer User Interface does not meet the standard of contemporary system user interface. Aside the user-friendly interface which makes some current Network sniffer difficult to use, most current Network sniffers are not cross platform and as such they are integrated to the kernel of operating system, which could pose a serious risk to the computer system. Furthermore, most current Network sniffers are poor in term of malware detection activity.

Wide Area Networks (WAN) have consistently grown in size, complexity and along withit the number ofusers from government parastatals, corporate organizations and individuals.Hence, network traffic flowing at each node has increased drastically. One thing worthy of note is that, some connections are aimed at hacking and defrauding other people connected to the network as well. This entirely compromises the pivot objective of computer networks which is to share resources and information. Furthermore, on simple LANs, some customers perform malicious third-party activities while others connect to networks and download very large files-an action that is capable of reducing network transmission speed, its overall reliability and performance. Capturing packets transfer with promiscuous mode will not necessarily be sufficient to see all traffic on the network.

According to Stallings (2000), different categories of attacks can target the flow of packets from source to destination. They include: interruptions, interceptions, modifications, and fabrications.

II. Literature Review

Magers (2002) identified the following sniffer components: Hardware, Drive program, Buffer and Packet analyser. The hardware is required when working with snifferat times for analysing hardware related problems like voltage fluctuation and poor cabling. The main component of sniffer is the drive program which captures traffic in network and filters it to restrict data. The buffer on the other hand is a storage device where captured data from network is stored. While Packet analysis can be done on real time or stored in buffer, retrieve later for analysis. It is possible when data is stored in buffer for analysis to be done both on the header and actual data, when we store data in memory or we perform real time analysis, decoder is used to decode the data store in packets.

Bhandari and Ailawadhi (2017) categorisedsniffing techniques into client side: where web page of sniffer uses java, script interpreted by user agent to web servers; server side where sniffer attacks from server side using http communication protocol; browser sniffing where websites and applications are used to misinterpret html, etc. And aid malicious users steal private data; Content Sniffing where alteration is made in the stream of bytes which changes the format of the file to malicious contents; Password sniffing aims to crack passwords and login information saved in data packets.

Nevertheless, network packet sniffers can access and inspect data packages flowing through network interface of a machine running a network sniffing software. For instance, the PRTG sniffing software uses a switch that offers a monitoring port – also called port mirroring configuration. This switch sends a copy to the monitoring port of all data packages travelling through the switch and analyses traffic entirely as it travels through the switch.

Similarly, study reports from Frieden (2008) show that network service providers are willing to pay for packet sniffing enhancements that provide more protection, privacy and reliability. In a bid to accommodate an increasing demand for higher bandwidths and less tolerance for dropped, delayed or lost packet delivery, equipment that can examine and prioritise internet traffic at an increasingly granular level and inspect network traffic on a packet-by-packet basis are being developed. As such, network service providers now diversify service on the basis of allocated bandwidth, routing priority and performance guarantees.Packet sniffing in this

case also entails actively examining packet headers that provide traffic routing information and identify content characteristics contained thereof in the packet sniffed.

There are a lot of works done in packet sniffing for LAN and WAN to monitor network traffic as well as its user's activities to keep the network smooth and efficient. Packet sniffers are of two types: Active and Passive (Menga and Timm, 2006). Passive packet sniffers do not respond back, i.e. they only collect data and are impossible to detect them (Gandhi *et al*, 2014) and are useful in areas such as telecommunication, Radar systems, medical equipment, etc. Active packet sniffers can send the data in the network and hence could be detected by other systems through different techniques (Rupam *et al*, 2013). For example, active packet sniffer can fake replies tothe broadcast or can forward it to a legitimate host. Top passive packet sniffers in used are: Tcpdump, Wireshark, ColasoftCapsa etc.(Gandhi *et al*, 2014).

The rapid increase in the deployment of computer networks by organisations and individuals has increased traffic flow and created more demand for network administrator; thereby creating more need for packet sniffers to cope with attacks by cyberpunks. To effectively evaluate an attack against system security and choose appropriate mitigation mechanisms, specific security policy defining properties are essential (Kruegel *et al*, 2005). When packets are transfer from source to destination, the packets pass through many intermediate devices. A node whose NIC is set in the promiscuous mode receives all information travelling on network (Ansari, Rajeev and Chandrasekhar, 2002). Each NIC has a unique physical address and network. When packets arrive at NIC, hardware address of frames match with the physical address of NICs, but if set in promiscuous mode then all packets will arrive at that interface. When switches which already pass filtered data are used, then some methods to capture all data of network are performed. When NIC accept packets, packets are copied to driver memory then it passes to kernel and kernel passes it to user application.

Von, Wulf, Schröder, & Wolf, (2016)presented a real time capable UWB sniffer. Their main design objectives were easy setup and usage, and packet timestamps that are as precise as possible. To accomplish these goals, they employed the generally famous Wireshark and the hardware timestamping unit of the DW1000 which offers a resolution of 15 ps. Their UWB Sniffer comprises of three major components. First the DW1000 UWB transceiver which is linked to a STM32 ARM based microcontroller via a SPI interface. The microcontroller is connected to the PC through a USB link. On the PC they developed custom application software that is capable of reading the captured packets from the USB and forwarding same to Wireshark. This Software is also used as a configuration interface for the DW1000.

Talekar, Tidake, & Shinde, (2011) designed a network sniffer with data mining techniques. Their network sniffer differs from the previous existing IDS because it automatically detects the type of the user such as normal user, spy, unauthorized user, and intruder from the network traffic graph. The system can be used as a utility for anti-hacking, Mobile Agent, LAN monitoring as well as for controlling the entire network.

Portoles-Comeras, Requena-Esteso, Mangues-Bafalluy, & Cardenete-Suriol, (2006) presented a performance assessment of a common tool used in wireless networking testing: off-the-shelf wireless sniffers. In their work, they first, analysed the performance of marketable devices in terms of packet generation rates, illuminating the sundry behaviours that can be detected conditional on the hardware deployed for communications. This first step serves, then, to analyse the fruition of wireless sniffers adopting the packet capture proportion or estimate as a performance parameter Secondly, they flaunt that sniffers can percolate full-rate capture when decorously calibrated. However, experimental observations show that Prism-based sniffers begin losing packets when they have to capture them at rates above a certain limit. They called this development *saturation loss*. They further demonstrated how, with the adoption of a correlation factor between traces acquired using autonomous sniffers, one can achieve a degree of excellence of the traces and the performance of respective sniffer. Lastly, they demonstrated how innumerable sniffers can work synchronously in the same device, auditing similar or diverse channels. However, in case of using Prism-based devices, they posited that the performance of such architecture is circumscribed or restrained at certain packet capture degrees.

Min, Kim, & Ping, (2012)developed a novel and robust multi-channel traffic management scheme in 2.4 GHz wireless automation networks architecture for a local automation system. Their system incorporates a channel resource optimization scheme using a centralized channel assignment algorithm and an intrusion discovery scheme using traffic extrapolation algorithms. To evaluate the performance of their system, they used the wireless data measured from an actual wireless test networks. A simulation result of their system demonstrated that the scheme is capable of managing the multi-channel status of the networks and present the analysis result to network administrator for the network resource configuration of the system. In their system, the security manager could detect the intrusion attack to improve the whole performance of the system, and the intrusion detection ratio is acceptable to enhance the security efficiently.

Cote, Wang, Zeng, & Shi, (2010) in their article, experimentally reconnoitred the proficiency and dependability of mote-class sniffers for sensor network watching. They computed the maintainable workload experimentally and guessed the buffer overflow likelihood using a waiting line model. In furtherance, they compared per-hop loss measurements by the sniffers and the receivers. Lastly, they compared fine-grain timing measurements from the sniffers to those from a logic analyser to evaluate the accuracy of delay measurements. In their result, they discovered that a sniffer can observe traffic at the rate of 60 packets / sec with diminutive buffer overspill. They also demonstrated that per-hop loss measurements from sniffers unveil disparities but are comparable to those at the receiver and that per-hop delay measurements from a sniffer are correct (with errors up to 300 μs). Their results showed that measurement quality by mote-class sniffers is suitable for many watching devotions.

As a limitation, their work did not compare their model to standard models like MicaZ, and Imote2. They also did not equate their model in terms of the loss and delay characteristics as well as capability and fidelity of diverse mote platforms. They did not also conduct experiments over all-encompassing networks but used simulated circumstantial traffic. Finally, they did not conduct akin experiments in union with electromagnetic interference from other shopper or engineering devices.

Appraisal of some Classic Packet Sniffer Software

Case 1: Tcpdump

Tcpdump isone of the oldest and most commonly used packetanalyser that uses command line statement and works only on the Linux based systems. It allows the user to capture and display TCP/IP and other packets being transmitted or received over a network. The modified version of tcpdump called windump is a free and open source windows base version of Tcpdump used for packet capturing, network analysis and protocol debugging in windows environment.TCPdumppackets can beaccessed remotely and this gives it an advantage overother packet sniffers. Hence, it is the most preferred analysis tool with small overhead cost for Network administrators intending to work from a different network (McCanne and Jacobson, 1992). Although TcpDump has advantage in terms of memory utilization as its installation filesize is just 484 KB, it has poor or noGUI as users may be required to study and get acquainted with the command prompt to make efficient use of the tool.

Tcpdump provides standard interface to all common (UNIX-based) operating systems such as Linux and FreeBSD as well as windows platform. It is also memory efficient, but has technological limitation in terms of GUI, platform dependence and malware detection.

Case 2: Wireshark

Similar to TCPdump, Wireshark is a free and open- source packet analyser with Graphical User Interface in addition to sorting and filtering features.This tool allows for configuration ofnetwork interface card in promiscuous modeto make visible both packets on that interface configured addresses and broadcast/multicast traffic. VoIP can also be captured and even played if only it is properlydecoded.It supportsbothLinux-based, windows and Mac operating systems. Wireshark provides option for network layer filtering with supports for geo-localisation of packets (Borja, 2011).

Wireshark has a user-friendly GUI, displaying the information inside packets in a meaningful manner, but its size of installation file is about 18 MB and has a running memory of 81 MB in Windows and aslarge as 449 MB in Linux. This make it more expensive when compared to TCPdumpin terms of memory requirements, and resource utilization. However, neither TCPdump nor Wireshark has intrusion detection function. They cannot generate alarms for attacks or hints when a passive attack or anything strange happens in the network.

Case 3: ColasoftCapsa

ColasoftCapsasupports most of the featuresof Wireshark but with powerful TCP flow analysis andits easier interpretation. It has versatile networktraffic, bandwidth and utilization analysis. It has in-depth packet decoding feature with multiplenetwork behaviour monitoring. It has a matrixrepresentation and eclipse visualisation of thenetwork. ColasoftCapsa extends the network diagnosis by detectingand locating suspicious hosts that may cause problemand alerts computer against network anomalies. One of the demerits of ColasoftCapsa is that it is quite expensive because it is a proprietary packet sniffer.

Case 4: Ethereal

This is an open-source and cross-platform network sniffer that supports quite a large number of network protocols and provides core and rudimentary statistical information. It was initially developed for Unix/Linux applications based on libPcap (Clincy and Abu-Halaweh, 2005). However, Ethereal is currently available for Windows platforms. Because of its "fine" and detailed documentation and superlative GUI, it is

easy to understand and use.The main function for which it was designed was to capture and analyse traffic passing through a network segment as well as study the architectures of all the protocols captured.

Furthermore, it has the capability of displaying the service response times, endpoints list, conversation lists, and other protocol statistics. Ethereal can show chats that occur among two hosts, group packets that are related to the same TCP connection, and display protocol spectrum spread (Clincy and Abu-Halaweh, 2005).

Ethereal comprises of three main panes. The topmost pane which displays a summary of captured packets; the middle pane, which displays the details of a selected captured packet and the third pane, which displays the data from the packet selected in the packet list pane and highlights the field selected in the packet details pane.

One major flaw of Ethereal is that it does not provide graphical representation of statistical data. Ethereal is not intended for use in intrusion detection systems. It cannot recommend network fixes. Furthermore, because it is open-source software, it lacks the required technical support for application of its kind.

In Network Sniffers such as tcpdump, packets that are blocked by network gateway firewalls may not be seen by the packet sniffer. This is quite a big flaw, considering the standard of developing sniffers. In packet sniffers like etherape, due to the fact that it requires root privileges to run, there could be risk to the machine running when connected to the internet.

Network Sniffer such as Snoop is disadvantaged because it is tightly integrated within the Solaris kernel; its use is largely limited to Sun-based systems. Though there have been hacks to port it to kernels like FreeBSD and Linux, most of them have been fairly limited by number.

III. Materials and Methods

The study adopted Object Oriented Modelling (OOM) design methodology. TheOOM design method is employed to bring out detailed description of the system as well as providing avenue for easy modification of the system using flowchart, Context Diagram, Data Flow Diagram to describe the system and gives out a clearer view of the modules, procedures, functions and sub-systems alongsidetheir respective relationships and the representation of the objects (data and processes) of the Network Sniffer.

Figure 1 shows the OOP model class diagram of the enhance proposed Network Sniffers

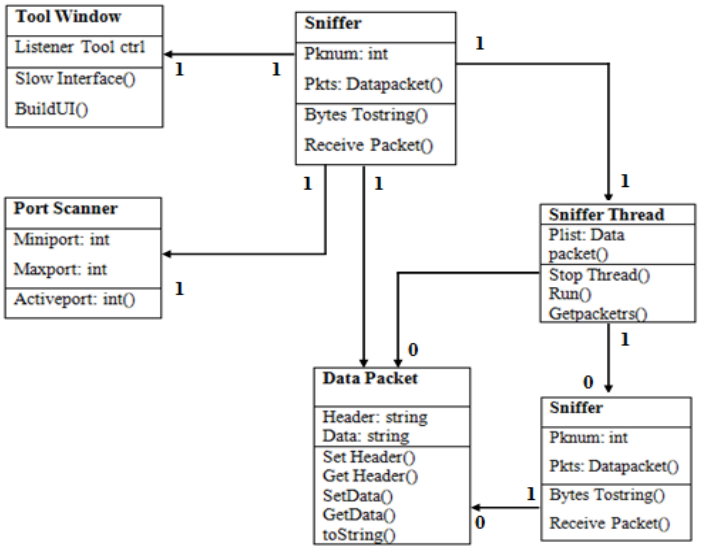

Fig 1: Class Diagram

As represented in figure 1,the designedsystem consists of six classes namely: Sniffer, window, Tool Control, Data packet, Sniffer thread and Port scanner class. Each of these performs different function in the system. While the Sniffer Classes the data capturing module and contains procedures that implement the packet capturing operations, the window class is used for GUI design of the application whichdisplay available menu options for user selection. On the other hand, the Tool Control Class is the application's main module responsible for handling the processing of the application's inputs, invocation of the relevant, application modules and formatting of the program output. The Data packet class is used to model the network data packet. The class defines the structures of a packetand providemember functions with the responsibility of handling a free list for objects of this format. The storing of information related to a single data packet on a network is also performed by this class. Another class in OOP model design is the Sniffer thread class. This class models a computer process that encapsulates the sniffer module thus enabling the data capture operation to run in the

background without blocking the user's interaction with the system. Finally, Port scanner class as the name implies is the module with routines for implementation of the port scanning operation.

The system is compartmentalized into five (5) different and independent modules which take care of different tasks efficiently. The compartments are as follows: User Interface Module, Packet Sniffing Module, Analyse layers Module, Free Memory Module, and Protocol Analysis Module.

User Interface Module

Actually, every application has one user interface for accessing the entire application. The user interface for the proposed system 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. The module was developed using HTML, CSS and PHP.

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.

Analyse Layers Module

This module contains the code for analysing the layers in the system. Mostly in this module we have to discuss about three layers Transport layer, 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.

Free Memory Module

This module analyses computer memory usage at the time of packet capturing. It can show the memory size in number format as well as graphical representation.

Protocol Analysis Module

This module analyses 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.

The sophisticated features of this system can handle both packet analysis, malware detection and many other features. These features enable:

i. Administrators to show statistics of received packets
ii. Administrators detect malicious IP addresses according to its number of ARP requests in previously specified time
iii. Administrators to view all network interfaces and enable them to capture data from that interface and consequently save captured packets.
iv. Administrators generate reports that aid effective and efficient decision making.

Architecture of the New System

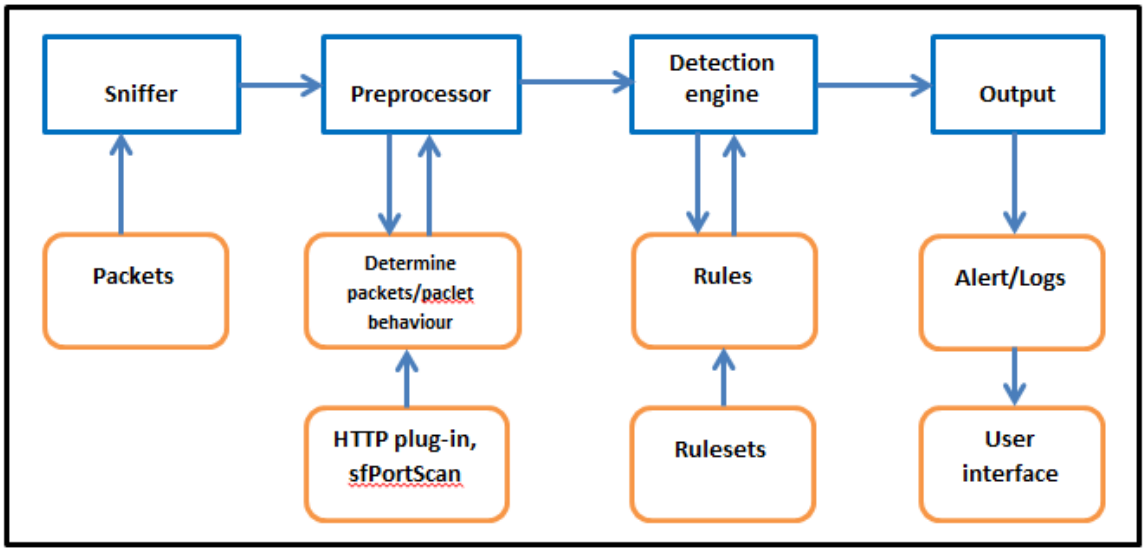

Fig 2: Architectural design of the Proposed System

Figure 2 is the architectural design. The diagram depicts the working of the system and the integration of relevant components. It consists of four fundamental components: sniffer, pre-processor, detection engine and output. The activities of the network sniffer are in sequence starting from sniffing, pre-processors, detection engine and finally output. That means, when the software sniffs the network packets, it forwards the packets to the pre-processor which is responsible for preparing and processing the packets for the detection engine to detect the actual packet. Upon detection of the packet, the software gives out output.

IV. Results and Discussion

Thedesigned network sniffer is platform independent i.e., it eliminates the need for system dependent packet capture modules in each application; providing detailed description of a source address, destination address and the type of traffic that has been detected. Also it has both Graphical User Interface (GUI) and Non Graphical User Interface and can understands and work in the structure of different network protocol.

Figure 5 is the application main menu with just two menus: File and View menu. The user can perform different routines built into the system by selecting the File or View menus for sub menus to appear for further selections. The Application Window also have three shortcut icons: Start, Stop and Clear as shown in figure 4.

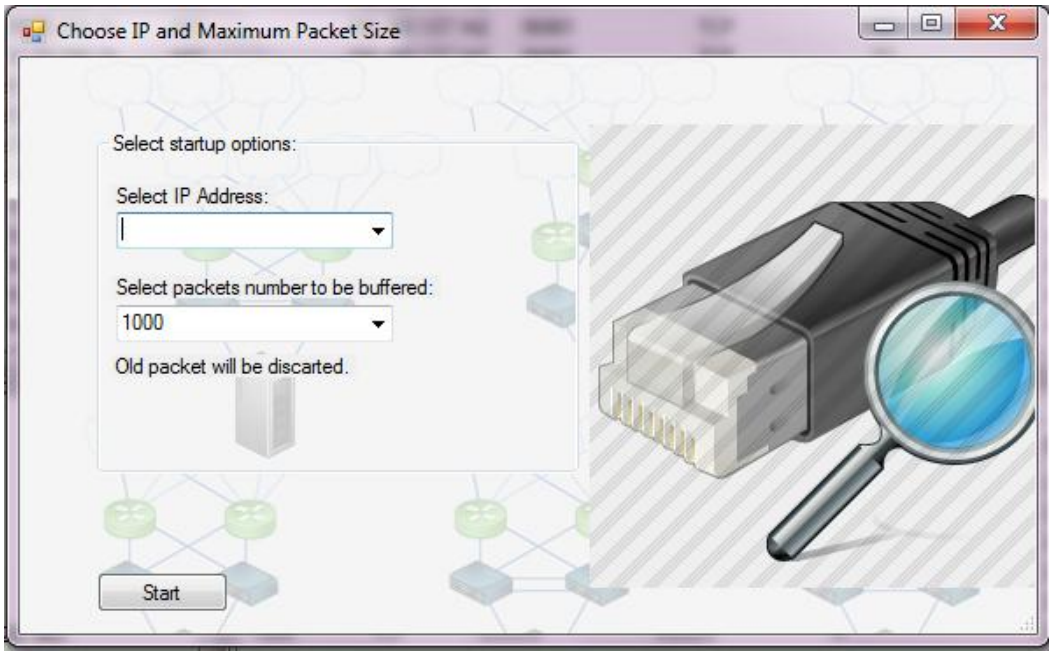

Figure 3: IP / Packet size Selection Window

The input format for the system consists of a combo box where users can select the IP address and the Packet number to be buffered. It also has a button called the Start where users can issue command. This is shown in figure3.

Details of the captured packets showing the Source Mac and IP addresses, Destination Mac and IP addresses and methods of system on the network at the time it was sniffed and is represented in pie chart as shown in Figure 5. The output of the proposed system is displayed inthe general packet analysis window shown in figure 5. It shows the full packet analysis as well as the IP, protocols and open ports analysis. It also display Packet No, Packet time, Packet source, Packet destination, Packet source port, Packet destination port, Packet protocol, packet size, system name and system.

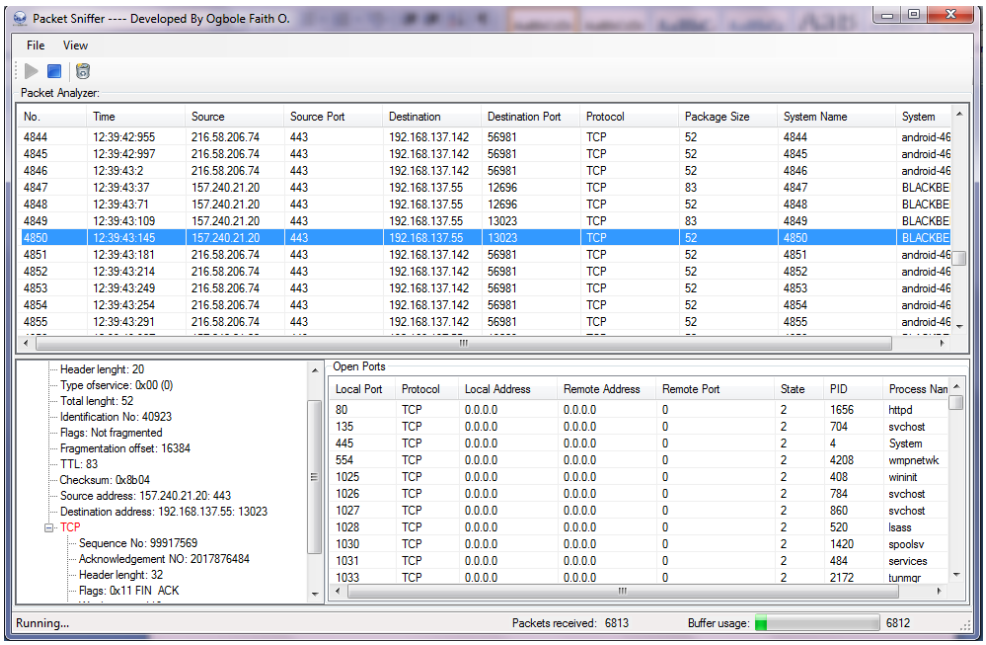

Figure 4: Packet Analysis Window

Figure 4 window is divided into two sections: the IP and Protocol sections. This IP window shows the protocol version, header length, type of service, total length, identification number, fragmented flags, not fragmented offset TTL, source address, destination address. The Protocol section displays the Sequence No., Acknowledgement No., Header length, Flags, Window size, checksum and message length. This description is for TCP. For UDP, the protocol section displays length and checksum only. Details of the captured packets of the protocol used on the transport layer at the time the network was sniffed is represented in a pie chart in Figure 5. The Pie chart shows percentage, total packets and size of packets used on the transport layer.

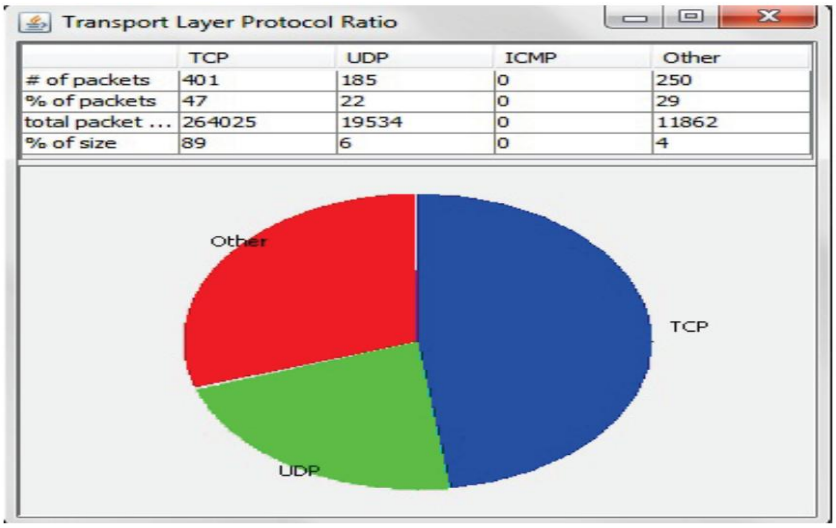

Figure5: 3D pie chart showing received packet characteristics on transport layer

V. Conclusion

The result from this research work shows that the develop network sniffer have the capacity to reduce time wasted on manually monitoring of network connected node, as well minimized overhead of packet sniffing to enable network administrators analyse the network and evaluation of any network for better performance and security. Data sniff from the network was used for analysis. Real time data was collected from a life network environment and the pattern of the traffic was analysed based on used system resourceslike memory and processor which was less compared to other existing system. The packet loss was also less.

References

[1]. Ansari, S., Rajeev, S., and Chandrashekar, H. (2002). Packet Sniffing: A BriefIntroduction. ***IEEE Potentials***, 21(5), 17-19.

[2]. Asrodia, P., Sharma, V. (2013) Network Monitoring and Analysis by PacketSniffing Method.***International Journal of Engineering Trends and Technology,(4)5.*** [Online] Available from <http://www.ijettjournal.org/volume-4/issue-5/IJETT-V4I5P160.pdf> [July 2018]

[3]. 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]

[4]. Borja, F. (2011) Traffic analysis with Wireshark [online] available from <https://www.csirtcv.gva.es/sites/all/files/downloads/cert_trafficwireshark.pdf > [August, 2018]

[5]. 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

[6]. Esin, J. O. (2017). System overview of a cyber-technology in a digitally connected global society. Bloomington, USA: AuthorHouse. P.67.

[7]. Magers, D. (2002). "Packet Sniffing: An Integral Part of Network Defense",May 09, 2002 SANS Institute 2000 – 2002.

[8]. Forouzan, B.A. (2007). Data Communications and Networking. (4th Edition), McGraw Hill Higher Education, New York.

[9]. Gandhi, C., Suri, G., Golyan, R., Saxena, P., Saxena, B. (2014) Packet Sniffer – A Comparative Study.***International Journal of Computer Networks and Communications Security, (2)***5, 179–187 [online] available from < www.ijcncs.org > [May 2018]

[10]. McCanne, S., & Jacobson, V. (1992).The BSD packet filter: A new architecture for user-level packet capture [online] available from <https://www.tcpdump.org/papers/bpf-usenix93.pdf> [July, 2018]

[11]. Kevin, W. (2002) A Layered Model for Internet Policy, 1 J. TELECOMM. & HIGH TECH. L. 37

[12]. Kevin J. C. (2003). Law of Internet Security and Privacy. Aspen Publishers: California

[13]. Kruegel, C., Valeur, F., Vigna, G. (2005). Intrusion Detection and Correlation Challenges and Solutions London:Springer

[14]. Rupam, Verma, A. And Singh, A. (2013) An Approach to Detect Packets Using PacketSniffing ***'International Journal of Computer Science & Engineering Survey (IJCSES)*** **(4)3** [online] available from <http://airccse.org/journal/ijcses/papers/4313ijcses02.pdf >[June 2018]

[15]. 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]

[16]. Clincy, V. A. and Abu-Halaweh, N. (2005). A Taxonomy of free Network Sniffers for teaching and research. ***Journal of Computing Sciences in Colleges,***(21)1, 64-75.

[17]. Talekar, S. A., Tidake, V. S., & Shinde, P. V. (2011). Intelligent network sniffer. ***Proceedings of the International Conference & Workshop on Emerging Trends in Technology-ICWET 2011. 258-261***. doi:10.1145/1980022.1980081. Retrieved from sci-hub.tw/10.1145/1980022.1980081on 16th December, 2018

[18]. Portoles-Comeras, M., Requena-Esteso, M., Mangues-Bafalluy, J., & Cardenete-Suriol, M. (2006). Monitoring wireless networks: performance assessment of sniffer architectures. ***2006 IEEE International Conference on Communications.*** doi:10.1109/icc.2006.254780. Retrieved from sci-hub.tw/10.1109/ICC.2006.254780 on 16th December, 2018

[19]. Menga, J. and Timm, C. (2006) CCSP: Secure Intrusion Detection and SAFE Implementation Study Guide: Exams 642-531 and 642-541 John Wiley & Sons: London

[20]. Min, W., Kim, K., & Ping, W. (2012). A traffic management scheme using multi-channel sniffer for secure wireless networks. ***Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication-ICUIMC 2012.*** doi:10.1145/2184751.2184768. Retrieved from sci-hub.tw/10.1145/2184751.2184768 on 16th January, 2018.

[21]. Cote, J., Wang, B., Zeng, W., & Shi, Z. (2010). Capability and Fidelity of Mote-Class Wireless Sniffers. ***2010 IEEE Global Telecommunications Conference GLOBECOM.***doi:10.1109/glocom.2010.5683714. Retrieved from sci-hub.tw/10.1109/GLOCOM.2010.5683714 on 16th January, 2019

[22]. Tillers, J. S. and Fish, B. D. (1999). Packet Sniffers and Network Monitors. Harold, F. Tipton and Micki Krause (eds.). ***Information Security Management Handbook, 4th (ed.) Vol. 2.*** USA: Auerbach Publications, p117-145.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Bukie, P. T. "An enhanced Sniffing Tool for Network Management" IOSR Journal of Computer Engineering (IOSR-JCE) 21.1 (2019): 26-34.