IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 10, Issue 2, April - May 2022
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Java, there is a Packet Sniffer (PSniffer) application for network security.

Dr. Suman Singh[1],Prashant K Aryan[2],Rajarshi Sanyal[3],Avijit Bose[4]

[1]Associate Professor Department Of Information Technology Monad University,Hapur, Uttar Pradesh

[2,3,4] Assistant Professor,Department Of Information Technology Monad University,Hapur, Uttar Pradesh

Abstract

This paper presents the full implementation of the Psniffer application software that captures network data as well as provides sufficient means for the decision making process of an adminis- trator. The aim of this application is to rewrite C language sniffer into Java, and also develop an application that consumes little memory on the hard disk. This work illustrates the requirement needed to design a new application; it was developed in Java and it consumes little memory on the hard disk. This application comprises of five independent modules that handles different tasks efficiently. This program can monitor network traffic, analyzes traffic patterns, identify and trou- bleshoot network problems. This application does not transmit any data onto the network, uses 1MB of the hard disk space, friendly GUI and it is very easy to install.

Keywords: network traffic, packets, packet capture, packet analyzer/sniffer.

1. Introduction

Bundles in PC correspondences can be characterized as an amount of information of restricted size. In-ternet all traffic goes as parcels, the whole record downloads, Web page recoveries, email, this large number of Internet correspondences generally happen as bundles. In the web, bundle is an organized unit of information conveyed by a parcel mode in PC organization

This work foster a PSniffer (Awodele and Otusile, 2012) network analyzer that can catch net-work traffic and examine it and permits client to accept just the element on a case by case basis with little memory use for establishment and store it in a document to utilize it later in his work, then this will decrease the memory that is utilized to store the information dissimilar to accessible devices can catch network traffic with-out examination, while some require huge memory size for establishment. Furthermore, have an easy to use control interface (Awodele and Otusile, 2012).

Network sniffing is an organization layer assault comprising of catching bundles from the organization sent by different PCs and perusing the information content looking for touchy data like passwords, meeting tokens and private data. This should be possible utilizing instruments called network sniffers; these devices gather bundles on the organization and, contingent upon the nature of thetool, break down the gathered information like supportive of Consent to make advanced or paper duplicate of part or these works for individual or study hall use is conceded without charge

A bundle analyzer in some cases called an organization analyzer, convention analyzer or sniffer or Ethernet sniffer or remote sniffer (Spangler, 2003), is a PC program or a piece of PC hard-product that can catch and log traffic disregarding part of an organization (Chan, 2002)

Sniffer is utilized as a partner of organization the board in view of its checking and dissecting highlights which can assist with investigating organization, recognize interruption, control traffic or administer network contents.

2. Why the Use of A Network Sniffer

The data going through networks is an important wellspring of proof for network adminis-trators to fish out gatecrashers or bizarre associations. The need to catch this data has lead to the improvement of parcel sniffers.

Various examination works exist in the improvement of bundle sniffers. Nonetheless, the quest for the ideal parcel sniffer proceeds. Psniffer will accompany extra functionalities, for example, 3D pie outlines, a GUI and with little memory necessities.

Psniffer when introduced in an organization will assist with observing organization traffic and keeps log of all con- nections to the organization, which is then broke down for the location of dubious exercises.

Packet Sniffer Tools

A few instruments exist that can screen network traffic, generally such devices will put the organization card of a PC into unbridled mode, this empowers the PC to pay attention to the whole traffic on that segment of the organization. Sifting of this parcels should be possible in view of the IP related header information present in the bundles, generally such separating determines basic standards for the IP locations and ports present in the bundles. These uninvolved organization sniffing programs have been created for either wired or remote organization estimation; the most popular are tcpdump and Wireshark.Tcpdump by McCanne, Leres and Jacobson

It is quite possibly the most famous parcel sniffer. Tcpdump is joined by the libpcap library. It was initially written in 1987 at the Lawrence Berkeley National Laboratory and distributed a couple of years after the fact and immediately acquired clients consideration.

Libpcap is a C library for catching parcels. The methodology remembered for libpcap give a stand-ardized point of interaction to all normal (UNIX-based) working frameworks, including Linux and FreeBSD. The connection point of the libpcap is usable considerably under Windows however there the library is called winpcap.

Tcpdump is a typical parcel analyzer that runs under the order line and parsing device ported to a few stages. It permits the client to block and show TCP/IP and different bundles being sent or gotten over an organization to which the PC is joined. Tcpdump works by catching and showing bundle headers and matching them against a bunch of rules.

It runs on most UNIX-like working frameworks - for example Linux, BSD, Solaris, Mac OS X, HP-UX and AIX among others utilizing the libpcap library to catch bundles.

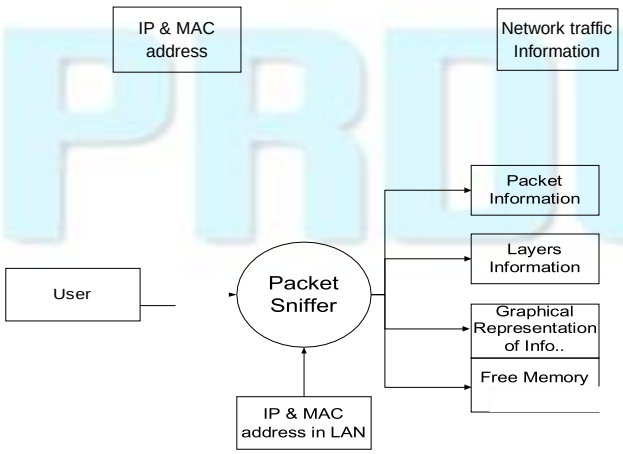
Wireshark by Gerald Combs

Wireshark is a free and open- source packet analyzer and it is written in C. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named **Ethereal**, in May 2006 the project was renamed Wireshark due to trademark

with a packet analyzer in promiscuous mode on a port on a network switch, not all of the traffic traveling through the switch will necessarily be sent to the port on which the capture is being done, so capturing in promiscuous mode will not necessarily be sufficient to see all traffic on the network. Port mirroring or various network taps extend capture to any point on net; simple pas- sive taps are extremely resistant to malware tampering.

Dataflow Diagrams (DFDs)

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 a DFD's are as follows:



Level-1 DFD

Figure 1: Data flow diagram of the psniffer

3. Use Case Diagram

In computer programming, a utilization case outline is a sort of social chart characterized by and creat-ed from a Use-case examination. Its motivation is to introduce a graphical outline of the usefulness given by a framework regarding entertainers, their objectives (addressed as use cases), and any dependen-cies between those utilization cases.

The fundamental reason for a utilization case graph is to show what framework capacities are performed for which entertainer. Jobs of the entertainers in the framework can be portrayed in figure 2 underneath:

Use case diagram of the Psniffer

Component Diagram

A component diagram depicts how components are wired together to form larger software sys- tems. Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component.

An assembly connector is a "connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port."

Deployment Diagram:

A deployment diagram serves to model the physical deployment of artifacts on deployment tar- gets. It shows "the allocation of Artifacts to Nodes according to the Deployments defined be- tween 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 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 cluster-ing.

Deployment diagram of the Psniffer

Architecture of the Proposed System

The design of the proposed system discusses the various requirements that will make up the ap- plication. By conducting the requirements analysis we listed out the requirements that are usefulto restate the problem definition.

- Analyze network Layer.
- Analyze Transport Layer.
- Analyze Application Layer.
- Analyze UDP Protocol
- Analyze TCP Protocol
- Analyze HTTP Protocol
- Analyze Free Memory Size
- Show Line-graph Representation.
- Show Pie chart Representation.
- Find out the Packets over network.

The Features of PSniffer

Psniffer is a customized software application that has a number of features. These features enable:

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

The Psniffer is developed in Java™. This application is designed into five (5) independent mod- ules which take care of different tasks efficiently.

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

User Interface Module

Actually every application has one user interface for accessing the entire application. The user interface for the Psniffer 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.

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 goes 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.
- A host running a sniffer sets its NIC in promiscuous mode. If any host's NIC is running in promiscuous mode, it will receive all packets either those packets targeted to it or not. This way of sniffing is effective in an environment which is broadcast in nature like hub,access point and bus Local Area Network (LAN) environments.
- ARP cache poisoning is also used for sniffing. This way of sniffing is effective in an environment, which is not broadcast in nature. Address Resolution Protocol (ARP) cache poisoning depends on local ARP cache maintained by each host of network. This cache contains IP with corresponding Media Access Control (MAC) addresses of recently ac- cessed hosts.
- makes sense of ARP store harming process that would be utilized for execution. In this graph, 'C' have performs ARP reserve harming assault. 'C' have sends an ARP poison parcel to target have 'A' which contains have 'C' MAC address in source MAC address field and host 'B' IP address in source IP address field of ARP poison bundle. At the point when target have 'A' gets this pack-et, it harms neighborhood ARP reserve esteem either by adding bogus passage or refreshing old section with new one. Same interaction is rehashed with have 'B'. This cycle defiles the neighborhood ARP reserves of host 'A' and 'B' which are displayed in Figure 5. After the culmination of harming process, the two hosts can't discuss straightforwardly with one another. Each host sends a bundle to sniffer host and sniffer have reroutes parcel back to genuine objective. Sniffer have should have IP parcel steering empowered so it could send bundle back to real objective subsequent to getting secret data.
-
- Benefits of Psniffer

Psniffer has many benefits over the existing models. Listed below are the benefits.

- It captures the live packet information in promiscuous and non-promiscuous mode.
- It shows all the network interfaces and enables to capture data from that interface.
- It also shows the statistics of the received packets.
- It can save the captured packets.
- It can retrieve the contents of the previously saved packet capture (Pcap) file.
- It can show the TCP flow graph generated from the received TCP packets.

Interface of PSniffer

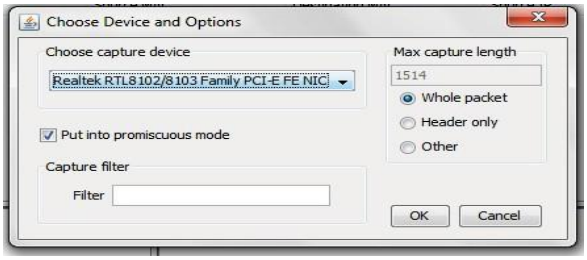


Figure 2: The main GUI of PSniffer

Figure 2 shows the user interface of the sniffer, where the device captured model will be deter- mined either WAN or

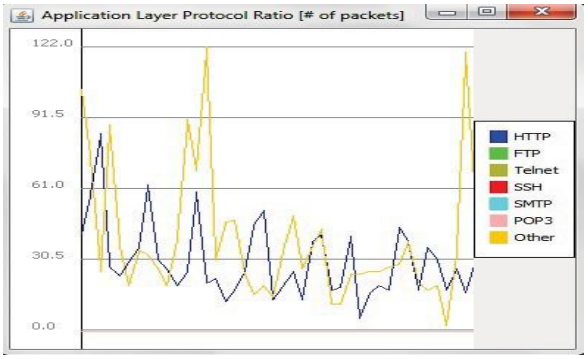


Figure 3: Graph showing the ratio of used supported application

The figure 3 shows the graph of the protocols (http, ftp, telnet) used on the application layer at the time it was sniffed.

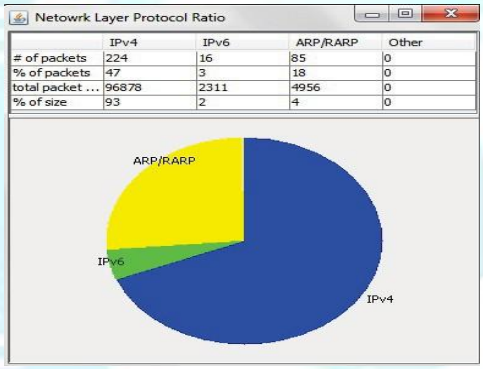


Figure 4: 3D pie chart showing received packet characteristics on Network Layer

The figure 4 show the Pie Chart (percentage, total packets) of the internet protocol type used on the network at the time the network was sniffed.

6. Conclusion

Contrasted with comparative works this application shows the layer associated with sniffing and the proto-cols. This Passive Sniffer would be introduced on an impact space that utilizes the switch as opposed to the transmission area (HUB). The crash area would be utilized since the utilization of HUBS in network setting is continuously diminishing because of its communicating nature.

Future Enhancements

It is not possible to develop a system that meets all the requirements of the user. User require- ments keep changing as the system is being used. Some of the future enhancements that can bedone to this system are:

- 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 technolo-gies.

7. Additional Sources

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

Asrodia, P., & Patel, H. (2012). Network traffic analysis using packet sniffer. *International Journal of Engineering Research and Applications (IJERA)* [www.ijera.com], *2*(3), 854-856.

Brozycki, J. (2010). *Capturing and analyzing packets with Perl.*

Dabir, A., & Matrawy, A. (2007). Bottleneck analysis of traffic monitoring using Wireshark. *4th Interna- tional Conference on Innovations in Information Technology, 2007*, IEEE Innovations '07 (pp. 158 –162)

Deri, L. (n.d.). *Improving passive packet capture: Beyond device polling*. Retrieved from http://www.net-security.org/dl/articles/Ring.pdf

Dhar, S. (2002). *Switchsniff*. Retrieved from http://www.linuxjournal.com/article.php

Fuentes, F., & Kar, D. (2005). Ethereal vs. Tcpdump: A comparative study on packet sniffing tools for educational purpose. *Computer Journal of Computing Sciences in Colleges, 20*(4), 169-176.

JFreeChart. (n.d.). *JFreeChart*. Retrieved from http://www.jfree.org/jfreechart/download.html Jpcap. (2011). *Jpcap*. Retrieved from http://jpcap.sourceforge.net/
JRE. (n.d.). JRE. Retrieved from http://www.oracle.com/technetwork/java/javase/downloads/index.html Kjell, B. (n.d.). *Introduction to computer science using Java.*
Lewis, J., & Loftus, W. (2001). *Java software solutions.* Addison Wesley.

McCanne, S., & Jacobson, V. (1992). *The BSD packet filter: A new architecture for user-level packet cap-ture*.

Muna, M., Jawhar, T., & Mehrotra, M. (2010). System design for packet sniffer using NDIS hooking. *International Journal of Computer Science & Communication, 1*(1), 171-173.

Niphadkar, S. (2006). *Analysis of packet sniffers – TCPdump VS Ngrep VS Snoop*

Parmar, R., & Patel, H. (2011). NetCap: A packet sniffer in Java. *International Journal of Computer Sci- ence and Technology, 2*(3).

Senthil, K. P., & Arumugam, S. (2012). Establishing a valuable method of packet capture and packet ana- lyzer tools in firewall. *International Journal of Research Studies in Computing, 1*(1), 11-20

Wireshark. (2009). Wireshark: introduction. Retrieved from http://www.wireshark.org/ TcpDump. (2009). Overview of TcpDump. Retrieved from http://www.tcpdump.org/ Winpcap. (2009). Sniffers: Wincap. Retrieved from http://www.wipcap.org/download

8. References

Awodele, O., & Otusile, O. (2012). The design and implementation of Psniffer model for network security. *International Journal of Electronics Communication and Computer Engineering,3*(6).

Chan, C. Y. (2002). *A network packet analyzer with database support*. Retrieved from http://www.cs.rpi.edu/~szymansk/theses/chan.ms.02.pdf