Volume 5, Issue 2, February 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Virtual Sniff

Shalini Bhagia, Riya Vora, Pranay Mali, Pallavi Jain
I.T, Pune University, Pune,
Maharashtra, India

Abstract—Virtual Sniff is aimed at developing PACKET BASED NETWORK SNIFFER. A packet sniffer is a piece of software or hardware that monitors all network traffic. Virtual Sniff presents a packet sniffer written in java that captures network data as well as provides sufficient means for decision making process of an administrator. The aim of this project is to develop a model that consume comparatively less memory on hard disk. ARP cache poisoning method is used for sniffing in this model.

Keywords—packets,network traffic,sniffing,sniffer,promiscuous mode

I. INTRODUCTION

Packet sniffing is the technique of monitoring and capturing packets of data flowing across a computer network. It is a network layer attack. It consists of capturing all the packets from the network and reading the content in search of sensitive data. The tool used to perform this activity is called a packet sniffer.A packet sniffer is a computer program or a piece of computer hardware that can intercept and log traffic passing over a network. Packet sniffers are used by network technicians to diagnose network-related problems and also used by hackers for less than noble purposes such as spying on network user traffic and collecting passwords .Packet Sniffers are of two types: active and passive . Active sniffers are the ones which modify the packet data due to which they can be detected easily. Passive sniffers do not modify the packet data and cannot be detected easily. Virtual Sniff is a passive sniffer i.e. it will be used only for detecting the traffic.

II. RELATED WORK

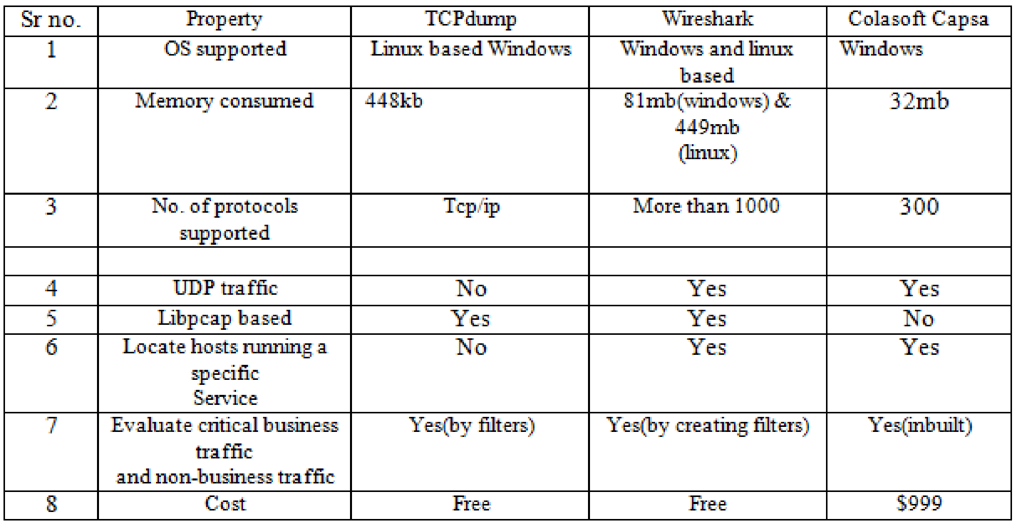
A. TCPdump

TCPdump [1] is the oldest packet sniffer based on command line interface. It works only on Linux based systems. TCPdump uses a libcap library. It can be used to read live capture or already captured log file. TCPdump requires third party tools to display graphs of the transactions [2].It requires 448kb disk space. It is harder to learn TCPdump and its filtering rules. It displays only TCP/IP packet. TCPdump is written in C language.

B. Wireshark

Wireshark[3] is another commonly known packet sniffer. It has a user-friendly graphical user interface (GUI). Wireshark is also written in C language. It has some integrated filtering and sorting options. Wireshark runs on linux as well as windows based systems. After installation Wireshark[2] consumes 81 MB in windows and 449 MB in Linux. Wireshark can also run in command-line mode.

TABLE I TCPDUMP VS WIRESHARK VS COLASOFT CAPSA



Sr no.	Property	TCPdump	Wireshark	Colasoft Capsa
1	OS supported	Linux based Windows	Windows and linux based	Windows
2	Memory consumed	448kb	81mb(windows) & 449mb (linux)	32mb
3	No. of protocols supported	Tcp/ip	More than 1000	300
4	UDP traffic	No	Yes	Yes
5	Libpcap based	Yes	Yes	No
6	Locate hosts running a specific Service	No	Yes	Yes
7	Evaluate critical business traffic and non-business traffic	Yes(by filters)	Yes(by creating filters)	Yes(inbuilt)
8	Cost	Free	Free	$999

C. Colasoft Capsa

Colasoft Capsa [4] is almost similar to wireshark. It also has an interactive GUI and it supports most of the features of wireshark. Colasoft Capsa runs only on the windows platform. Like TCPdump and wireshark this sniffer is also written in C language. It consumes 32 mb disk space. Colasoft Capsa cover only 300 protocols which is very less when compared to wireshark. Colasoft Capsa has an additional feature of notifying alerts by email.
The table below shows the comparison of the 3 sniffing tools discussed above

D. Promiscuous Mode

Each machine present in any network has its own unique physical address. This address differs it from the other machines [7]. When a packet is sent to a machine on this network, it will be transmitted to all the machines present on the network. According to the principle of Ethernet, all the machines on the network can see the traffic passing through the network, but would ignore the packets which do not belong to them. A machine would receive only those packets which match its physical address. When a packet is received by a machine, the machine’s network interface card (NIC) compares the MAC address of the packet to its own. If the MAC address matches then machine accepts the packet otherwise drops it. However, if the Network Interface Card (NIC) of the machine is set in promiscuous mode then it would receive all the packets travelling through the network irrespective of the destination address. Thus, the mode which allows a machine to accept all the packets travelling through the network is called promiscuous mode. The sniffers basically operate in the promiscuous mode due to which they receive all the packets which are travelling in the network.

E. ARP Cache poisoning

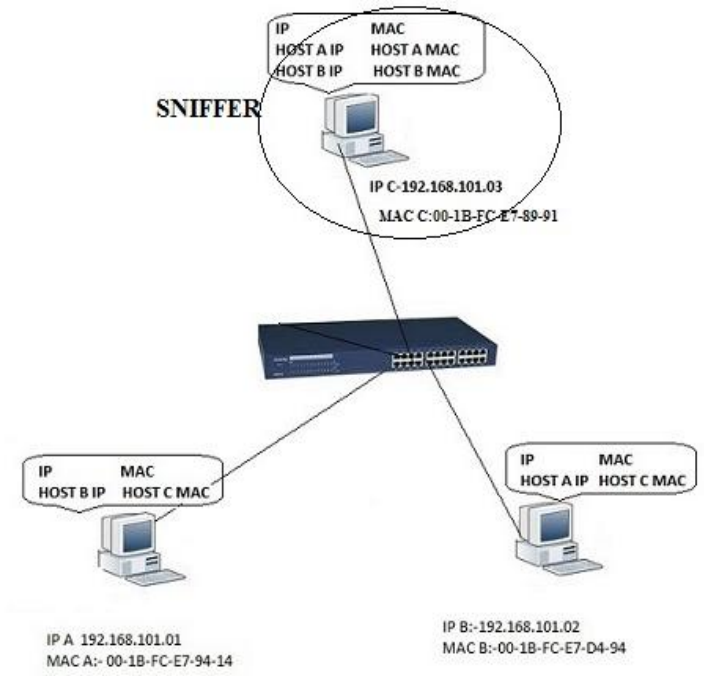


Fig.1 ARP cache poisoning technique

The above fig explains the ARP cache poisoning attack which is going to be used in the proposed system to operate the sniffer in non-promiscuous mode. As shown in fig 1 three hosts A, B and C are connected to a switch. The host C is the host performing the attack and A and B are the victim hosts. Host C sends an ARP poison packet to host A which contains IP address of host B in source IP address field and MAC address of host C in source mac address field. When host A receives this packet it poisons its local ARP cache value by entering false values. After this the host C send an ARP poison packet to host B which contains the IP address of host A in source IP address field and MAC address of host C in source MAC address field. Now host B also poisons its local ARP cache by entering false values. Due to this process the local ARP caches of both the hosts A and B get corrupted. Now after this process both the hosts A and B are unable to communicate with each other directly. Each of these hosts first sends the packet to the host C on which the sniffer is installed and the sniffer reroutes the packet to its original destination.

III. PROPOSED SYSTEM

The aim is to develop a sniffing tool which consumes less memory on disk and also captures all the packets on the network. Virtual sniff runs in two modes promiscuous mode and non-promiscuous mode. For non-promiscuous mode virtual sniff uses ARP cache poisoning. Virtual sniff allows the user to capture the whole packet or only the header of the packet. After selecting the device virtual sniff starts capturing the packets. Once the packets are captured the data of the packet is displayed in the hex and ASCII format. Also the other contents of the packets are displayed. Using this data the system can perform protocol analysis and layer wise analysis. Layer wise analysis is shown in the statistical as well as the graphical format. Pie-chart and line graph are used for graphical representation

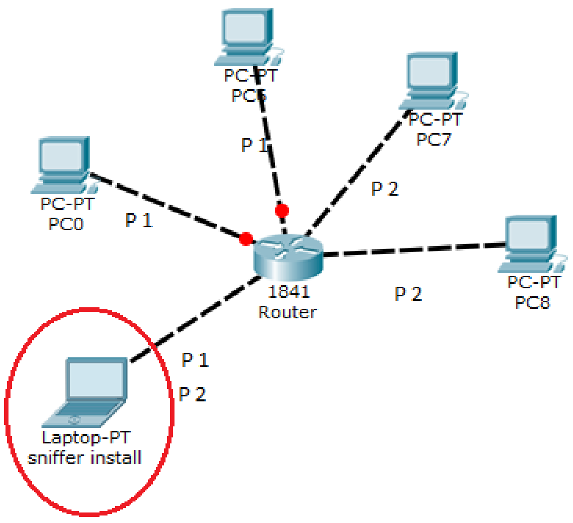


Fig. 2 Architecture of virtual Sniff.

Fig 2 shows the architecture of Virtual Sniff. As seen in the fig five machines are connected to router. The machine in red circle has Virtual Sniff installed on it. Packets are being transferred from one machine to another using router. Here PC0 is sending packet P1 to PC5 and PC7 is sending packet P2 to PC8 using router. As the machine in red circle is in promiscuous mode it captures all the packets travelling through network therefore the packets P1 and P2 are also captured by it. The virtual sniff in this way captures all the packets travelling through the network and analyses those packets.

A. Algorithm

```
//Input: Device (D) and Packets
//Output: Packet Data
//Pre-requisites: Install Sniffer and JPCAP/WINPCAP
    1)  Start
    2)  If D==NULL then
                Display message “Devices not found” goto 8

        Else
            a)  Select Devices
            b)  Select mode
                    If (mode==promiscuous)
                            Flag in NIC is set to 1
                    If (mode==! promiscuous)
                            Flag in NIC is set to 0
            c)  Select what to capture header/whole
                packets/others

    3)  The packets are captured and the data of the packets which is in the hex format is converted to ASCII format
        and displayed in both the hex as well as the ASCII format.

    4)  Statistical information
            a.  If opt==cumulative then
                        Pie-chart will be displayed goto 5
            b.  If opt==continuous then
                        Line Graph will be displayed goto 5
    5)    If  opt1=”Overall information”
                  Statistic view for all protocols is shown
          Else if opt1=“network layer protocol ratio”
                  Statistic view for network layer protocols is shown
          Else if opt1=”transport layer protocol ratio”
                  Statistic view for transport layer protocols is shown
          Else if opt1=”application layer protocol ratio”
                  Statistic view for application layer protocols is shown
    6)  To  save the log file click on save
    7)  To view previous log file click on open
    8)  Exit
```

The above algorithm shows the flow of the proposed system. In order to run virtual sniff the user must have JPCAP/WinPcap and Virtual Sniff installed on his/her machine. Once the user has finished installing these files he/she can start sniffing. Once the user starts sniffing the virtual sniff searches for the available devices. These devices are the device drivers which are present on the user's machine like the Ethernet driver, Wireless LAN driver, etc. The devices are located using the **Network Interface[] devices** function of JPCAP. If there are no devices present then the system displays a message of "Devices Not Found". Once the devices are found the user now has to select the device, mode in which to operate the sniffer and what to capture i.e. header, whole packet or others. If the user selects promiscuous mode then the flag in the network interface card is set to 1. The packets are captured using the **Jpcaptor** function. The size of header is 68 bytes and the size of whole packet is 1514 bytes. The others option allows the user to specify the size of the packet he/she has to capture. This size of others varies from 64 bytes to 1514 bytes. Capturing the whole packet displays the entire packet information whereas capturing only the header does not give the payload information. Once the packets are captured the packet data which in hex format is converted to ASCII and displayed. Along with the payload information other information like source address, destination address, frame size, protocol, date, time, etc. is also displayed. The proposed system shows the layer wise information of data link layer, Transport layer, network layer and application layer in a graphical format. In layer wise information the system shows the various protocols present at these layers and the no of packets using those protocols. It shows the information in the form of a line graph as well as a pie-chart. To display the information in form of a pie chart the user has to select the cumulative option and to display the information in line graph the user has to select continuous option. After selecting either of this option the user has to choose the layer of which he/she wants to see the information. After all this if the user wants if the user wants to save the captured packets he/she can do so by clicking the save option. The user can later open this file using the open option.

IV. ADVANTAGES

1. The sniffer is platform-independent.
2. It can capture packets in promiscuous as well as non-promiscuous mode.
3. It consumes very less memory on disk.
4. Easy to use due to rich friendly GUI.
5. It can save the captured packets as well as retrieve these saved contents when required.

V. CONCLUSION

Virtual Sniff is a passive sniffer. It would be installed on a collision domain which makes use of a switch instead of a hub. Virtual sniff captures network traffic and analyses it and allows the user to take only the features he needs. The system requires little memory size for installation and enables the user to store his/her selected features in a file for later use in his/her work. Consequently, this will reduce the memory that is used to store the data. Finally, Virtual Sniff contains additional functionalities like 3D pie chart, line graphs, etc. Virtual sniff is developed in java due to which it is platform independent.

VI. FUTURE SCOPE

The future enhancements which can be done to this system are:

1. The present application is only used in intranet therefore it can be extended to internet.
2. Based on the future security issues, the security of the system can be improved.
3. The system can be upgraded with the emerging technologies.

ACKNOWLEDGMENT

We take this opportunity to thank our project guide Prof. S.U.Ayachit and Head of the Department Prof S.V.Todkari for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of Jayawantrao Sawant College Of Engineering, Hadapsar for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the Principal Dr.M.G.Jadhav and the institute for providing the required facilities, Internet access and important books.

REFERENCES

[1] All about TCPdump [Online] Available http://www.TCPdump.org/.

[2] "*Packet Sniffer – A Comparative Study*" Dr. Charu Gandhi, Gaurav Suri, Rishi P. Golyan, Pupul Saxena, Bhavya K. Saxena

[3] All about Wireshark [Online] Available http://www.Wireshark.org/.

[4] All about Colasoft Capsa [Online] Available www.colasoft.com

[5] L. Garcia,"*programming with libpcap*," in Hacking- Practical protection hard core IT magazine, Vol 3, 2008, pp. 38-46.

[6] A. Shah, D. Bhatt, P. Agarwal, and P. Agarwal, "Effect of Packet-Size over Network Performance", International Journal of Electronics and Computer Science Engineering, Vol. 1, pp. 762-766, 2012

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

[8] Ansari, S., Rajeev, S., & Chandrashekar, H. (2002). Packet sniffing: A brief introduction. *IEEE Potentials*

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

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

[11] All about Tools [Online] Available: http://www. sectools.org