IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Capturing the Data Packet by Setting the NIC Card in Promiscuous Mode

R. Prajwala Rani
Associate Professor, Siddhartha Engineering College, Hyderabad, India
prajwalaravela@gmail.com

Abstract

Packet Sniffer is computer software. That can capture the data packets from the network. Whatever the data is capturing that will be decoded. the decoding or capturing the data can be done by using network interfacing controller in promiscuous mode. It also discusses ways to detect the presence of such software on the network and to handle them in an efficient way. Packet sniffer can able to capture the data of entire network that either wired network or wireless network. This paper presents practical results of capturing the data packet by setting the NIC card and results are verified.

Keywords: MAC, ARP, Packet sniffer.

Introduction

Packet sniffing is used within a network in order to capture and register data flows. Packet sniffing allows you to discern each individual packet and analyze its content based on predefined parameters. Packet sniffing allows for very detailed network monitoring and bandwidth usage analysis. It, however, requires a broader knowledge of networks and their inner functions, in order to be able to recognize the relevance of the data being monitored

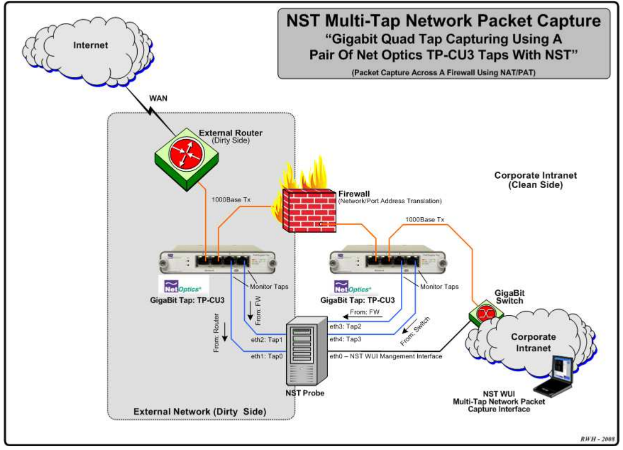
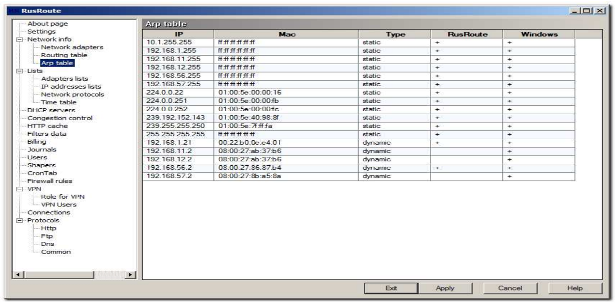


Fig1 Block Diagram of Packet Sniffer

In a switched network environment, packets are sent to their destination port by MAC address. This process requires that the systems on the network maintain a table associating MAC addresses to ports. In a switched environment, packets are only sent to devices that they are meant for. Even in this switched environment, there are ways to sniff other devices' packets. One such way is to spoof your MAC address and poison the arp table. Since arp keeps no state information, the arp cache can be overwritten (unless an entry is explicitly marked as permanent).



Arp table

IP	Mac	Type	RusRoute	Windows
10.1.255.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.1.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.11.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.12.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.56.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.57.255	ff:ff:ff:ff:ff:ff	static	+	+
224.0.0.22	01:00:5e:00:00:16	static	+	+
224.0.0.251	01:00:5e:00:00:fb	static	+	+
224.0.0.252	01:00:5e:00:00:fc	static	+	+
239.192.152.143	01:00:5e:40:98:8f	static	+	+
239.255.255.250	01:00:5e:7f:ff:fa	static	+	+
255.255.255.255	ff:ff:ff:ff:ff:ff	static	+	+
192.168.1.21	00:22:b0:0e:e4:01	dynamic	+	+
192.168.11.2	08:00:27:ab:37:b6	dynamic		+
192.168.12.2	08:00:27:ab:37:b6	dynamic		+
192.168.56.2	08:00:27:86:87:b4	dynamic	+	+
192.168.57.2	08:00:27:8b:a5:8a	dynamic		+

Arp cache poisoning puts the attacker in position to intercept communications between the two computers. Computer A believes it is communicating with Computer B, but because of the poisoned arp table, the communication actually goes to the attacker's computer. The attacker can then either respond to Computer A (pretending to be Computer B), or simply forward the packets to its intended destination, but only after the packet information is captured and logged for later use by the attacker. Likewise, the response from Computer B can be captured and logged by the attacker, who has also used Arp poisoning to make Computer B think the attacker's computer is Computer A. This type of attack is known as Man in the Middle attack. A *network sniffers* monitor's data flowing over computer network links. It can be a self-contained software program or a hardware device with the appropriate software or firmware programming. Also sometimes called "network probes" or "snoops," sniffers examine network traffic, making a copy of the data but without redirecting or altering it. Some sniffers work only with TCP/IP packets, but the more sophisticated tools can work with many other protocols and at lower levels including Ethernet frames.

a) Media Access control address:

A MAC address, or *Media Access Control* address, is a 48- or 64-bit address associated with a network adapter. While IP addresses are associated with software, MAC addresses are linked to the hardware of network adapters. For this reason, the MAC address is sometimes called the hardware address, the burned-in address (BIA), or the physical address. MAC addresses are expressed in hexadecimal notation in the following format: 01-23-45-67-89-AB, in the case of a 48-bit address, or 01-23-45-67-89-AB-CD-EF, in the case of a 64-bit address. Colons (:) are sometimes used instead of dashes (-). MAC addresses are often considered permanent, but in some circumstances, they can be changed. There are two types of MAC addresses:

b) Universally Administered Address

The UAA, or Universally Administered Address, is the most commonly used type of MAC address. This address is assigned to the network adapter when it is manufactured. The first three octets define the manufacturer, while the second three octets vary and identify the individual adapter. All network adapter manufacturers have their own code, called the Organizationally Unique Identifier (OUI). For example, in the MAC address 00-14-22-01-23-45, the first three octets are 00-14-22. This is the OUI for Dell. Other common OUIs include 00-04-DC for Nortel, 00-40-96 for Cisco, and 00-30-BD for Belkin. Most large manufacturers of networking equipment have multiple OUIs.

c) Locally Administered Address

The LAA, or Locally Administered Address, is an address that changes an adapter's MAC address. The LAA is a type of administered MAC address, and it is possible to change the LAA of a network adapter to any address of allowed length. When the LAA is set, the network adapter uses the LAA as its MAC address. Otherwise, the network adapter uses the UAA as its MAC address. All devices on the same subnet must have different MAC addresses, however. MAC addresses are very useful in diagnosing network issues, such as duplicate IP addresses, so it is a good practice to allow devices to use their UAAs instead of assigning LAAs, unless there is a compelling reason to do so. MAC addresses are useful for security purposes, as they are only rarely changed from the default. IP addresses can change dynamically, especially on networks using DHCP to assign IP addresses, so MAC addresses are often a more reliable way to identify senders and receivers of network traffic. On wireless networks, MAC address filtering is a common security measure to prevent unwanted network access. In MAC address filtering, a wireless router is configured to accept traffic from certain MAC addresses. In this way, as white listed devices are assigned new IP addresses through DHCP, they retain their ability to communicate on the network. Any intruder attempting to impersonate a valid user on the network by masquerading with a faked IP address will not be able to do so because the computer's MAC address will not match any of those in the white list. This security method is only minimally successful, however, as a determined intruder can fake a MAC address almost as easily as an IP address.

d) nic in promiscous mode:

"Promiscuous mode" is a network interface mode in which the NIC reports every packet that it sees. If you're using the Wires hark packet sniffer and have it set to "promiscuous mode" in the Capture Options dialog box, you might reasonably think that you're going to be seeing all the traffic on your network segment. This is not necessarily the case, and there could be several reasons for it. So before you use this tool to draw conclusions about traffic on your Windows network, it's worth seeing if you're really capturing what you think you're capturing. If you're connected to a switch as opposed to a hub, broadcast traffic and multicast traffic will go to all ports, but unicast traffic does not. Check your switch to see if you can configure the port you're using for Wires hark to have all traffic sent to it ("monitor" mode), and/or to "mirror" traffic from one port to another. (Here's one of the benefits of those more expensive managed switches.) .You might think that you could revert to using an old-style hub, given that hubs don't segment network traffic as switches do; and this "hubbing out" method might work, but even hubs don't necessarily pass all traffic. For example, on some multispeed hubs, listening on a 100 Mbps port may not capture traffic on ports operating at 10 Mbps. Separate from any hub and switch issues, some network interfaces do not allow themselves to be thrown into promiscuous mode. So if you think your network plumbing should permit promiscuous mode, Sometimes there's a setting in the driver properties page in Device Manager that will allow you to manually set promiscuous mode if Wires hark is unsuccessful in doing so automatically. Some network interfaces even have a driver setting that permits an administrator to *permanently* disable promiscuous mode on that adapter! So before you make any grand pronouncements about the results of your Wires hark research, make sure you inform yourself about the ways in which the traffic that you're capturing may not be showing the whole picture. This tool is easy to use for capturing traffic in and out of one specific host, but beyond that, there are a lot of variables to consider

Fig 2 NIC Card Hardware structure

Sniffer Mechanism

Packet sniffing may sound like the latest street drug craze but it's far from it. Packet sniffers or protocol analyzers are tools that are commonly used by network technicians to diagnose network-related problems. Packet sniffers can also be used by hackers for less than noble purposes such as spying on network user traffic and collecting passwords . Packet sniffers come in a couple of different forms. Some packet sniffers used by network technicians are single-purpose dedicated hardware solutions while other packet sniffers are software applications that run on standard consumer-grade computers, utilizing the network hardware provided on the host computer to perform packet capture and injection tasks .Packet sniffers work by intercepting and logging network traffic that they can 'see' via the wired or wireless network interface that the packet sniffing software has access to on its host computer. On a wired network, what can be captured depends on the structure of the network. A packet sniffer might be able to see traffic on an entire network or only a certain segment of it, depending on how the network switches are configured, placed, etc. On wireless networks, packet sniffers can usually only capture one channel at a time unless the host computer has multiple wireless interfaces that allow for multichannel capture. Once the raw packet data is captured, the packet sniffing software must analyze it and present it in human-readable form so that the person using the packet sniffing software can make sense of it. The person analyzing the data can view details of the 'conversation' happening between two or more nodes on the network. Network technicians can use this information to determine where a fault lies, such as determining which device failed to respond to a network request . Hackers can use sniffers to eavesdrop on unencrypted data in the packets to see what information is being exchanged between two parties. They can also capture information such as passwords and authentication tokens (if they are sent in the clear). Hackers can also capture packets for later playback in replay, man-in-the-middle, and packet injection attacks that some systems may be vulnerable to. Just like everybody else, both network engineers and hackers love free stuff, which is why open source and freeware sniffer software applications are often the tools of choice for packet sniffing tasks. If your a network technician or administrator and you want to see if anyone on your network is using a sniffer tool, check out a tool called Anti sniff. Ant sniff can detect if a network interface on your network has been put into 'promiscuous mode' (don't laugh that's the actual name for it), which is the required mode for packet capture tasks. Another way to protect your network traffic from being sniffed is to use encryption such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS). Encryption doesn't prevent packet sniffers from seeing source and destination information, but it does encrypt the data packet's payload so that all the sniffer sees is encrypted gibberish. Any attempt to modify or inject data into the packets would likely fail since messing with the encrypted data would cause errors that would be evident when the encrypted information was decrypted at the other end. Sniffers are great tools for diagnosing down-in-the-weeds network problems. Unfortunately, they are also useful for hacking purposes as well. It's important for security professionals to familiarize themselves with these tools so they can see how a hacker might use them against their network.

Conclusion

This packet sniffer can be enhanced in future by incorporating features like making the packet sniffer program platform independent, filtering the packets using filter table, filtering the suspect content from the network traffic and gather and report network statistics. A packet sniffer is not just a hacker’s tool. It can be used for network traffic monitoring, traffic analysis, troubleshooting and other useful purposes. However, a user can employ a number of techniques to detect sniffers on the network as discussed in this paper and protect the data from being sniffed.

References

[1] g. varghese, “network algorithmic: an interdisciplinary approach to designing fast networked devices”, san francisco, ca: morgan kaufmann, 2005.

[2] j. cleary, s. donnelly, i. graham, "design principles for accurate passive measurement," in proc. pam 2000 passive and active measurement workshop (apr. 2000).

[3] 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

[4] s. ansari, rajeev s.g. and chandrasekhar h.s, “packet sniffing: a brief introduction”, ieee potentials, dec 2002- jan 2003, volume:21, issue:5, pp:17 – 19

[5] daiji sanai, “detection of promiscuous nodes using arp packet”, http://www.securityfriday.com/

[6] ryan spangler , packet sniffer detection with antisniff, university of wisconsin – whitewater, department of computer and network administration, may 2003

[7] zouheir trabelsi, hamza rahmani, kamel kaouech, mounirvvfrikha, “malicious sniffing system detection platform”,vproceedings of the 2004 international symposium on applications and the internet (saint’04), ieee computer society

[8] hornig, c., “a standard for the transmission of ip data grams over ethernet networks”, rfc-894, symbolic cambridge research center, april 1984.