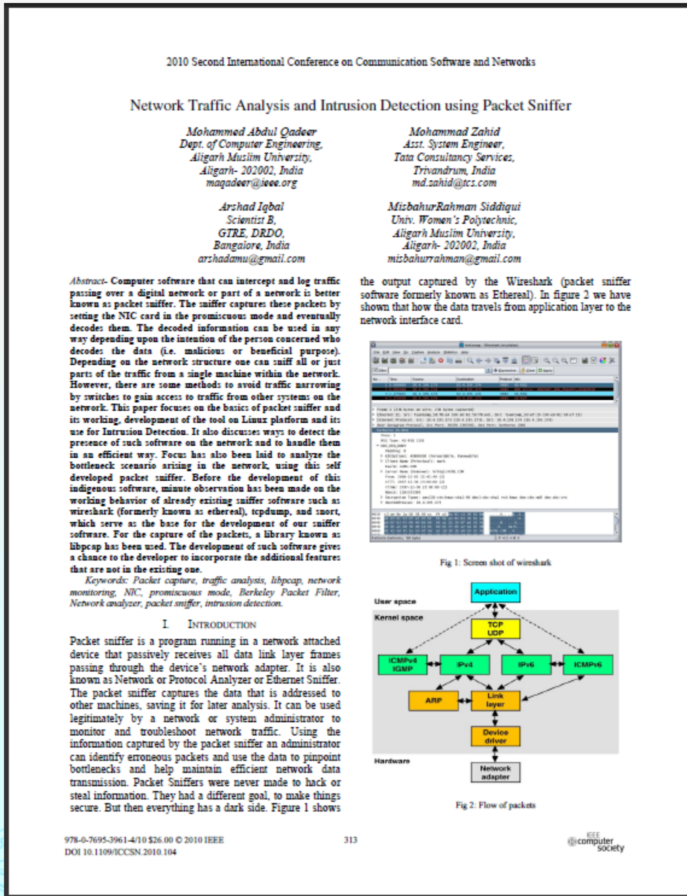
Network Traffic Analysis and Intrusion Detection using Packet Sniffer

Guanqun Wang

Supervisor: Prof. Nirmalya Roy

Introduction

- The paper is from “2010 Second International Conference on Communication Software and Networks”;
- Basic Idea
  - Introduce the basics of packet sniffer and its working;
  - Develop a packet sniffer tool on Linux Platform;
  - Intrusion detection and analysis of the bottleneck scenario using this designed packet sniffer.



2010 Second International Conference on Communication Software and Networks

Network Traffic Analysis and Intrusion Detection using Packet Sniffer

Mohammed Abdul Qadeer
Dept. of Computer Engineering,
Aligarh Muslim University,
Aligarh- 202002, India
maqadeer@ieee.org

Mohammad Zahid
Asst. System Engineer,
Tata Consultancy Services,
Trivandrum, India
md.zahid@tcs.com

Arshad Iqbal
Scientist B,
GTRE, DRDO,
Bangalore, India
arshadamu@gmail.com

MisbahurRahman Siddiqui
Univ. Women's Polytechnic,
Aligarh Muslim University,
Aligarh- 202002, India
misbahurrahman@gmail.com

Abstract- **Computer software that can intercept and log traffic passing over a digital network or part of a network is better known as packet sniffer. The sniffer captures these packets by setting the NIC card in the promiscuous mode and eventually decodes them. The decoded information can be used in any way depending upon the intention of the person concerned who decodes the data (i.e. malicious or beneficial purpose). Depending on the network structure one can sniff all or just parts of the traffic from a single machine within the network. However, there are some methods to avoid traffic narrowing by switches to gain access to traffic from other systems on the network. This paper focuses on the basics of packet sniffer and its working, development of the tool on Linux platform and its use for Intrusion Detection. It also discusses ways to detect the presence of such software on the network and to handle them in an efficient way. Focus has also been laid to analyze the bottleneck scenario arising in the network, using this self developed packet sniffer. Before the development of this indigenous software, minute observation has been made on the working behavior of already existing sniffer software such as wireshark (formerly known as ethereal), tcpdump, and snort, which serve as the base for the development of our sniffer software. For the capture of the packets, a library known as libpcap has been used. The development of such software gives a chance to the developer to incorporate the additional features that are not in the existing one.**

Keywords: Packet capture, traffic analysis, libpcap, network monitoring, NIC, promiscuous mode, Berkeley Packet Filter, Network analyzer, packet sniffer, intrusion detection.

I. Introduction

Packet sniffer is a program running in a network attached device that passively receives all data link layer frames passing through the device`s network adapter. It is also known as Network or Protocol Analyzer or Ethernet Sniffer. The packet sniffer captures the data that is addressed to other machines, saving it for later analysis. It can be used legitimately by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. Packet Sniffers were never made to hack or steal information. They had a different goal, to make things secure. But then everything has a dark side. Figure 1 shows the output captured by the Wireshark (packet sniffer software formerly known as Ethereal). In figure 2 we have shown that how the data travels from application layer to the network interface card.

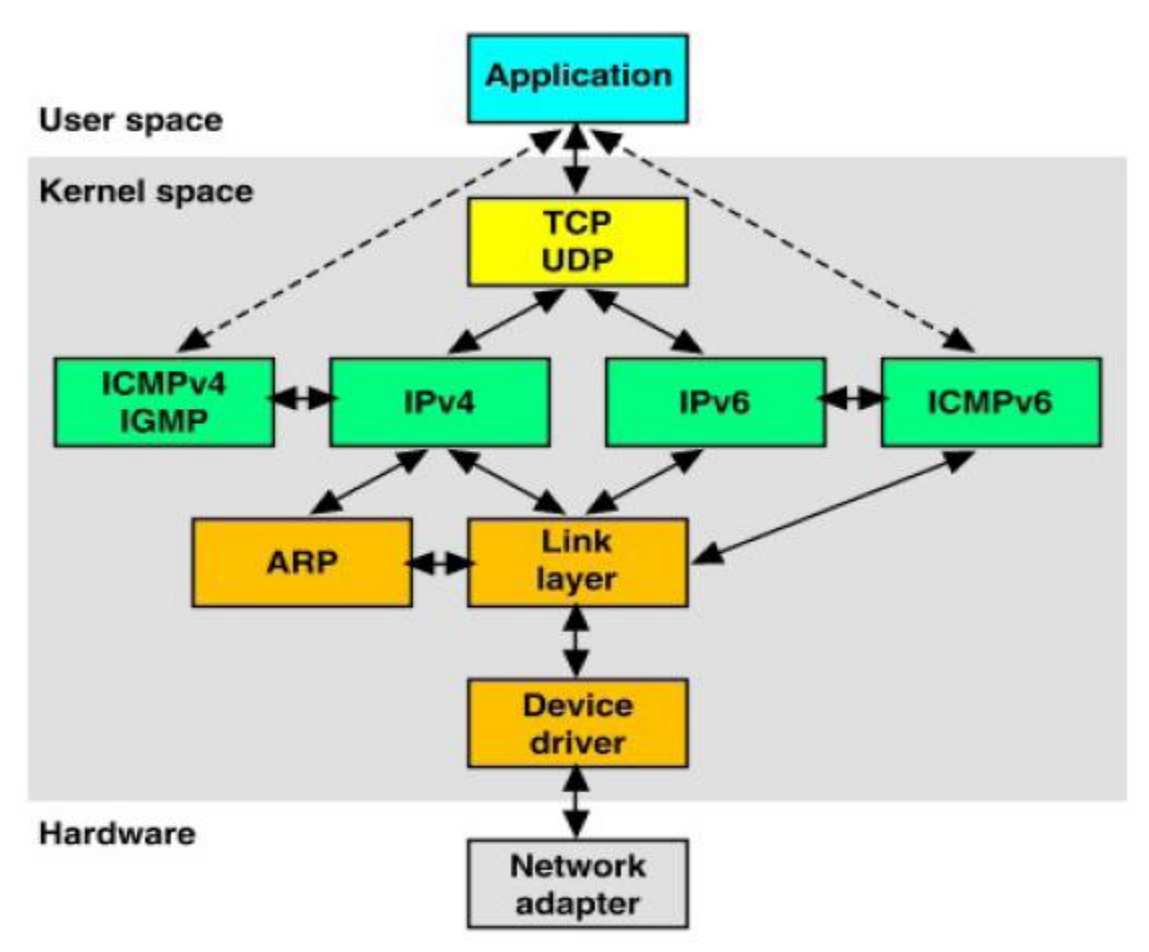
Fig 1: Screen shot of wireshark

Fig 2: Flow of packets

978-0-7695-3961-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCSN.2010.104

Flow of Packets

- Packet sniffer is a program running in a network attached device that passively receives all data link layer frames passing through the device's **network adapter**.



Standard Packet Capture Library: Libpacp

- Libpacp is a widely used standard packet capture library that was developed for use with BPF (Berkely Packet Filter) kernel device;
- Libpcap is a C language library that extends the BPF library constructs;
- Libpcap is used to capture the packets on the network directly from the network adapter;
- Libpcap is an in built feature of the operating system. It provides packet capturing and filtering capability.

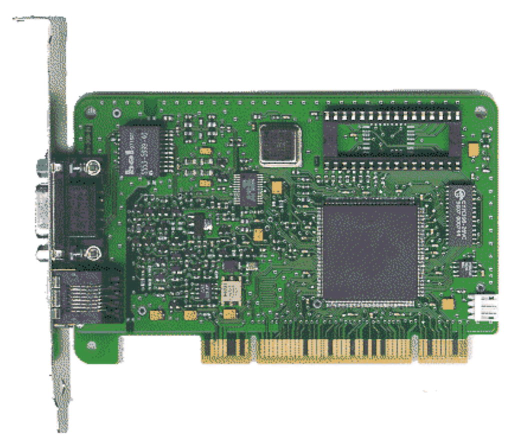
Network Interface Card (NIC) and Promiscuous Mode

- It is BPF (Berkely Packet Filter), which enables communication between operating system and NIC.
- When a packet is received by a NIC, it first compares the MAC address of the packet to its own.
- NIC works in two modes:
  - Non-promiscuous mode

    If the MAC address matches, it accepts the packet otherwise filters it.

    Each network card is minding its own business and reading only the frames directed to it.
  - Promiscuous mode.

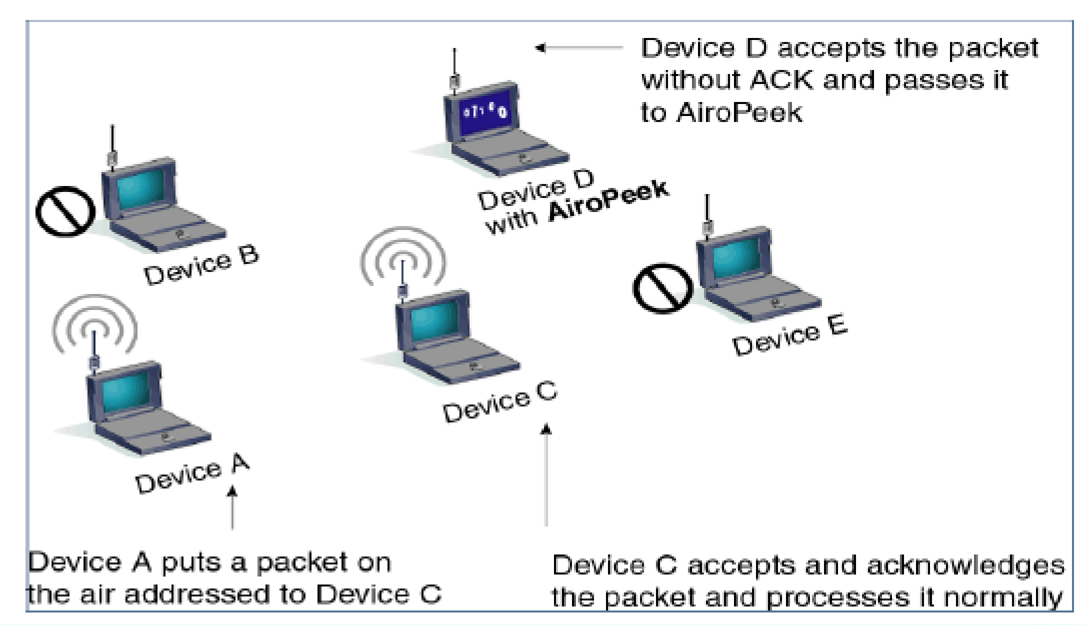
    receives all packets even they are not intended for it
- In order to capture the packets, NIC has to be set in the promiscuous mode.



Network Interface Card

Network Interface Card (NIC) and Promiscuous Mode

- The figure shows that the data sent by device A to device C is also received by device D which is set in promiscuous mode.



Device D accepts the packet without ACK and passes it to AiroPeek
Device D with AiroPeek
Device B
Device E
Device C
Device A
Device A puts a packet on the air addressed to Device C
Device C accepts and acknowledges the packet and processes it normally

SNIFFER WORKING MECHANISMS

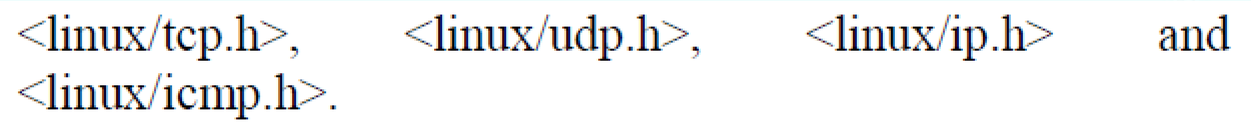
- A node whose NIC is set in the promiscuous mode tends to receives the packets passing through it.
- The packet arriving at the NIC are copied to the device driver memory, which is then passed to the kernel buffer from where it is used by the user application.
- Each socket has two kernel buffers associated with it for reading and writing.
- A single packet is handled by the buffer at a time for the application processing before next packet is copied into it.
- The new approach taken in the development of our packet sniffer is to improve the performance of packet sniffer, using Libpcap to use same buffer space between kernel space and application.

THE DEVELOPMENT OF PACKET SNIFFER ON LINUX PLATFORM

- Step A: Socket creation
  - Socket is a bi-directional communication abstraction via which an application can send and receive data

- Step B: Set NIC in promiscuous mode
  - When a socket is created, a socket stream, similar to the file stream, is created, through which data is read

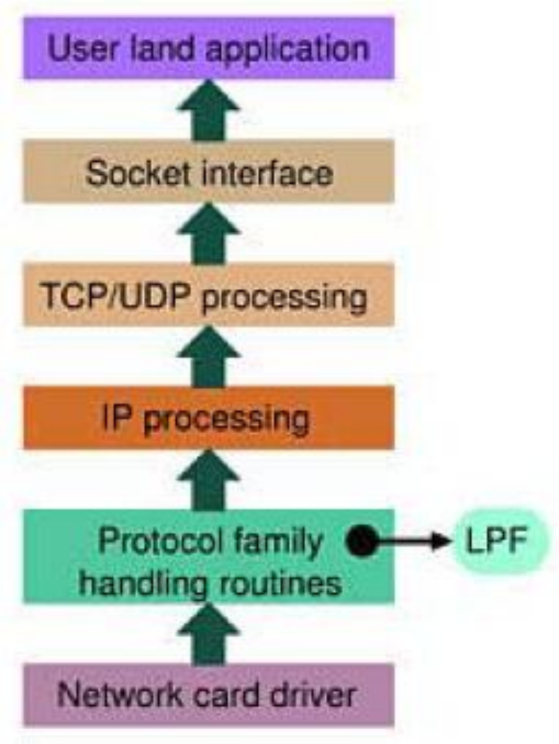
- Step C: Protocol interpretation
  - Interpreted the protocols such as IP, TCP, UDP, ICMP protocols by including the headers as



<linux/tcp.h>, <linux/udp.h>, <linux/ip.h> and <linux/icmp.h>.

Linux Filter

- Filter received packets and print out information only on those we are interested in;
- Insert an “if statement'' in the sniffer's source —— inefficient
- The Optimal solution to this problem is to put the filter as early as possible in the packet-processing chain



METHODS TO SNIFF ON SWITCH

- Method A: ARP spoofing
  - When you want to sniff the traffic originating from a machine, you need to ARP spoof the gateway of the network;
  - Another trick that can be used is to poison a hosts ARP cache by setting the gateway's MAC address to FF:FF:FF:FF:FF:FF (also known as the broadcast MAC).
- MAC flooding
  - a switch can intelligently route packets from one host to another, but it has a limited memory for this work;
  - MAC flooding bombards the switch with fake MAC addresses until the switch can't keep up.

BOTTLENECK ANALYSIS

- On the arrival of the packet at NIC, they have to be transferred to the main memory for processing;
- As we know that the PCI bus has actual transfer of not more than 40 to 50 Mbps;
- Bottleneck is created in writing the packets to disk in traffic sensitive network;
- To handle the bottle neck we can make an effort to use buffering in the user level application

DETECTION OF PACKET SNIFFER

- Packet sniffer can be made of malicious use;
- Ways for detection of packet sniffer:
  - Method A: ARP Detection Technique

    Sniffing host makes mistakes by responding to such packets that are supposed to be filtered by it;

    if an ARP packet is sent to every host and ARP packet is configured such that it does not have broadcast address as destination address and if some host respond to such packets, then those host have there NIC set into promiscuous mode;

    In Linux we can analyze the behavior of filter by examining the source code of this OS.

DETECTION OF PACKET SNIFFER

- Ways for detection of packet sniffer:
  - Method B: RTT Detection

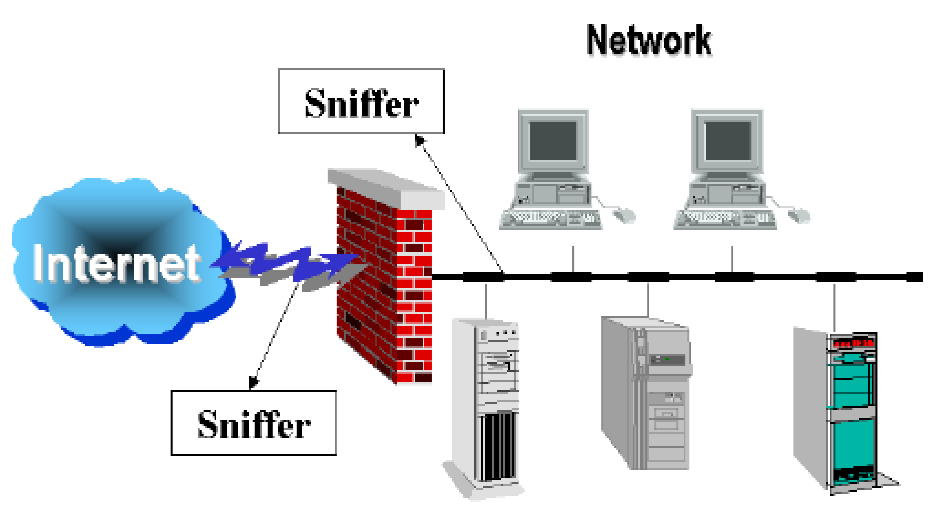
    Round Trip Time (RTT) measurement increases when the host is in promiscuous mode;

  - Method C: SNMP Monitoring

    SNMP: Simple Network Management Protocol;

INTRUSION DETECTION USING PACKET SNIFFER

- Packet Sniffer can be used for intrusion detection also;
- The Intrusion Detection software is placed on the system to read and analyze all traffic
- Looks for specific types of network attacks, such as IP spoofing and packet floods.
- Notify to the administrator by various mode such as console, beeping a pager, sending an e-mail, or even shutting down the network session.



Conclusion

- 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.
- 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.