# Packet Sniffer for Users End Network Performance Monitoring using Python Programming

**G**nyanesh **N**aikar

[**216**]

Course : MSc Information Technology

Semester : Sem-III

Year : 2021

## I. INTRODUCTION

Communication networks are more computerized than ever and rapidly increasing at a significant rate both in size and complexity. Monitoring the information or data transmitted over these networks is getting progressively difficult. As such, networks are subject to neglect and exposed to malicious attacks which could prove disastrous in protection and securing of data. Designing software that would have the capability to monitor and check/analyze the numerous data been transmitted over a computer network is one way to proffer solution to the challenge of monitoring systems. The software will make available the possibility of capturing data in terms of its formatted unit of data known as packet. The captured data which is in its binary format is the converted to readable format and displayed making it possible for network administrators or security professionals to monitor the information being transmitted over the network. Packet sniffer, also known as packet analyzer/ protocol analyzer/network analyzer is one of such software which has the ability to capture data passed over a network, converting it to readable format and reading of data content. It is known to have numerous versatile uses as the packet sniffer can be used legitimately by a network or system administrator to monitor and

troubleshoot network traffic. It is also a vital tool in identifying intruders on a network. It has had a large role in securing of network systems, resolving issues that arise in the network and identifying inconsistent connections. Packet sniffer has been defined in various ways. It has been defined as a tool used to monitor, intercept and decode data packets as they are transmitted across networks. It has been described as a tool that utilizes the process of gathering traffic from a network and storing them for later analysis.

The packet sniffer, in the present world has several significant uses that most security experts utilize in an effort to make network secure for data transmission without worry of an attack. It is relied upon to monitor and filter network traffic and it is an effective tool for testing protocols, diagnosing network problems, identifying configuration issues. Information technology- teams rely on packet sniffers to discover network misuse of any kind and also resolve bottleneck issues. In an intrusion monitoring system, packet sniffers have played a vital role as the monitoring of data is first carried out before detection and prevention of suspicious or malicious activity is performed.

Each machine on an immediate network has its own hardware address which differs from the other. When a data is sent across the network, it is sent in the form of packets. These packets are the chunks of data are directed to the certain designated system though will pass through some nodes on the network. Normally a particular system in a network is designed to receive and read only those data which are intended for it, so, for this reason, the network interface card (NIC) works in non-promiscuous mode and promiscuous mode. When a packet is received by a NIC, it first compares the destination address of the packet to its own. If the MAC address matches, it accepts the packet otherwise filters it. This operation mode where network card discards all packets that do not contain its own MAC address is called non promiscuous mode which basically means that each NIC reads only the frames meant for it. Hence, packet sniffer captures the packets by setting the NIC card of its own system into promiscuous mode and when the packet arrives at the NIC, it is copied to the device driver memory, then passed to the kernel.

The rest of this article is divided into four sections. Literature review of the study was carried out in section II, while the materials and methodology of research is

presented in section III. This followed by implementation and testing discussions in section IV. The paper is concluded in section V.

## II. LITERATURE SURVEY

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 [5]. Using information captured by packet sniffer, an administrator can identify erroneous packets and use the data compiled to pinpoint bottlenecks and help maintain efficient network data transmission. Data addressed to other machines can be captured using the packet sniffer unlike standard network hosts that receive traffic sent specifically to them. Packet sniffers were formerly known to be very expensive dedicated hardware devices, but new advances in technology have allowed for the development of software network analyzers making it more convenient and affordable.

Packet sniffers can be standalone hardware device with specialized software, or it can simply be software that can be installed on a computer system [8]. They are available both free and commercially with the major difference depending on features such as the number of supported protocols that can be decoded, the user interface and statistical capabilities. A packet sniffer is a combination of hardware and software.

Analysis of a network can be carried out for either good or to harm a network and as such, packet sniffers which are tools, and like all tools they can be used to perform these nefarious acts. When used by malicious individuals, sniffers can present a significant threat to the security of a network [8]. On the contrary, packet sniffers can also be used in penetration testing which has been defined as the legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making systems more secure from malicious attacks/threats. Penetration testing is known to determine how a system reacts to an attack, whether or not a system's defense can be breached and what information can be acquired from the testing. The passive nature of the packet sniffer makes detecting them on the network difficult.

As a program, the packet sniffer runs in a network attached device that passively receives all data link layer frames that passes through the device's network

adapter. The packet sniffer as stated captures data addressed to other machines, saving it for later analysis. Over the years, several approaches have been taken into consideration to determine the best way to keep track of network. The packet sniffing method is one of such ways and has been used in several ways. There are different types of network sniffing tools depending on the network, application or protocols are available in markets. Here, the primary and most useful packet sniffer is considered. Wireshark by Gerald Combsis a packet analyzer used for network troubleshooting, analysis. Originally named Ethereal, in May 2006 the project was renamed Wireshark due to trademark issues. It is cross-platform, using pcap to capture packets and runs on various Unix-like operating systems, Solaris, and on Microsoft Windows. It allows the user to put the network interfaces that support promiscuous mode into that mode, in order to see all traffic visible on that interface, not just traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic [9][11]. Wireshark is software that "understands" the structure of different networking protocols. Thus, it is able to display the encapsulation and the fields along with their meanings of different packets specified by different networking protocols. It provides users the capability of capturing the packets traversing over an entire network on a particular interface at a particular time. However, Wireshark has limitation of not reporting some malicious intents performed on a network [10][12].

Tcpdump by McCanne, Leres and Jacobson is a common packet analyzer that runs under the command line. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. It works on most Unix-like operating systems: Linux, Solaris, BSD, and Mac OS. In those systems, tcpdump uses the libpcap library to capture packets. The port of tcpdump for Windows is called WinDump and uses WinPcap-the Windows port of libpcap. It analyzes network behavior, performance and applications that generate or receive network traffic. It can also be used for analyzing the network infrastructure itself by determining whether all necessary routing is occurring properly, allowing the user to further isolate the source of a problem. It is also possible to use tcpdump for the specific purpose of intercepting and displaying the communications of another user or computer [9].

Capsa is a network analyzer for both LAN and WLAN which performs real-time packet capturing, 24/7 network monitoring, advanced protocol analysis, in-depth packet decoding and automatic expert diagnosis. It provides a comprehensive and high-level visibility to your entire network, helps network administrators or network engineers quickly pinpoint and resolve various application problems, and therefore enhance end user experience and guarantee a productive network environment. It can identify and analyze more than 300 network protocols, as well as network applications based on the protocols [10][12].

**Table1: Characteristic Comparison of TCPdump, Wireshark, Capsa and EtherApe**

| S/N | Property | Wireshark | TCPDump | Capsa | Ether Ape |
|---|---|---|---|---|---|
| 1. | OS Supported | Windows and Unix | Unix based | Windows | Unix based |
| 2. | Disk Usage | 81MB (Windows) & 449MB (Unix) | 448KB | 32MB | |
| 3. | Cost | Free | Free | Charged | Free |
| 4. | Open Source | Yes | Yes | No | Yes |
| 5. | Display protocol in OSI Layer structure | Yes | No | Yes | Yes |
| 6. | Libpcap used | Yes | Yes | Yes | Yes |
| 7. | User Interface | GUI/CLI | CLI | GUI | GUI |
| 8. | Creator | The Wireshark Team | The Tcpdump Team | Colasoft | Juan Toledo |

Common types of sniffing methods in use are IP based sniffing, MAC based sniffing and ARP based sniffing which does not put the NIC in promiscuous mode. It is an effective method for sniffing in switched environment [10][13]. Several factors or parameters influence the performance of packet capturing. Packet size is one of such factors yet there is no right answer to the question of which packet size gives best performance [11][14]. Short packets are known to increase load on

devices while long packets increase load on the network which means the less the long packets, the less stressed a network. Long packets mean high ratio of packet payload and packet headers. Tools which support maximum medium sized packets are better tool on this benchmark. Moreover, throughput refers to the amount of data a system processes in bits per second (bps). So, the tool with higher throughput would give better performance but those of large range of rapidly changing throughput leads to random changes in the throughput are not good with respect to the network performance. On the other hand, tools with a range in a pattern are good for the network and show a consistent behavior [15].

Other performance indicators are packet drop rate and response time. Reasons for excessive packet loss could be due to lack of buffer so that all packets coming to the NIC could not be saved [12][13]. However, response time is the time taken to receive acknowledgements between the communicating nodes. Less response time indicates a smaller number of retransmissions and less response time implies better performance. Many network administrators spend a lot of time trying to know what is degrading the performance of their network [14][16]. A typical solution to congestion problem could be an upgrade to the network infrastructure, that is, replace servers with high end servers and increase the bandwidth. However, this solution is expensive, short term and is not entirely scalable seeing that after the upgrade is performed; the congestion problem improves for a while and will later gradually deteriorate as the users change their behavior in response to the upgrade.

The alternative solution to this problem is to deploy a scalable network traffic monitoring and analysis system, in order to understand clearly the dynamics of the traffic and the changes in the internet together with the overall stability of the network. In addition to knowing the health status of the network, monitoring of network activity also has the benefits of detecting denial of service (DoS) and bandwidth theft attacks. In order to conduct analysis of wide range of network behaviors, it is necessary to collect network traffic on a continuous basis rather than as a onetime event which only captures transient behaviors that provides insight into network problems. Collection of long-term network traffic data will provide valuable information for improving and understanding the actual network dynamics [17][18].

# III. SYSTEM REQUIREMENT SPECIFICATION

## a. Aims and Objectives of Paper

The main aim of this paper is to develop a packet sniffer that can help some infosec professionals. Actually, I learned scapy on my college days and wrote several useful tools using python and scapy. These tools are extremely useful for me today as a security professional. Hence, today I want to share some of the tools I developed. Some of the key subjects that will be addressed in this paper are:

- Brief Explanation of Packet Sniffers and Use Cases of them
- Development of packet Sniffer with sequential steps on a layman term
- Availability of all codes on the github repository

today as a security professional. Hence, today I want to share some of the tools I developed

## b. Expected Knowledge on Readers

I am not going deep into describing all the protocols and detailed elaboration of packet sniffer. I have elaborated all the details of network protocols and packet sniffer on my previous paper which is available here. Hence, some sort of scripting python and network knowledge is expected on the readers of this paper. Also, scapy readers are expected to have basic knowledge of scapy framework. The detailed documentation of scapy is available here. Having knowledge of scapy always assists to develop security tools in python.

## c. Scope of the Paper

This paper is written to address the development of a quick packet sniffer using python and scapy. In this paper we are going to classify all packets using layer composition. The packet sniffer will sniff all the incoming packets and outgoing packets from the host machine from all interfaces. Packets will be classified based on TCP, UDP and ICMP protocols. On each protocol classification they are divided into incoming packets and outgoing packets. Some of the information eject on the console are source ip, destination ip, source port, destination port, geo location

etc. The packet sniffer does not have GUI interface and is executed from command console.

# IV. SYSTEM REQUIREMENT

## Software Environment:

| | | |
|---|---|---|
| Operating Platform | : | WINDOWS XP |
| Front End | : | Visual Studio XML |
| Back End | : | PSQL Server |
| Language | : | Python Programming |

## Hardware Environment:

| | | |
|---|---|---|
| Processor | : | Intel Core i5 4th Generation |
| RAM | : | 4GB |
| SSD | : | 256GB |
| LAN | : | ENABLED |

# V. PROBLEM DEFINITION

According to Nepal telecommunication authority, Nepal's internet penetration rate is 63% as of 2018. With this increasing number, the responsibility of network monitoring has increased for network and security professionals. They are highly dependent upon the traditional packet sniffer tools like Wire shark, tcp dump. However, the data provided by such tools is very large and sometimes even

network professional have difficult time to filter and get the required result. Also, these industry standard tools require sound knowledge of networking protocols which makes them unsuitable for laymen and end users.

## PROPOSED SOLUTION

After, reviewing some of the problems from diverse range of internet background, we can conclude that http protocol is excessively used in Nepal's internet space for transferring web credentials (Shodan, 2019). This definitely justifies that majority of end users are unknown about basic security concepts about the ssl and encryption. Similarly, majority of data provided by traditional packet sniffer are almost useless. In order to capture a basic cookie or password in network packets traditional tools provide data of whole seven layers. It is quite difficult to filter if the sniffers are operated for a long time to get secret confidential values. Hence, packet sniffer to sniff secret credentials can be a handy tool for network and security professionals either for troubleshooting or penetration testing purpose.

# VI. FEASIBILITY STUDY

A feasibility study is a preliminary study undertaken to determine and document a project visibility also known as feasibility study

The term feasibility study is also used to refer to the resulting document. The results of this study are used to make a decision whether or not to proceed with the project. If it indeed leads to a project being approved, it will-before the real work of the project succeed. It is an analysis of possible alternative solutions to a problem and recommendation on a best alternative. It, for example, can decide whether an order processing be carried out by a new system more efficient than the previous one.

A feasibility study is an important part of creating a business plan for a new enterprise, since it has been estimated that only one idea in fifty is commercially viable.

If a project seems to be feasible from the result of study, the next logical step is to proceed with it. The research and information uncovered in the feasibility study will support the detailed planning and reduce the research time.

## a. Operational feasibility:

There are two aspects to check the operational functionality of the system. One is of the technical performance and another is its acceptance. 32 Technical performance deals with the fact whether the system produces the correct and timely output required by the end user. Application can be tested with variable inputs and the outputs as desired. This system will help the user to enter the new suspect details, view all suspect details, add suspect's to most wanted list, get location of suspect, and edit suspect's details.

## b. Technical feasibility:

Technical feasibility refers to the ability of the process to take advantage of the current state of the technology in pursuing further improvements. Our project is developed in Python. This application can be accessed by a desktop or laptop. Ms Access Database serves as backend as well as frond end in few forms.

## c. Economic feasibility:

The economic feasibility is carried out to know the financial viability of the project in terms of the amount of investment in the system and the output expected. It also includes the cost involved at the time of development of the system as well as future cost in terms of maintenance and other miscellaneous expenditure. Since the hardware and software requirements are easily available at affordable cost, cost of development is very low.

# VII.DESIGNING

## a. INTRODUCTION TO DESIGNING PHASE

Software engineering is defined as a discipline whose aim is the production of fault free software that satisfies the user's needs and that is delivered on time

and within budget. In order to achieve this goal, appropriate techniques have to be used in all phases of software production, including specification (analysis), design, and maintenance. Software engineering addresses all phases of the life cycle and incorporates aspects of many different areas of human knowledge, including economics and the social science.

## b. DESIGNING PHASE

The specification undergoes two consecutive design processes. First comes architectural design in which the product as a whole is broken down into components called modules. Then each module is designed; this process is called detailed design. The two resulting design documents describe, “How the product does it”. The design phase is the art of designing the system components and interrelationships between those components in the best possible way to solve some well specified problems. The aim of the design phase is to map the functional requirement of the application to the hardware and software environment. The results of design phase are programming specifications and plans for testing, conversion, training, and installation. In addition, the design phase may result in the prototyping part, or all of the application.

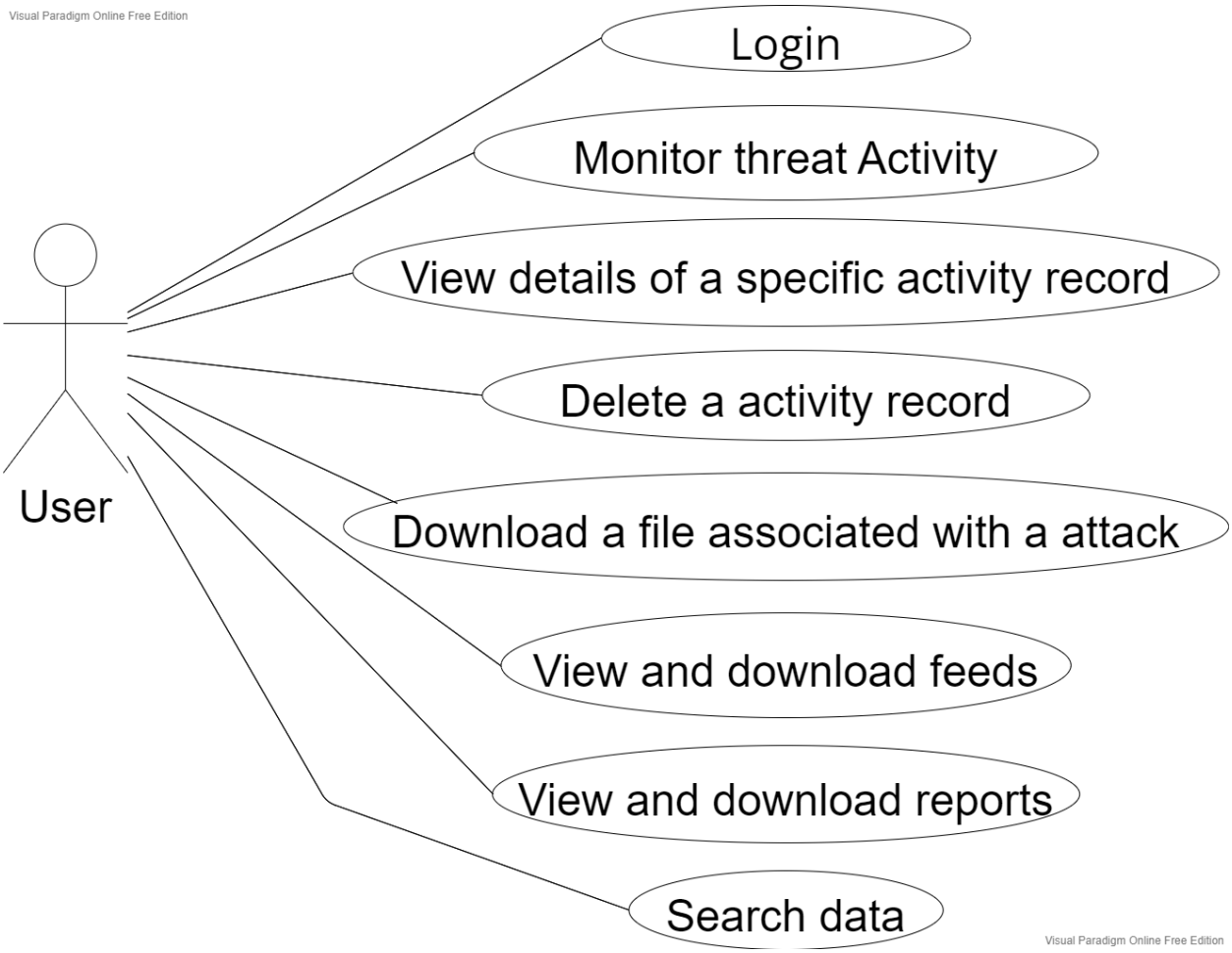
During the design phase, the software engineering team creates documents and verifies them. The software architecture part identifies and defines programs, modules, functions, rules, objects, and their relationships. The exact software architecture depends on the method used during the design phase. Software components and module specifies detailed contents and functions of software components, including, but not limited to inputs, outputs, screens, reports, data, files, constraints, and processes. Interfaces stare the detailed contents, timing, responsibilities, and design of data exchanged with other applications or organizations.

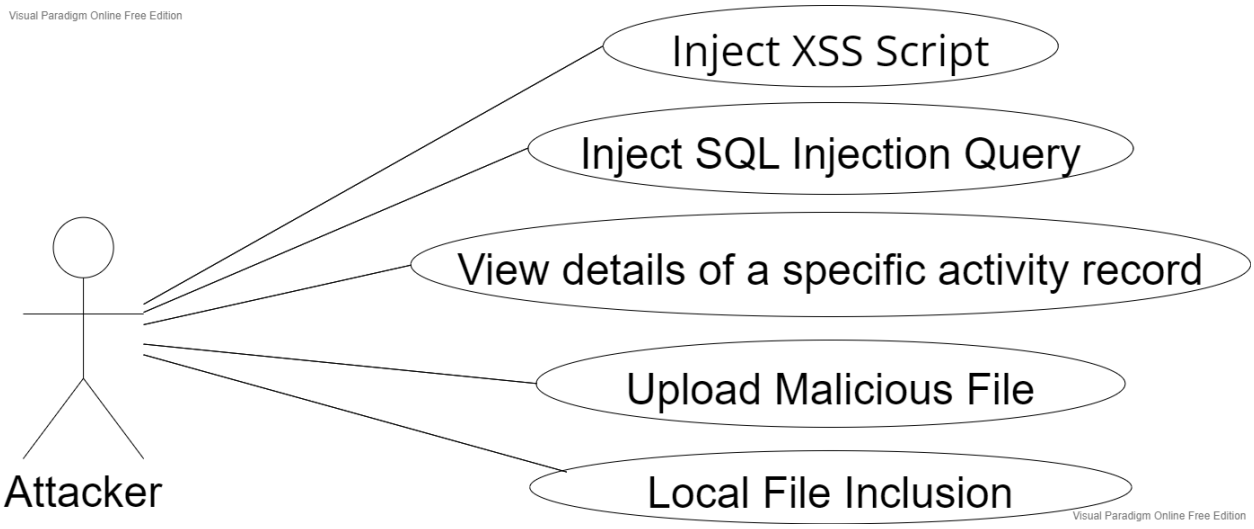
There are various diagrams which accomplishes the designing phase. Out of which we have implemented few for the project.

These are the following diagrams, which are been briefly explained as well as the corresponding system diagrams are associated with it.

## c. USE CASE DIAGRAM

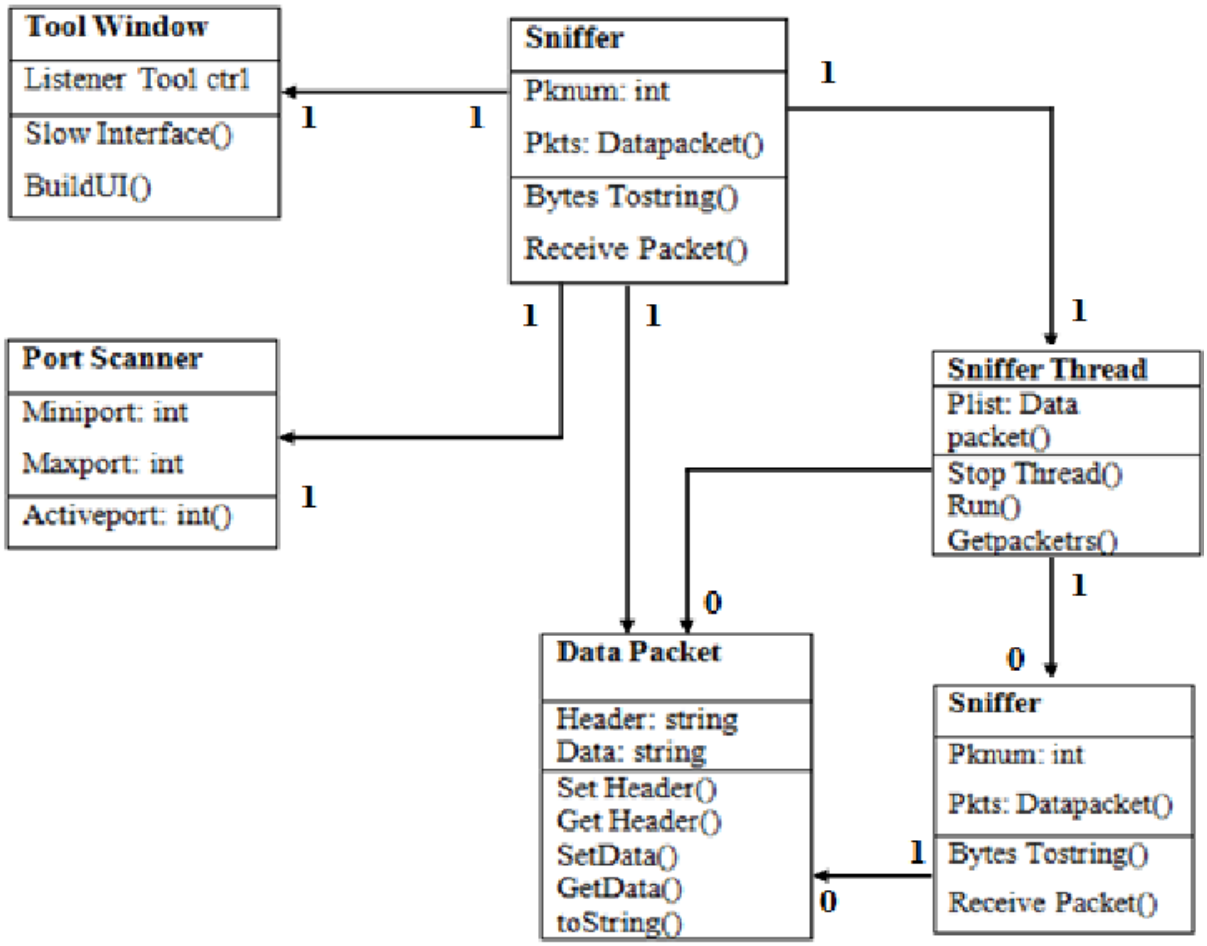
A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.




Visual Paradigm Online Free Edition
Inject XSS Script
Inject SQL Injection Query
View details of a specific activity record
Upload Malicious File
Local File Inclusion
Attacker
Visual Paradigm Online Free Edition

## d. CLASS DIAGRAM

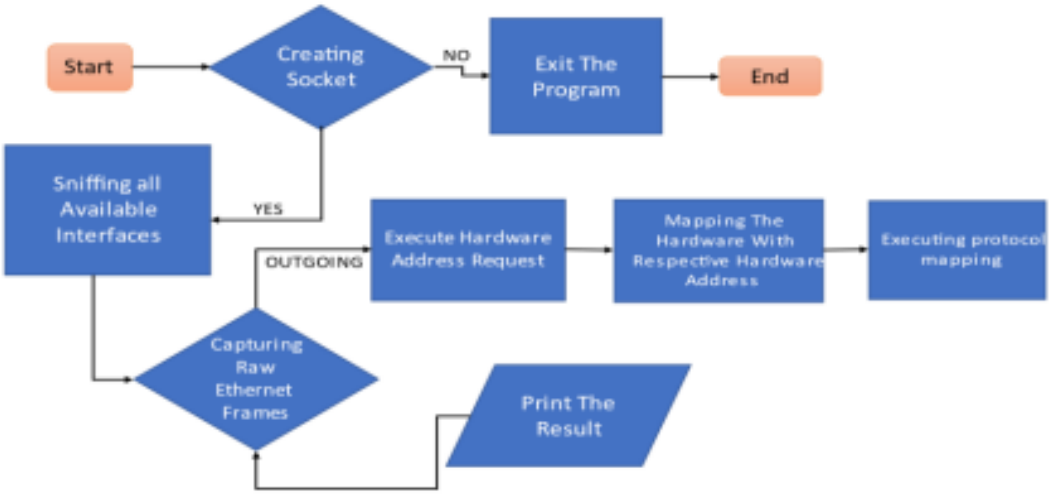
In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. Explanation of the Class Diagram with respect to City Surveillance System: The pivot classes which will be created are, the databases will also be managed for the same.




## e. FLOW CHART

A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation can give a step-by-step solution to a given problem. Process operations are represented in these boxes, and arrows

connecting them represent flow of control. Data flows are not typically represented in a flowchart, in contrast with data flow diagrams; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.




*FlowChart(capturing incoming packets only)*