NETWORK POLICE

(Traffic Identification for Segregation & Analysis of Suspicious Encrypted VoIP Traffic-Grey Traffic)

By

Ayesha Khaliq

Ammara Sajid

Sarmad Nisar

Osamah Karim

Submitted to the Faculty of Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

JULY 2011

ABSTRACT

NETWORK POLICE

Telecom policies in every country are determined by National Telecom Regulator (NTR). In many countries around the world, NTRs has imposed call termination taxes on national calls as well as international calls landing in that country. In many third world countries, every year almost 90 % international traffic by-passes regulatory check, causing a great revenue loss to the country. The use of illegal gateway to bypass the Voice Clearing Houses to terminate international traffic using VoIP gateways, GSM / local line branch exchanges or other related equipment is called “Grey traffic". Use of encryption techniques is the easiest way to hide the VoIP traffic from any Clearing house. In this paper we have described a way to identify this illegal activity over the Internet. We have used traffic analysis techniques coupled with statistical anomaly based Intrusion Detection system and behavior analysis to detect, segregate and qualify VoIP traffic into different categories (targeted, suspicious and Grey). We have performed traffic analysis for IP, IPSec, PPtP, TCP, UDP, TLS, SSL and any application layer encryption protocol for VoIP.

Network Police is a robust and dynamic application for PTA and other Telecommunication Companies facing the wrath of Grey Traffic. It has extensively and successfully been tested under stress conditions and results have been double checked through rigorous testing.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

ACKNOWLEDGEMENTS

First and foremost, we are eternally grateful to Allah Almighty for giving us the strength and courage to undertake and complete this project.

This project would not have been complete without the constant guidance and persistent effort of our project supervisor Maj. Ch. M. Asim Rasheed. We would like to thank him for his continuous support, inspiration and motivation. We would like to show our gratitude to the administration of Department of Computer Software Engineering, MCS for their help and support.

We would like to offer our admiration to our families for their patience and tolerance to bear with us in this time of hard work.

We would like to show our deepest appreciation for our parents for having unflinching faith in us, for their continuous payers and for supporting us in every way possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction

CHAPTER 1

1.1 Introduction

This chapter provides an introduction to its readers, future developers, software testers and project evaluators about the problem statement, proposed solution for this project, an overview of the system to be developed, motivation behind doing this project, scope of work in order to accomplish project objectives, research domain in which this project falls and summary of project objectives and goals.

Network Police is a robust and dynamic application for PTA and other Telecommunication Companies facing the wrath of Grey Traffic. It is an effective and efficient method of dealing with the problem.

1.2 Problem Statement

There is a prevalent problem of illegal internet traffic landing in Pakistan which is incurring substantial monetary loss not only for the operators and private companies but also for the National Telecommunication Regulator of Pakistan (PTA). The problem statement can be defined as to develop a system capable of detecting Grey traffic over the network without hindering the normal flow of the traffic. The end goal is to assist PTA to identify internet addresses involved in international Grey traffic (VoIP) termination in Pakistan using encryption.

1.3 Motivation

PTA has announced that taxes worth billions of rupees are evaded every year by illegal VoIP traffic activity. International VoIP traffic is being illegally terminated in the country, resulting in tax evasion and business loss. As per PTA statistics, approximately 6 billion minutes of international voice calls are terminated in Pakistan every year. Currently, 90 % international traffic by-passes regulatory check, the PTA calls such traffic “Grey Traffic”. Grey traffic is a menace that not only incurs revenue loss to national exchequer but also denies level playing field to LDI (Long Distance & International) operators who are the only legitimate carriers to terminate and originate international traffic to/from telecommunication networks, such as Land-line or Cellular phones. PTA has taken the initiative of developing technical solution to curtail Grey traffic. **Network Police** project aims at using statistical techniques for categorizing the suspicious encrypted 'Grey Traffic', hence directly benefiting Motherland Pakistan with a significant revenue increase.

1.4 Proposed Solution

The proposed solution consists of a stand-alone module that detects encrypted packets over the internet and analyzes their behavior over a period of time to categorize them. The project deals with mirroring traffic and then short-listing packets of interest. The analysis is made based on similarity of captured traffic with the standard observed VoIP traffic patterns. Packet features will be observed and decision will be made based on these statistical measures. The end goal is to mark IP's using encrypted VoIP traffic as suspicious.

1.5 Scope

As per PTA statistics, approximately 6 Billion international minutes are terminated in Pakistan every year. As a legal means, PTA assigns LDI licenses to different operators to originate and terminate international voice traffic from / for telecommunication networks. Every minute terminated in the country brings 10.5 cents, which is shared among Government (as tax) and operators. PTA has installed ICH at Internet landing stations of Pakistan Internet Exchange (PIE) and Trans World (TW-1) to monitor international traffic. Considering the involvement of huge finances, many culprits (including legal operators) try to bypass ICH to hide amount of actual traffic terminated in Pakistan. Such termination, which causes huge revenue loss to state, is known as “Grey Traffic”. As per PTA statistics, currently, 90 % of international traffic bypasses regulatory check.

1.5.1 In Scope

The simplest technique to bypass ICH is use of encryption for VoIP at Network layer, Transport layer or even at Application layer. Network Police will provide an add-on module for ICH to identify encrypted VoIP traffic at Network layer, Transport layer or Application layer. The project will work to identify and segregate encrypted VoIP traffic from normal traffic on basis of statistical behavior of the packets. Features of the packet are extracted and behavior of packet is determined by applying various logic algorithms.

1.5.2 Out of Scope

Conclusions on whether the traffic is 'Grey', 'Suspicious' or 'Targeted' are the end result of the project. The steps taken against such suspicious users are not in this project’s scope.

1.6 Diagrammatic Overview

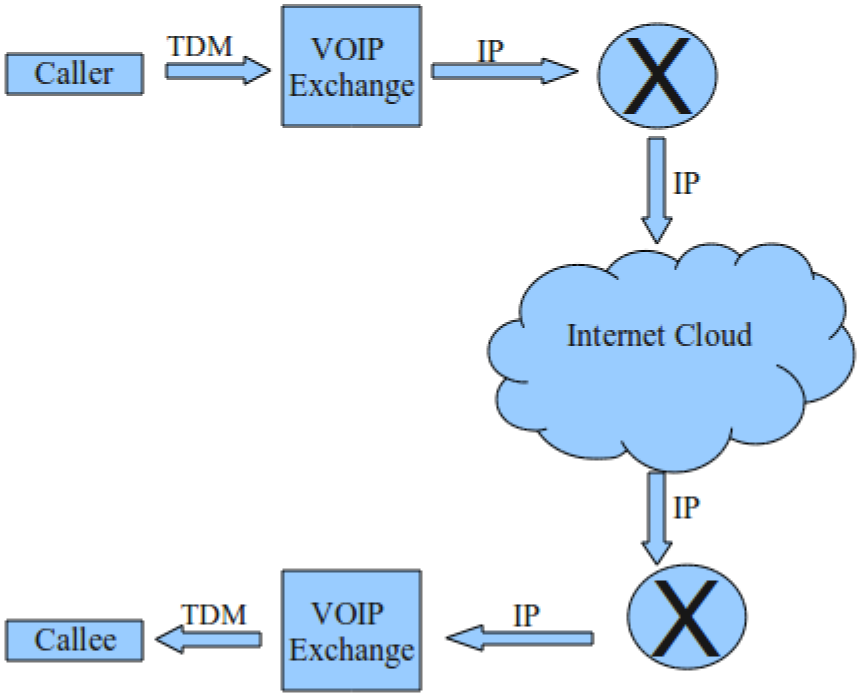


Figure 1-1: Diagrammatic Overview

1.7 Terms and Abbreviations Used

PTA – Pakistan Telecommunication Authority

ICH – International Clearing House

VoIP – Voice over Internet Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

SIP – Session Initiation Protocol

IP – Internet Protocol

IPSec – Secured Internet Protocol

TLS – Transport Layer Security

SSL – Secure Socket Layer

SSH – Secure Shell

ISP – Internet Service Provider

1.8 Summary

This chapter described the details about problem statement its objectives and project goals to be implemented. Moreover it highlighted an overview of the project scope of work, its structural diagram and the key parameters which needs to be considered. Research work related to the project and the main objectives and structural analysis has been discussed.

Literature Review

CHAPTER 2

2.1 Introduction

In this chapter current technical approaches to related problems and their shortcomings are discussed. Also, current approaches dealing with problems related to different modules of the project are discussed for in depth knowledge of the problem domain.

2.2 Network Police Unique Features

Network Police extracts information from packet headers and detects encryption at Network, Transport and Application Layers. The captured traffic is then matched with patterns and standards established for VoIP traffic flow over the internet. The analyzed traffic is categorized, where each category shows the level of suspicion. This way, even the VoIP traffic disguised by encryption can be detected. Previously the problem was approached by imposing taxes on unencrypted VoIP traffic, where as encrypted, disguised VoIP traffic passed through unchecked. Network Police identifies encrypted VoIP traffic by on calculating probability depending on the statistical behavior of encrypted traffic.

2.3 Current Approach to the Problem

International Clearing House (ICH) was set up by PTA in 2008 to check the illegal VoIP traffic over the internet. The mitigation component of the facility has been automated wherein any IP address carrying voice but not authorized to do so would be automatically blocked. ICH checks for packets which have their VoIP flag on but do not pay the due taxes to the government. It stops such traffic from landing in Pakistan.

2.3.1 Shortcomings

Illegal vendors have found a way around the detection policy imposed by ICH. VoIP traffic is encrypted such that it can easily pass the ICH without getting recognized as carrying VoIP inside the encrypted packet. This way illegal VoIP traffic can still land in Pakistan, evading taxes and resulting in huge losses.

2.3.2 Comparison between ICH and Network Police

Network police identifies and detects disguised VoIP traffic over the network rather than looking for packets with their VoIP flags set to ‘On’. The detected encrypted traffic is analyzed over a period of time and its statistical behavior is matched to that of standard VoIP traffic. This way, illegal VoIP traffic which is encrypted to by-pass ICH can be detected and the IP addresses involved can be efficiently and effectively identified.

2.4 Existing Approaches to Related Problems

2.4.1 Packet Analyzer

Packet sniffers like Wireshark have been developed and are already in wide use in the market. These packet sniffers can capture packets and scan their headers to extract relevant information and can intercept and log traffic passing over a digital network or part of a network they might have graphical front ends with information filtering or sorting. They can separate traffic based on standard protocols used by the packets.

2.4.1.1 Limitations

These packet analyzers only work on standard protocols defined but not on propriety protocols that may be used by an end user. This is a major drawback since end users with malicious intentions may use non-standard propriety protocols which will not be detected. Moreover, such analyzers operate on per packet basis, rather than giving session information and type of flow. Network Police is designed to overcome this limitation so that propriety protocols can be easily incorporated into the system at any time.

2.4.2 Deep Packet Inspection

Deep packet inspection examines the data part of the packet (and optionally the header) as it passes the inspection point and may decode it to detect protocol non-compliance, viruses, worms, intrusion or for the purpose of collecting statistical information. It is often used by service provides to gain statistical information about the users, how one user's pattern varies from the other, etc.

2.4.2.1 Limitations

Deep Packet Inspection normally inspects the data/payload of the packet and derives results based on these inspections. This might be a drawback since encryption in a packet will require decryption to effectively read the data. This incurs a lot of overhead in term of processing since decryption is a resource expensive process, especially in cases like grey traffic where all international traffic has to be analyzed and monitored. This would incur very great processing and storage overheads which will affect the end result. Another limitation can be the fact that it collects statistical information that is useful to differentiate one user pattern from another but does not develop standard protocol patterns. Network Police is designed to overcome these hurdle to get accurate and useful results. VoIP traffic pattern is established and anomalies to this developed profile are observed and detected. Also, Network Police does not in any way depend on decryption of packets but focuses solely on statistical analysis.

2.4.2 Network Monitor

Network monitors can help manage networks efficiently. They can help visualize network traffic, log network activity and network throughput, create history reports and calculate traffic over time. Traffic between days and specific timing can be calculated, graphs for network usage can be made, sources for any network slow-down can be found in a matter of seconds.

2.4.2.1 Limitations

Network Monitors can log information and filter it according to time and dates but it does not use profile based matching to detect a certain type of traffic. It gives overall network statistics and usage but a single protocol might not be identified. Also, encryption layer/level is not detected which can prove to be an important information for specific protocol detection.

Software Requirement Specification

CHAPTER 3

3.1 Introduction

This chapter lays out the project functionality and the work lay-out. The project perspective and its importance with respect to present environment are briefly mentioned. The project's learning objectives are also discussed. Functional and non-functional requirements form a major section in the chapter. These requirements give an overview of the functions that project will accomplish.

3.2 Learning Objectives

This project will provide in-depth study of following:-

- Voice over IP
- Network Programming
- Network based Application Design
- Network Sniffing
- C Language
- Linux Environment

3.3 Functional Requirements

A **functional requirement** defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. **Network Police** provides the following functions:

3.3.1 Packet Capture and Header Information Retrieval

Traffic is captured from the core ingress router for any network, such as ICH and information from packet header is extracted and logged (stored) in a raw file format with the necessary fields of the packet’s header enlisted.

3.3.2 Segregation of Encrypted VoIP Traffic

A process model is developed against the network model and encrypted VoIP traffic is segregated. Encryption on Network, Transport and Application Layer is detected and encrypted packets are segregated from unencrypted traffic.

3.3.3 Session Maintenance

Maintaining multiple sessions is one of the distinguishing features of VoIP traffic, therefore, maintaining necessary session information is a vital part of the project. Network Police provides mechanism for extracting and analyzing relevant information from packet headers (like Source and destination IP addresses and Ports) to maintain session information for IP pairs.

3.3.4 Traffic Analysis based on Statistical Behavior

The retrieved information is observed and analyzed over a period of time. Long term behavior of the captured traffic is analyzed based on statistical measures.

3.4 Use Case Diagram

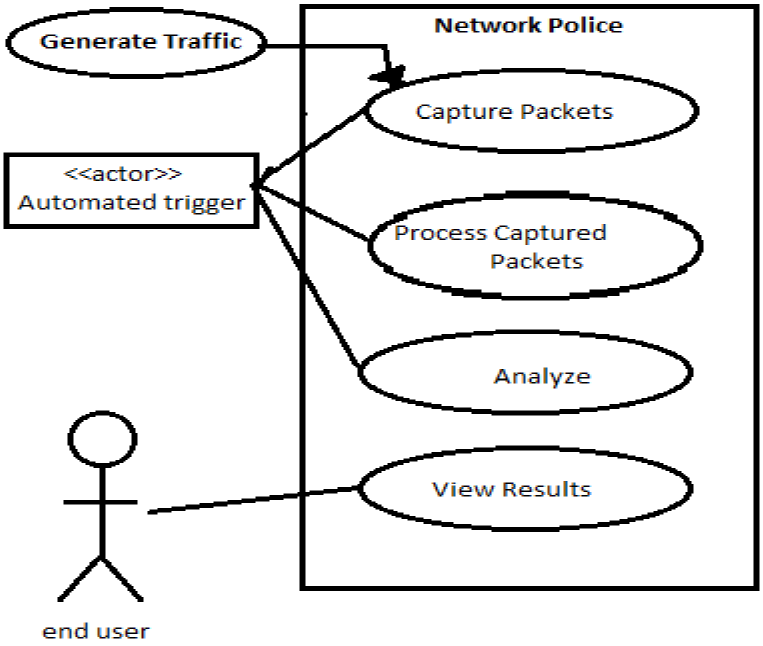


Figure 3-1: Use Case Diagram

3.4.1 Actors

There is only a primary actor involved in this scenario. The actor (end user) can use the system by initiating it and viewing the end result, the rest entire system is automated and does not need user interaction.

3.5 Non-Functional Requirements

A **non-functional requirement** defines a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

Non-functional requirements of Network Police include:

3.5.1 Efficiency

The project is intended to be efficient and fast enough as to not create bottle necks in the process. It will compute a large number of calculations as a fairly large number of packets are captured and stored.

3.5.2 Accuracy

The project is also intended to be accurate so that people with non-suspicious traffic activity are not affected by the whole process.

3.5.3 Reliability

The system should be reliable and must not break down under stress or greater traffic load.

3.5.4 Availability

24 hours availability is necessary as traffic may be flowing at any time of the day. Missing packets might result in inaccurate results.

3.5.5 Interoperability

The system is designed such that there is ease of use in integrating it with other modules/systems, if necessary.

3.5.6 Maintainability

The system will easily accommodate changes with evolving technologies. It will be designed so that further protocols can be incorporated into the system at a later time.

3.6 Operational Environment

The project is aimed to be deployed at the International Clearing House of Pakistan. It will mirror traffic from the ICH and then run, analyze and segregate the captured traffic. It will mirror live traffic but information extracted is stored in form of files on the system that the project runs on. This helps in evaluating traffic behavior and drawing conclusions at a later time (offline) rather than on spot analysis which can hinder/burden traffic flow.

3.7 Hardware Required

1- 6 x Computers for processing purposes

2- 1 x Router for traffic capturing

3.8 Tools & Technologies Requirements

1- Linux based GCC for C code compilation (Network Programming)

2- QT Creator for GUI Development

3.9 Summary

The main tasks in terms of requirements specification need to done in order to accomplish desired project goals that have been highlighted in this chapter. These requirements have given the clear understanding of system design which can be implemented for the Network Police application development. Graphical interface measures need to be adopted while designing the proposed Network Police underlying structure. Non-functional requirements mentioned in this chapter have reflected upon us a clear picture of performance and quality attributes which must be considered while designing this system to ensure its efficiency, reliability, integrity, availability and security.

System Design

CHAPTER 4

4.1 Introduction

Software Design specification for **Network Police** comprising of key processing functionalities, high level and low level design architecture, graphical user interface design have been discussed in this chapter. It also focuses on modular design for the system. Furthermore it describes major design constraints related to data design, graphical interface and schema design of Network Police.

4.2 System Architecture

System architecture can be defined in two ways; high level architecture and low level architecture

4.2.1 High Level Architecture

The high level architecture shows the overall design for the system. It covers the 5 modules of Network Police and their basic dependencies.

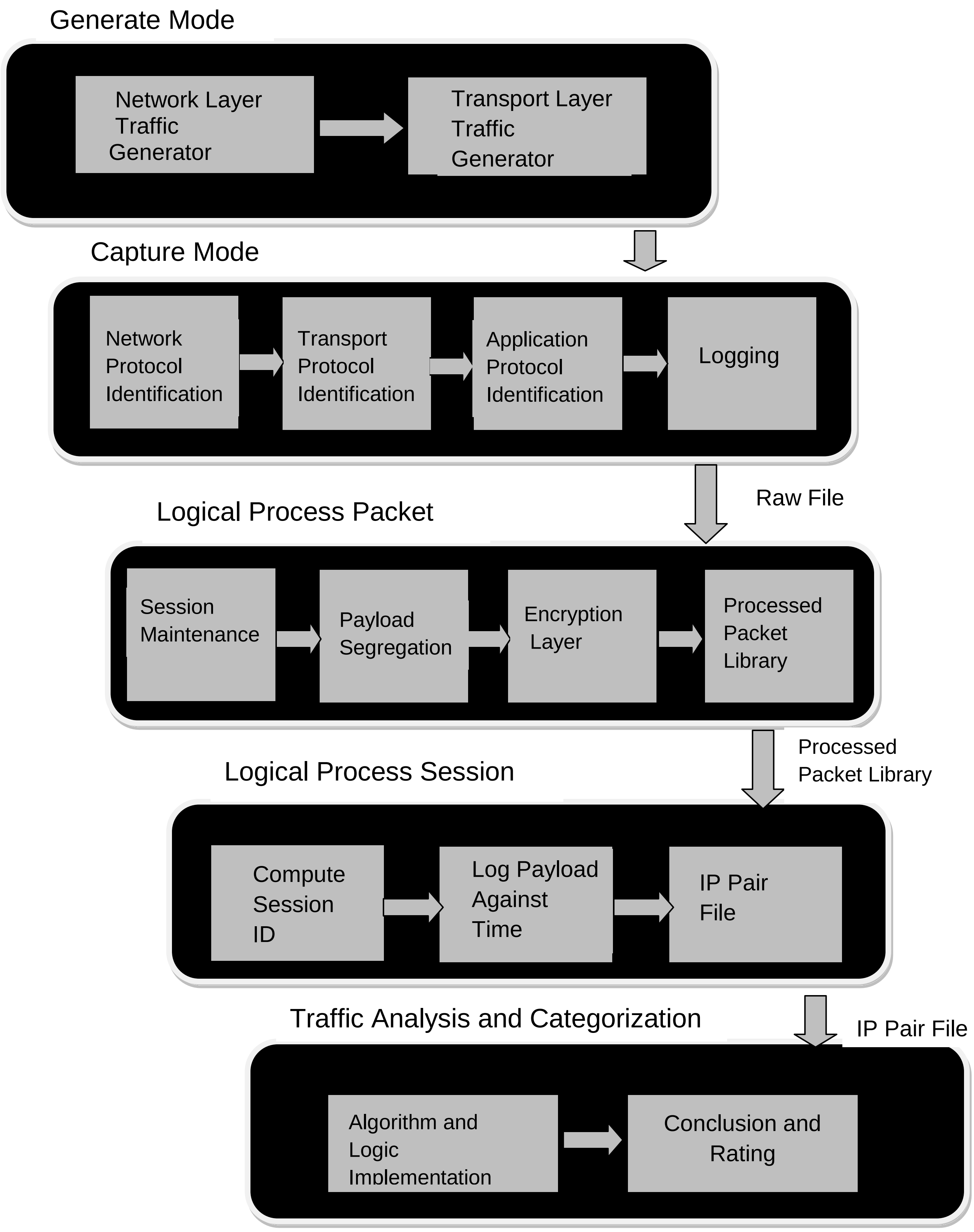


Figure 4-1: System Architecture

4.2.2 Low Level Architecture

There are various types of architectures to represent the context. Basically the architectures can be divided as system architecture and application architectures. The system architecture has been described above as Architecture Context diagrams. There are many ways to represent application architectures such as call and return, pipes and filters, Object oriented architecture etc. In our case modular architectural design is shown using Data Flow Diagrams.

4.2.2.1 Data Flow Diagrams

4.2.2.1.1 Level 0

The design shows the basic flow of the entire project. The main modules and their relationships, along with their inputs and outputs are represented.

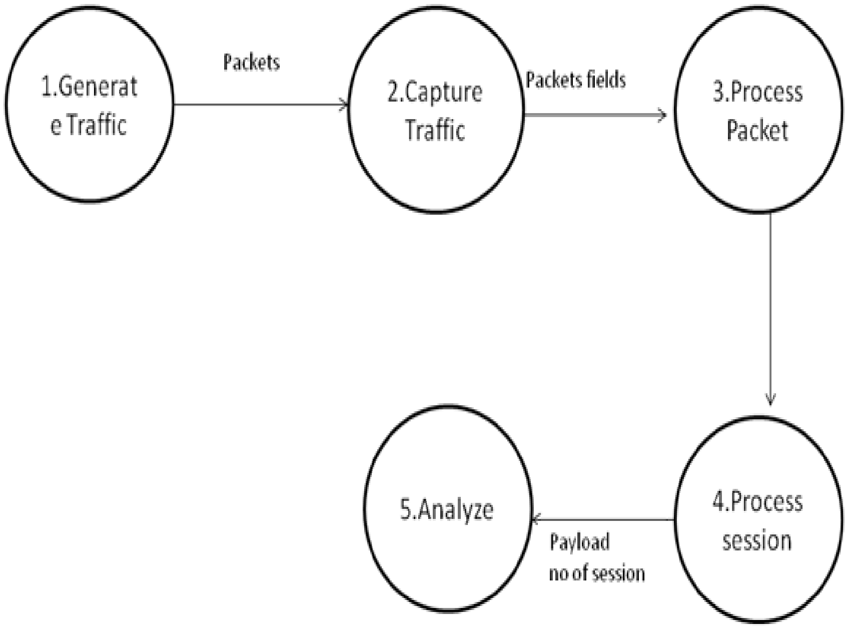


Figure 4-2: Level 0 DFD

4.2.2.1.2 Level 1

4.2.2.1.2.1 Capture

Packets are captured by opening port on the device and header of captured packet is parsed and maximum possible information is extracted. The extracted information is logged into Raw File.

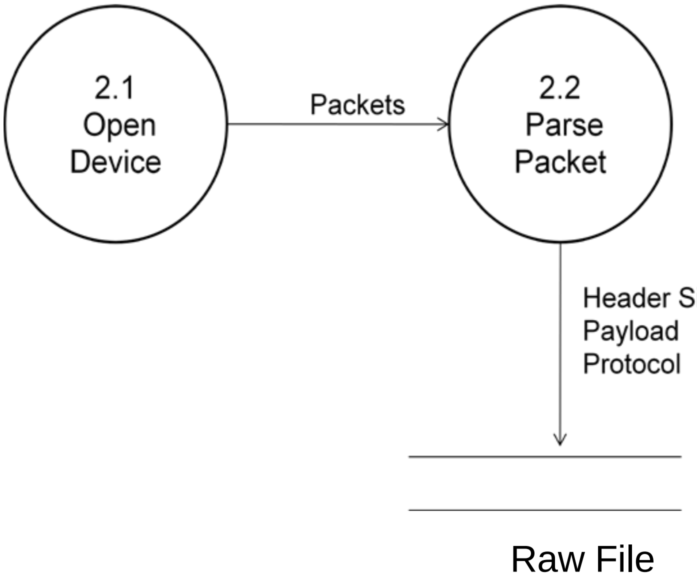


Figure 4-3: Level 1 DFD, Capture

4.2.2.1.2.2 Process Packet

The output of capture module acts as input to process packet module. Session numbers are computed based on IPs and Port numbers. The computed session number along with the last known header aids in payload computation for each session, and this information is then logged.

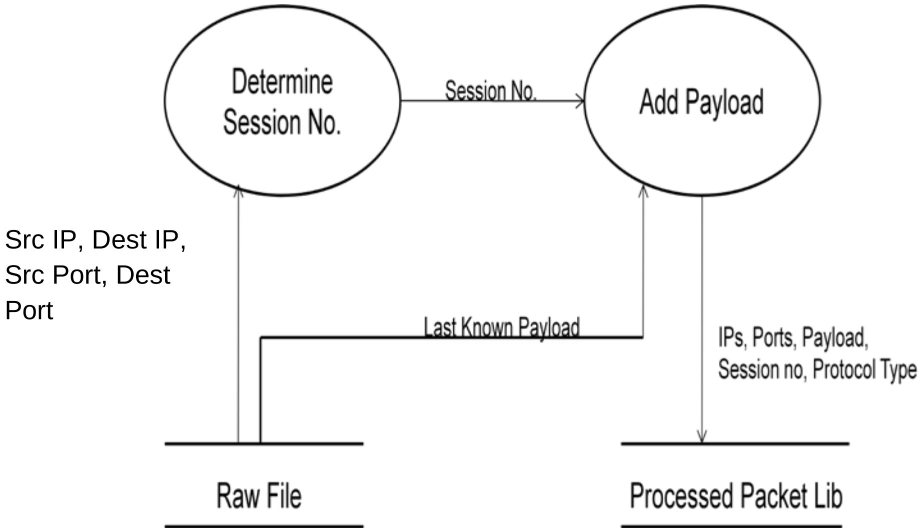


Figure 4-4: Level 1 DFD, Process Packet

4.2.2.1.2.3 Process Session

File name is computed at run time based on the data read from the file. The session and payload from process packet module acts as input to compute session ID. Payload is computed based on information extracted from both processed packet library and IP Pair file.

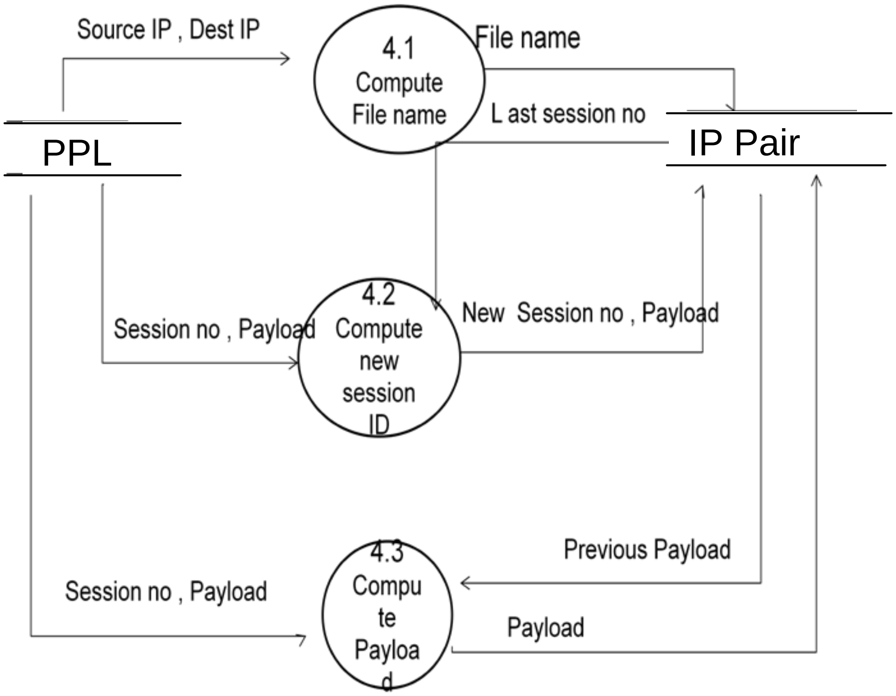


Figure 4-5: Level 1 DFD, Process Session

4.2.2.1.2.4 Analyze

Packets are analyzed based on defined rules. Each rule is separately applied and rating is done based on specific weight age of each rule.

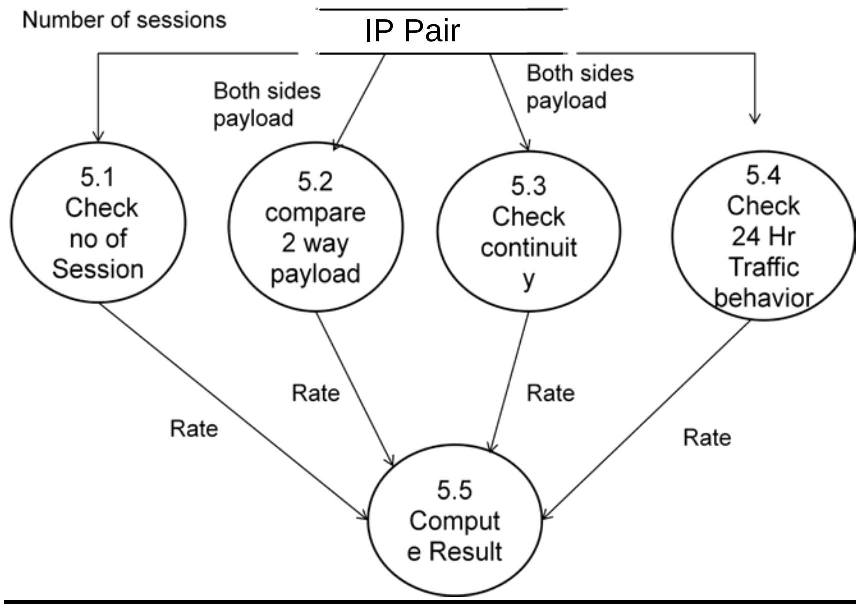


Figure 4-6: Level 1 DFD, Analyze

4.2.2.1.3 Level 2

4.2.2.1.3.1 Analyze

Information is extracted based on header fields for each layer. Header sizes, payload sizes, protocols on each layer, next protocol (in case of network layer) etc. are among the many fields extracted.

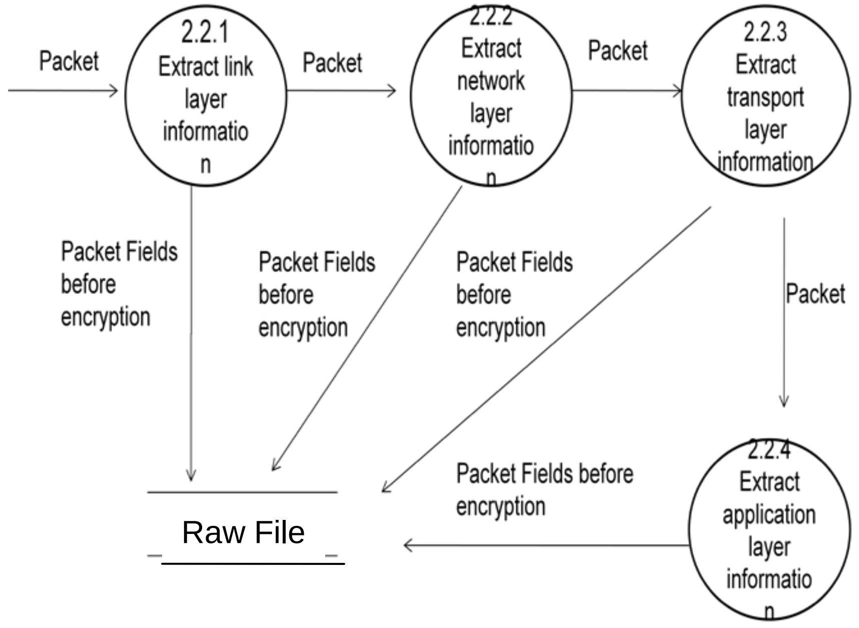

Packet
2.2.1
Extract link layer informatio n
Packet
2.2.2
Extract network layer informatio n
Packet
2.2.3
Extract transport layer information
Packet Fields before encryption
Packet Fields before encryption
Packet Fields before encryption
Packet
Packet Fields before encryption
2.2.4
Extract application layer informatio n
Raw File

Figure 4-7: Level 2 DFD, Analyze

4.2.2.2 Flow Charts

4.2.2.2.1 Processed Packet

This flowchart represents the data flow on pseudo code level. Data is read from file (output of module 1) and sessions are computed based on source IP, destination IP, Source Port and Destination Port. Last Known Header (LH) is extracted based on logical operations. Session maintenance is achieved and logged/updated.

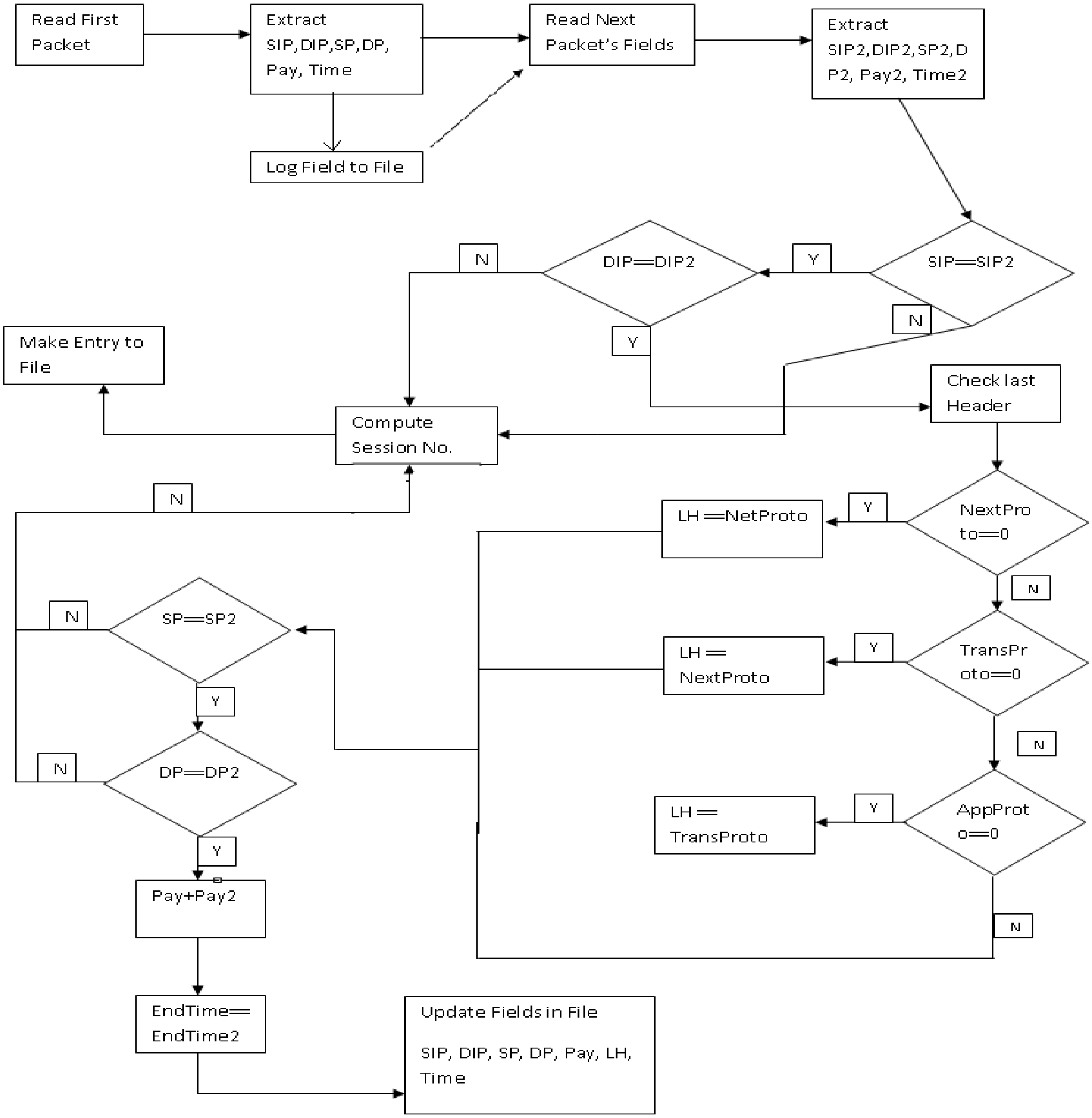


Figure 4-8: Flow Chart Process Packet

4.2.2.2.2 Processed Session

This diagram represents processing of session on pseudo code level. Data is read from the output file of module 2 and further processed. Payload is logged against time after regular intervals and session IDs are logged.

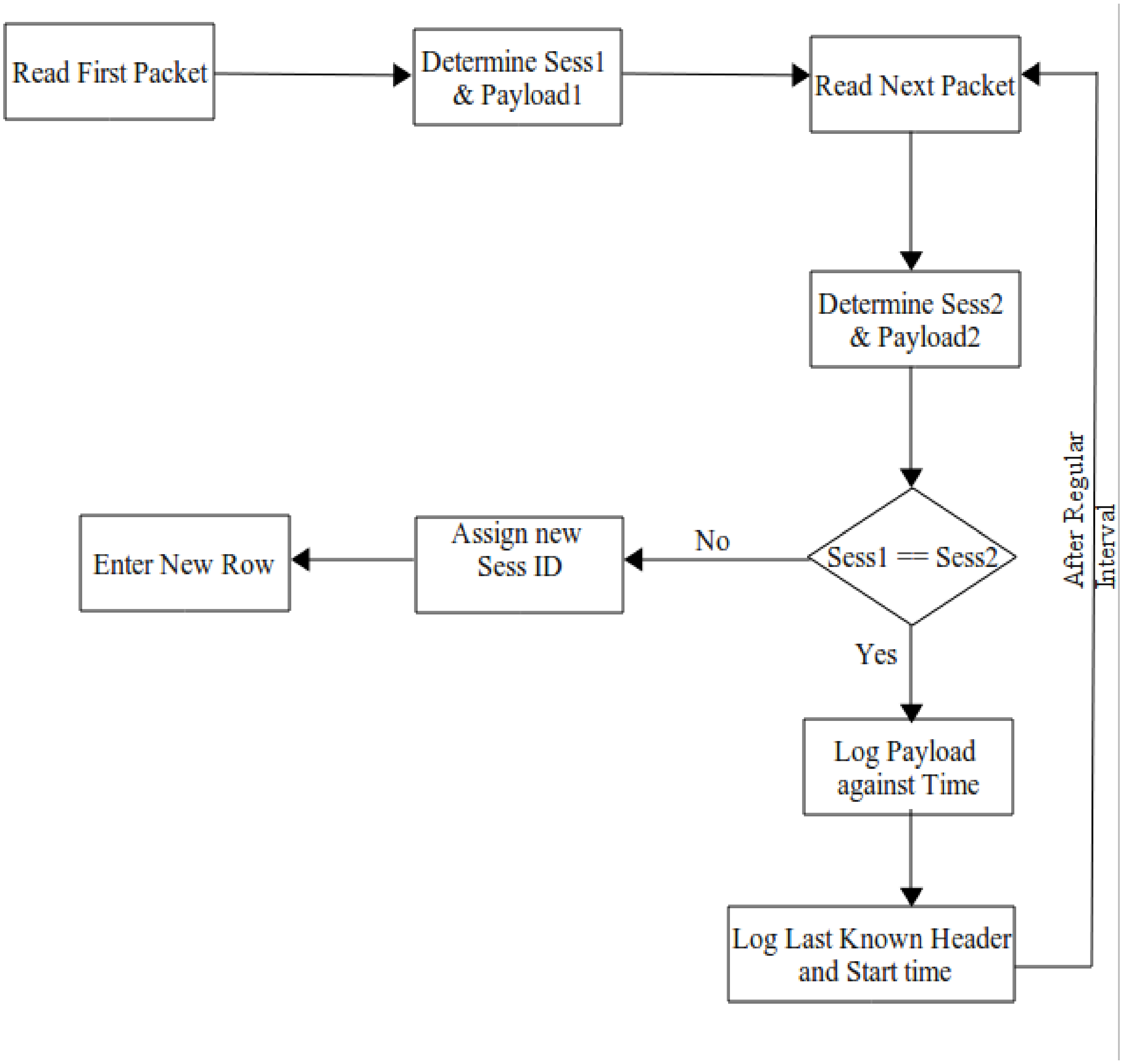


Figure 4-9: Flow Chart, Process Session

4.3 Graphical User Interface Design

The Graphical User Interface (GUI) for Network Police has been developed in QT Creator. It is a simple and user-friendly interface that shows a list of IP addresses and their categorization (low-suspicious, moderate suspicious, highly-suspicious or Grey). On selection of an IP address pair by the end-user, the interface displays the graphs for Payload and sessions against time for that particular pair.

4.4 Summary

Major design specifications needed to be adopted in order to implement the core functionalities of Network Police have been highlighted in this chapter. High and low level architecture provided the basic structure and approach to implement functions in accordance with the design requirements. Graphical interface design reflected an idea of developing aesthetic look and user friendly interface to enhance the productivity and usage of our interface. Furthermore this chapter has focused on development and implementation of each component to achieve the top priority functionality of system first.

System Implementation

CHAPTER 5

5.1 Introduction

Implementation details of the project have been discussed in this chapter. The coding for the system has been done in C language in GCC, Linux. As illustrated in the prior chapter, the major functionalities of the system are divided into different Modules for flexible development and integration of inter-dependant Modules. This chapter presents the implementation details of the project.

5.2 Implementation

System implementation has been highlighted in the Figure 5-1 showing the overall architecture and design implementation of Network Police.

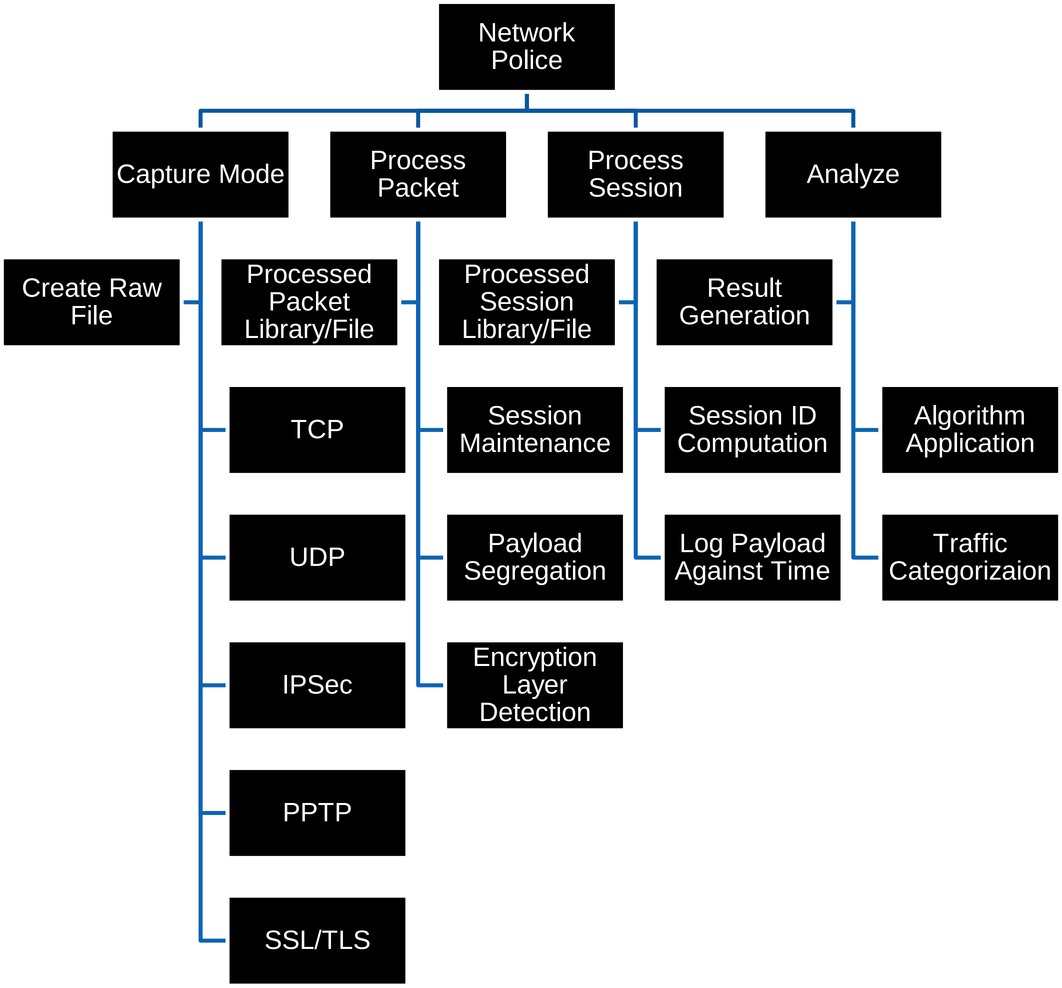


Figure 5-1: Implementation Overview

5.3 Implementation Language

The implementation language chosen for the development of the system is C. it is preferred over other languages because

1- It provides a low-level access to memory which provides language constructs that map efficiently to machine instructions. This is a major benefit since the project is designed to be implemented on routers.

2- It encourages cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code.

5.4 Distribution of Modules

5.4.1 Module 1 – Generation Mode

Artificial Network Traffic generation module is used to simulate real-time network traffic. It is not a formal part of the project, but is incorporated for the purpose of generating base line conditions for protocols that are included into the system, till access to real-time traffic is available. For testing purposes, traffic using different protocols is generated.

5.4.2 Module 2 – Capture Mode

The module is designed to capture real-time network traffic packets over the internet connected router to our system. This is the only module that runs online. Real-time traffic is captured using PCAP library. This is achieved by putting the system in promiscuous mode. The traffic landing on the main router (ICH) is mirrored and captured. Every packet is parsed thoroughly and maximum retrievable information from the headers is extracted.

Headers for the highest accessible layer in each packet are scanned. Ethernet Packet details (like packet header length) are extracted using built-in PCAP functions. The extracted details are then checked for link layer information. This returns the “ethertype” field form ethernet header. It checks the “ethertype”. If it is “IP”, the packet’s network header is scanned to determine the “Next Protocol” or “Protocol ID” field which contains information about the next protocol used. This can be a transport layer protocol or another network layer protocol, depending on whether encryption at network layer is applied. Network Header for “IP” is included in <netinet/ip.h>.

The protocol id then takes the processing to a next level. If encryption at network layer is applied (e.g. IPSec) then the protocol id gives a network layer protocol, but if network layer encryption is not done, then the processing goes to the next layer i.e. Transport Layer.

At Network Layer “IPSec” is checked. This includes the “Mode” in which traffic flows (transport or tunnel) and the encryption type used (AH or ESP). At Transport Layer TCP and UDP are checked. Also, SSL and TLS are catered for in the project.

The extracted fields computed by Capture Mode include

- Time
- Data link
  - Header Size
  - Payload Size
- Network
  - Protocol Type
  - Payload Size
  - Header Size
  - Source IP
  - Destination IP
  - Next Protocol
  - Next Protocol Header
  - Next Protocol Payload
  - Next Protocol Type
- Transport
  - Transport Protocol
  - Source Port
  - Destination Port
  - Payload Size
  - Header Size
- Transport Security
- Sequence Number
- Acknowledgment Number
- Application
  - Application Protocol
  - Payload Size
  - Header Size
- Application Reserved

Table 5-1: Fields Included

Capture Mode logs this retrieved information from headers into a “Raw” File. This Raw file acts as input to the next module.

5.4.3 Module 3 - Process Packet Library

The input to Process Packet Library is the output of Capture Mode i.e. Raw File. The information logged in module2 is further processed in an off-line mode to get meaningful results. Process Packet Library maintains IP Sessions. For every entry in the Raw file, it checks the source and destination IP address and port numbers. It segregates payload based upon sessions. In a packet, if source and destination IP are same, then it is checked for the source and destination port. If ports are different from the ports of the previous entries, then the entry is considered to be a new session between an IP address pair. For every new session, the number of sessions between an IP pair is incremented by 1. For each session, the total payload and duration of the session is computed. Also, the encryption layer on the packet is found, based on the last accessed header. This gives the layer (network or transport) on which encryption is used in the packet.

Pseudo Code

```
For first entry in Raw File
For every entry(r) in raw file
             Match source and destination IP with every entry(s) of PPL
           If (r.IP matches s.IP)
                      {
                      Match ports
                      If(r.ports match s.ports)
                                {
                                Add r.payload to s.payload;
                                Add r.endtime to s. endtime;
                                Compute last accessed header;
                                }
                      else
                                {
                                Increment s.session number by 1;
                                Compute Start and end time;
                                Compute last accessed header;
                                }
                      }
```

```
else
    {
    Make new entry to PPL;
    Set start and end time;
    Compute last accessed header;
    }
```

Table 5-2: Pseudo Code

The output of module 3 is Processed Packet Library (PPL) which contains source IP, destination IP, source port, destination port, Session start time, session end time and encryption layer.

5.4.4 Module 4 - Process Session Library

The output of module 3, Processed Packet Library, is the input to module 4. Module 4 utilizes the information computed by Processed Packet Library and further computes useful results from it. It processes information based on sessions, rather than individual packets. For each IP pair identified in Processed Packet Library, Process Session Library creates a pair of files in comma separated values (.csv) format, one for each direction of traffic flow. E.g. if the IP pair is a.b.c.d and e.f.g.h, then two files will be created, one for traffic flow from a.b.c.d to e.f.g.h and the other for e.f.g.h to a.b.c.d. This is necessary since bi-directional flow is a primary criterion for checking VoIP flow.

The files contain all the sessions between an IP pair, encryption layer and the payload for each session logged against time. Start time of session is also logged. The file is updated after a fixed time interval and the payload is recorded. Processed Session Library reads from Processed Packet

Library at the end of each time interval and updates the payload according to time. If no payload flows, the entry for that particular time period is 0.

This also plays a role in easy manipulation of data for analysis purpose. Different sessions of the same IP pair may start and end at different times. Processed Session Library handles this effectively by noting the start time of the first session and then creating new entries accordingly i.e. if session 2 does not start for half an hour after session 1, the entries for session 2 are made 0 for that initial half an hour. This helps in computations regarding session durations and number of sessions maintained during a time interval.

The output of this module is a .csv format file (processed session library) which is easily handled for result computation in the next module.

5.4.5 Module 5 – Analysis

The input to this module is the output of processed session library. This module evaluates and categorizes traffic depending on standard VoIP behavior and flow pattern. Decision is made by checking if the traffic satisfies the VoIP flow pattern. Each criterion is assigned weight and probability of traffic being suspicious is computed based on these weights.

5.4.5.1 Criteria for Analysis

5.4.5.1.1 24 Hour Traffic Flow

This condition checks if the traffic between an IP pair flows for 24 hours. This is checked for the IP pair file. Each session in the file is parsed and number of vacant/empty time slots is calculated for the IP pair. The threshold for meeting this criterion is a minimum 22 hour traffic flow between the two addresses, since there are chances that the stop/pause in flow between different sessions in a same IP pair may coincidently be at the same time. Traffic flow can be checked by calculating the number of zeros in the session payload, i.e. the time (according to the specified interval) for which no data transfer took place.

5.4.5.1.2 Continuous Traffic Flow

This condition checks if the traffic flow between the IP pair is continuous or not. Continuous here means that the sum of payload for all session in an IP pair should not be zero, for duration longer than the threshold, during any time of the connection. Null or Zero in VoIP traffic flow indicates that a connection has been established but both end parties are not transferring any data (e.g. call on hold). This, coupled with the fact that multiple VoIP sessions have a very low probability of being on hold at the precise same moment, is a very unlikely scenario in a VoIP connection. The threshold in this case is 2.5 minutes at a stretch i.e. for a time interval of 30 seconds, 5 empty slots are encountered.

5.4.5.1.3 Multiple Session Maintenance

Usage statistics of the internet show that ISPs providing VoIP connections have a higher probability of maintaining multiple sessions simultaneously than the other traffic.

5.4.5.1.4 Bi-directional Traffic Flow

Bidirectional flow of traffic is a defining feature of VoIP traffic. This emphasizes on the fact that in a VoIP connection, the traffic load in both directions is almost equal. In voice communication, the flow is not one sided. Amount of payload flowing from A to B should almost equal the amount flowing from B to A. the threshold for this condition is 65-35, i.e. approximately no more than 65% of the total data transferred during the connection is sent from one side or each party must at least contribute 35% of the data transferred during the established VoIP connection.

5.4.5.2 Weight-age for Criteria

The defined criteria for analysis are a perfect case scenario and all VoIP traffic may not exactly fit the criteria. This is the reason why each criterion is given weight according to its importance in the detection of VoIP traffic. The most important out of these criteria is “Bi-directional flow of traffic”, since this is the distinguishing characteristic of VoIP traffic.

Besides assigning each criterion a specific weight, additional weight is given if a combination of criterion is met. This proves to be helpful in scenarios where two criterion of different importance are met. E.g. if two most important or two least important criterion are met, both these cases must not be given the same weights, so all possible combinations are also assigned weights.

5.4.5.2.1 Individual Weights

1- Bi-Directional Traffic Flow = 25; eq 1

2- Twenty Four Hour Flow = 20; eq 2

3- Continuous Traffic Flow = 15; eq 3

4- Multiple Sessions = 15; eq 4

5.4.5.2.2 Weights for Combination of Criteria

1- Bi-directional + Continuous + 24Hour
+ Multiple Session = 25; eq 5

2- Bi-directional + 24Hour + Multiple Session = 23; eq 6

3- Bi-directional + Continuous + 24Hour = 23; eq 7

4- Bidirectional + Continuous +
Multiple Session = 13; eq 8

5- Bi-directional + 24Hour = 18; eq 9

6- Bidirectional + Multiple Session = 8; eq 10

5.5 Summary

This chapter showed the detail of every module that is incorporated in the system. The distribution of the five modules the system has been divided into, the tasks they perform, the inputs and the outputs to that particular module and the functionality. This chapter also justified the use of the chosen programming language and its underlying benefits.

Graphical User Interface

CHAPTER 6

6.1 Introduction

GUI of a system plays very important role in the success of that system. A very well off project having an absurd GUI won't do any good for that project. GUI needs to be user friendly, easy to use and above all it needs to depict all the functionality of the system as it's the GUI the user has to interact with in order to operate the system. The GUIs for desktop application needs to be helpful for the administrator or whosoever is using it. It should provide the basic functionality in a way so that even with little training, the system can be fully understood.

6.2 Graphical User Interface

The GUI for Network Police has been developed in QT Creator (Linux). The system is automated so does not need elaborate user interaction for commands and running but shows results in graphical format to aid the user in quick detection of IP addresses involved in malicious activity.

The user can select an IP address pair form the list to see its full detail on the interface. On selection, the interface displays the graphs for payload and sessions, the encryption type used and the conditions fulfilled. The User Interface provides the end user with the following functionalities

6.2.1 Categorized IP Address List

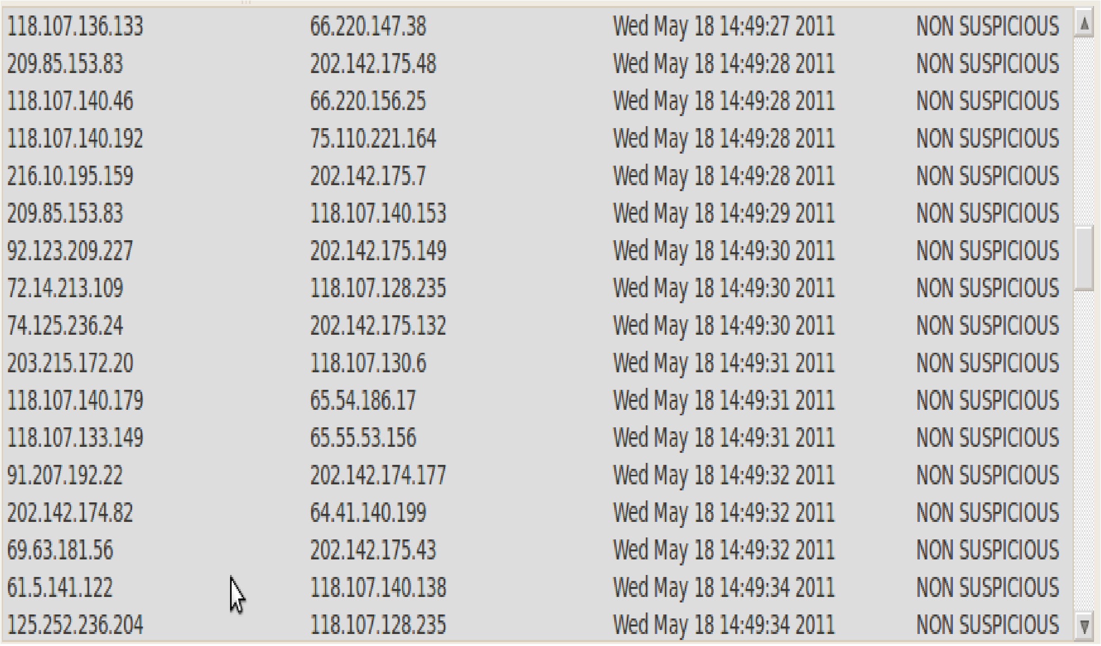
The user interface shows a list of IP addresses detected in the network and the suspicion level they fall in. The list shows four possible suspicion levels based on suspicion scale (0-100) where 100 shows a probability of 1 for traffic being Grey and 0 shows a probability of 0.

1- Non-Suspicious (calculated weight < 60)

2- Low-suspicious (60 < calculated weight < 79)

3- Moderate-suspicious (80 < calculated weight < 100)

4- Grey Traffic(calculated weight == 100)

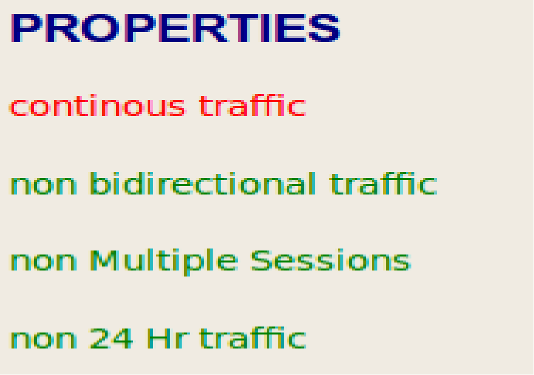


118.107.136.133	66.220.147.38	Wed May 18 14:49:27 2011	NON SUSPICIOUS
209.85.153.83	202.142.175.48	Wed May 18 14:49:28 2011	NON SUSPICIOUS
118.107.140.46	66.220.156.25	Wed May 18 14:49:28 2011	NON SUSPICIOUS
118.107.140.192	75.110.221.164	Wed May 18 14:49:28 2011	NON SUSPICIOUS
216.10.195.159	202.142.175.7	Wed May 18 14:49:28 2011	NON SUSPICIOUS
209.85.153.83	118.107.140.153	Wed May 18 14:49:29 2011	NON SUSPICIOUS
92.123.209.227	202.142.175.149	Wed May 18 14:49:30 2011	NON SUSPICIOUS
72.14.213.109	118.107.128.235	Wed May 18 14:49:30 2011	NON SUSPICIOUS
74.125.236.24	202.142.175.132	Wed May 18 14:49:30 2011	NON SUSPICIOUS
203.215.172.20	118.107.130.6	Wed May 18 14:49:31 2011	NON SUSPICIOUS
118.107.140.179	65.54.186.17	Wed May 18 14:49:31 2011	NON SUSPICIOUS
118.107.133.149	65.55.53.156	Wed May 18 14:49:31 2011	NON SUSPICIOUS
91.207.192.22	202.142.174.177	Wed May 18 14:49:32 2011	NON SUSPICIOUS
202.142.174.82	64.41.140.199	Wed May 18 14:49:32 2011	NON SUSPICIOUS
69.63.181.56	202.142.175.43	Wed May 18 14:49:32 2011	NON SUSPICIOUS
61.5.141.122	118.107.140.138	Wed May 18 14:49:34 2011	NON SUSPICIOUS
125.252.236.204	118.107.128.235	Wed May 18 14:49:34 2011	NON SUSPICIOUS

Figure 6-1: IP Pair List, UI

6.2.2 List of Conditions

The interface also shows a list of properties that the selected IP address pair fulfils. The list comprises of the four conditions discussed in detail in section 5.4.5. The properties that the traffic fulfils are highlighted.



PROPERTIES

continous traffic

non bidirectional traffic

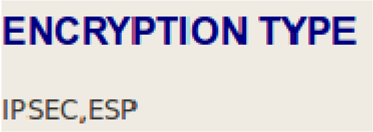
non Multiple Sessions

non 24 Hr traffic

Figure 6-2: List of Conditions, UI

6.2.3 Encryption Type

The interface also specifies the encryption type that is used by the traffic. By type it means the layer at which encryption is used and the encryption, e.g. IPSec using Authentication Header, or SSL, or TLS etc.



ENCRYPTION TYPE

IPSEC,ESP

Figure 6-3: Encryption Type, UI

6.2.4 Graphs

The user interface displays the graphs for the selected IP pair. The two graphs show payload against time and number of sessions against time. The payload graph shows traffic for both ways in different colored lines. It is an effective visual aid that helps understand the ratio of traffic flow from each side. The session graph helps observe the “multiple sessions” condition.

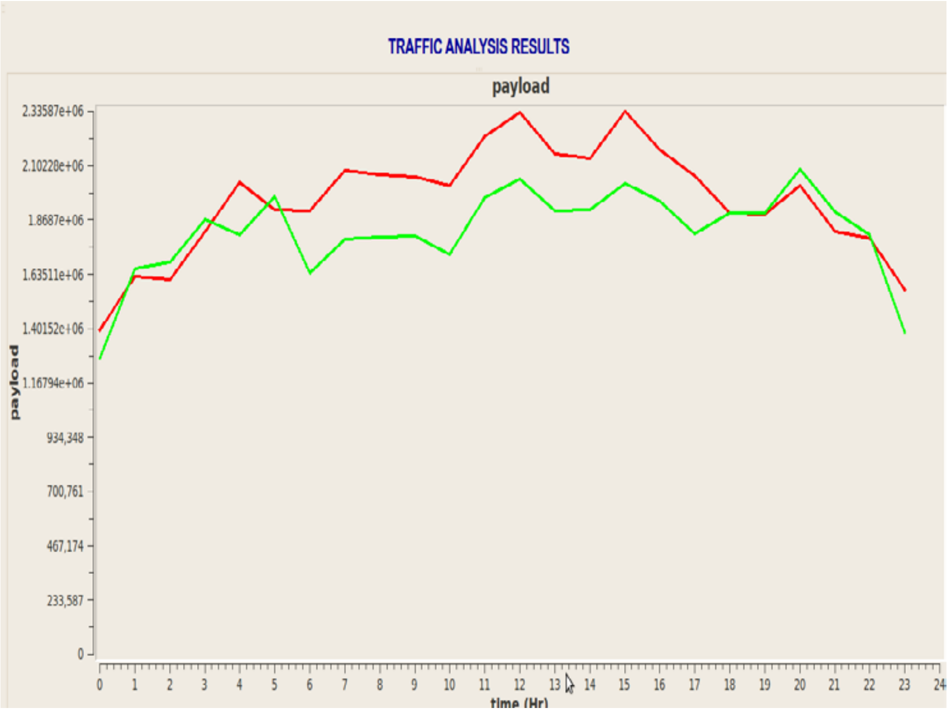


Figure 6-4: Payload Graph, UI

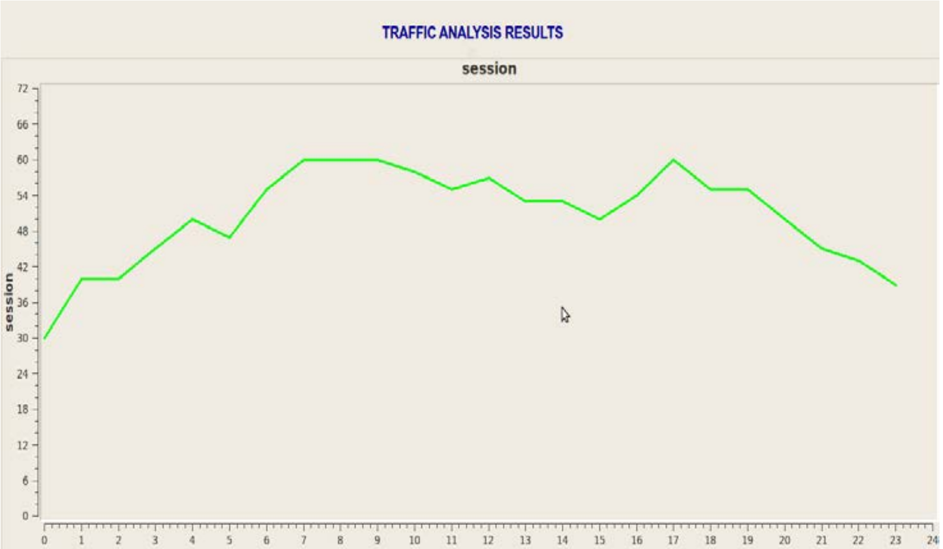
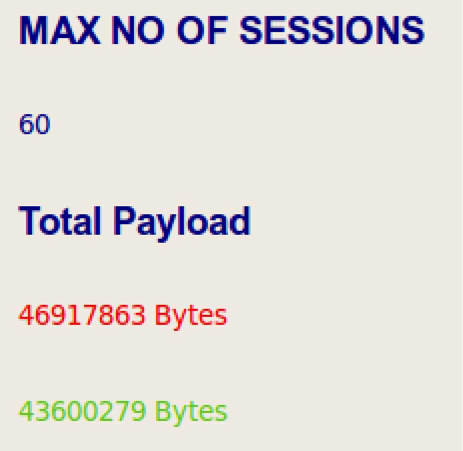


Figure 6-5: Session Graph, UI

6.2.5 Session and Payload Information

The user interface also displays the number of session and the total throughput between the IP address pair. This helps summarize the graph as well as better understand the conditions fulfilled.



MAX NO OF SESSIONS

60

Total Payload

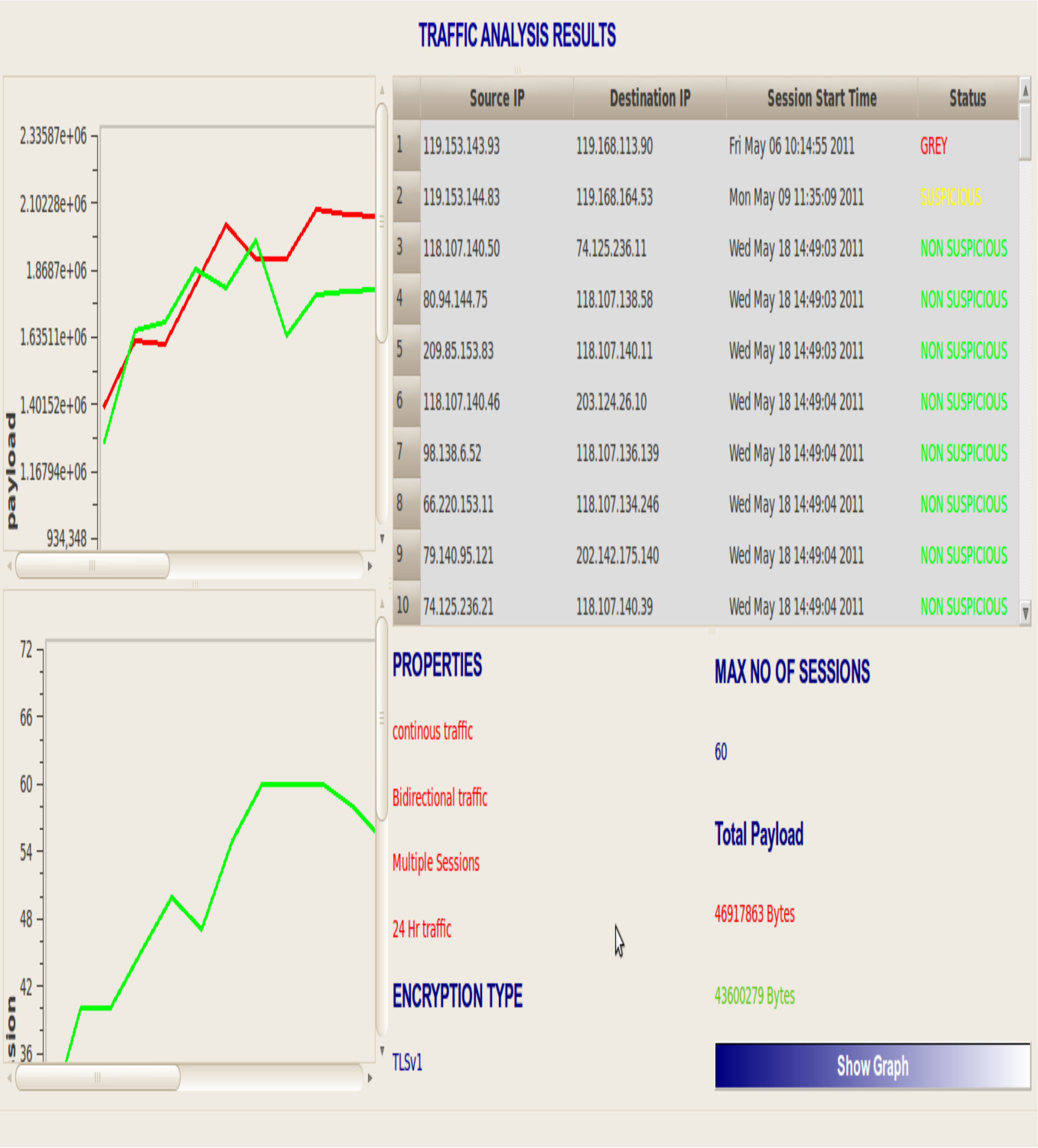
46917863 Bytes

43600279 Bytes

Figure 6-6: Session & Payload Info

6.2.6 User Interface

The user interface displayed shows a brief and precise overview of the nature of the traffic. It helps easily identify IP pair suspected as grey through color coding and to easily view related graphs and details.



TRAFFIC ANALYSIS RESULTS

	Source IP	Destination IP	Session Start Time	Status
1	119.153.143.93	119.168.113.90	Fri May 06 10:14:55 2011	GREY
2	119.153.144.83	119.168.164.53	Mon May 09 11:35:09 2011	SUSPICIOUS
3	118.107.140.50	74.125.236.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
4	80.94.144.75	118.107.138.58	Wed May 18 14:49:03 2011	NON SUSPICIOUS
5	209.85.153.83	118.107.140.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
6	118.107.140.46	203.124.26.10	Wed May 18 14:49:04 2011	NON SUSPICIOUS
7	98.138.6.52	118.107.136.139	Wed May 18 14:49:04 2011	NON SUSPICIOUS
8	66.220.153.11	118.107.134.246	Wed May 18 14:49:04 2011	NON SUSPICIOUS
9	79.140.95.121	202.142.175.140	Wed May 18 14:49:04 2011	NON SUSPICIOUS
10	74.125.236.21	118.107.140.39	Wed May 18 14:49:04 2011	NON SUSPICIOUS

PROPERTIES

continous traffic

Bidirectional traffic

Multiple Sessions

24 Hr traffic

ENCRYPTION TYPE

TLSv1

MAX NO OF SESSIONS

60

Total Payload

46917863 Bytes

43600279 Bytes

Figure 6-7: User Interface

6.3 Summary

In this chapter the Graphical User Interface of the system was discussed in detail. It showed the functionality of the entire system from the end user's perspective as well as different functions that the system performs.

Software Testing

CHAPTER 7

7.1 Introduction

In this section of the document, system testing and load balancing test has been performed using different test cases. The purpose of software testing is to assess and evaluate the quality of work performed at each step of the software development process. The goal of testing is to ensure that the software performs as intended, and to improve software quality, reliability and maintainability. Software testing can also be stated as the process of validating and verifying that a software program/application/product:

- meets the business and technical requirements that guided its design and development;
- works as expected; and
- can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

7.2 Unit Testing

Testing of individual software components or modules is called Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. It may require developing test-driver modules or test harnesses.

Unit testing was a major part in testing the project. As discussed, the project is divided in modules. In Unit Testing, every module was tested independently to ensure that it functions according to needs and requirements. The developers of the current and subsequent module normally carried out the tests, since the output of one is the input of the next module.

7.2.1 **Test Case 1** - Capture Mode was tested in Special Communication Organization (SCO) by capturing internet traffic from a special mirrored link provided by SCO.

7.2.2 **Test Case 2** - Processed Packet Library was tested by providing as input the data captured by Test Case 1 and processing it to see the performance and reliability of the module.

7.2.3 **Test Case 3 –** Processed Session Library was tested by output of Test Case 2. Also, the output files very thoroughly checked and verified to ensure correct results.

7.2.4 **Test Case 4 –** Analysis module was fed with the output file of Test Case 3. The results were verified by matching with the expected output computed.

7.3 Integration Testing

Testing of integrated modules to verify combined functionality after integration is called Integration Testing. Modules are typically code modules, individual applications, client and server applications on a network, etc.

All modules were tested to find compatibility issues with other modules. Output of modules was made standard to avoid and remove compatibility issues since it acted as input for the next module.

7.4 Summary

This chapter has reflected the tests performed on the system to ensure its proper functioning and error removing. It also gives an overview of module interdependencies and their integration.

Result And Analysis

CHAPTER 8

8.1 Introduction

In this section of the document, we analyze our application and then compare it with applications having similar attributes. This does not include testing techniques rather it just gives the analysis. Analysis of Network Police's approach to the problem of grey traffic is done based on comparison with current techniques and approaches.

8.2 Network Police Salient Features

Network Police has achieved the following desired goals and outcomes

8.2.1 Selectively identify voice traffic on any IP network.

8.2.2 Identify the IP addresses involved in Grey Traffic in Pakistan.

8.2.3 The ability to produce comprehensive information on call duration and connection time maintained.

8.2.4 International carrier rates and settlement payments will stabilize as Grey traffic is controlled.

8.3 Network Police Comparison with State-of-the-art Application

Feature	Network Police	Packet Analyzer (Wireshrak, MaaTec etc.)	DPI	Network Monitors
Traffic Capture from Network	Yes	Yes	Yes	Yes
Profile based Anomaly Detection of Traffic	Yes	No	No	Yes
Encryption Detection	Yes	Yes	Yes	No
Encryption Layer Detection	Yes	Yes	No	No
Detection of Encrypted VoIP	Yes	No	No	No
Identification of IP involved in grey traffic	Yes	No	No	No

Table 8-1: Comparison

8.4 Summary

Network Police was compared with pre-existed state-of-the-art traffic analysis and monitoring tools and technologies in terms of functionality and limitations in this chapter. It has reflected the salient features of **Network Police** and evaluated it with respect to its commercial applications.

Conclusion And Future Enhancements

CHAPTER 9

9.1 Introduction

This chapter describes the possible future enhancements in the project. There are a lot of enhancements that can be done in order to make the system more effective in the face of providing flexibility and efficiency for better results.

9.2 Commercial Application and Use

Network Police is a tool for identifying IP addresses involved in Grey Traffic in Pakistan. This can prove to be beneficial for the PTA, as well as LDI operators and other telecommunication companies of the country. It can prove to be financially helpful to these companies, but above all, it can help PTA to overcome the existing problem of tax evasion that our country is facing because of the curse of Grey Traffic.

9.3 Future Enhancements

The project addresses a current problem and proposes an efficient solution for it. The proposed approach to the problem is novel and can be developed further in future for the purpose of meeting new and upcoming needs regarding the problem.

Future Enhancements can be related to the development of code for high capacity core routers that are solely bound to perform VoIP intrusion detection on run time. Capturing data packets at run time is an expensive process that not only requires better hardware resources and processing power but also an efficient and thoroughly tested code that can perform such powerful tasks without failing under pressure and without missing data at a high incoming speed. Furthermore, the processing needed to categorize traffic at run time also requires a great amount of intelligence at the code’s part so that decisions can be made in online mode rather than the currently proposed offline mode.

Run time traffic detection would mean automating the filtering process basing on past behavior of the traffic or the targeted IP address pair. This filtering process should be fast and reliable enough to judge the best possible targeted traffic in the least possible time and also, in time. Run time decision would increase the load and there would be greater constraints for better results. The code should be up to date and efficient to handle such problems and the sort of dynamic data handling involved.

9.4 Conclusion

The growth of illegal VoIP activity is a lucrative platform for culprits to earn huge revenue and causes huge financial losses to the state. VoIP intrusion detection against commercial traffic is a tricky affair as VoIP is legally used by many domestic internet users for point to point direct communication, such as internet messengers etc. It requires thorough statistical and behavioral analysis over a period of time to identify and detect IP addresses involved in this crime and to separate illegal IPs form innocent VoIP users. In addition, involvement of encryption techniques and availability of high throughput internet connections at cheap rates, has made the problem complicated many folds.

We have targeted mainly the network layer, transport layer and application layer encryption. In this project, instead of encryption analysis, we have endeavored to deploy VoIP based Intrusion Detection System using statistical flow analysis coupled with behavior analysis. It deals with profile development and anomaly detection based on variation form this profile of VoIP traffic based on observed patterns. Encryption is detected and layer on which encryption is done is found but decryption of that traffic is not in any way a part of the project. Encryption layer helps identify the type of encryption technique used and can prove to be helpful in computing statistical parameters for better decision making.

The tests show significant success on applied approach, however there is a need to implement the algorithm for high capacity core routers. Network Police is aimed to be more robust and dynamic to serve different organizations, with special interest in solving the problems caused by Grey traffic for the country due to which national exchequer is suffering.

Appendices

Appendix A

USER MANUAL

MCS

2011

Ayesha Khaliq
Ammara Sajid
Sarmad Nisar
Osamah Karim

[NETWORK POLICE]

Traffic Identification for Segregation & Analysis of Suspicious Encrypted VoIP Traffic-Grey Traffic

TABLE OF CONTENTS

USER MANUAL

INSTALLATION

The product is a software with easy to install components and modules. It comprises of four basic modules which need to be copied to the system. The development language is C, thus the product is portable. It has been developed in GCC Linux so care regarding compatible header types must be taken to install in a different operating system.

ABOUT THIS MANUAL

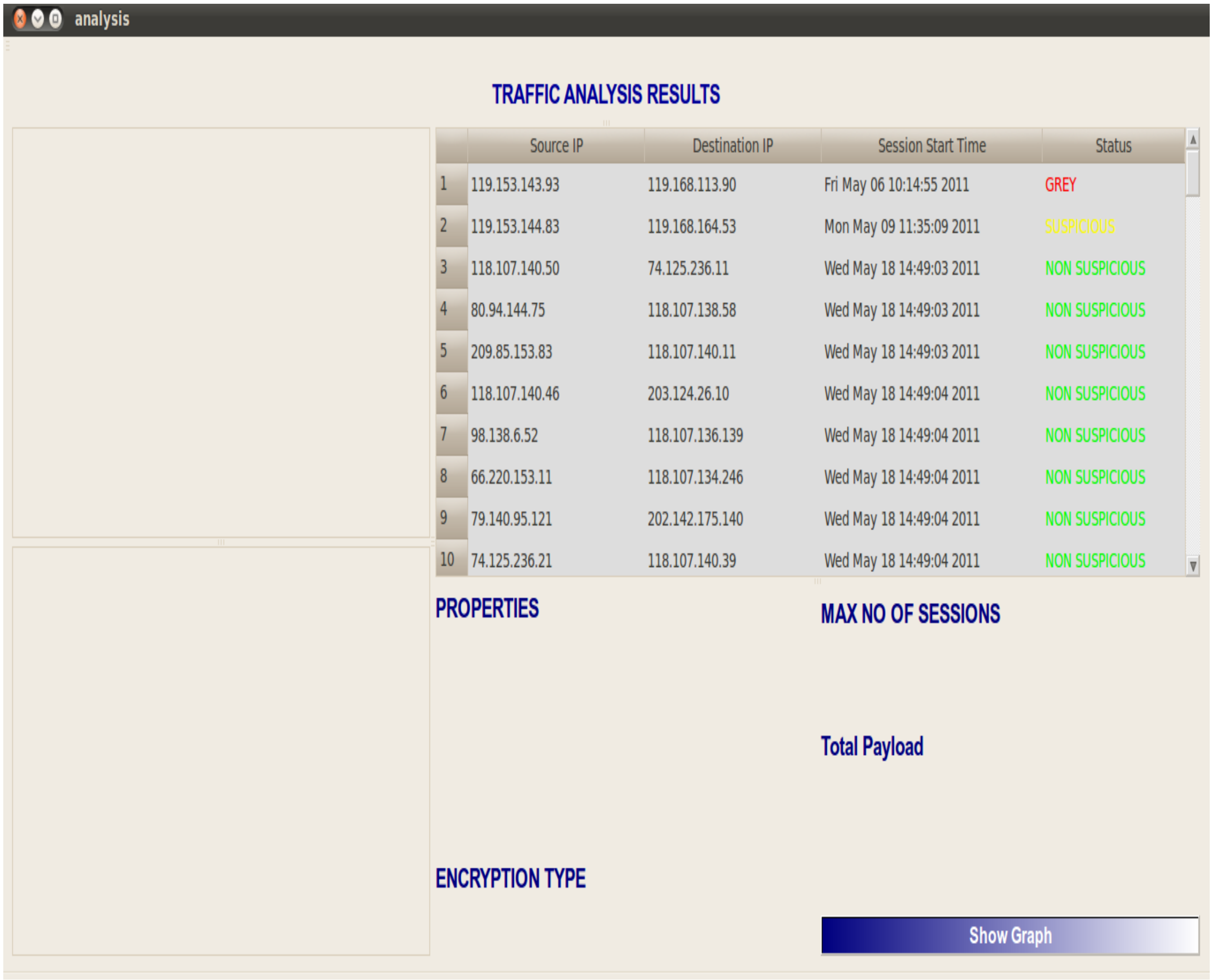
This manual describes the working and usage method of the software.

PROCEDURE TO USE THE SOFTWARE

It's a simple two step, user friendly process. Select and Display.

STEP 1 ➔ Select IP address pair from IP list

The IP address pair list shows the IP addresses detected over the network along with their category. Select the desired IP pair by simply clicking on the entry in the list.

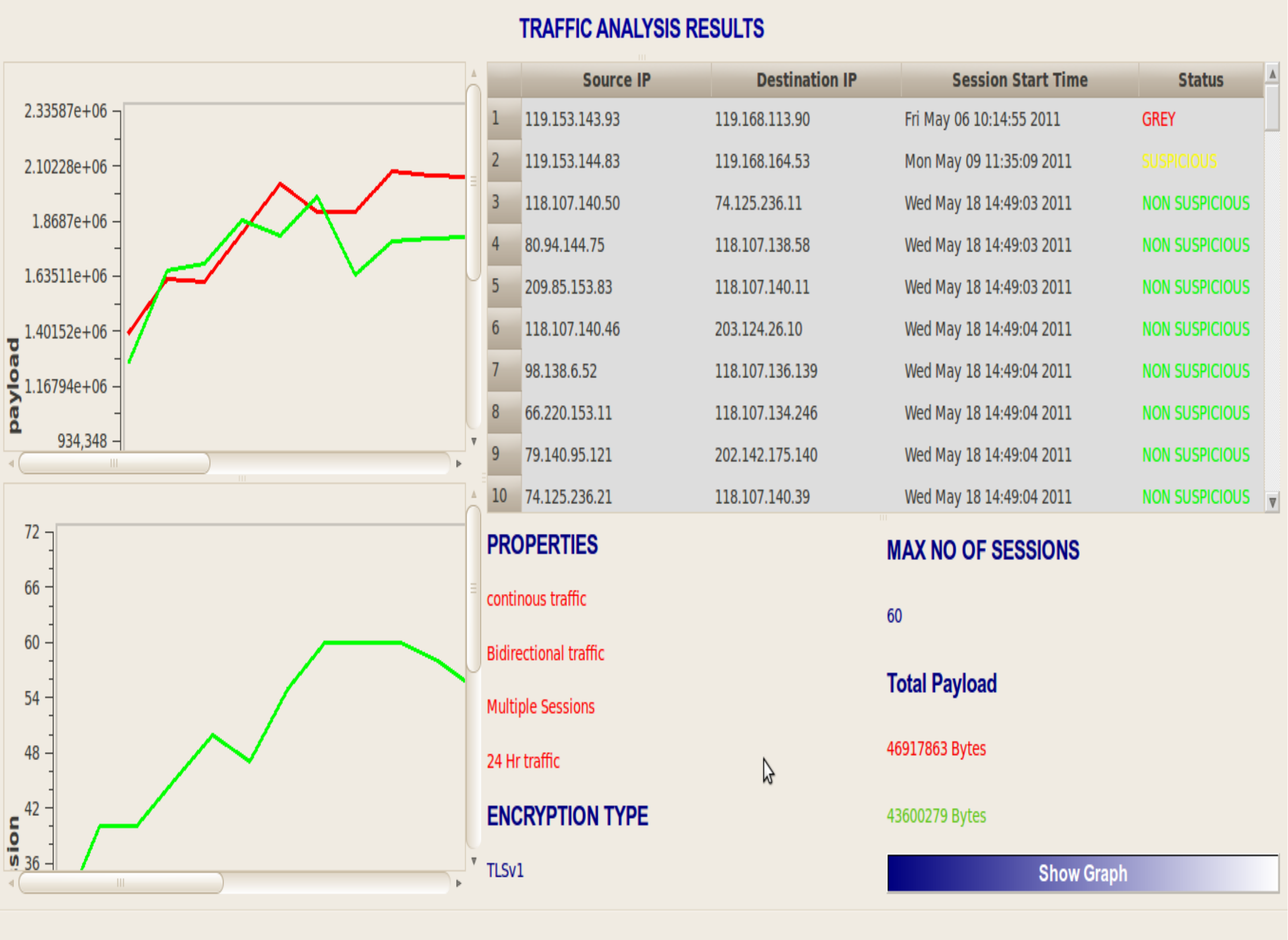


TRAFFIC ANALYSIS RESULTS

	Source IP	Destination IP	Session Start Time	Status
1	119.153.143.93	119.168.113.90	Fri May 06 10:14:55 2011	GREY
2	119.153.144.83	119.168.164.53	Mon May 09 11:35:09 2011	SUSPICIOUS
3	118.107.140.50	74.125.236.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
4	80.94.144.75	118.107.138.58	Wed May 18 14:49:03 2011	NON SUSPICIOUS
5	209.85.153.83	118.107.140.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
6	118.107.140.46	203.124.26.10	Wed May 18 14:49:04 2011	NON SUSPICIOUS
7	98.138.6.52	118.107.136.139	Wed May 18 14:49:04 2011	NON SUSPICIOUS
8	66.220.153.11	118.107.134.246	Wed May 18 14:49:04 2011	NON SUSPICIOUS
9	79.140.95.121	202.142.175.140	Wed May 18 14:49:04 2011	NON SUSPICIOUS
10	74.125.236.21	118.107.140.39	Wed May 18 14:49:04 2011	NON SUSPICIOUS

STEP 2 ➔ Display Details

Press the "Show Graph" Button the bottom of the interface to view details regarding the selected IP address pair.



TRAFFIC ANALYSIS RESULTS

	Source IP	Destination IP	Session Start Time	Status
1	119.153.143.93	119.168.113.90	Fri May 06 10:14:55 2011	GREY
2	119.153.144.83	119.168.164.53	Mon May 09 11:35:09 2011	SUSPICIOUS
3	118.107.140.50	74.125.236.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
4	80.94.144.75	118.107.138.58	Wed May 18 14:49:03 2011	NON SUSPICIOUS
5	209.85.153.83	118.107.140.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
6	118.107.140.46	203.124.26.10	Wed May 18 14:49:04 2011	NON SUSPICIOUS
7	98.138.6.52	118.107.136.139	Wed May 18 14:49:04 2011	NON SUSPICIOUS
8	66.220.153.11	118.107.134.246	Wed May 18 14:49:04 2011	NON SUSPICIOUS
9	79.140.95.121	202.142.175.140	Wed May 18 14:49:04 2011	NON SUSPICIOUS
10	74.125.236.21	118.107.140.39	Wed May 18 14:49:04 2011	NON SUSPICIOUS

PROPERTIES

continous traffic

Bidirectional traffic

Multiple Sessions

24 Hr traffic

ENCRYPTION TYPE

TLSv1

MAX NO OF SESSIONS

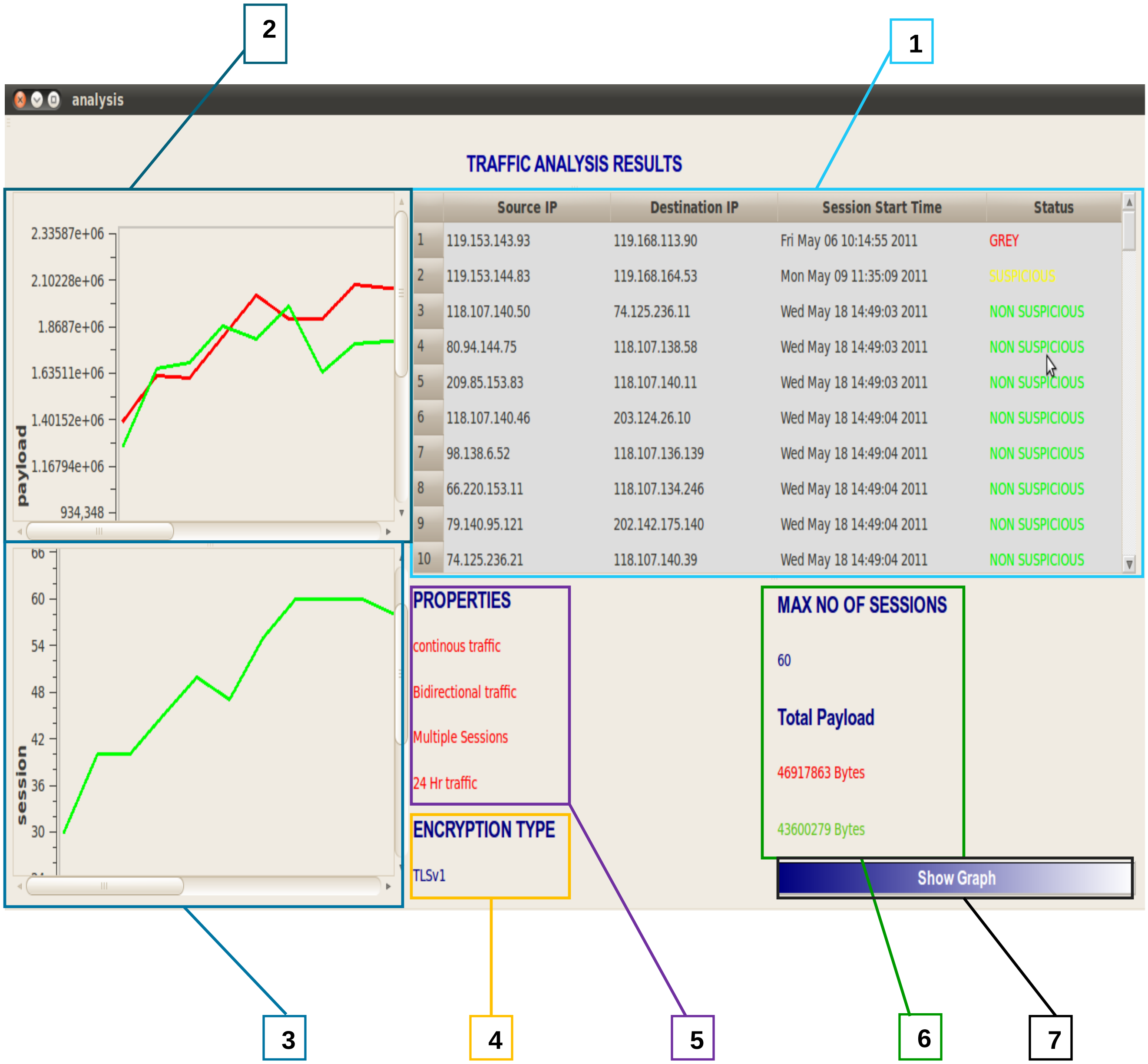
60

Total Payload

46917863 Bytes

43600279 Bytes

INTERFACE PARTS AND DESCRIPTION



TRAFFIC ANALYSIS RESULTS

	Source IP	Destination IP	Session Start Time	Status
1	119.153.143.93	119.168.113.90	Fri May 06 10:14:55 2011	GREY
2	119.153.144.83	119.168.164.53	Mon May 09 11:35:09 2011	SUSPICIOUS
3	118.107.140.50	74.125.236.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
4	80.94.144.75	118.107.138.58	Wed May 18 14:49:03 2011	NON SUSPICIOUS
5	209.85.153.83	118.107.140.11	Wed May 18 14:49:03 2011	NON SUSPICIOUS
6	118.107.140.46	203.124.26.10	Wed May 18 14:49:04 2011	NON SUSPICIOUS
7	98.138.6.52	118.107.136.139	Wed May 18 14:49:04 2011	NON SUSPICIOUS
8	66.220.153.11	118.107.134.246	Wed May 18 14:49:04 2011	NON SUSPICIOUS
9	79.140.95.121	202.142.175.140	Wed May 18 14:49:04 2011	NON SUSPICIOUS
10	74.125.236.21	118.107.140.39	Wed May 18 14:49:04 2011	NON SUSPICIOUS

PROPERTIES

continous traffic

Bidirectional traffic

Multiple Sessions

24 Hr traffic

ENCRYPTION TYPE

TLSv1

MAX NO OF SESSIONS

60

Total Payload

46917863 Bytes

43600279 Bytes

1

Categorized IP Address List

Shows a list of IP addresses detected in the network and the suspicion level they fall in.

2

Payload Graph

Shows the graph for payload/ throughput against time for a selected IP address pair. Traffic from both sides is represented in different colors.

3

Session Graph

Shows the graph for number of sessions against time for the selected IP address pair.

4

Encryption Type

Shows the encryption type that the traffic is using (if any).

5

Properties

Shows a list of properties that the selected IP address pair fulfils. The properties that the traffic fulfils are highlighted.

6

Maximum Sessions and Payload

Shows the maximum number of sessions and the total throughput for the selected IP address pair. The color codes correspond to the color code in payload graph.

7

Show Button

Displays details for the selected IP pair on the interface.

Bibliography

BIBLIOGRAPHY

[1] RFC 2246 - TLS Protocol version 1, January 1999
http://www.ietf.org/rfc/rfc2246.txt

[2] RFC 2401 - Security Architecture for the Internet Protocol, 1998
http://www.ietf.org/rfc/rfc2401.txt

[3] RFC 2637 - Point-to-Point Tunneling Protocol (PPTP), 1999
http://www.ietf.org/rfc/rfc2637.txt

[4] RFC 791 - Internet Protocol, Protocol Specifications, 1981
http://www.ietf.org/rfc/rfc791.txt

[5] RFC 793 - Transmission Control Protocol (TCP), 1981
http://www.ietf.org/rfc/rfc793.txt

[6] RFC 2543 - SIP: Session Initiation Protocol, 1999
http://www.ietf.org/rfc/rfc2543.txt

[7] RFC 3261 - SIP: Session Initiation Protocol, 2002
http://www.ietf.org/rfc/rfc3261.txt

[8] RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1, 1999
http://www.ietf.org/rfc/rfc2616.txt

[9] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications, 2003
http://www.ietf.org/rfc/rfc3550.txt

[10] PCAP library, 2003
http://www.tcpdump.org/pcap3_man.html

[11] Wireshark Open Source Code, 2011
http://anonsvn.wireshark.org/viewvc

[12] Toshiya Okabe, Tsutomu Kitamura and Takayuki Shizuno, “Statistical Traffic Identification Method Based on Flow-Level Behavior for Fair VoIP service”, IEEE, 2006.

[13] Bing Li, Zhigang Jin and Maode Ma, “VoIP Traffic Identification Based on Host and Flow Behavior Analysis”, IEEE, 2010.

[14] Ruining Guo, Tianjie Cao, Xuan Luo, “Application Layer Information Forensics based on Packet Analysis”, IEEE, 2010.

[15] Maznan Derarnan, Jalil Md Desa Zulaiha Ali Othman, “Multilayer Packet Tagging for Network Behaviour Analysis”, IEEE, 2010.

[16] Roni Bar-Yanai, Michael Langberg, David Peleg, and Liam Roditty , “Realtime Classification for Encrypted Traffic”, IEEE, 2010.

[17] Brian Beej Jorgensen Hall, “Beej's Guide to Network Programming Using Internet Sockets”, 2009.

[18] Neil Matthew and Richard Stones , “Beginning Linux® Programming 4th Edition”, 2007.