Integrated Network Traffic Visualization for Threat Detection, Analysis and Reporting

A thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

by

Amit Kumar

(90703502)

under the guidance of

Dr. Maninder Singh

Associate Professor

Computer Science & Engineering Department

Computer Science & Engineering Department
Thapar University, Patiala-147004, INDIA

MAY 2015

DEDICATED

TO

MY FATHER

“Late. Sh. Purushotam Lal Ji”

Certificate

I hereby certify that the work which is being presented in this thesis entitled *Integrated Network Traffic Visualization for Threat Detection, Analysis and Reporting*, for award of degree of Doctor of Philosophy submitted to the Computer Science and Engineering Department of Thapar University, Patiala, is an authenticate record of my own work carried out under the supervision of Dr. Maninder Singh and refers other researchers works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other University.

(Amit Kumar)

Reg. No. 90703502

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Maninder Singh)
Associate Professor,
Computer Science & Engineering Department
Thapar University, Patiala-147004, Punjab, INDIA

Acknowledgements

I am thankful to the Almighty God, who gave me the opportunity and strength to carry out this work.

I feel privileged to express my sincere regards and gratitude to my supervisor Dr. Maninder Singh, Associate Professor, Computer Science and Engineering Department, Thapar University, Patiala for his expert guidance, valuable suggestions, support, advice and continuous encouragement throughout the course of my research work.

I am highly obliged to Professor Padmakumar Nair, Director, LM Thapar School of Management, all members of Doctoral Committee, faculty members of LM Thapar School of Management and Computer Science & Engineering department, for their motivational words and staff for providing me necessary facilities for carrying out my research work. I also wish to pay my special regards to Dr. Seema Bawa and Dr. Anil K. Verma for his guidance and support.

I am also grateful to Dr. O.P Pandey, Dean of Research and Sponsored Project, for his constant encouragement that was of great importance in the completion of the thesis. I am also thankful to Professor P. K. Bajpai for his encouraging words.

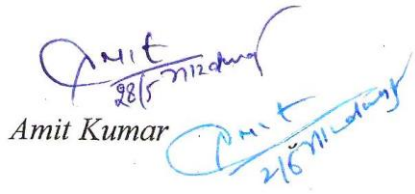
I extend my thanks to Dr. Prakash Gopalan, Director, Thapar University, Patiala for his valuable support that made me consistent performer.

A special thanks to my family friend Dr. Amit Kumar for his consistent support and motivation. I would like to thank Dr. Deepak Garg, Dr. Tejo Prakash, Dr. Prateek Bhatia, Ms. Sanmeet Bhatia, Dr. S.S. Kasana, Dr. Rudra Rameshwar, Dr. Gurparkash Singh, Mr. Ankit Mahindroo,

Mr. Gaurav Goyal, Mr. Ravinder Kumar, Dr. Inderpereet Singh, Dr. Perminder Singh Reel and all other Research Scholars (especially Sukhpal Singh and Ms. Ratinder kelly) of Computer Science and Engineering Department for their timely help and the moral support they provided me during my research work.

I would like to thank all researchers whose work I have cited in this research work.

A deep sense of gratitude is always in my heart for my parents who formed part of my initial vision and inculcated in me the real value of education which really matters in life. I would like to give my special thanks to my mother, in-laws and brothers for their constant support and encouragement that was of great importance in the completion of the thesis. I would like to acknowledge the time scarified by my cute kids (Mannat and Akash Jai) and lovable wife (Bindu Bhardwaj) a great source of motivation and support.



Abstract

In this speedy and voluminous digital world the threat detection and reporting is a challenging task for a revert action. This research work highlights 5 Vs (Volume, Velocity, Variety, Vulnerability, Visualization based security system) of computer networking, which are getting attention of network administrators and network security providers. First 4 Vs are posing big challenges to handle the mischievous practices over the Internet, and fifth 'V' visualization based security solution (VizSec) is a promising solution, standing against network threats and having advantage over conventional approach of log based security. This research work encourage the network security analyst to adopt a new approach i.e. VizSec, which may lead to the development of a security solution based on visualization such as visualization based firewall. This research work discusses visualization schemes given by key researchers in the domain of VizSec in detail and highlights motivation to develop a VizSec.

Traditional schemes by many researches like Flodar, Wisconsin Netpy, IDS Rainstrom, IDGraphs, Rumint, NfSen and VIAssist are capable to visualize upto third layer data of OSI model and some namely NVisionIP, VISUAL, VisFlow Connect-IP, SIFT, NetViewer, PortAll, TNV, InetVis, FlowTag, Flamingo, FloVis, NAV, ScanViewer, CCScanViewer and InfoVis, are capable to give data visualization of third and fourth layer of OSI model.

Visualization systems reported in literature lack in one way or other to capture packets, tokenize, parse and then visualize them in an integrated interactive manner upto Layer 7 in real-time guided and unguided media form.

Based on literature review it was clear that capability needs to be researched and developed to produce interactive visualization of network traffic in a seamless manner.

In this work, a novel approach integrated network traffic visualization system (INTVS) is proposed, developed and validated which can capture, tokenize, parse, detect and report the threats in visual form based on data mining. INTVS is having component based architecture that provides the flexibility to add/remove any component.

INTVS demonstrates three novel schemes- Grid view, Listmap view and Platter view. The grid view can display network traffic in different classified grids, based on application layer protocols. Listmap view gives a holistic view of all detected nodes in a network, whereas Platter view based on data mining deals with the visualization of campus area network traffic on single screen while identifying and mentioning the compromised networks and machines based on defined network policy. Two - dimensional analysis of whole campus area network facility is also offered, which is unique in its presentation - a user can have a drill down analysis of network to know, which VLAN is under attack or consuming maximum network resources? which machine in a VLAN or which application layer protocol is consuming maximum bandwidth? which machine is under attack.? INTVS is validated in both online and offline mode.

INTVS as framework is bundled into live distribution CD, which can work as plug and play system with little human intervention.

The usage of fuzzy set theory, fault tree analysis approach and fuzzy fault tree analysis of INTVS are used for validation. A FTA of INTVS is prepared, followed by fuzzy fault tree analysis of INTVS. A fuzzy data set is used to determine the reliability of INTVS .

List of publications

Bhardwaj A. K. and Singh M., Data mining-based integrated network traffic visualization framework for threat detection, Neural Computing and Applications 26 (1), 117-130, 2014.

Bhardwaj A. K. and Singh M., Network Traffic Threat Detection and Reporting System Validation through UML, Network and Complex Systems 5 (2), 21-29, 2015.

Table of Contents

Chapter 1

Introduction

Information Visualization (InfoVis) is communication of abstract data through the use of interactive visual interfaces. Additionally, InfoVis is a branch of Computer Science concerned with modeling complex data using interactive images, which is an emerging field and has great utility in depicting lots of information as graphical content which in turn will help administrator to take quick actions/decision out of voluminous logs. InfoVis is one of the most informative and precise security approach which helps network administrator to get data in pictorial form for quick analysis, understanding vulnerabilities and give response against emerging threats/attacks.

Visualization for Cyber Security (VizSec) tools are using visualization schemes as an integral part of the network security because "a picture depicts thousands words [34]". Human brain responds rapidly to a picture than a text information because the eye and the visual cortex of the brain form a massively parallel processor that provides the highest bandwidth channel into human cognitive centers [34].

This research work highlights basic and important features of security system based on five Vs i.e. ***Volume, Velocity, Variety, Vulnerability and Visualization*** of network traffic. In which, first four Vs are posing big challenges to handle malicious intents over the Internet. Remaining fifth 'V' visualization based security scheme (VizSec) is a promising solution, applicable against network threats and having advantage over conventional approach of security tools.

This will encourage the network security fraternity to adopt new nomenclature for security approaches like InfoViz: information visualization, FireViz: Visualization based firewall, IDSViz: visualization based IDS. These approaches must incorporate features of integrated network traffic visualization system (INTVS) framework, which is outcome of this research.

The main objective of this thesis is to know how to manage voluminous and variety of network traffic effectively in high velocity and vulnerable environment with visualization based security awareness of network assets.

1.1 Internet: Volume & Velocity

In digital world, nearly 640TB of data is getting transferred over Internet in a span of one minute, which constitutes approximately 100k tweets, two million Google search queries, 1.3 million videos viewed over YouTube and 204 million e-mails sent across by different email service providers globally. Since 2000 to 2014 the numbers of Internet users have increased by 3938%. There are 67,29,85,183 websites and 2,75,61,98,420 Internet users recorded in 2013. Due to larger population of users, voluminous data is accessed through Internet with high velocity for variety of web services. Through variety of web services, vulnerable application software leads to the network attacks. VizSec based security solutions provide in-depth analysis and reporting for network traffic and threats for quick decision making.

Figure 1.1 displays the exponential growth of the Internet users and Websites from 1 website in 1991 to 672,985,183 websites in 2013.

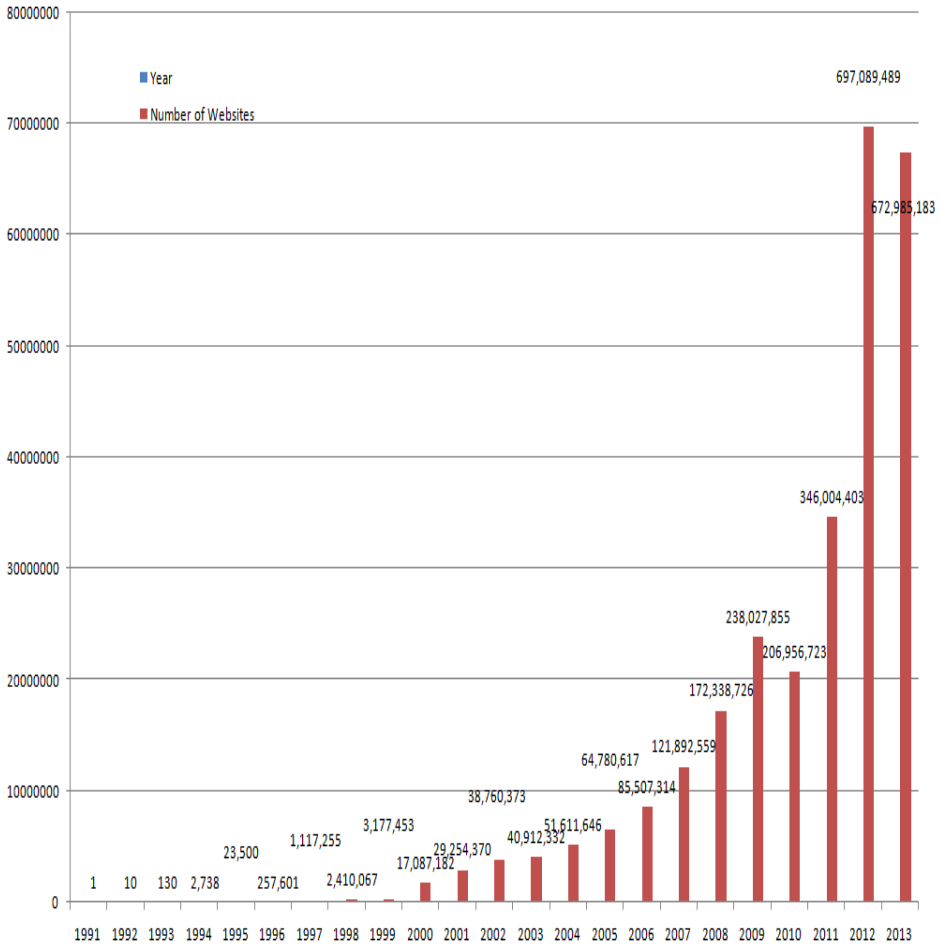


Figure 1.1: Growth of Internet Users and Websites [130]

A Web site (website is a unique hostname which gets resolved by a name server into a IP address) is a related collection of World Wide Web (WWW) files including a home page file. The purpose of home page is, to address the Website as well as to interlink all the other pages on that Website.

Figure 1.2 is showing population penetration by the Internet, and North America is leading among the geographic region with 87.7% penetration rate. It means a very large part (Volume) of population is using the Internet service.

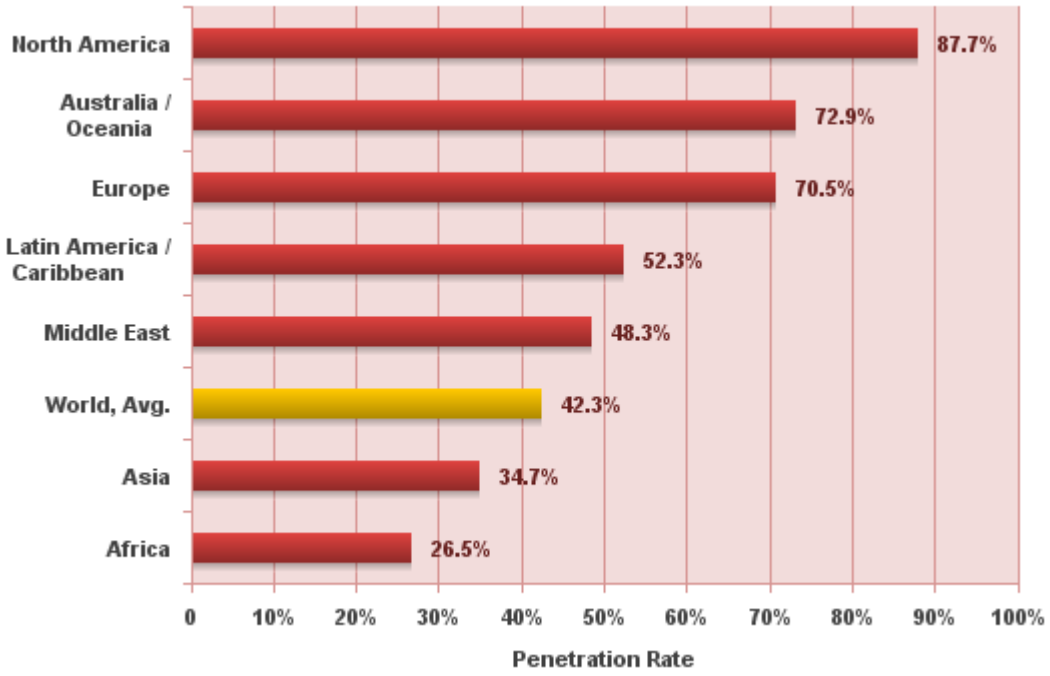


Figure 1.2: World Internet penetration [131]

As shown in Figure 1.3, there are 1,386,188,122 Internet users in Asia, which means that 45.7% of Internet user community is from Asia.

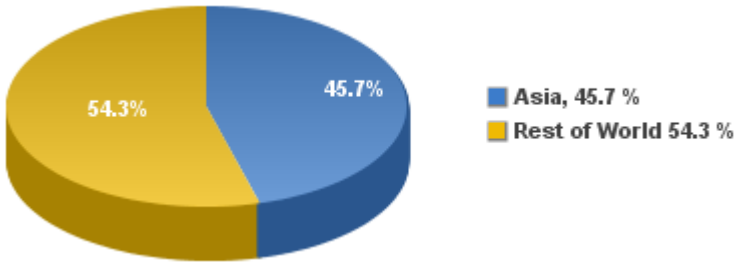


Figure 1.3: Internet user - Asia versus world [132]

Further, in Asia, it is China who is having maximum Internet users (642.3 million) followed by India with 243 million Internet users as shown in Figure 1.4.

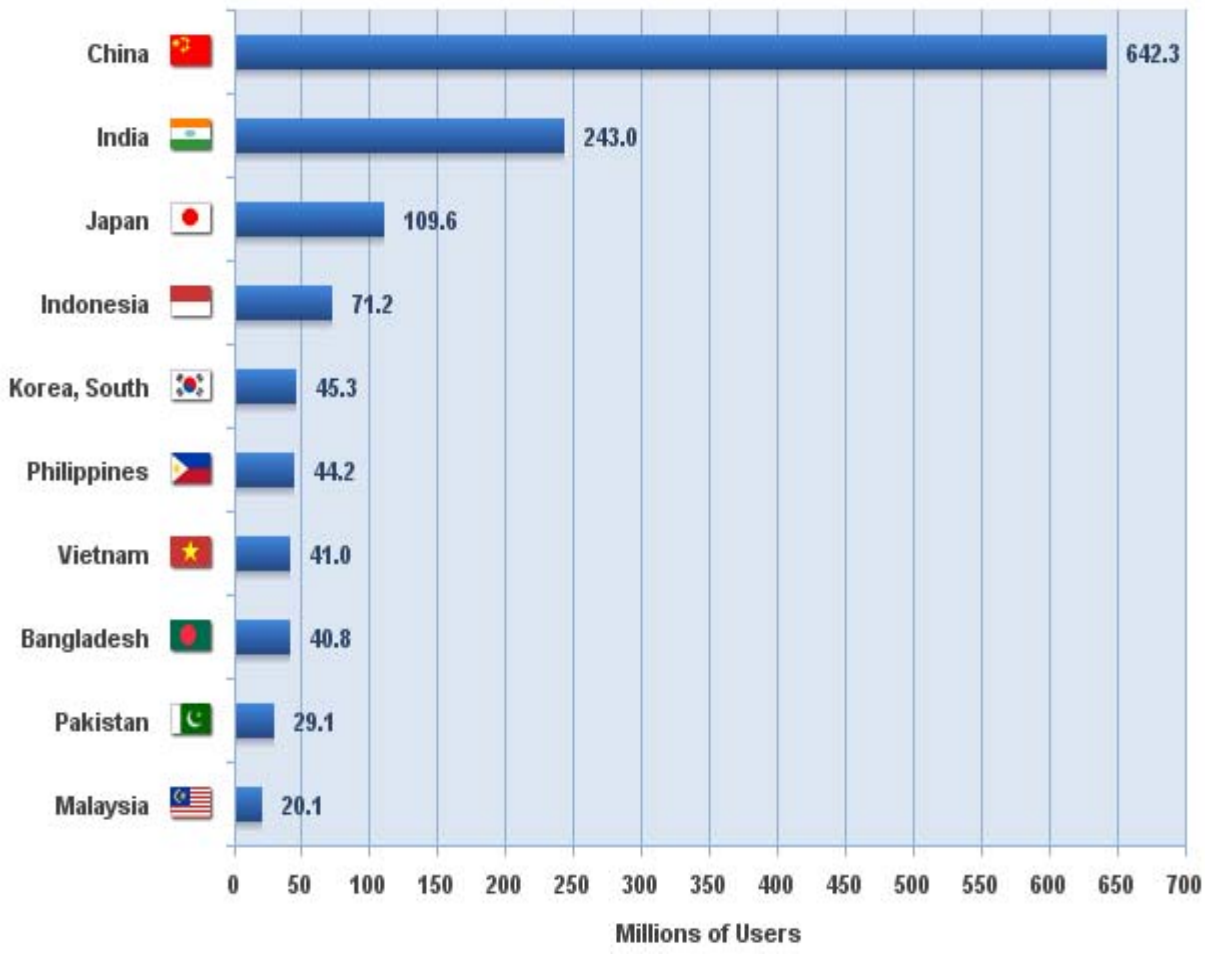


Figure 1.4: Top Internet user country of Asia [132]

To fulfill the Internet services demand and make them connected with Internet community requires a high velocity (speedy) Internet infrastructure. Singapore is top in providing the highest Internet velocity as shown in Figure 1.5.

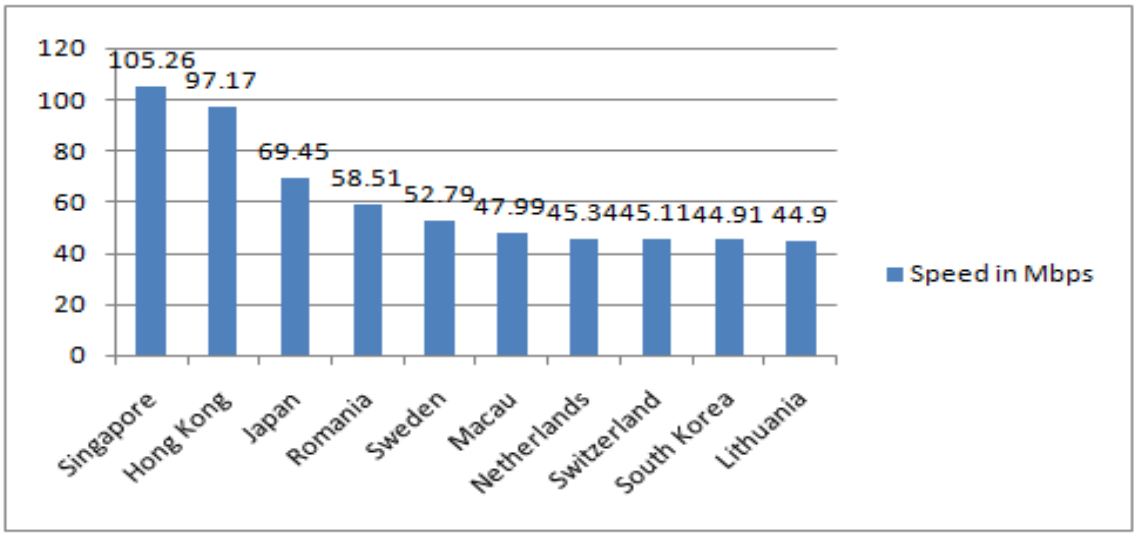


Figure 1.5: Top ten countries w.r.t. Internet speed [133]

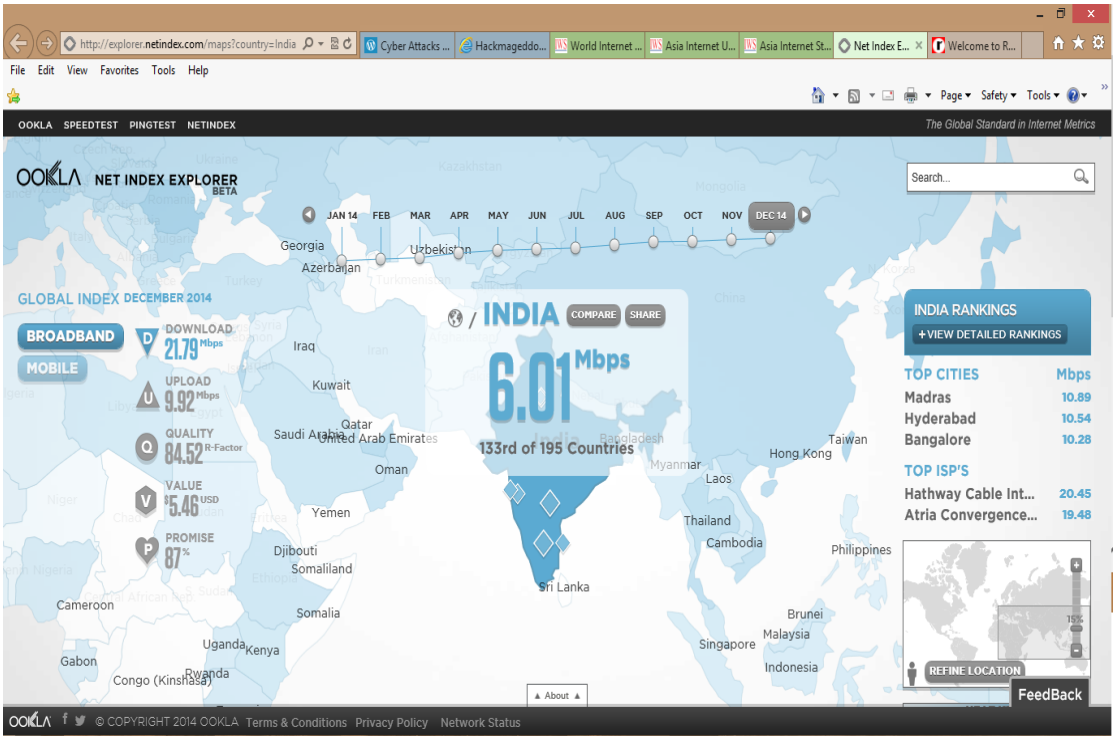


Figure 1.6: Rank of India w.r.t. Internet speed [134]

As shown in Figure 1.6, India is ranked 133rd of 195 countries w.r.t. Internet velocity (speed) i.e. 6.01Mbps for download. Further, Hathway Cable Internet is best Internet service provider (ISP) in India with 20.45Mpbs speed as shown in Figure 1.7.

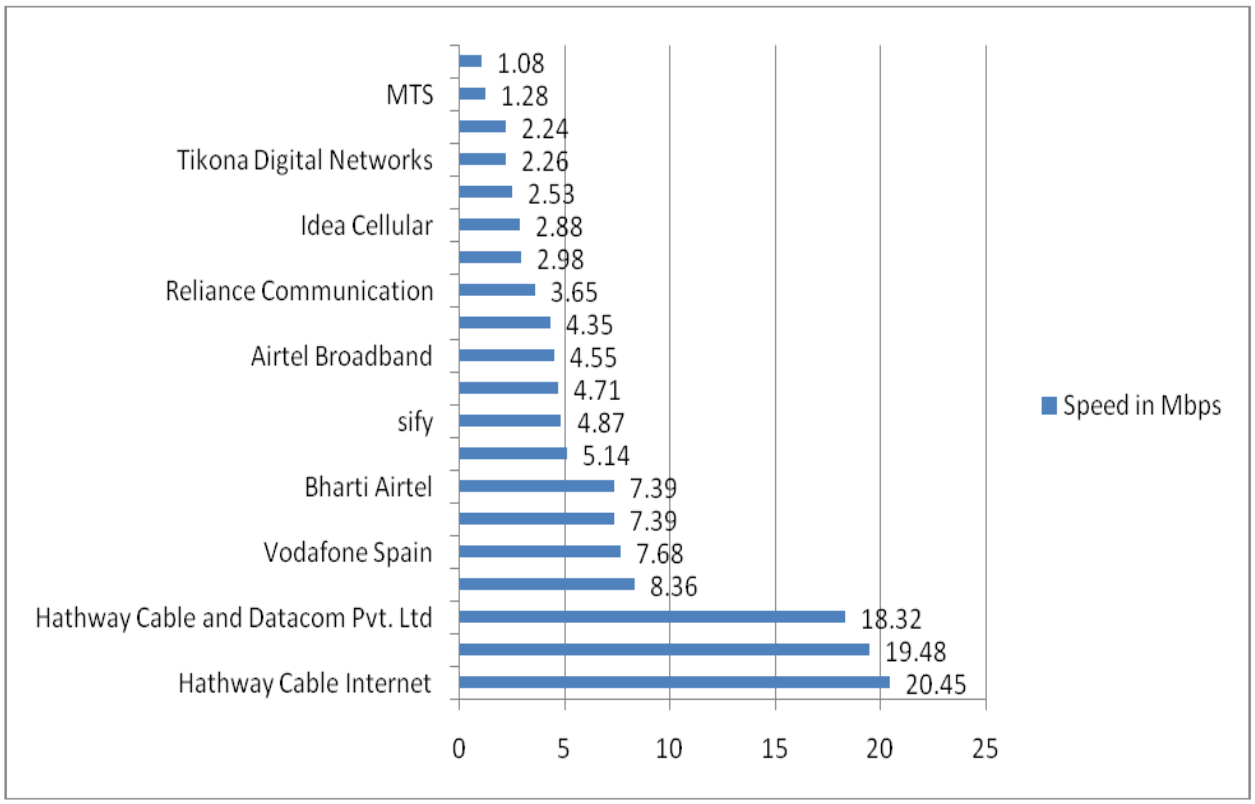


Figure 1.7: Best Internet speed providers in India [134]

1.2 Internet: Variety of Traffic

The Internet is a "network of networks" that consists of millions of smaller private, public, domestic, academic, business, and government networks that transmit data by packet switching using the standard Internet Protocol. A network is collection of interlinked computers, may be connected in the scope of local area network (LAN), campus area network (CAN), metropolitan area network (MAN), wide area network (WAN), global area network (GAN). There are certain

reasons of interconnecting 3,035,749,340 Internet users through their computers worldwide access variety of information and numerous services, such as Googling (Google is most popular website as shown in Figure 1.8) for searching some data/ information, chatting and sharing feeling through social media website's (facebook.com, twitter.com, linkedIn.com etc.) electronic mail (gmail.com, mail.yahoo.com), online buying (Amazon.com, ebay.com, snapdeal.com) and selling of goods and services, online chat, video conferencing, groupware, virtual organization, file transfer (text, picture, audio video) and the interlinked web pages and other resources of the World Wide Web. The Internet facilitates it's for 24 hours, seven weeks, 365 days in year. As shown in Figure 1.8, every month Google is getting visited by 11 million unique visitors.

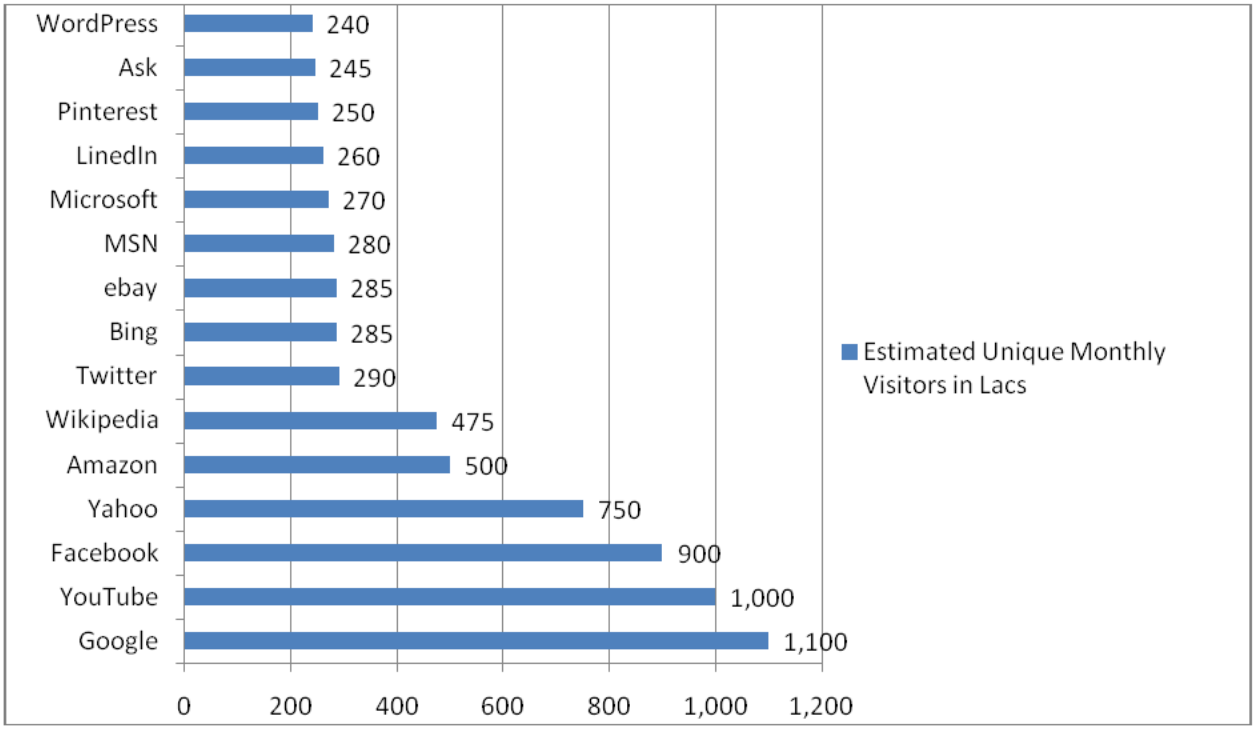


Figure 1.8: Visitors versus most popular websites [135]

Even though usage of Internet is not limited to education and military only, since last two decade it touches almost every domains of information demanded by human being, by sharing at social and business front. The different variety of Internet services changed the dimension of today's business like e-marketing, e-commerce, e-auction (forward and reverse), solving the customer's query online. The facility of e-banking, e-learning, e-governance, m-commerce and m-ERP, all are possible due to Internet. Figure 1.9 is showing the top 15 business websites visited by the Internet user, Yahoo finance is leading company with 75 million visitors followed by Forbes with 65 Million visitors. A large volume of Internet users are visiting these business websites to fulfill their business demands/ requirements.

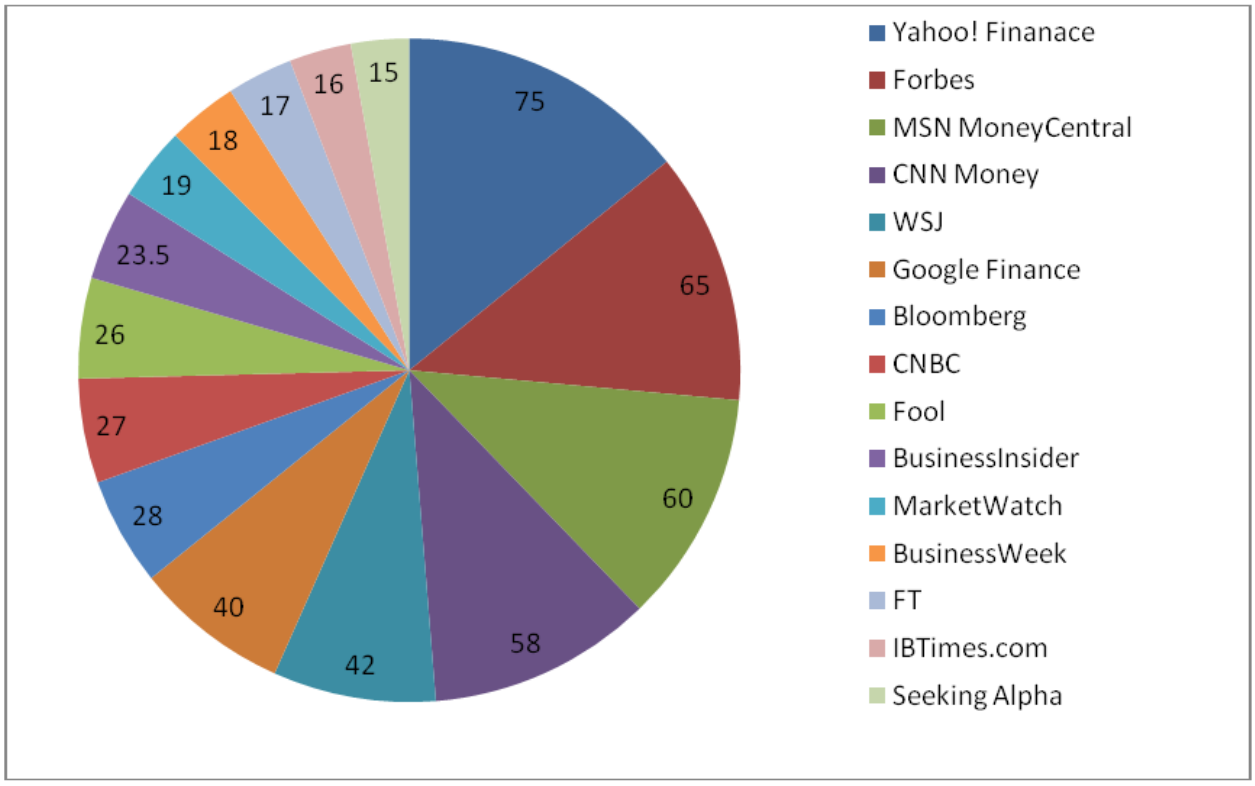


Figure 1.9: Top 15 most popular business websites [136]

In the environment of cut throat competition, where every business leader wants to use e-business as a weapon to conquer the market, through virtual organization and cloud computing (giving plenty of opportunities to most of the companies to adapt the new paradigm of e-business). The backbone of e-business is intranet (a LAN) and extranet (a WAN - network which allow the stakeholders to access the company's local resources (web services) from public environment while using their issued credentials).

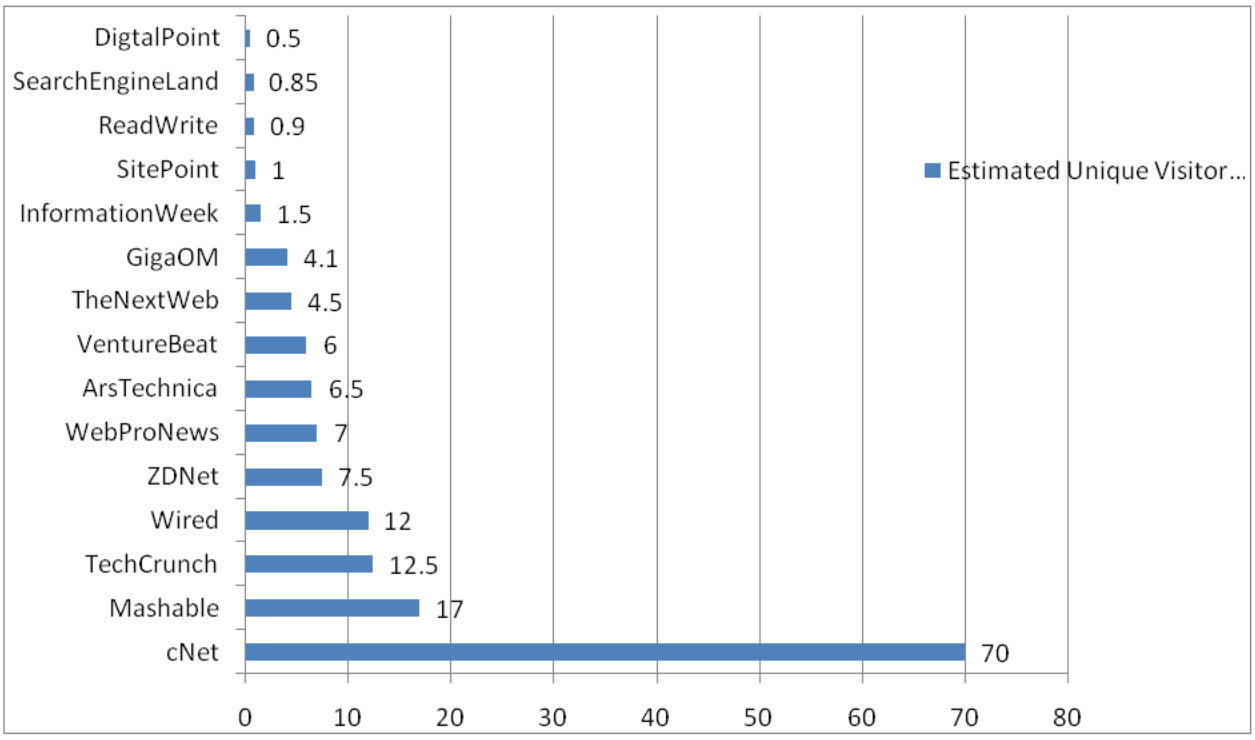


Figure 1.10: Top most 15 most popular e-Business websites [137]

To provide the accessibility of resources to millions of users, *web applications are used,* Figure 1.10 is shows top 15 popular e-business website, in December 2014, cNet had 70 million unique visitors followed by Mashable with 17 million.

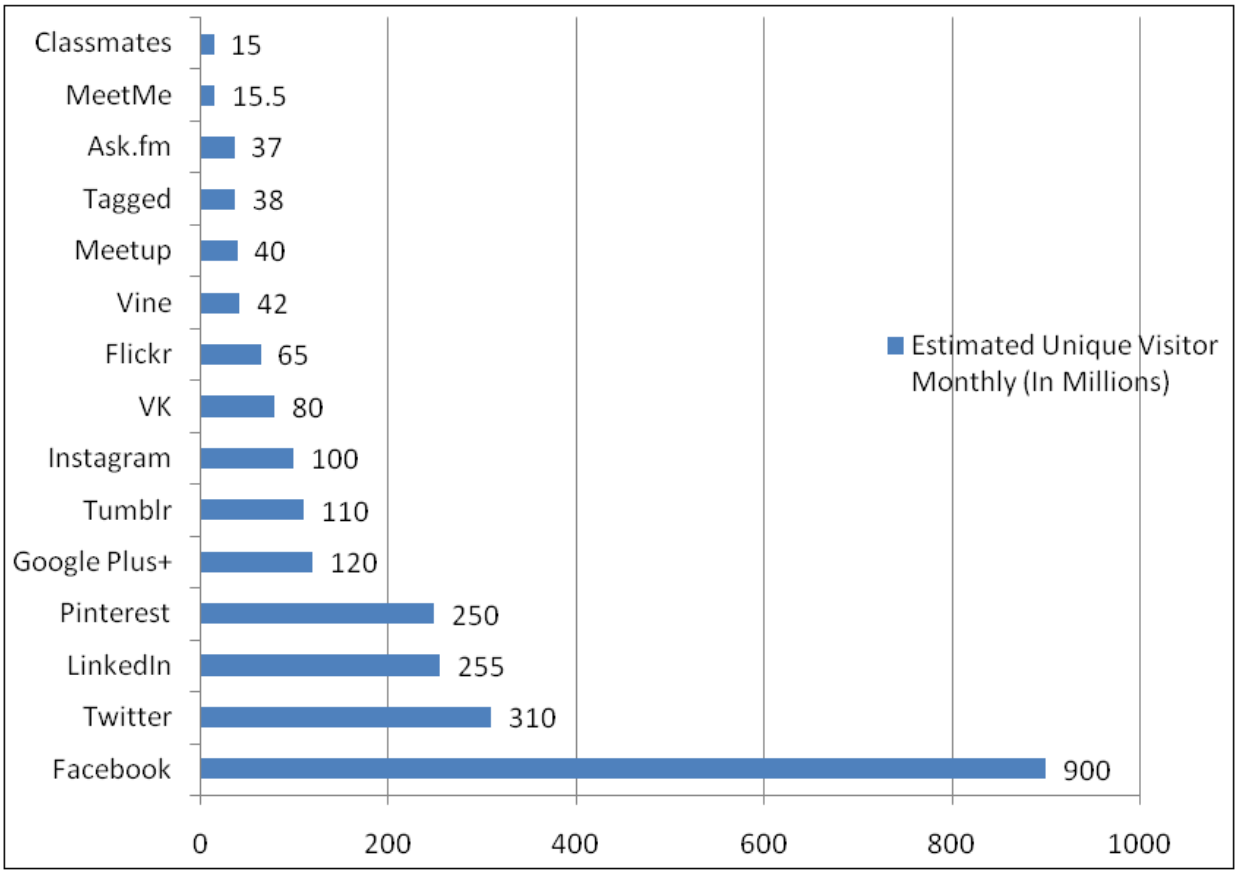


Figure 1.11: Top 15 most popular social networking sites [138]

Internet facilitates social group to get connected in better manner while binding them with their website with lots of user friendly web applications which help them to reach their friend, group or society in fast and economical way. Figure 1.11 is shows top 15 social media websites, and there are 900 million unique facebook user visiting the website monthly.

Among the huge volume of Internet users (more than 3 Billion), different websites (more than 6 billion) and variety of different web applications, speedy transactions, sometime it happens that a

vulnerable web application may cause failure of the business system. The issue of bulky data analysis and vulnerable applications is big challenge over Internet.

Even security of network resources is big challenge in such scenario where network traffic *is voluminous, with high velocity and variety of network traffic over Internet*. It became a big challenge for network administrator to find out a tiny malicious packet of few kb from terabytes data transmission over their network.

1.3 Internet: Vulnerability and Attacks

This section highlights the current trends of the Internet attacks and techniques, used by the hackers due to vulnerable network. According to Kaspersky Lab in 2014, there were 1,432,660,467 web based attacks launched from online resources located all over the world and 38.3% of computers were attacked at least once, when their owners were online. Kaspersky Lab security solutions claims that a total of 16,552,498 notifications of malicious activity to steal money via online access to bank accounts were identified and registered in 2014. Sony Entertainment Network's server hacking a cause great tension between USA and North Korea, the main page for movie title "The Interview" was not available on 22 December 2014, as the server was hacked and movie was downloaded and uploaded on many other servers.

According to [139] more than 36000 website of different countries are affected by cybercrime, cyber war, cyber espionage and hacktivism daily. Figure 1.12 shows distribution of attacks in October 2014. The USA is most targeted country, except in cyber espionage attack.

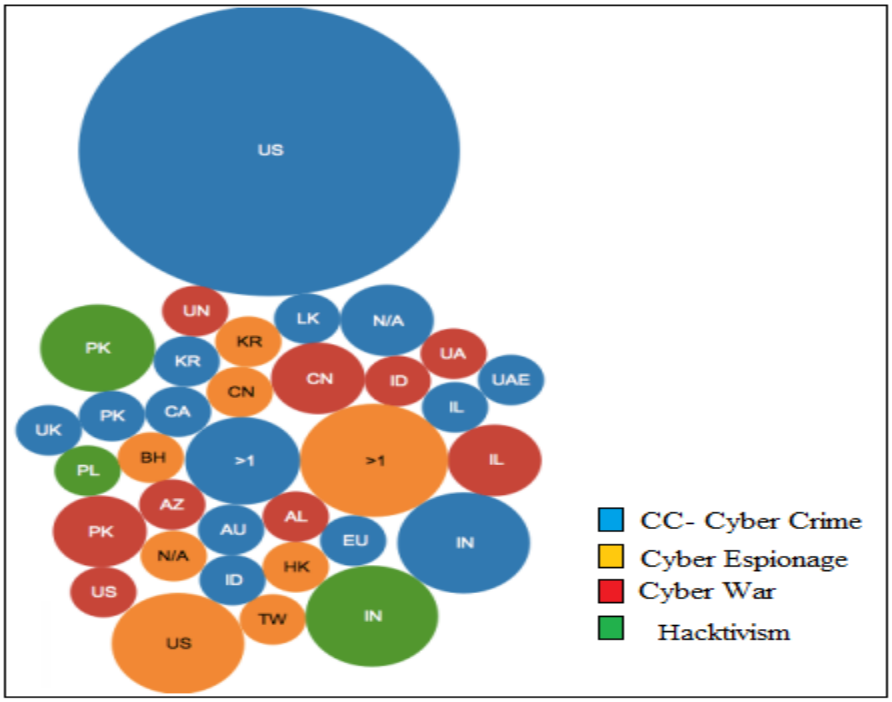


Figure 1.12: cybercrime, cyber war, cyber espionage and hacktivism attacks

Further, the study [139] shows through Figure 1.13 that the cybercrime (near to 60%) is main motivation behind these attacks, followed by the cyber espionage (17.2%) hacktivism (13.8%) at 3rd place and Cyber War (9.2%) at 4th place.

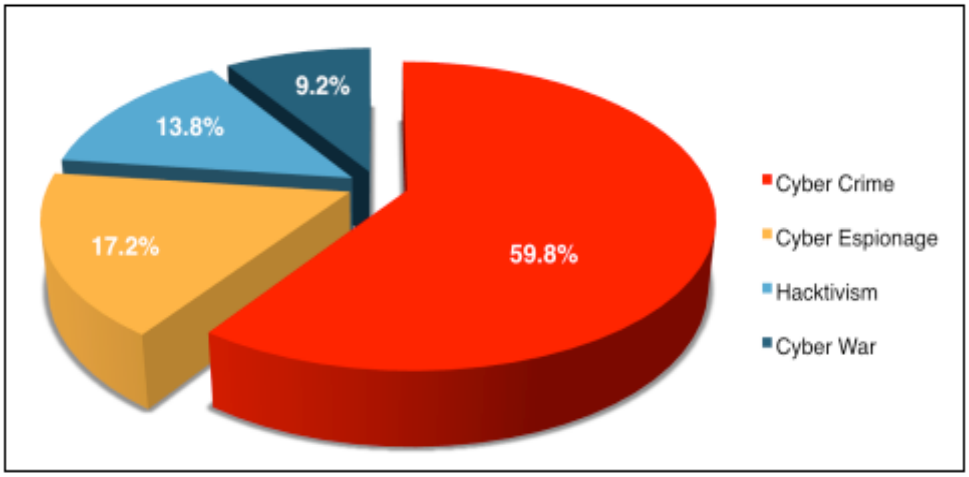


Figure 1.13: Motivation behind the Internet attacks [139]

In Figure 1.14, it can be seen that in October 2014, SQL Injection attack was used as most popular attack techniques. The defacement attack techniques and DDoS attacks contributed 11.5% and (3.4%) respectively.

Further, Figure 1.15 shows that in October 2014, it is industry oriented websites which were targeted most (28.7%), followed by governmental websites (21.8%). Even single individuals were also targeted (used as a part of botnet) and ranked third (10.3%), slightly ahead of attacks against organizations (9.2%).

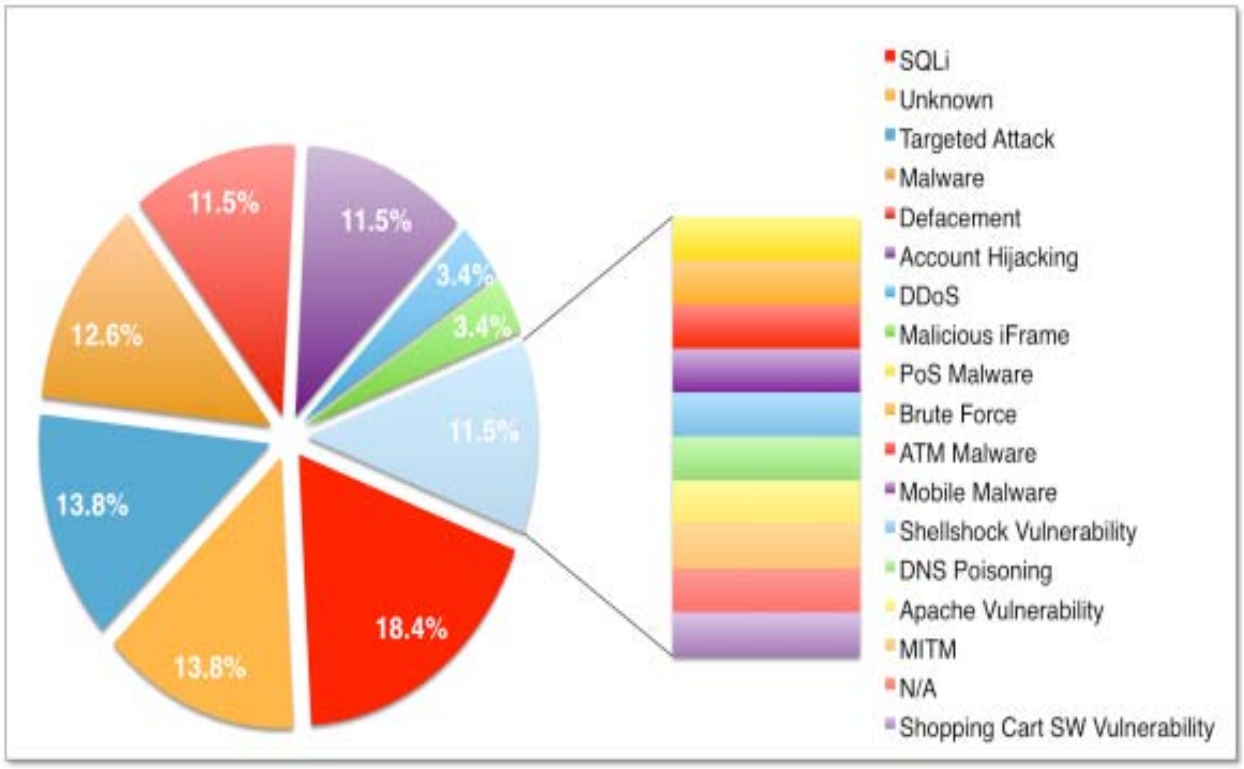


Figure 1.14: Attack techniques used by attacker [139]

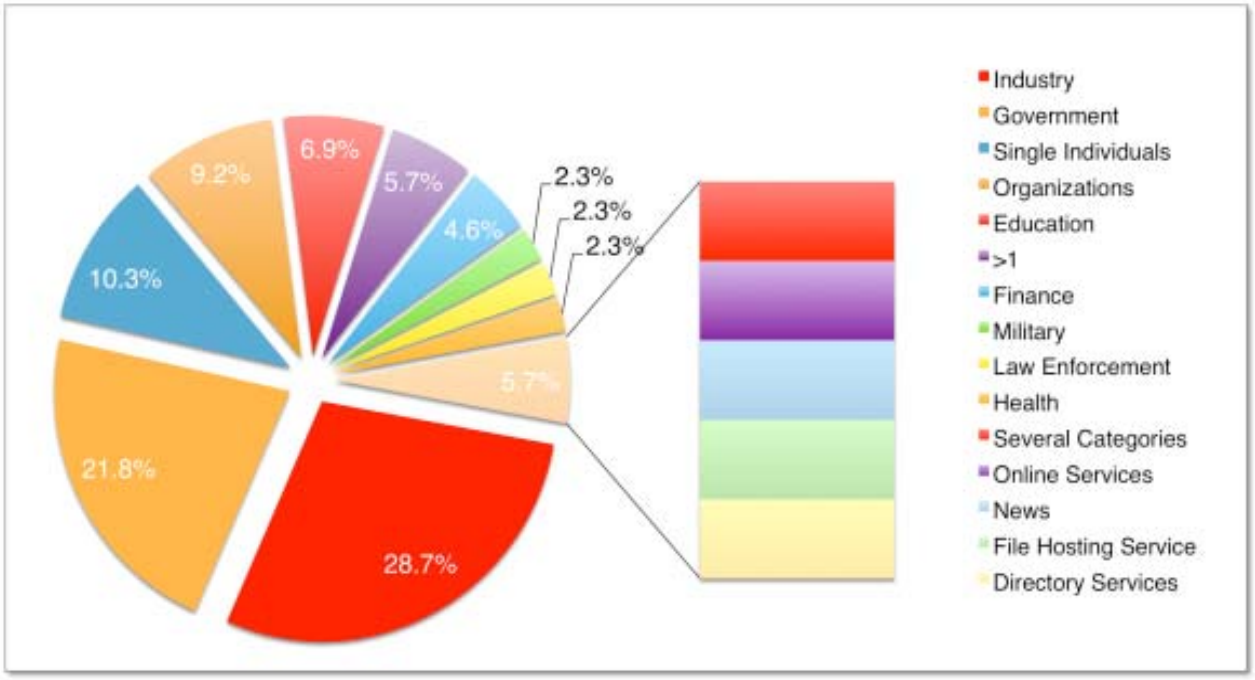


Figure 1.15: Distribution of targets of Internet attacks [139]

Figure 1.16 display a deeper view at the distribution of the industrial targets, shows a majority of E-Commerce (24%) websites targeted by the attackers.

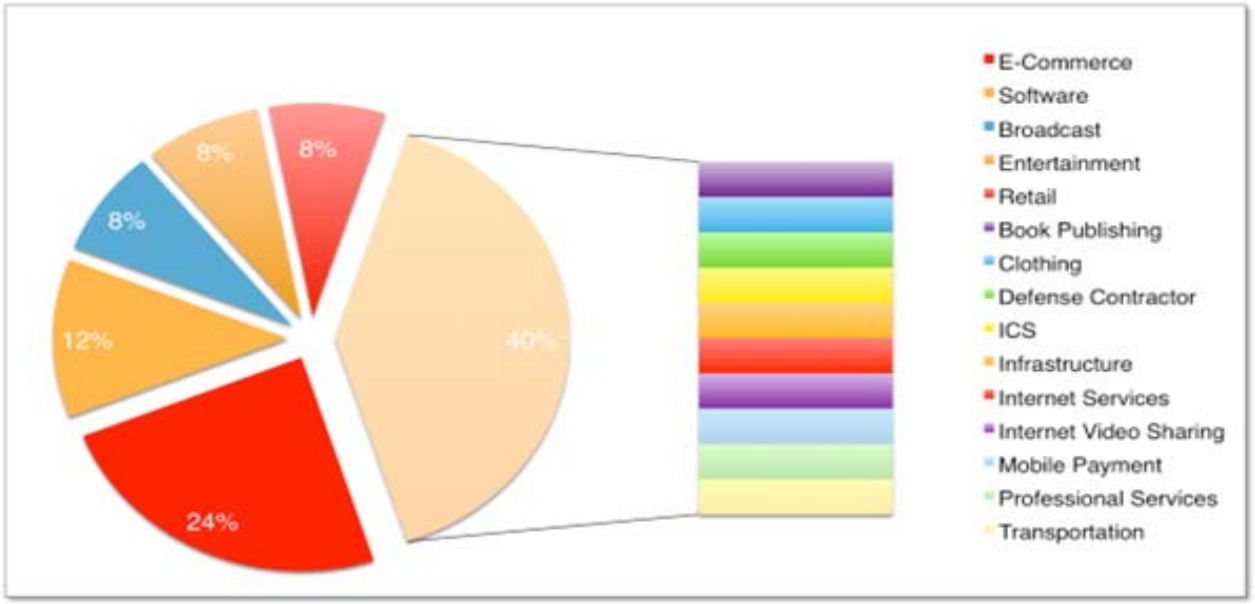


Figure 1.16: Targeted industry [139]

Figure 1.17 depicts that Brazil is most affected country having maximum number of web banking user, who were attacked by the attackers to steal their money. In this category, Brazil is followed by Russia and Germany. India ranked 4th in this category.

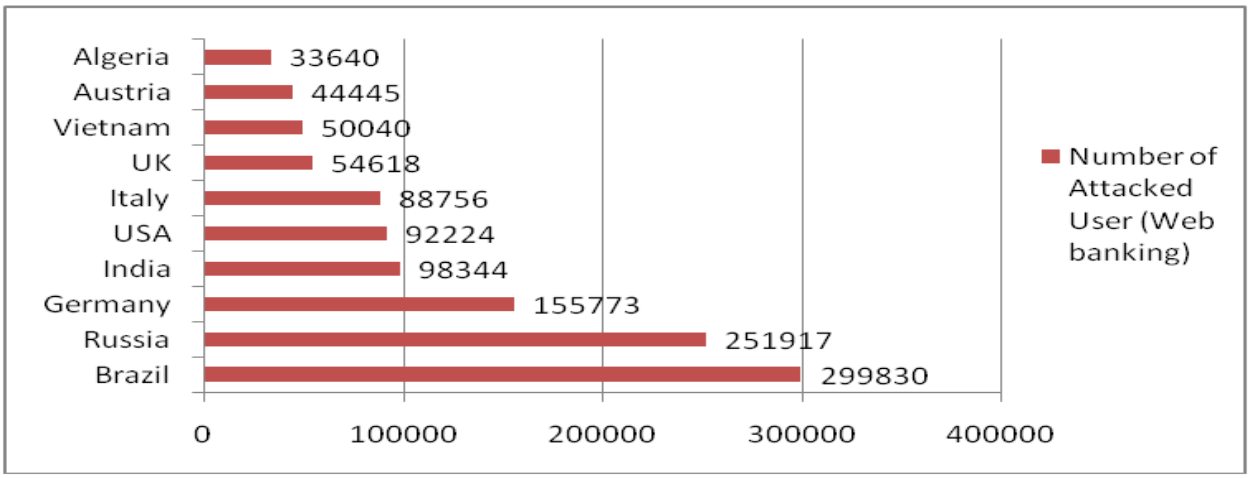


Figure 1.17: Web Banking attack [91]

All this happened throughout the year with Web banking users due to top 10 banking malwares as enlisted in Table 1.1 used by the attacker.

Table 1.1: Top ten banking Malware list [91]

	Name	Number of attacked users
1	Trojan-Spy.Win32.Zbot	742,794
2	Trojan-Banker.Win32.ChePro	192,229
3	Trojan-Banker.Win32.Lohmys	121,439
4	Trojan-Banker.Win32.Shiotob	95,236
5	Trojan-Banker.Win32.Agent	83,243
6	Trojan-Banker.AndroidOS.Faketoken	50,334
7	Trojan-Banker.Win32.Banker	41,665
8	Trojan-Banker.Win32.Banbra	40,836
9	Trojan-Spy.Win32.SpyEyes	36,065
10	Trojan-Banker.HTML.Agent	19,770

Zeus (Trojan-Spy.Win32.Zbot) is the most widespread banking Trojan. It is ranked 1st in the top 10 banking malware Trojan for the year 2014 followed by Trojan-Banker.Win32.Che-Pro, and Trojan-Banker.Win32.Lohmys. Both families have the similar utility and are spread via spam messages with a theme related to online banking. Many key researchers discussed about the attacks, worms and vulnerabilities as follows:

The Internet remains the major source of malware for users in most countries. Malicious websites are deliberately created by cybercriminals; infected sites include those with user-contributed content (such as forums) as well as legitimate resources that have been hacked.

1.3.1 Denial-of-service (DoS) attacks [114][120]: Due to this attack the intended users suffer a lot and cannot use their allocated resources. DoS maybe the worst, and most difficult to address. These are worst, because they're very easy to launch, difficult to track. A single computer can be used to send many illegitimate requests or many computers can be used to send many illegitimate requests to just exhaust the server.

1.3.2 Executing Commands Illicitly- An unauthorized user escalate his rights over a server and is able to execute commands of an administrator, which is clearly unwanted.

1.3.3 Unauthorized access - The goal of these attacks is to access such resources of network which are not permitted to users but still they (attacker) are able to manage the accessibility of network resources without taking the permission in legal manner.

1.3.4 Destructive behavior - Negative behavior of the attackers, there are two major categories of this attack. Data diddling – Changing the contents of a record after or before processing the data. Data destruction - these attacks are simply used to delete the data/ records.

1.3.5 Confidentiality Breaches – everybody wants confidentiality of their data, on a computer a compromised user can be threat for all other user.

1.3.6 Financial Crimes - Here attackers perform unlawful conversion of the ownership of property. Financial crimes involves frauds like (credit card fraud, cheque fraud, medical fraud, mortgage fraud, securities fraud (including insider trading), corporate fraud, bank fraud, payment fraud, theft, tax evasion, embezzlement, identity theft, bribery, money laundering for terrorist activities, forgery and counterfeiting.

1.3.7 Cyber Pornography - Cyber pornography [129] is about using the cyberspace to display, distribute and publish pornography contents, especially depicting children involved in sexual acts with adults. Cyber pornography is a criminal offense, categorized as causing harm to persons. Even the attacker use the pornography contents as baits while distributing virus/ worms/ trojans through steganography in their pornography material which is available free over websites.

1.3.8 Online Gambling- In India gambling is unlawful activity. The Internet has however made gambling easier. A myriad of sites have sprung up in recent years offering countless gambling opportunities: online lotteries, bingo, or the more 'traditional' card games such as blackjack or poker, providing opportunities to gamble in real-time, 24/7.

1.3.9 Email Spoofing- Attacker, used e-mail spoofing while forging the header of E-mail to change the actual source of origin.

1.3.10 Email Bombing - an email bomb is an outline of net abuse consisting of sending enormous volumes of email to an E-mail address to overflow the e-mailbox or bring down the server where the email is hosted.

1.3.11 Email Frauds- An attacker through email fraud intentionally deceive an email receiver for personal gains or to harm him/her through email.

1.3.12 Salami Attacks - salami attack is about small attacks sum up to one major attack that can go concealed due to the nature of this type of cybercrime, such as taking advantage of round of money on each financial transaction.

1.3.13 Virus/Worm Attacks- Computer virus is un-wanted program for any user but attackers used these to damage the hardware, software or files. To spread a computer virus from computer to another computer, it requires human intervention. On the other side, worm is similar to a virus by design and unlike in capability as worm has the capability to travel, replicate itself without any intervention.

1.3.14 Trojans & Key loggers- Trojan is an illegitimate software which acts as useful software but will actually do damage on installation or running on victim's computer. Some Trojans are designed to create a backdoor on victim's computer. Keyloggers are such hidden spy tool which are regularly used, to log all the key strokes on a victimize machine.

1.3.15 Cyberwarfare - Cyberwarfare is Internet-based clash linking politically motivated attacks on information and information systems. Under Cyber warfare attacks the attackers can hinder official websites and networks, steal or alter classified data, disrupt or disable essential services, and cripple financial systems.

1.3.16 Cyber Terrorism - The usage of Internet to cause destruction and harm for personal objectives (political or ideological) in the form of terrorism, like ISIS is using Internet to channelize its terrorist activities and to terrorize the world.

1.3.17 Hacktivism - Politically or socially motivated act of hacking is called Hacktivism, the individual who involved in hacktivism is called as hacktivist.

1.3.18 Cyber Stalking - Harassing a victim by using electronic means is a crime and is called as Cyber stalking, such as sending an e-mail or instant messaging (IM), or messages posted to a social Website or over a blog (discussion groupware). A cyber stalker relies upon the secrecy given by the Internet to allow them to stalk (follow) their victim without getting detected.

1.3.19 Cyber Defamation - The act of publishing (over a website, discussion groups, Intranets, bulletin boards or sending emails) of defamatory material against another person with the help of computers or internet is known as Cyber defamation.

1.3.20 Web Defacement - The act of changing the visual appearance of the site or a webpage by an attacker (typically system cracker, breaks web server security and replace the hosted website with one of their own) without taking any permission of the owner.

1.3.21 Intellectual Property Crimes - These includes copyright infringement, software piracy, theft of computer source code, trademark violations.

1.3.22 Forgery - In this, cyber criminals use the sophisticated computer, printers and scanners to produce counterfeit of currency notes, postage and revenue stamps, mark sheets, academic certificate, etc.

1.3.23 Internet Time Theft - In this an unauthorized person use the Internet hours without paying anything or/for which some amount is paid for by some another person.

1.3.24 Web Jacking- In this attack hacker gains access and control over a website. The attacker can even change the information on that Website.

1.3.25 Bots- This is a new technique used by the attackers to create a network of compromised nodes (compromised node is called a bot and network of compromised nodes is called as botnet) in which compromised node are self-propagating the malware intended to infect a host and connect back to a central server or servers that act as a command and control (C&C) center for an entire botnet. Attackers are using botnets, to launch big, "remote-control," flood-type attacks against their victim machines/ target(s). Botnet are very dangerous, having the capability of worm (self-propagating), ability to log keystrokes, collect passwords, gather financial information, relay spam, open back doors and launch DoS/ DDos attacks.

1.3.26 SQL Injection attack - In this attack, an attacker compromises the database server while using code injection technique to attack data-driven applications, in which malicious SQL commands are inserted.

1.3.27 DNS poisoning - DNS poisoning, is about the changing domain name system (DNS) cache mischievously while changing an Internet server's DNS table by replacing mischievous address of another Internet server's address with that. In this visitor want to visit a particular website, but diverted towards a mischievous website.

1.3.28 Man-in-Middle-Attack - In this attack, an attacker has the ability to observe and amend or insert messages in a communication between two nodes and intercept all relevant messages passing between these two victims.

1.3.29 Cyber Espionage - Cyber spying, or cyber espionage, is an act or practice of revealing secrets (personal, economic, political or military) without taking the permission of the owner of the information.

1.3.30 Zero day attacks [104]- In zero day attacks, the attackers exploits the zero day vulnerability (which refers as a hole in software, unidentified by vendor) before the vendor becomes aware of it and fixes it. P2P network attack [115], SMTP attack due to anomaly [112], email attack [113], cross site scripting (XSS) attacks [119] and DDoS attack [114], [120] are main cause of damage.

1.4 Network Security

Section 1.3 elaborated on various types of attacks and techniques used by the attackers. Security consists of the provisions made in an underlying computer network infrastructure for safety policies adopted by the network administrator to protect the network and is allowing network-accessible resources from unauthorized access. “Network security” [38] is combination of two words “Network” and “Security”. Network may be defined as interconnected nodes, with a motive of optimum utilization of its resources under the scope of network. Here “resources” are concerned with electronics devices like data storage, peripherals, application servers, software and bandwidth.

Security means assuring the availability of the authorized resources to a legitimate user keeping it confidential and integrated without any damage. Network security relies basically on AAAAIC.

Availability: It assure the availability (ability to use the information or resource as per need/desire) of intended resources for the legitimate user.

Authentication: It assure the verification and validation of the users based on their credentials (user_name and password).

Authorization: The authorization (the process of enforcing policies) refers to the privileges of authorized users, who is validated through their authenticity, are entitled to use the network resource, or services as per their privileges.

Accountability: The accountability measures the resources, a user (authorized user) used during access. Accounting is based on logging of session statistics and is used for billing, authorization control, resource utilization, trend analysis, and resource planning activities for optimum utilization of network resources.

Integrity: Integrity refers to the trustworthiness of data or resources in terms of preventing improper and unauthorized changes.

Confidentiality: Confidentiality is assuring the privacy of the use of network resources as per network policy.

Today every organization needs to decide for itself, between the two extremes of total security and total access. A network policy needs to be articulated, and then define how that will be enforced with practices. Everything that is done in the name of network security, must enforce that policy uniformly.

1.5 Network security approaches

To stop the attacks discussed in section 1.3, network administrator usually take the help of following network security approaches.

1.5.1 Firewall - A firewall system [100] [88] defines a single access point for traffic between a private network and public network. It inspects network traffic passing through it, and denies or permits passage based on a set of rules. A firewall's basic task is to regulate flow of traffic between computer networks of different trust levels. There are three major classes:

a) Packet filter firewalls examine endpoint identifiers in datagrams passing through a link to determine if each packet should be allowed to proceed. They can filter on IP addresses and can also block specific ports and protocols but cannot prevent spoofing.

b) Stateful inspection firewalls have the capability to track the status of a connection. If an arriving ACK packet claims to be from an established connection, the stateful firewall would deny it if it did not have a record of the three-way handshake ever taking place.

c) Proxy firewalls act as a mediator between two devices across the firewall. They terminate flow at one side of the firewall, provide some proxy service to examine the data within the flow, and then re-create the flow in the network at the other side of the firewall.

While a firewall provides the first line of safety against the attacker, many catastrophic incidents have resulted in failure of firewall, to identify and to stop a malicious data packet. It is difficult for firewall to identify malicious packet from high volume traffic which is coming with new exploit. Once the firewall is penetrates whole network comes under the risk. But still personal firewall [88] as well as network firewall is useful.

1.5.2 Intrusion Detection System (IDS) - In web access case firewall has to allow incoming traffic at port 80 or 8080 from the Internet without any restriction. To run these services the

programming parts are often complex. This complexity some time leads the programming bugs, which can be exploited by the expert intruders [106].

An IDS [89] generally a monitoring system detects unwanted manipulations of computer systems specifically the log files, mainly through the Internet. The manipulations may take the form of attacks by crackers. An intrusion detection system is used to detect several types of malicious behaviors that can compromise the security of a computer system [106]. This includes network attacks against vulnerable services, anomaly based [92][93] data driven attacks on applications in real time [101][107], host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware.

a) A network intrusion detection system is an independent platform which identifies intrusions by examining network traffic and monitors multiple hosts. Network Intrusion Detection Systems (NIDS) gain access to network traffic by connecting to a hub, network switch configured for port mirroring, or network tap. An example of a NIDS is Snort.

b) A protocol-based intrusion detection system consists of a system or agent that would typically sit at the front end of a server, monitoring and analyzing the communication protocol between a connected device (a user/PC or system).

c) An application protocol-based intrusion detection system consists of a system or agent that would typically sit within a group of servers, monitoring and analyzing the communication on application specific protocols. For example; in a web server with database this would monitor the SQL protocol specific to the middleware/business-login as it transacts with the database.

d) A host-based intrusion detection system consists of an agent on a host which identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files, access control list databases) and other host activities and state.

e) A hybrid intrusion detection system combines two or more approaches. Host agent data is combined with network information to form a comprehensive view of the network. An example of a Hybrid IDS is Prelude.

1.5.3 Reactive IDS/ Intrusion prevention system (IPS) - A step ahead to IDS, IPS not only detect the malicious activity based upon log file etc. but also takes the action against the attack. An intrusion prevention system is a computer security device that monitors network and/or system activities for malicious or unwanted behavior and can react in real-time to block or prevent those activities. Network-based IPS, for example, will operate in-line to monitor all network traffic for malicious code or attacks. When an attack is detected, it can drop the offending packets while still allowing all other traffic to pass.

But still to make the policy to detect the new exploit and to react against them is really a difficult job, action against false positive where the data transmission rate are in terabytes. To capture all packets and investigate these packets deeply, need lots of time and high computational devices.

1.5.4 Honeynet - A honeynet is a network of honeypots, machines that are intended to be compromised, designed to provide the system administrator with intelligence about vulnerabilities and compromises within the network [105]. A honeynet is placed behind a reverse firewall that captures all inbound and outbound data. The reverse firewall limits the amount of malicious traffic that can leave the honeynet. This data is contained, captured, and controlled. Any type of

network device can be placed within the honeynet, to include configurations identical to production machines elsewhere on the network. These standard production systems are used on the honeynet in order to give an attacker the alluring look and feel of a real system. Since honeypots do not offer any legitimate services to Internet users and the Internet addresses of the honeypots are not openly known, most traffic on the honeynet is doubtful. These qualities result in an unusually high signal to noise ratio.

1.5.5 Information visualization: Information visualization (InfoVis) is a branch of computer science concerned with modeling complex data using interactive images, which is an emerging field since last decade. Information visualization is also one of the most accurate security approach which helps the network administrator to get the data in graphical form for quick analysis, understanding and respond to emerging threats and vulnerabilities.

InfoVis approach provide the dataset of network traffic in graphical, pictorial form which can make administrator's response to security threats faster and more effective.

1.5.5.1 Challenges in network data visualization

Network traffic data visualization is a powerful technique that taps into the high bandwidth visual recognition capability of the human and is well suited to resist network attack. Unfortunately, most information visualization systems are designed without a clear emphasis on protecting the human from malicious activity.

a) **Volume of network data-** Every organization is having their own website, running their own web server to share the information through http. As discussed earlier, huge amount

of data is transmitted per second over the Internet, finding out a malicious activity in such scenario is like seeking the needle from the sea.

b) **Failure of firewall and IDS due to new attacks and weak policies.** Sometime firewall misjudge the packet and let it go in the network which may have some file with a logic bomb or a Trojan horse. Generating false positive and false negative alarm by snort (IDS) requires network administrator to observe the data continuously.

c) **Variety of new attacks and exploit:** Every day network administrator faces new kind of virus, worms and spyware backdoor or Trojan, attacks in spoofing mode.

d) **Lack of Resources:** Network Administrators are supposed to scan gigabyte to terabytes of data per second. To scan each packet to do real time analysis and to take necessary decision timely, high computing infrastructure is required.

Further, network administrator do not have much effective network data visualization tools to draw graphs of all incoming packets in different format like:

(a) IP Address wise

(b) Port wise (application oriented)

(c) Protocol wise (services oriented)

(d) Main resource wise and priority wise.

1.5.5.2 Data visualization - To achieve the information visualization target one must understand the basics of data visualization:

Data visualization - Visualization is the graphical presentation of information with the goal of providing the viewer with a qualitative understanding of the information contents. Graphical

presentation may entail manipulation of graphical entities (points, lines, shapes, images, text) and attributes (color, size, position, shape).

1.5.5.2.1 Characteristics of Data

Data may be numeric, symbolic (or mix), scalar, vector, or complex structure. Data is having various units like, Kb, MB, GB and Tb etc. Data may be of discrete or of continuous type. It may be accurate or approximate, ordered or non-ordered, disjoint or overlapping, binary, enumerated, multilevel, independent or dependent, multi-dimensional, single or multiple sets, may have similarity or distance metric and may have intuitive graphical representation.

Graphical entities and attributes: In data visualization, entity and data attributes must be consider

- Entity: point, line, polyline, glyph, surface, solid, image and text.
- Attribute: color/intensity, location, style, size, relative position/motion

A good data visualization system should possess following attributes:

- Accuracy: It should give results correctly for quantitative evaluation.
- Efficiency: Able to minimize data-ink ratio and chart-junk, show data and maximize data-ink ratio.
- Effectiveness: Viewer must get it easily (ease of understand/ interpretation).
- Aesthetics: Must not offend viewer's senses.
- Adaptability: can adjust to serve multiple needs.

1.5.5.2.2 Visualization Common Techniques

To display the data in meaningful way various common and conventional techniques are popular such as charts: bar or pie, graphs: good for structure, relationships, Plots: 1- to n-dimensional, Maps: one of most effective, Images: use color/intensity instead of distance (surfaces), 3-D surfaces and solids, translucency and animation. Furthermore there is a variety of conventional ways to visualize data viz. tables, histograms, pie charts and bar graphs.

The much better, profound, creative and absolutely fascinating ways to visualize data schemes are discussed below :

Thumbnail toolbar: The thumbnail toolbar provides a real-time overview of each visualization window in a thumbnail size display. Doubling as a menu, users may bring the full size window by clicking on a thumbnail [73].

Scrolling text display: The scrolling text display presents network packets, one per horizontal row, in a user selectable encoding [73].

Parallel coordinate plot display: This visualization uses the parallel coordinate plot technique to display scaled values [74] [75].

Detail* display*: The detail window displays the selected packet's contents in a traditional hex/ASCII format [73].

***Glyph-based animation display*:** The glyph-based display combines three display panes to animate any two attributes of network traffic [73].

Binary rainfall visualization: The binary rainfall visualization displays packet contents, one per line [73].

Scatter plot display: The scatter plot visualization allows users to select any two attributes and plots them on a traditional X, Y display. Header field values are scaled to match the dimensions of the display window well explained by Tom Goldring [82].

Byte frequency display: The byte frequency visualization displays the presence and frequency of bytes within each packet [73].

1.5.6 Types of network traffic visualization

In this sub-section, contribution of the key researchers is discussed, based on the scope of their proposed solution viz. location based (geographical) visualization, conceptual (types of network traffic) based visualization and scheme (plot) based visualization.

1.5.6.1 Location based visualization: In this type of visualization represents the physical locations of nodes (machines) and their connectivity as well as relation among them. A node represents the location of a machine and an edge represents the direction.

1.5.6.2 Conceptual based visualization: In this type to network traffic visualization category network data is shown w.r.t. a node- link instead of location (geographic location) of the node. Hence, in conceptual based visualization, machines are more readable and meaningful, able to mention that which particular type of data is being received and emitted by a node in a network.

1.5.6.3 Scheme (plot) based network visualization:

Scheme (plot) based visualizations says about the presentation of data in traditional graph form viz. line graph, histogram, scatter plot etc. Generally Scheme (plot) based visualizations represent

a single network point in a network. It also have some approaches to present a grid of several plots.

Visualization as silver line: In other words *visualization* is fifth V, presenting in this work to monitor and control the rest 4 Vs: Volume (used for data size), Variety (used for different web/ network application data), Velocity (fast network data transmission speed) and Vulnerable (prone to attack network activities). So this thesis (study) advocate the usage of visualization based security solution (VizSec) or infoViz and result in an integrated network traffic visualization system (INTVS), which is very promising and to facilitate the network administrator in very effective manner and demonstrated in chapter 4 & 5.

1.6 Organization of the thesis - Rest of the structure of this thesis is as follows: literature review is presented in Chapter - 2. In chapter - 3, the gap analysis, problem formulation, objective of the research work, research methodology and framework architecture and composition of INTVS are discussed. Further, in the same chapter, UML *modeling of INTVS*, in which the static, architectural and behavioral aspects of INTVS are discussed. In chapter - 4 the implementation of INTVS framework is discussed in real time and off-line. Chapter - 5 discusses the validation of results, reports and reliability analysis of INTVS. Chapter - 6 discussed the contribution of the research work, its limitations and future scope.

Conclusions:

This chapter highlighted the concept of 5 Vs w.r.t. Internet traffic. Five Vs has been discussed w.r.t. ***voluminous*** and ***velocity*** (high speedy transmission of data), ***variety*** (different web-applications/ services thorough different media), ***vulnerability*** and ***VizSec***. Volume specified the huge number of Internet users, accessing huge number of websites and transmitting huge Internet data. Velocity talked about speed of data transmission which is facilitating the end user to get response in minimum time. Variety dealt with different types of application software, Web-services, Websites, vulnerability, threats and attacks while Vulnerability elaborated the weakness of different network resources which posed the threat to the networks from the different attacks. Fifth 'V' is VizSec (Visualization based security solution) which assures the mitigation of these threats and attacks in easy manner and having the advantages over the conventional security solution. Basic types of visualization schemes are also discussed w.r.t to their utility in location (geographic based), conceptual (types of network traffic) abstract and type of presentation. Fifth 'V' visualization based security solution inspired this study and gives the direction to work in the VizSec domain.

Chapter 2

Literature Review

Previous chapter gave introduction to growth of Internet based on 5Vs, followed by network attacks, role of visualization and techniques. This chapter discuss work carried out by other researchers and find out gaps in present study which eventually this work is going to address. This chapter is divided into three sections a) section 2.1 discusses VizSec (Visualization based Security solution) b) section 2.2 discusses UML based analysis and designing of security solutions c) section 2.3 discusses use of fuzzy based approaches to improve the reliability of the security systems.

2.1 VizSec: In this section work of many key researchers is discussed pertaining to

a) Security through data visualization

b) Security through data mining techniques

c) Security through data visualization based on data mining

2.1.1 Security through data visualization:

Swing [1] developed a tool, named as Flodar, able to display high-level view of the network and servers available within a network. In this scheme, data servers are shown by rectangular standing bars in different circles as shown in Figure 2.1.

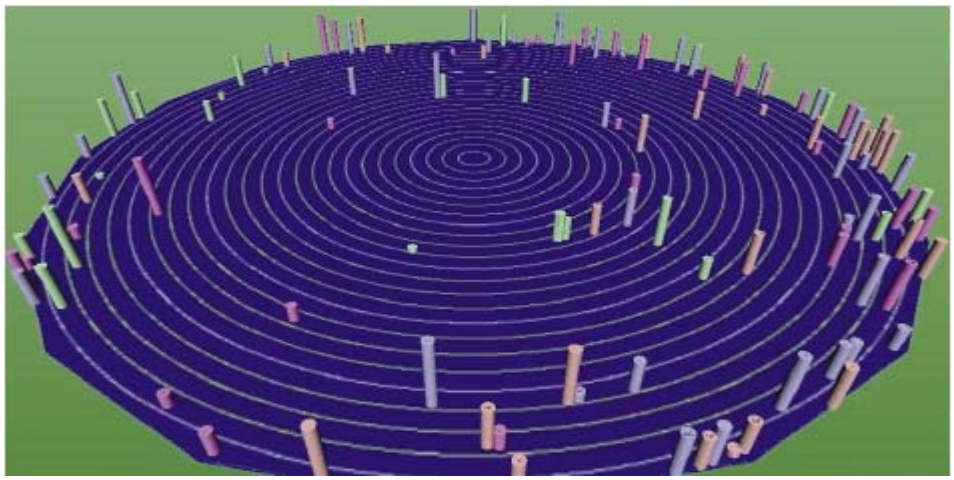


Figure 2.1: Platter display [1]

Estrin et al. [2] developed a tool named as 'Nam'. This tool works in offline mode. 'Nam' is capable of showing the packet-level animations, from network simulator (ns) simulation package, which generates traces. These trace files contains required information for static network layout and for dynamic events. This tool is equipped with video cassette recorder (VCR) like buttons as shown in Figure 2.2, which are helpful to view a particular portion of the data.

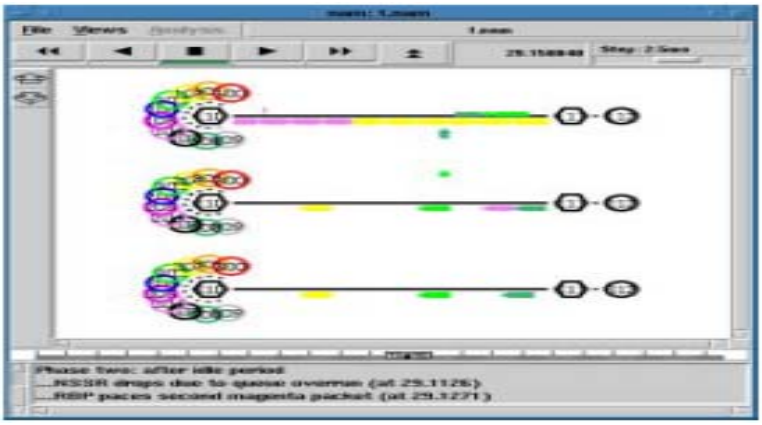


Figure 2.2: Nam [2]

Yin et al. [5] developed 'VisFlowConnect -IP', a network security tool, which is a supplement part of security incident fusion tool (SIFT). The main purpose of 'VisFlowConnect-IP' is to

represent the flow between internal networks visually. It is also facilitating filtering and drill-down. Yin et al. used converter and anonymizer for investigating netflow events (CANINE) tool to synchronize the different Netflow formats (the Cisco NetFlows and Argus NetFlow). VisFlowConnect-IP used parallel coordinate plots to display the connectivity between different machines based on IP address. The VisFlowConnect-IP used the parallel coordinate plots to represent the connections between IP addresses. There are three parallel vertical axes used, central axis represents the IP address of source machines, and the other two axes used to display destination and subnet machines IP address shown in Figure 2.3. In this parallel connection scheme, the symmetrical connections indicate the regular established session between two machines and asymmetric connections point out scanning activities. Thickness or transparency of the lines indicates the uploading /downloading links.

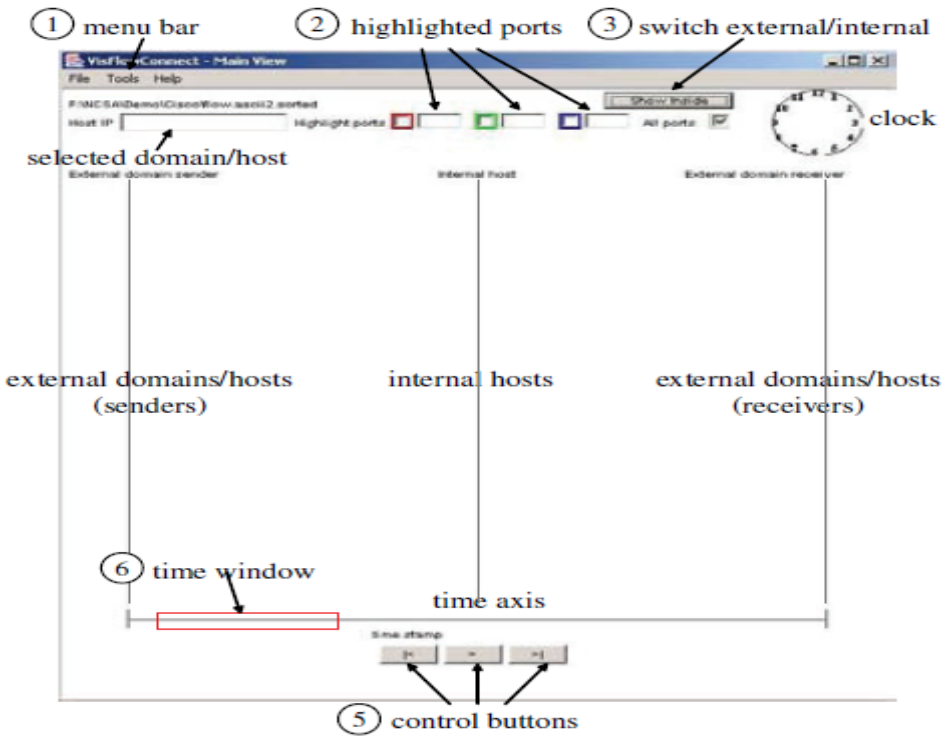


Figure 2.3: VisFlowConnect-IP [5]

Ball et al. [4] developed a tool named VISUAL (Visual Information Security for Administrator Live). The VISUAL is capable in visualizing the connectivity between centric network host and 10,000 external hosts in 3D grid square display as shown in Figure 2.4. VISUAL is also equipped with multiple port view, interactive filtering based on protocol (TCP, UDP and ICMP) and time. VISUAL is using tcpdump and ethereal for data source and generating 3D grid square display. However, VISUAL is limited to network layer visualization only. In VISUAL there is no real time data visualization facility.

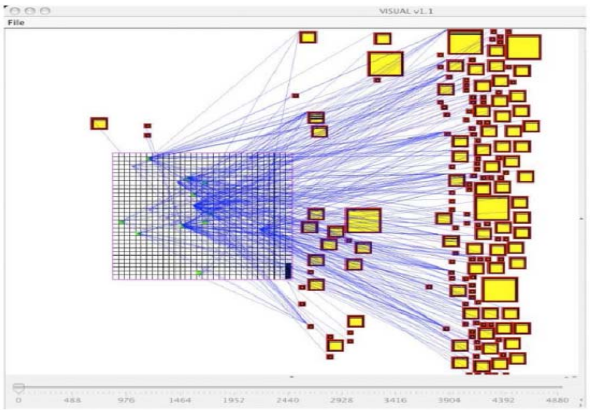


Figure 2.4: VISUAL [4]

Fink A. et al. [8] proposed a network visualization tool named as 'Portall', a prototype of a Network Eye framework. This prototype is an extended version of previous prototype of Network Eye framework of VISUAL.

TNV (The Network Visualizer or Time-based Network Visualizer) is developed by Goodall et al. [11] in Java, a network security portable solution can run on any platform like Windows, Linux etc. The philosophy behind the TNV is to help the network analyst especially intrusion detection analyst to analyze the details of an attack without losing the context as TNV provides the view of packet level data. The matrix displaying network activity of hosts over time, is a main component

of TNV, shows the connection between hosts as shown in Figure 2.5. All hosts suppose to be colored according to its level of activity and multiple linked views w.r.t. port activity and details of packets. TNV is an open source network security visualization tool.

Figure 2.5: TNV [11]

Kim et al. [6] developed 'NetViewer'. There are five main component of 'NetViewer', the packet parser, the signal-computing engine, the detection engine, visualization engine and alert engine. 'NetViewer' used libpcap, Cisco's NetFlow and NLANR as input source. 'NetViewer' is capable to produce the real time picture of the network traffic with the help of images to detect and identify malicious traffic patterns based on multiple dimensions of network traffics such as IP addresses, port numbers as shown in Figure 2.6.

Estan et al. [7], developed Wisconsin 'Netpy' an extended version of Netpy in Politehnica University of Bucharest with collaborative efforts of University of Wisconsin-Madison U.S.A., to evaluate the network traffic in run time. 'Netpy' is having four main components: GUI, database, console and analysis engine. It uses Netflow data source and stores data in database for further analysis. The GUI is developed using wxPython UI tool kit. 'Netpy' is an interactive tool for

analysis of source addresses, destination addresses, filters and time series plots that separate the traffic into various categories as shown in Figure 2.7. This tool supports the run time interactive drill-down facility for network layer protocol, port and time based filters.

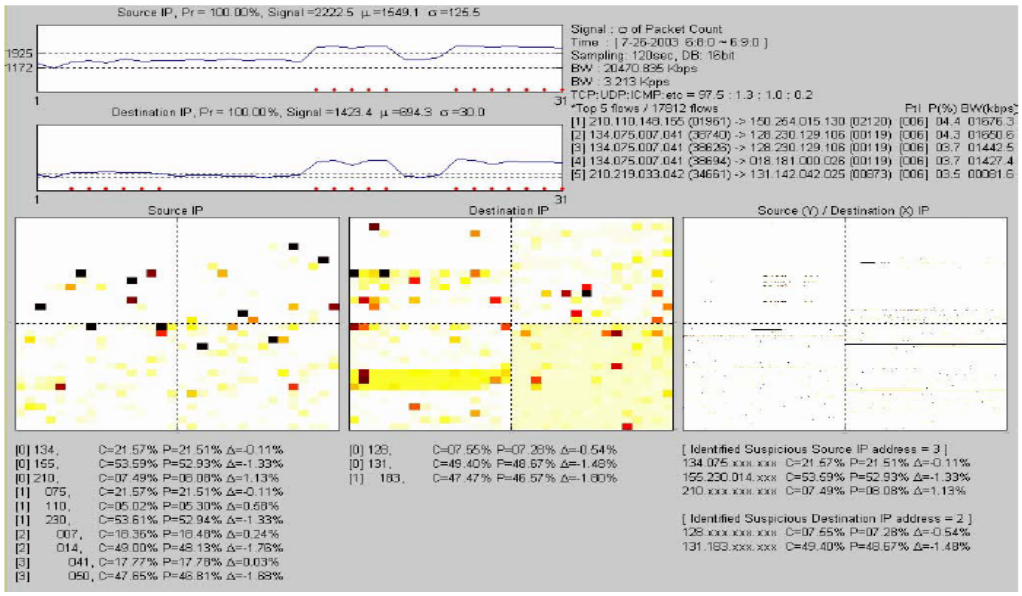


Figure 2.6: NetViewer [6]

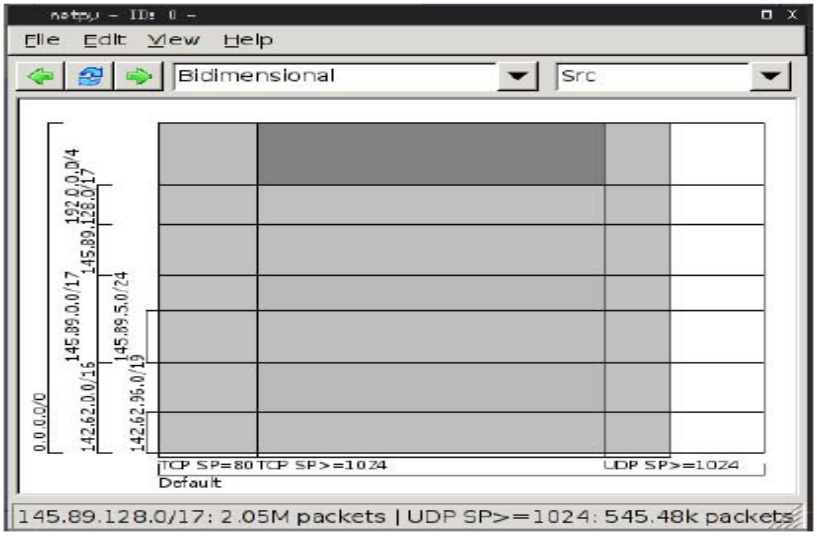


Figure 2.7: Netpy [7]

Ren et al. [10] developed the IDGraphs an intrusion detection based on interactive visualization

system support. IDGraphs system works in real time over network router data-set (Netflow as data source) while identifying the temporal features of the intrusions in netflows and recognition of correlated narrative attacks. In Netpy, scatter plot visualization scheme is used in which time is shown at x-axis and netflow at y-axis as Figure 2.8. IDGraphs facilitates its user through histographs visualization. Histograph is used for selecting a data point with zooming and brushing facilities to mention an event. IDGraphs is helpful in identifying the DoS and DDoS attacks.

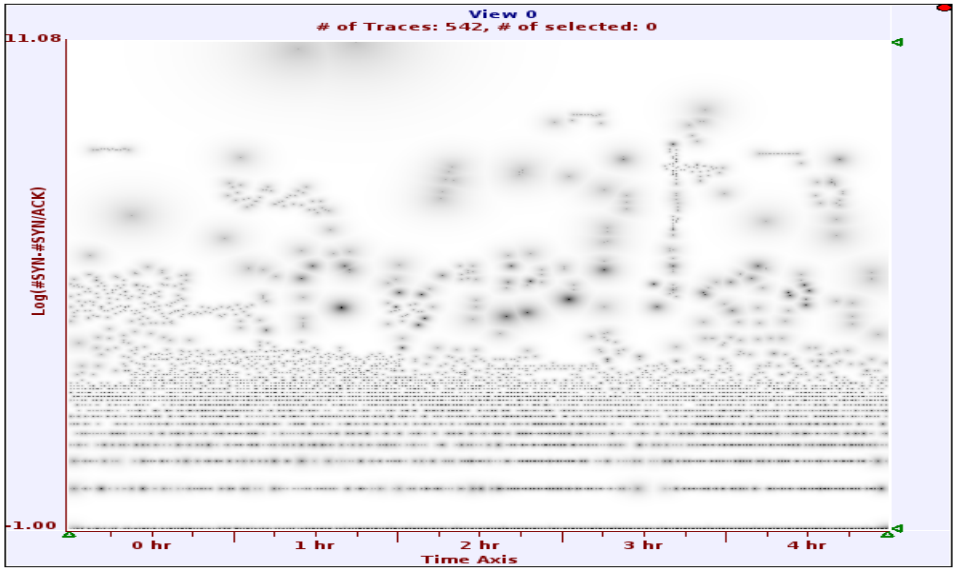


Figure 2.8: IDGraphs [10]

Abdullah et al. [9] proposed IDS Rainstorm to overcome the problem of text based evaluation of log files of IDS at Georgia Institute of Technology. The IDS Rainstorm can display 163,830 IP address on a single visualization while using pixels and colors to represent alarms as shown in Figure 2.9. This tool is able to represent network layer based connectivity and their alarms according to network policy. This tool facilitates to see the details of a particular IP by filtering and zooming. The source of input of this tool is alert logs, generated by Snort. IDS Rainstorm

works offline and displays scattered plot and parallel coordinate plot visualization. In this scheme multiple parallel y-axes are assigned to IP addresses and a single x-axis is used as the time line for alarms. Apart from the main view, a zoom-in view is also given by Abdullah et al. [9] to its user to selects an area to zoom in. There are two y-axes preserved in zoom-in view viz. the internal IP addresses (victims) and external IP addresses (attackers) shown by left y-axis and right y-axis respectively. The alarms are also visually presented. Even, color schemes are used to expose the sternness and the number of alarms.

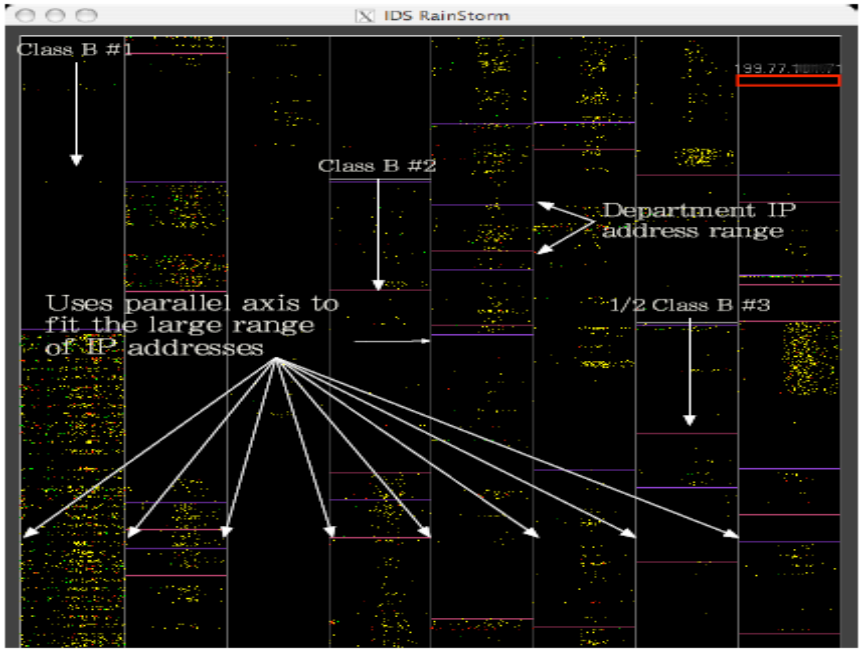


Figure 2.9: IDS RainStrom main view [9]

Conti et al. developed tool named as 'Rumint' [12] to meet the overloaded information of security. 'RUMINT' can analyze 30000 packets (max) at a given time. This tool is capable of visualizing binary file and display port vulnerability assessment reports as shown in Figure 2.10. RUMINT is also helpful in Dissecting port scans, Nessus vulnerability assessments and Metasploit attacks. This tool is able to display the logs of firewall and IDS. VCR like animation

facility makes this tool more effective to interact with historical data in effective manner through scatter visualization.

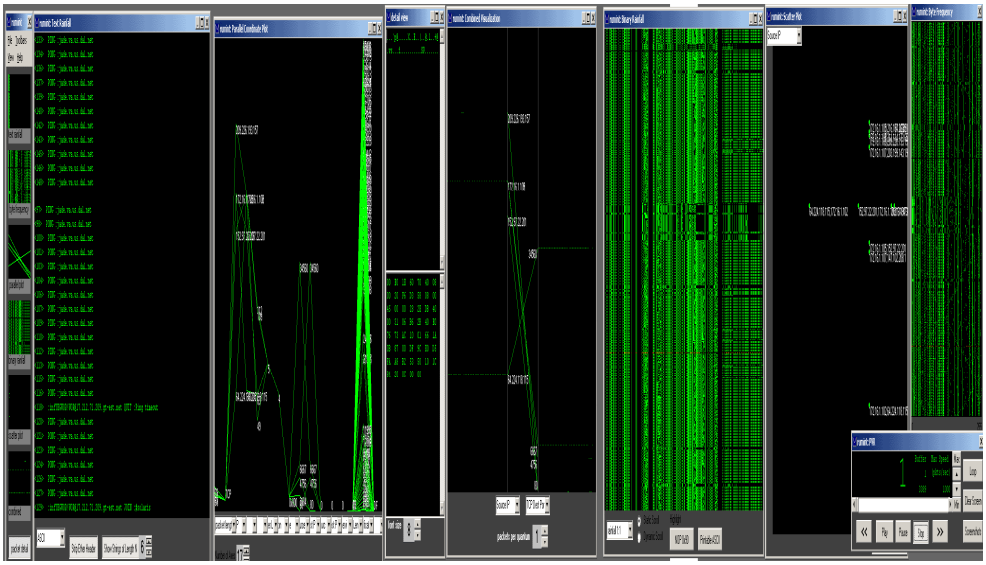


Figure 2.10: A panoramic view of the RUMINT [12]

Though Conti et al. in [12] summarize the interaction techniques applied in the two complementary tools as domain-specific semantic zooming, interactive encoding and dynamic querying. But lacking in direct interaction on the visual objects, which run off the existing visualization (RUMINT) with several problems, which delay the exploration of data and reduce the efficiency.

Raffael Marty's developed 'AfterGlow' [13] using PERL for displaying the relationships between the entities of the data input. AfterGlow uses pcap files and log file of sendmail used as input source. 'Afterglow 2.X' developed in java, uses network graph and Treemap visualization scheme to display email Logs, firewall Logs, firewall rule sets, web logs, IDS logs, IPS logs, and operating system logs as shown in Figure 2.11. It produces CSV file after parsing the tcpdump. This CSV file is used to produce dot file, which is further used as input of Graphviz. Graphviz is

another tool used to produce graphs of the logs using spring models. Graphviz also facilitates its user to view data in real time of a specific area. Marty expanded this work in 2010 and further proposed the Data Visualization and Analysis Linux (DAVIX) framework [14], which is equipped with 25 tools and also freely available for data visualization.

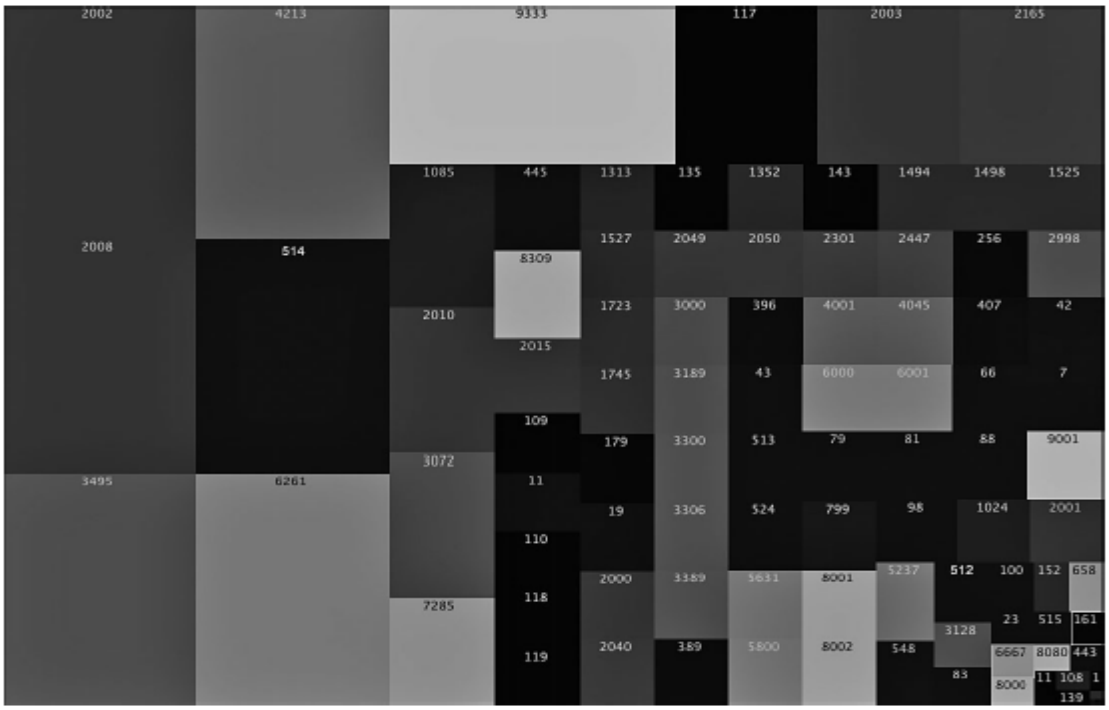


Figure 2.11: Treemap showing traffic on ports by using AfterGlow [13]

Reil et al. [15] developed 'InetVis', a 3D scatter plot visualization tool, to analyze the suspicious activity present in Internet traffic for C-class networks. 'Nmap', libpcap and tcpdump are used as input source to capture the network data. InetVis is capable to visualize ICMP and UDP in real time as shown in Figure 2.12. InetVis facilitate user with adjustable time window, navigation (for deeper exploration), and dynamic filtering. InetVis also provides the feature of color encoding by selecting a specific traffic properties, resizing of data points, and manipulation of 3D space such as moving, rotating and zooming. These features are helpful in identifying the anomalies. Even

InetVis can replay data while adjusting its rate. However, InetVis is limited to visualize the network layer data only.

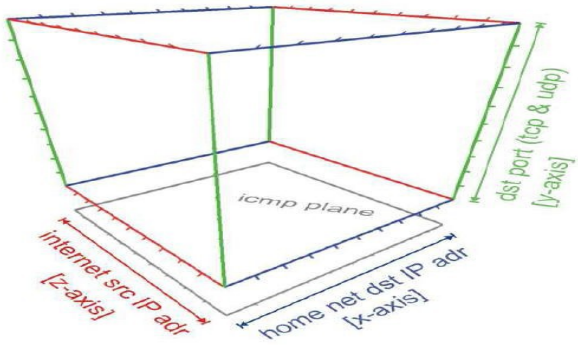


Figure 2.12: InetVis [15]

Oberheide et al. [17] developed 'Flamingo', a client / Server based tool, which is capable of displaying the live data while using OpenGL dataset. Flamingo uses parallel axis concept to represent individual IP address and their respective load by using bar height as shown in Figure 2.13. This tool also represents the load per port -tree square layouts at different heights along the cube. Volume of aggregated traffic based on source IP prefix is shown in left window in 3D representation and 2D representation in right window.

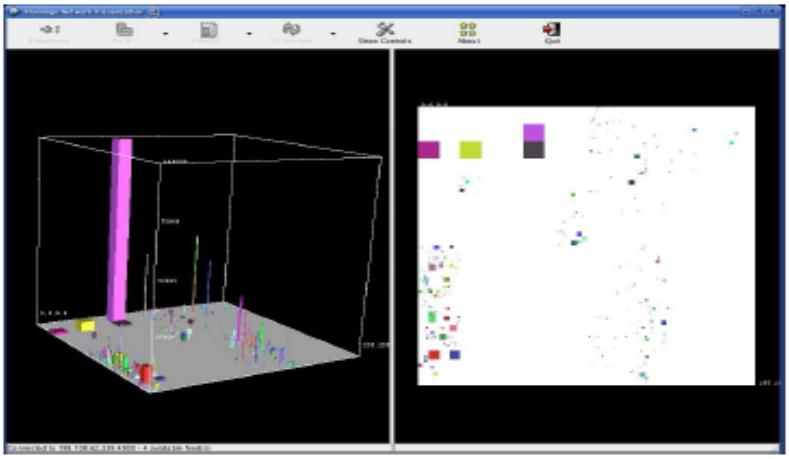


Figure 2.13: Flamingo [17]

Decker et al. [18], developed NfSen, using various plugins to detect the mischievous data in Netflow, based on PHP (front-end plugins). NfSen also facilitate its users with filtering, aggregation, customized time window and statistical summary to understand the properties of Netflow data as shown in Figure 2.14. While focusing on flows of data, packets and bytes, Nfsen is helpful in detecting DoS and DDoS attacks.

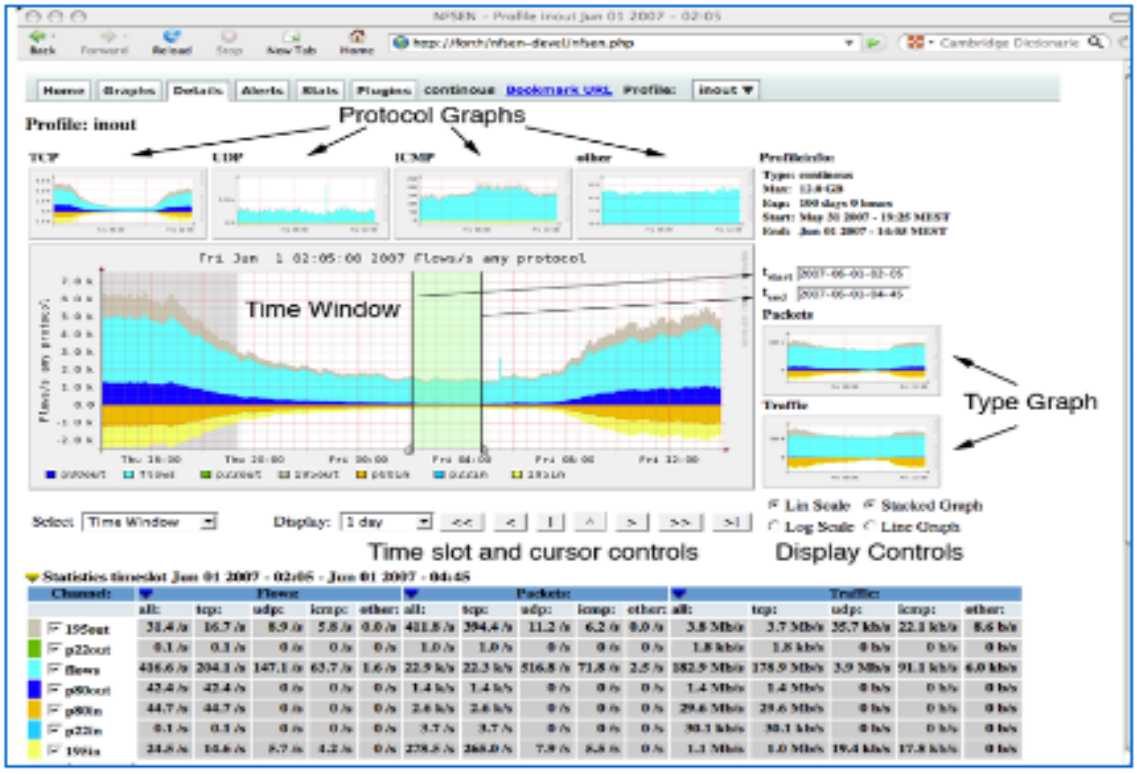


Figure 2.14: NfSen [18]

Godinho et al. [19] developed 'PRISMA' a multiple coordinated views information visualization tool, developed in Java, which had been experimented with car dataset. PRISMA an interactive, portable, scalable tool. In PRISMA there are three visualization techniques are used: treemap, scatterplot and parallel coordinates as shown in Figure 2.15.

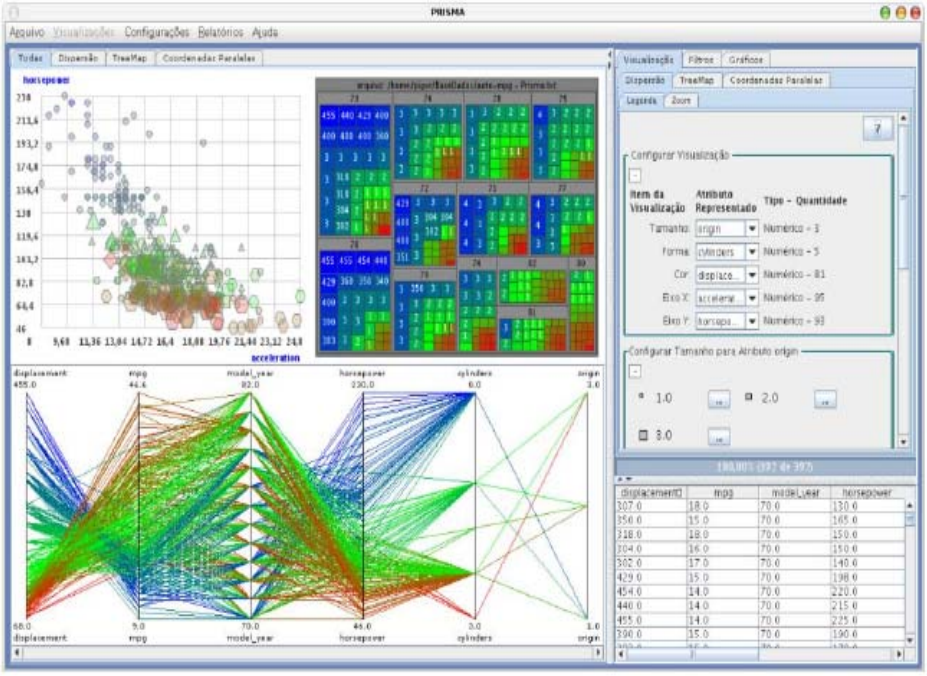


Figure 2.15: PRISIMA Interface [19]

Makanju et al. [20] developed the LogView to visualize the log of application layer services like IMAP, POP3, SSH and HTTP. LogView can display event log clusters based on clusters, which are produced by simple Log file clustering tool (SLCT) a third party tool. LogView is developed in Java language (Java Swing toolkit) and also used visualization toolkit. LogView generates treemap layout shown in Figure 2.16, also facilitate its user to analyze logs while selection, filtering and searching on event log data sets for application layer services.

Frei & Rennhard [21] developed the Histogram Matrix (HMAT) a log file visualization tool to detect the anomaly while analyzing all kind of textual log files. Frei et al. [21] tested the logfile of email server and Samba server. This scheme allows the user to identify an anomaly in easy manner while clicking the corresponding circle as shown in Figure 2.17, it automatically clusters the log messages behind the circle, to identify the log messages which are reason of an anomaly.

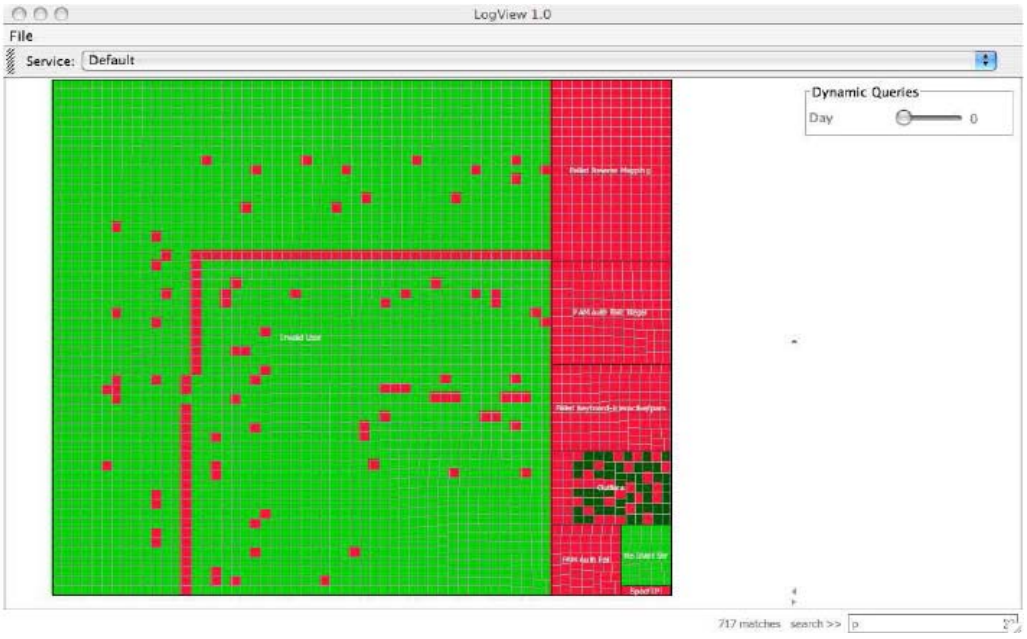


Figure 2.16: LogView Interface [20]

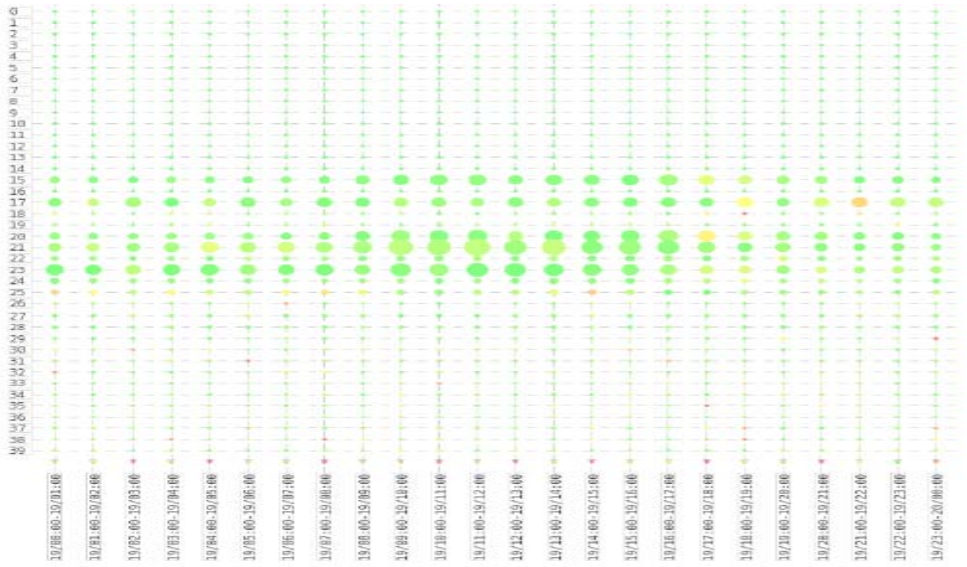
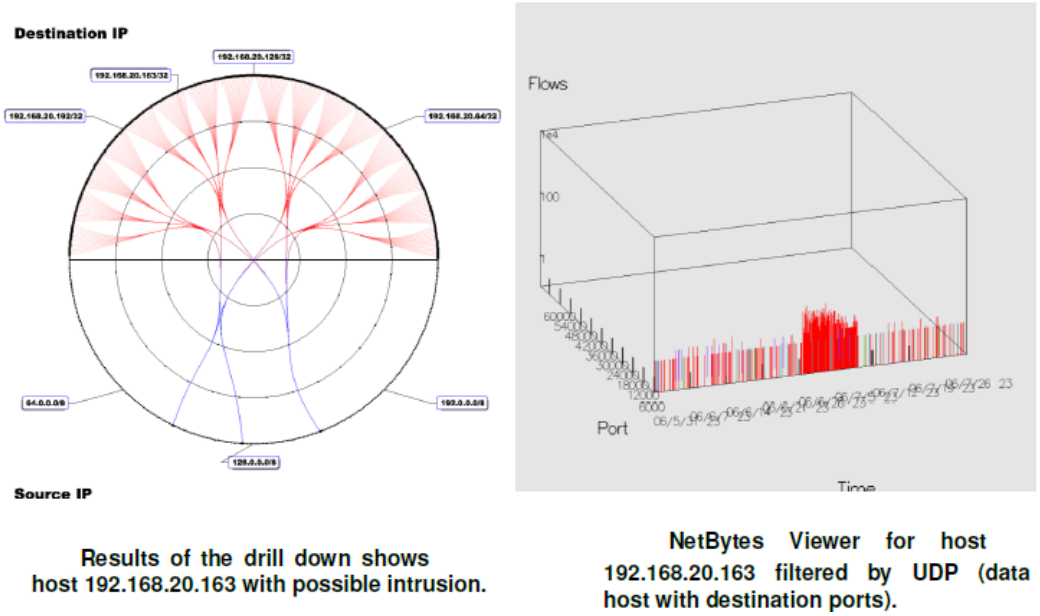


Figure 2.17: Histogram matrix of mail server [21]

'FloVis' developed by Taylor et al. [22], is based on client and server architecture and supports Windows, Linux and UNIX platforms. It uses SiLK repositories to store the Netflow and IDS

alerts. In FloVis, there are three different kind of visualizations: Activity viewer, FlowBundle, and NetBytes viewer. The purpose of Activity viewer is to display a selected part of sequential behavior of a random group. The 'FlowBundle' displays the flow of interactions between hosts or subnets in bundles. The NetBytes viewer is used for detailed behavior analysis of an individual machine based on time parameter as shown in Figure 2.18.



Results of the drill down shows host 192.168.20.163 with possible intrusion.

NetBytes Viewer for host 192.168.20.163 filtered by UDP (data to host with destination ports).

Figure 2.18: FloVis [22]

Jiawan et al. [23] developed NetViewer, capable in detecting the DDoS attacks, network scans, and port scans. NetViewer is having three main subsystems: capture subsystem, database and network security visualization. The WildPackets and OmniPeek are used as capture subsystem, helpful in connection oriented analysis. Network security visualization subsystem is capable in IPv4 network display and analysis through interactive analysis as shown in Figure 2.19.

Allen et al. [24] developed 'NAV' Network Analysis Visualization in JAVA to visualize the high-level network events, to investigate the malicious activities for large network, while operating on network and transport layer visualizations. The input source of this tool is log files with IP

address of high-level devices. NAV facilitates to have overview and detailed view of network traffic with IP wall view, Port scan view, service view, open dialog, textual filter, time filter, and color brushing feature to drill down as shown in Figure 2.20.

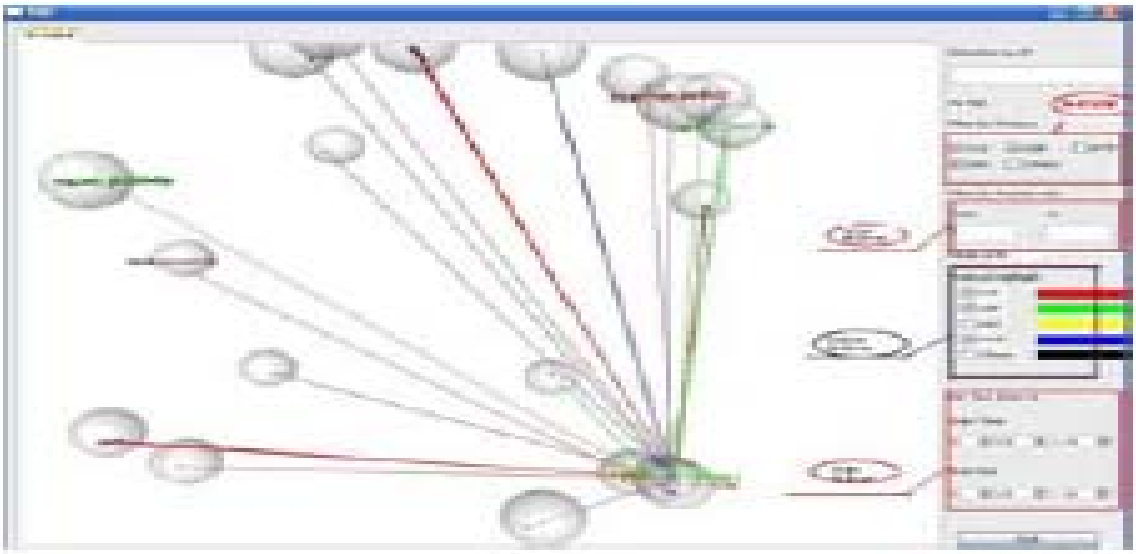


Figure 2.19: NetViewer [23]

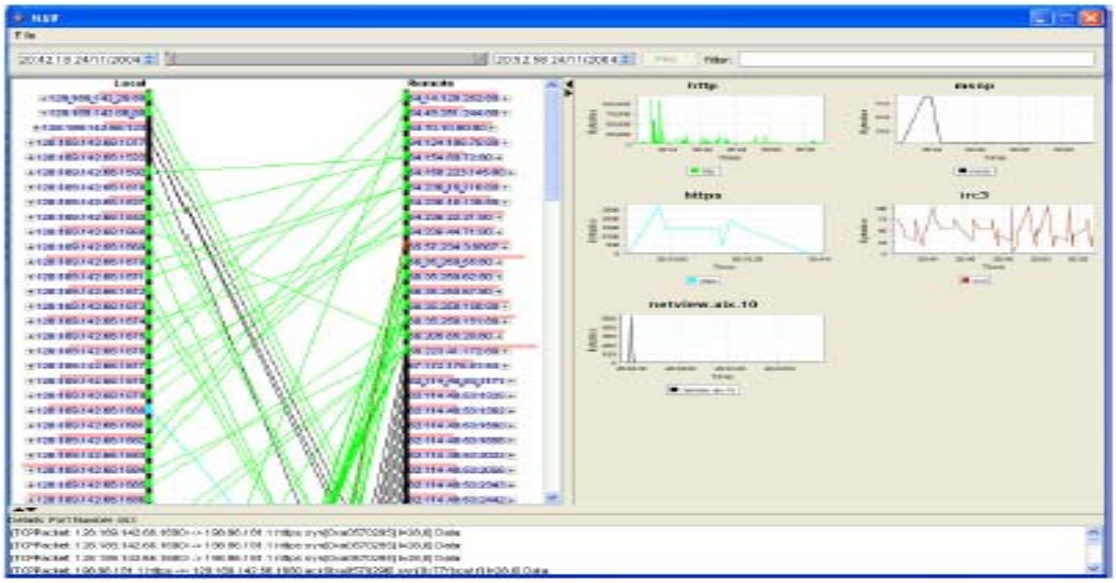


Figure 2.20: An overview of NAV Source [24]

Goodall et al. [25] developed 'VIAssist' a visualization tool, developed using Microsoft's .NET framework. The comprehensive cognitive task analysis (CTA) used by network security analysts for optimized operating efficiency. This tool produces big-picture overview of the network with the help of an intuitive dashboard shown in Figure 2.21. The drill down facility helps in selecting a particular event and displaying it with multiple visualizations with the help of automatic Smart aggregation. VIAssist uses the histogram, parallel coordinate plot with Glyph encoding for in-depth event analysis.

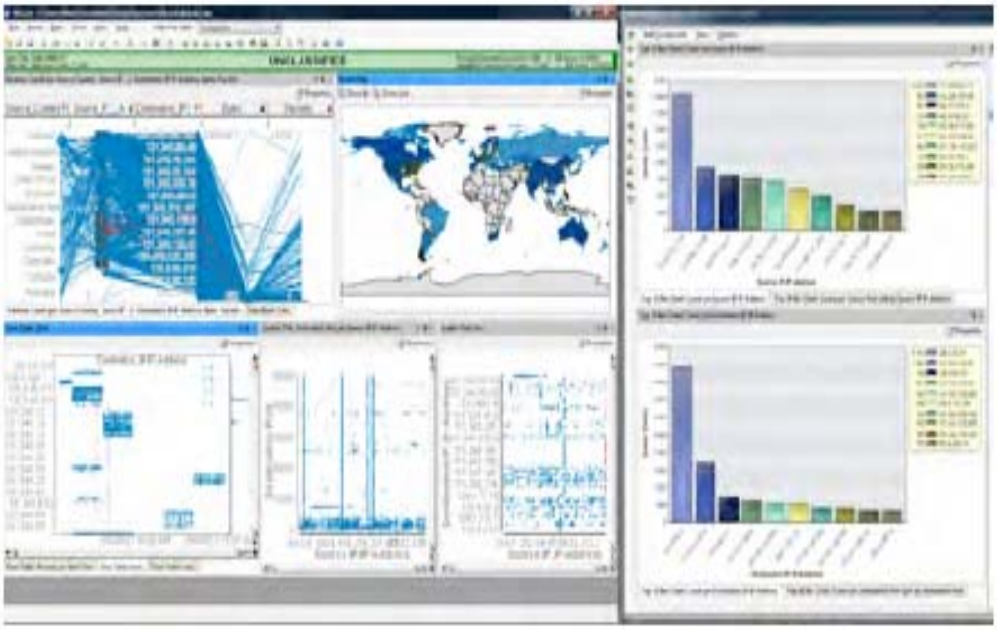
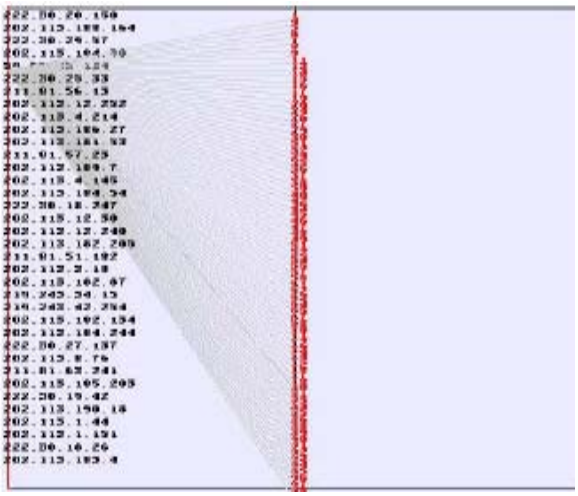
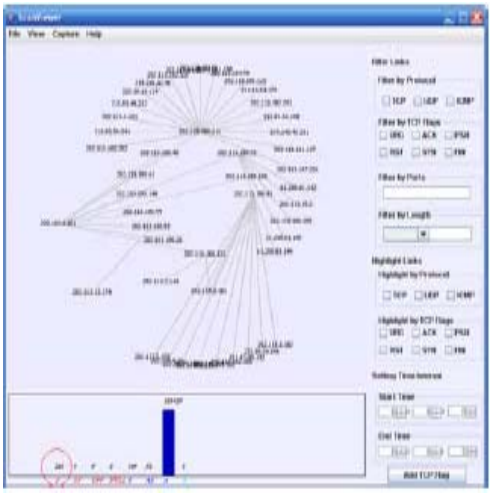


Figure 2.21: VIAssist [25]

Jiawan et al. [26] developed 'ScanViewer', designed to capture large-scale ports information. ScanViewer can detect hidden scans along with network scans, port scans and distributed port scans. ScanViewer generate two views: the global view and detail view shown in Figure 2.22. The global view represent the network flows with help of multitude of lines and the nodes stand for different hosts in the web.



Patterns of Port Scans in ScanViewer



Patterns of hidden Scans in ScanViewer

Figure 2.22: ScanViewer [26]

Lu et al. [27] proposed a network scan tool named as 'CCScanViewer' developed in Java. Lu et al. [27] conclude that polycurves display is better than parallel coordinated view after experimenting with 160 different datasets, further able to detect DDoS attack in real time. Although interactive but still lacks in ease of use and drill down clarity. A normal networking pattern displayed in CCScanViewer shown in Figure 2.23.

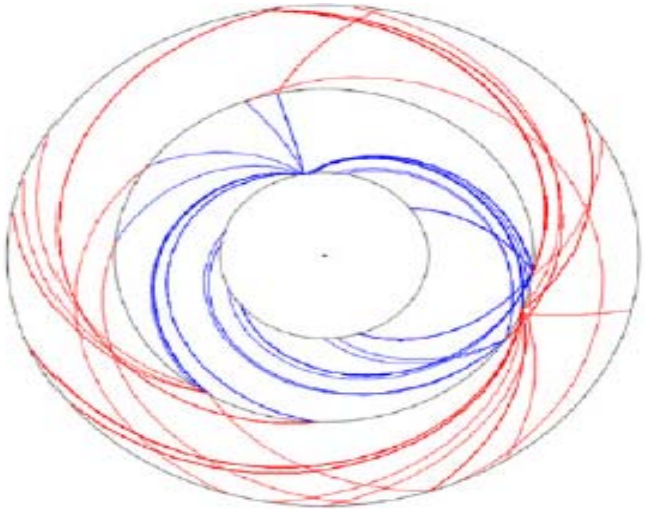


Figure 2.23: CCScanViewer [27]

Osborne et al. [28] developed Explore, Investigate and Correlate (EIC) framework to investigate digital forensic information from a large volume of data by using tool named Infovis. Infovis is equipped with zooming, filtering and visualization facility. Osborne et al. [29] discusses a refinement of EIC (Explore, Investigate, Correlate) process, which will take care of data complexity and volume issues effectively through infovis.

Osborne et al. [29] categorized the visualization techniques into explorative visualization and correlation visualization used in EIC process to express the high level overviews and behavioral relationships within a dataset respectively shown via visualization as shown in Figure 2.24. In EIC investigation techniques are facilitating its user in searching and filtering (based on keywords) in real time and interactive manner. The input source used in this model is either raw digital evidence data or data from industry standard tools (Encase or Forensic Toolkit). The Ruby (an object oriented language), Rails (web based open source platform for programmer), JavaScript and HTML5 canvas technologies for AJAX, JSON data manipulation and interactive infovis are used as development platform to implement refined EIC model.

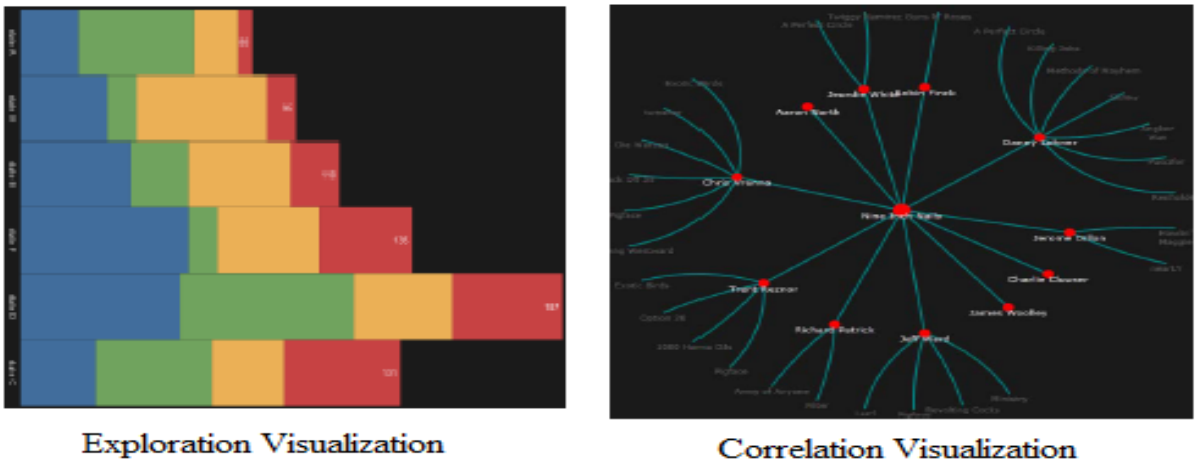


Figure 2.24: EIC process model [29]

Figure 2.25: EIC [30]

Osborne et al. continue EIC process model in this research work for the development of infovis, which is an outcome of their previous work [30]. Osborne et al. [30] focused on explore, investigate and correlate so that network security visualization tools can give a overall view, a selected view and even explains the linkage between nodes cognitively shown by Figure 2.25.

Shurkhovetskyy et al. [31] discuss the role of network security visualization to understand situation of a network with investigation. They used a customized infovis toolkit (IVTK) [31] (an open source visualization toolkit), which can read CSV , XML and many other input source. Shurkhovetskyy et al. [31] used interactive security visualization analytic tools of IVTK to demonstrate the situation of network thorough heatmap visualization for IP addresses involved in IRC traffic, Time series visualization for IP addresses involved in FTP / SSH / IRC traffic, parallel coordinated for IDS events and port scan visualization for firewall logs shown in Figure

2.26. Emphasis is on linking the expert knowledge with outcomes from the visualization tools, so that network analysts make well learned decisions about relevant events.

Donahue et al. [32] explained the problem of detecting the malware in portable executable (PE) files through reverse engineering (RE) and conclude it can be easy in visual form as compared to handling thorough Hex editors (A tool to examine the PE files in hexadecimal and ASCII formats) The attacker packed the malware by compressing and encrypting code while using UPX (an automated packer tool) to frustrate the cyber security analyst. In addition to this Donahue et al. [32] generate two visualization techniques to analyze a PE file and detecting the malware through navigational PE file structure and Markov Byte Plot as shown in Figure 2.27.

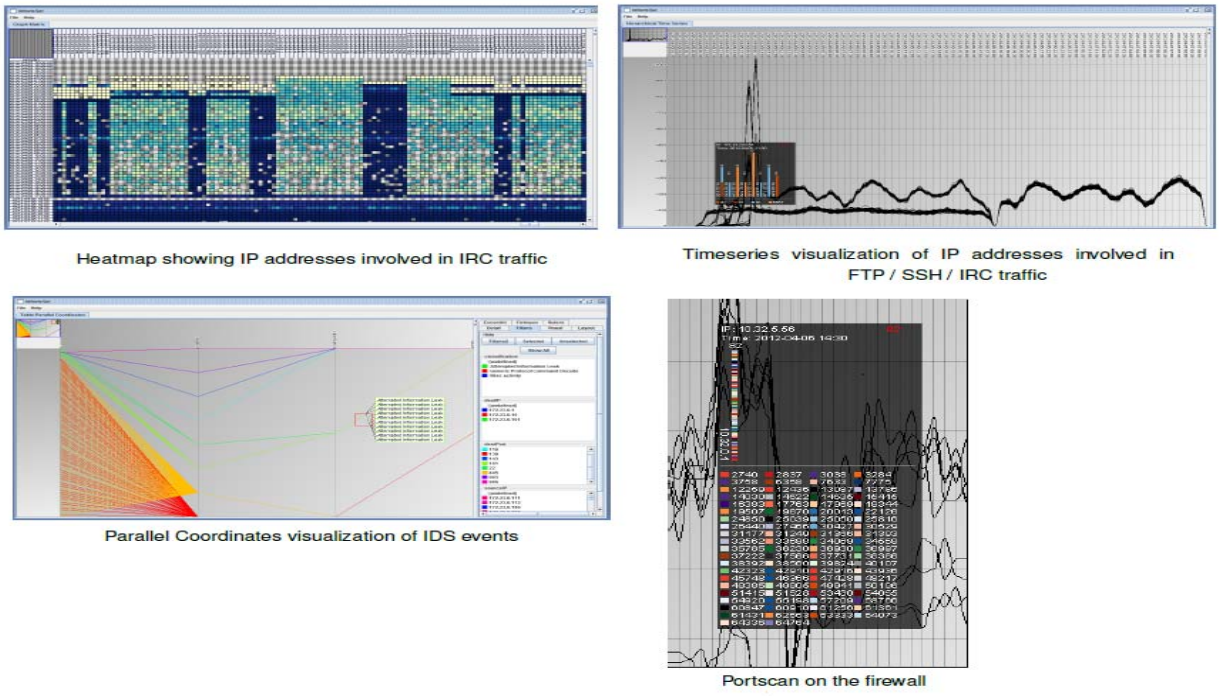
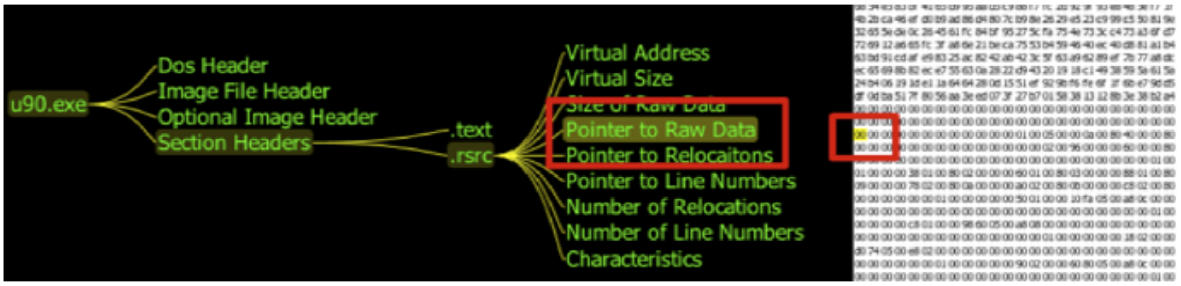
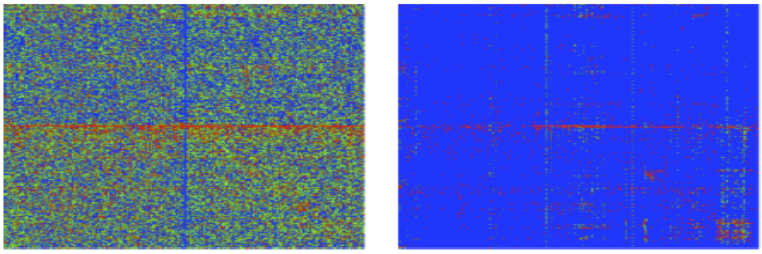


Figure 2.26: EIC [31]



Interactive Hex Editor



Packed and Unpacked version of Beagle Malware

Figure 2.27: PE file analysis and Malware detection [32]

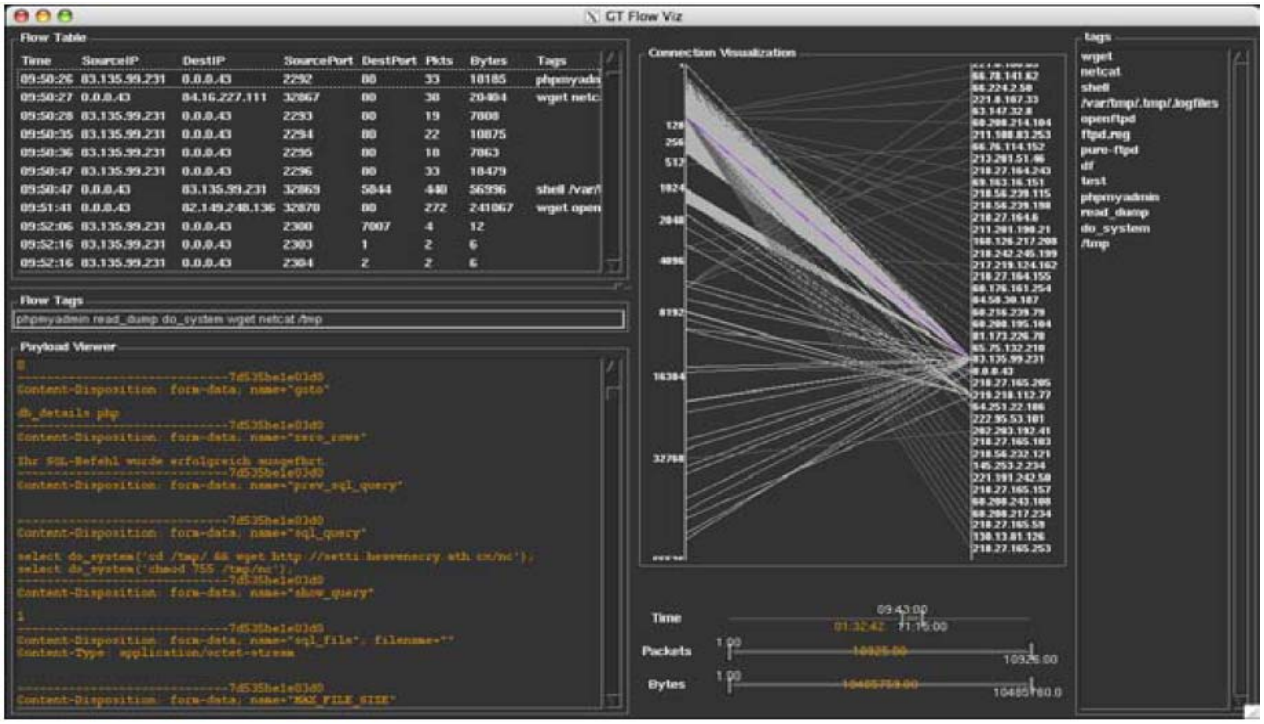


Figure 2.28: FlowTag [16]

Lee et al. [16] developed the FlowTag, a visualization scheme for their Honeynet while enabling its users in tagging (selection and filtration) the flow of data (ethereal, IDS and Logs) to support analysis and collaboration. Lee et al.'s tagging scheme allows the network analyst to label key elements to minimize cognitive load of investigation during the analytic process to maintain context shown in Figure 2.28. Even tagging on social networking and social bookmarking Websites become popular and widely accepted.

2.1.2 Security through data mining techniques

This subsection deals with literature review of data mining techniques usage in network security solution.

Thuraisingham et al. [121] discussed the utility of data mining in cyber security application especially about intrusion detection system. Thuraisingham et al. [121] mentioned that computers and their networks are being intruded by unauthorized individuals but data mining techniques such as for classification and anomaly detection are being used extensively to detect such unauthorized intrusions. Thuraisingham et al. [121] also mentioned the utility of data mining in biometrics, pattern recognition and machine learning techniques.

Grégio et al. [94] discussed the problems in analysis of logs of network services due to huge volume and proposed an effective data mining based approach for intrusion detection. Grégio et al. [94] applied and evaluated (K-Nearest Neighbors, Artificial Neural Networks and Decision Trees) data mining techniques in order to classify traffic logs as normal or suspicious through a real network as well as on a honeypot.

Lee et al. [95] [96], proposed the utility of data mining framework for constructing intrusion detection models. Their work discusses the strengths of classification, meta-learning and

association rules. Lee et al. utilize auditing programs to extract an extensive set of features which explain each network connection and apply data mining scheme to learn the rules that capture the behavior of intrusions and normal activities. Further these rules were used for misuse detection and anomaly detection.

Muda et al. [97] proposed a hybrid learning approach while collating the K-Means clustering and Naïve Bayes classification. This approach gives the cluster data into the matching groups, before applying a classifier for classification purpose. Oreku et al. [98] discusses ideas of using data mining techniques to implement IDS while discovering the consistent and useful patterns to compute (inductively learned) and recognize anomalies.

Zhang et al. [123] proposed a frameworks based on data mining algorithm called in misuse, anomaly, and hybrid-network-based IDSs while building automatically by the random forests algorithm over training data. Also evaluate their approaches over the Knowledge Discovery and Data Mining 1999 (KDD'99) dataset. Apart from computer network security.

2.1.3 Security through data visualization based on data mining

Singh et al. [36] used k-means clustering algorithm to analyze and visualize the flow of data to predict threats based on three attributes of flow data i.e. protocol, IP and ports. They used third party tool for capturing data and used 'Weka' a data-mining tool for analysis.

Ahmad et al. [37] used ANN, based on multiple layered perceptron to detect attacks over the network. Data source for their work was Kddcup99 dataset. They used the back-propagation for training and testing of their system. ANN is having advantage over other data mining techniques for data cleansing, flexibilities of making association, sequence and linkage among the networks. But Ahmad et al. did not provide a visualization based solution. Golnabi et al. [39] used data

mining to update rules of firewall. Vaarandi [40] proposed data mining based IDS solution for real time alerts generator against threats.

Lakkaraju et al. [3] developed 'NVisionIP', a visualizing tool using data mining, for entire Class B network node visualization on a single screen. It facilitates drill down and collects information in more details about a particular host in a network. There are three different levels of visualization galaxy view, small multiple views and machine view. The galaxy view is a scatter plot the subnet address at the horizontal axis with and the host address at vertical axis. NVisionIP also facilitate its user in zooming and leading a dynamic query. The other two detailed views explains statistical characteristics of netflow according to a particular properties thorough histogram shown in Figure 2.29. NVisionIP uses Argus netflow data and Cisco netflow data as its input data source.

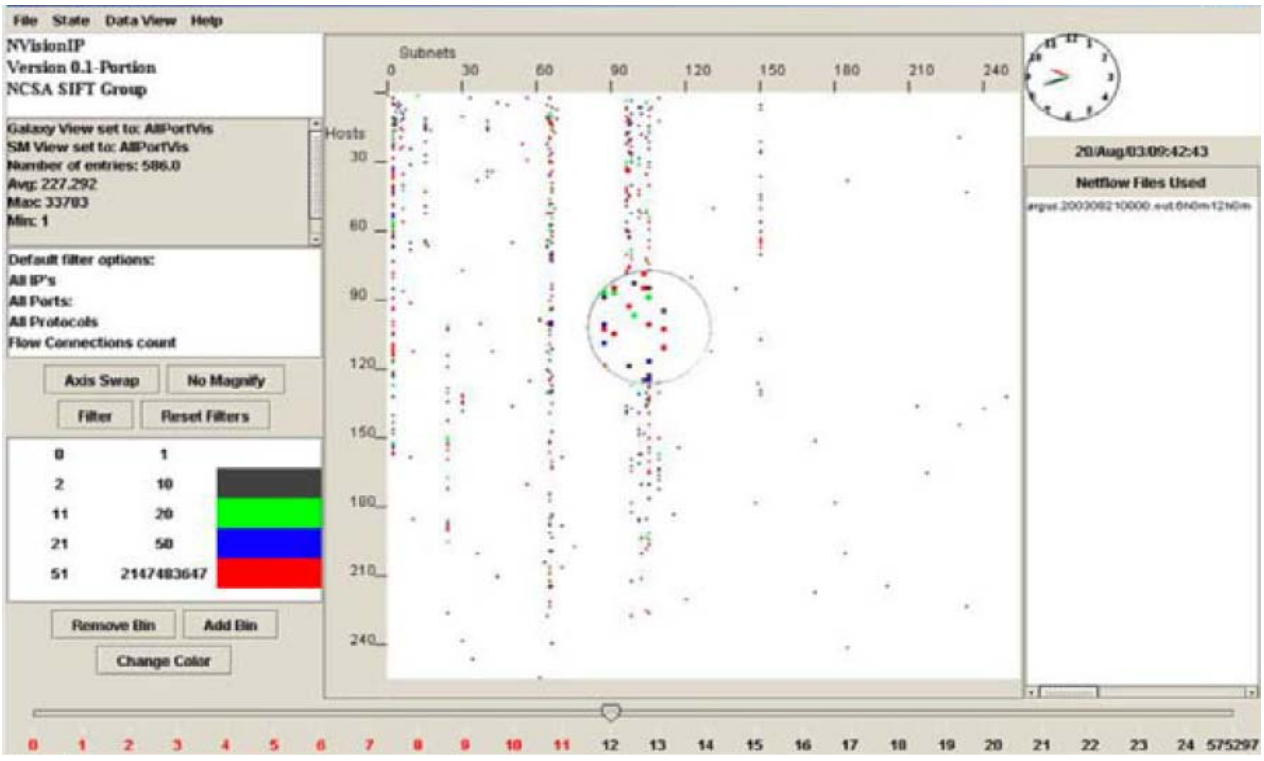


Figure: 2.29 NVisionIP [5]

Creese et al. [41] proposed a situational awareness based visualization tool named as 'CyberVis'. This tool is able to visualize the enterprise network attacks and their subsequent potential consequences combining traditional network diagram icons with business process modeling and notation. This tool facilitates drill down and helps in forensic analysis using movie-style playback.

Table -2.1 Comparison of Network Traffic Visualization tools

Tool	**Media**	**Real time**	**OSI Layer**	**UI**	**Visualization Scheme**
Flodar [1]	NM	No	NM	No	Platter display, 3D spinning cube, Building Display
Nam [2]	Both	No	Up to 7th	Yes	Link graph, monitor boxes, Time event graph
NVisionIP [3]	NM	No	3rd and 4th	Yes	Bar Charts for small multiple view, Machine view
VISUAL [4]	NM	No	3rd and 4th	Yes	3D grid square display
VisFlow-Connect [5]	NM	No	3rd and 4th	Yes	Parallel axes

NetViewer [6]	NM	Yes	3rd and 4th	Yes	Simple X-Y graphs and real time traffic visualization
Wisconsin Netpy [7]	NM	Yes	3rd	Yes	time series plots and Heatmaps
PortAll [8]	Wired	No	3rd and 4th	Yes	Time series graph, Scatterplot, 3D bar graphs
IDS Rainstrom [9]	NM	No	3rd	Yes	Scatter plot and Parallel coordinated plot
IDGraphs [10]	NetFlow	Yes	3rd	Yes	Histograph
TNV [11]	NM	Yes	3rd and 4th	Yes	Histogram, Link visualization for Network and Port
RUMINT [12]	Both	Yes	3rd	Yes	Binary Rainfall, Scatter Plot, Parallel Coordinate Plot
AfterGlow [13]	Both	Yes	7th	none	Network Graph, Treemap
InetVis [15]	NM	Yes	3rd and 4th	Yes	3D Scatter Plot

FlowTag[16]	Logs, IDS, Ethereal	No	3 & 4th	Yes	FlowTag Connection Visualization
Flamingo [17]	NM	Yes	3rd and 4th	Yes	Parallel line, bar graph in quad-tree structure
NfSen [18]	NM Nfdump	Yes	3rd	Yes	Time series graph
PRISIMA [19]	NA	NA	NA	Yes	Treemap, scatter plot and parallel coordinates,
LogView [20]	NM (log files)	No	7th	Yes	Treemap
HMAT [21]	Log files	No	NA	Yes	Histogram Matrix
FloVis [22]	Both	Yes	3rd and 4th	Yes	Activity viewer, FlowBundle, NetBytes viewer
NAV [24]	NM	Yes	3rd and 4th	Yes	Line graph, Netmap
VIAssist [25]	NM	No	3rd	Yes	Histograms, Parallel coordinated plot, Glyphs,
ScanViewer [26]	NM	Yes	3 & 4th	Yes	Parallel coordinated for

					port scanning, Link graph
CCScanViewer [27]	NetFlow	Yes	3 & 4th	Yes	Polycurves
InfoVis (EIC process Model) [28][29][30]		Yes	3 & 4th	Yes	Exploration, Correlation, Sunburst, Bar chart, Details on demand
IVTK [31]	Logfiles of firewall, IDS	Yes	Upto 7th Layer	Yes	Heatmap, Time series, Parallel Coordinated, portscan
VizSec based on PE and RE [32]	Portable executable files	NM	NM	Yes	PE file structure and Markov Byte Plot

***NM -Not mentioned, *NA- Not applicable, *VizSec- Visualization Scheme, *UI- User interactive**

Gap analysis of above discussed visualization schemes (mentioned in Table 2.1) is discussed in chapter -3. There is a need to understand the framework working of these visualization systems, in lieu of this framework given by Schiavone et al. [122] and Usmani et al. [124] was studied. Schiavone et al. proposed a 'Enterprise Ontology' and 'Information Security Capability-Driven Framework' to ensure the appropriate use of security. Schiavone et al.'s business value model is incorporating the function of financial, operational and security-based quality assurance

measures. Usmani et al. [124], proposed a three independent layer based Internet traffic monitoring framework, which can cyber crimes, cyber terrorism, and cyber crimes incidents in better manner. Layer 1 consist of stakeholders involved in implementing the framework. Layer 2 is the core layer deals with the mechanism for its implementation, and layer 3 mentions the results of the Internet Traffic Monitoring Framework.

Further, this subsection deals with literatures of eminent researchers who contributed in the field of network traffic visualization based on location, conceptual and plotting schemes as discussed in Chapter 1 subsection 1.5.5.

Location based visualization - Becker et al. [34], proposed SeeNet, in which they used "Link Map". Link Map is very basic way to represent the connectivity between paired nodes with help of line segment (edges) in two dimensions. In Link Map (Figure 2.30a), color and thickness of line is used to show information about the link. But this scheme has its limitation w.r.t., number of links i.e., under a large network the map becomes jumbled and unreadable.

Becker et al. [34] also gave another kind of location based visualization named as 'Node map' from SeeNet. In this approach, they used Glyphs (Figure 2.30b) to visualize the node oriented information. In Glyphs, circles can be used for one dimensional information, rectangles for two and more complex glyphs for higher dimensionality and colors schemes used for extra dimensions. Overload nodes are shown by rectangular glyphs, horizontal rectangular are showing the incoming connections and the vertical outgoing connections respectively.

Cox et al. [59 - 61] proposed an interactive tool named as 'SeeNet3D', which is an outcome of extended work of Becker et al. [34] as a 3-dimensional visualizations [59 - 61]. Cox et al. [59 - 61] showcased the Internet traffic information on a globe with help of SeeNet3D (Figure 2.30c).

Histogram bars are used to represent the total number of packets originating from each country in the network. The color-coded arcs shows inter-country links, varying height of these arcs highlights the amount of traffic.

Melissargos and Pu [62] gave a different approach of the 'Link map' to represent load-per-link in a location based network structure, load is shown by dark grey color on links as shown in Figure 2.30d.

Lamm and Reed [54] developed a system named as Avatar system to visualize point-to-point communication traffic in globe-based view with arcs, which highlights communication using color, height and line thickness. With the help of histogram bars, their system present data from the origin of WWW requests, also using colored bands and different heights to represent the different parameters of the data [54]. With the help of height relative number of packets or requests originating at a given location are represented and the colored bands represent the distribution of document type, domain classes, servers or time intervals between requests (Figure 2.30e).

Koutsofios et al. [116-118] reported location based visualization named as 'Swift-3D'. They used 3D histogram bars to represent flat projection of the located area, where these bars represent traffic volume at various points in the network as shown in Figure 2.30f. Initially they aimed at telephone data but latter used it for network traffic.

Concept based visualization - Eick and Wills [57] developed a tool named as HierNet used for visualizing email communication within a company's large hierarchical networks. In this visualization nodes and links are used to represent communication (Figure 2.30g), the number of

messages received and sent are represented by area of each node and color of each node represents role of a user in the company.

Huffaker et al. [65] developed Otter, useful for core BGP routing, reach-ability and delay displays. Otter facilitate its user to customize data in an interactive manner. Otter visualize data in two steps, first placing the outer node and then rest nodes. The placement of node is in a circular layout (around the circumference) as shown in Figure 2.30h.

Cheswich et al. [69] in their experiment, used a huge test data sets (88k nodes, 100k links) for data visualizations layout of routing and reachability shown in Figure 2.30i. For small size data set, IP address, domain information, location, ISPs and locations of firewalls are clearly shown and colored where as for huge dataset graph becomes cluttered.

Erbacher and Frincke [108-109], visualize the behaviour of users, using Hummer IDS log files, to detect malfunction activities like intrusion and misuse. In this approach, nodes are representing machines and links represent their connections with each other. The style and color combination of the links represent the type of connection, viz. anonymous FTP connections is represented by short dashed links. The intensity of the color shows the last accessibility and protruding spikes (Figure 2.30j) are showing the number of users on that node.

Munzner and Burchard [125] developed a tool to display the structure of the World Wide Web in 3D hyperbolic visualizations. As hyperbolic space has ‘more room’ than Euclidean space, more information can be visualized (see Figure 2.30k). The visualization is constructed to show the hypertext links between web pages.

Takada and Koike [127] developed a tool named as ’Tudumi’, also known as a log visualization system, which helps to detect intrusion and anomalous behavior, while auditing log files.

This system is mainly concerned about the servers which are being accessed by small group of users. It takes care of three main activities: server's remote access, logging into the server and switching user accounts. The visualization of this data is represented as a series of layered concentric disks. The bottom disk displays information about user switching and the remaining disks represent network access and log-in information. Each user is represented by a textured node (Figure 2.30l).

Plot based network visualization - Eick et al. [128] gave an approach to visualize the log file of switches. These log files can be pretty big in tune of 50000 and more lines. Each file is shown in rectangle shape with1-pixel thick line, which is showing a line of text as shown in Figure 2.30m. In this approach they provide facility of interactive filtration for message frequency, time correlation and patterns of messages. The lines are color-coded based on the metric of concern. These researchers also implemented SeeSoft tool, which increased the efficiency and decreases the analysis time over traditional methods.

Becker et al. [34] used SeeNet for Plot based Visualization, they named it 'Matrix Display', the main focus of this approach is on links in the network, which addresses two basic problems from geographic network display: (i) long link are overly high up and many links overlap each other (ii) looming readability- data presented in a matrix with all the nodes being represented in one row and one column. Cell entries are small squares, color-coded by latency. The order of the rows and columns are geographical from west-to-east on the horizontal axis and north-to-south on the vertical (Figure 2.30n).

Oetiker [83] reported multi router traffic grapher (MRTG). In this approach router's information is retrieved via SNMP and stored with the round robin database tool (RRD). It uses large size graphs which are covering arbitrary time periods.

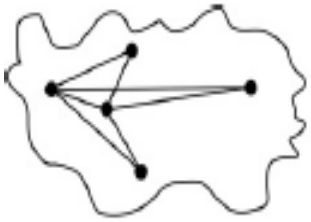
Plonka [99] used RRGrapher, to produce graphs. This work includes flow-per-second, packet-per-second, and byte-per-second as metric value over a default 48-h. In this approach Plonka used RRD and this approach is bit similar to MRTG. It gather the data from Cisco's NetFlow. The standard graphs provide view of traffic by a network or a subnet, by an application or service with help of color schemes.

Pongsiri et al. [103] presented a visualization of network traffic which use shows network trends and relationships among the traffic parameters like time of day, port and protocol on hourly bases for viewing long-term traffic features that characterize Internet-related traffic as shown in (Figure 2.30o).

Parish et al. [111] used figural deformity visualization (FDV). In this, approach the shape of an object is distorted in proportion to the values of a set of measurement data and parameters decided based on delay and loss. This works by deforming a given shape within a number of degrees of freedom in easily visible ways as shown in Figure 2.30p.

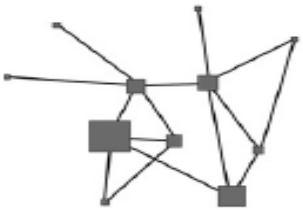
Lamm and Reed [54] used a three-dimensional scatter cube for the presentation of server access data. Their approach 'Avatar' produces two dimensional scatter plots, supporting analysis of high dimensional, non-grid based, and time varying data plots. In this data is presented as colored ribbons within a set of three-dimensional axes (Figure 2.30q).

Location based Visualization



a: Link Map [34]

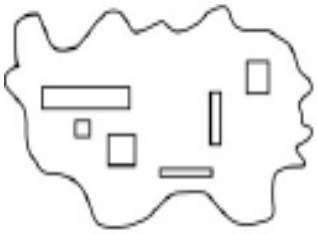
Conceptual based Visualization



g: HierNet [57]

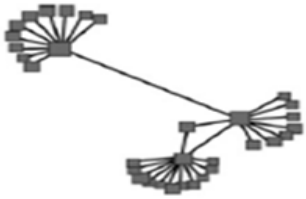
Plot based Visualization

m: SeeSoft [128]



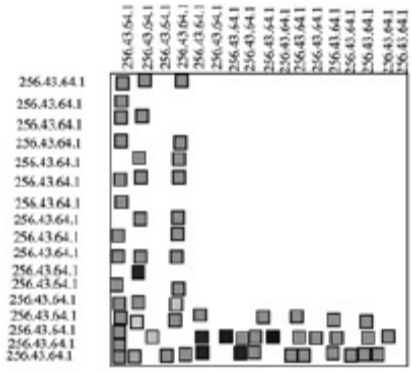
b: Glyphs

[34]

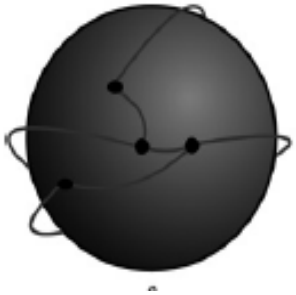


h: Otter

[65]



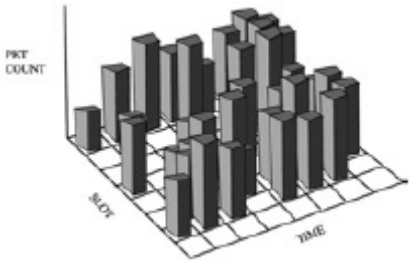
n: Becker et al. [34]



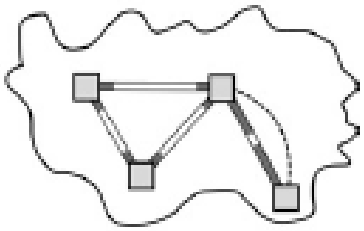
c: SeeNet3D

[59-61]

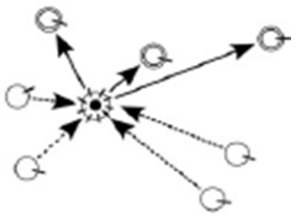
i: Simulated spring-force's outcomes [69]



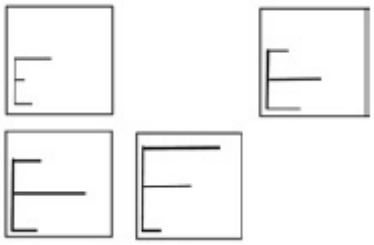
o: Pongsiri et al.'s tool's output [103]



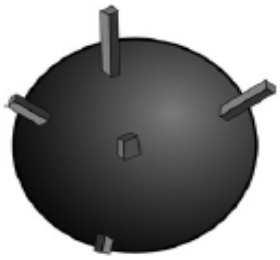
d: Link map [62]



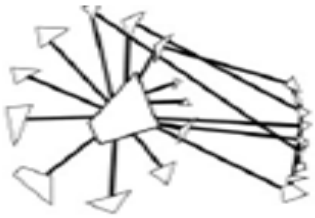
j: Erbacher and Frincke's tool's visualization [108-109]



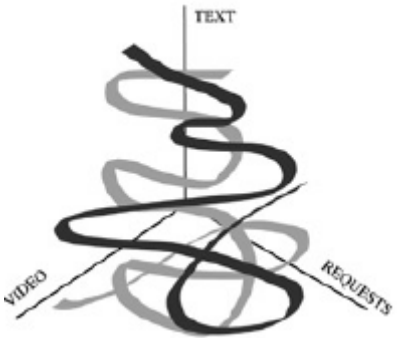
p: FDV [147]



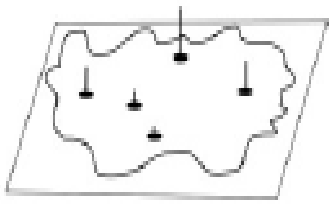
e: Avatar system [54]



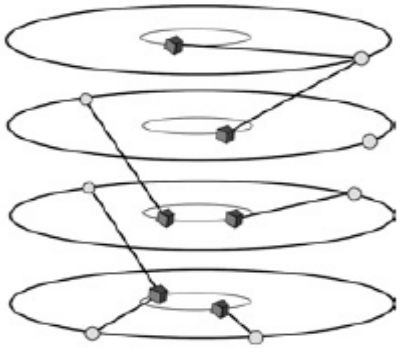
k: 3D hyperbolic visualizations [125]



q: Avtar [54]



f: Swift-3D [116-118]



l: Tudumi [127]

Figure 2.30: Location, conceptual and plotting based visualization

2.2 UML for analysis and designing security solutions

This section discuss the designing and validation of INTVS through three fundamental UML modeling techniques propounded by Booch et al. [42], Jacobson et al. [43], and Rumbaugh et al. [52]. Further a vast literature is studied related to UML usage to valid the proposed IT solution.

Dwyer et al. [45] demonstrate the network layout and details visualization based on UML class diagram and biological networks. Ray et al. [51] used UML based use case, state-chart and sequence diagram to reveal the attack model and its defense mechanism along with the usage of XML and pseudo-code. Wenhui et al. [53] exploited the UML use case diagram to validate the VPN service management system.

Devamalar et al. [44] used UML activity, class, collaboration, use case, sequence and state-chart diagram to validate the web centric intelligent health care diagnosis system and named it as web based diabetes expert system WEBDIACIN.

Hu et al. [46] used UML approach to designing Resource Oriented Architecture (ROA) through a case study on Smart Grid Home Area Network (HAN). This study helps to know the process. Furthermore, how to link it to an existing Service Oriented Architecture (SOA) environment through UML?

Jakimi et al. [47] suggested a requirement engineering process to get the global description of a required service of the system, validated through UML scenarios.

Jong [48] reveal the designing of an embedded real-time systems for a telecommunications application while showing the applicability of the flow to control and data-dominated types of systems, through UML and SDL.

Ming et al. [49] introduced integrated network solution to manage the multiple technologies and domain networks efficiently and cost effectively. They used use-case diagram, collaboration diagram, object diagram, sequence and class diagram of UML diagrams to define the information related to protocol used by the existing and emerging telecom networks and management technologies.

Kukkala et al. [50] developed and implemented a medium access control protocol named as TUTMAC while validating it with the help of UML.

Chandra et al. [110] validate the network protocols for their implementation over software and / or hardware in an efficient manner through UML in two phases. The first phase dealt with protocols modeling using UML, and the second phase is about the java based executable specification and prototyping of protocols. Lipsinky et al. [55] used UML, for space reliability modeling formalism of distributed systems and services compliant.

2.3 Fuzzy based approaches to improve the reliability of the security systems

To measure the reliability of a system fuzzy logic is good option, because fuzzy is more about varying values, contents and inputs conditions significantly according to context according to Verma et al. [78]. Chen [64] [71] presented a method for analyzing the fuzzy system reliability using fuzzy number arithmetic operations. Pan and Yun [67] used the fuzzy set theory for modeling the fuzzy system structure and proposed the new procedure to calculate the system reliability.

Gang Wang et al. [76] developed a new approach FC-ANN for intrusion detection system to detect the intrusion using Artificial Neural Networks and fuzzy clustering. Further fuzzy fault tree

approach is a leading mechanism of analyzing the reliability, which is not only limited to hardware-based system only like electrical, mechanical, civil engineering solution, etc. but also many researchers capitalized the importance of analyzing the reliability of software domains effectively like M. Al-Kuwaiti et al. [77].

Fault tree analysis (FTA) is a powerful diagnosis technique and is widely used for demonstrating the root causes of undesired event in system failure. The concept of fault tree analysis was developed by Bell telephone laboratory in 1961. It is widely used in many fields such as in LPG tank proposed by Wu Rong and Deng Xin [81], Liang and Wang [63] used fault tree analysis in microelectronics reliability, also used fault tree analysis by nuclear reactors, chemical and aviation industries.

Singer [58] presented a fuzzy set [56] approach for fault tree and the reliability analysis in which the relative frequencies of the basic events are considered as fuzzy numbers. Chanda and Bhattacharjee [68] presented a fuzzy fault tree based reliability analysis of an optimally planned transmission system. Ching-Trong et al. [66] developed hybrid fault tree analysis while using the fuzzy sets in 1997.

Chang et al. [70] developed a novel approach by incorporating digraph models, fault trees and fuzzy inference mechanisms in a unified frame work to relieve the online computation load.

Bai and Asgarpoor [72] described a fuzzy-based analytical method and a fuzzy-based Monte Carlo simulation (MCS) technique to obtain a possibility distribution of reliability indices for substations.

Volkanovski et al. [79] proposed a new method for power system reliability analysis using the fault tree analysis approach. Li Y and Song K [80] used FTA to develop the software for space

camera. Kumar et al. [85] used fuzzy sets to calculate the fuzzy reliability of marine power plant. Also, Kumar et al. [86] used vague sets to calculate the fuzzy reliability. Kumar et al. [84] used intuitionistic fuzzy fault tree approach to calculate the reliability of computer security system. Mahmood et al. [87] reviewed the effectiveness of FFTA.

After having a broad discussion on existing VizSec, utility of data mining, usage of UML and fuzzy fault tree approach to improve the reliability of security solution, one can be in a position to understand the problem with existing tools, further able to identify the objectives to be achieve, and how to achieve these objectives through a methodology (discussed in Chapter -3) and develop a unique framework which can eliminate the existing problems.

Conclusions:

This chapter dealt with literature of a) security through data visualization b) security through data mining techniques and c) security through data visualization based on data mining. The contribution of key researcher has been discussed, and the features of various tools including Flodar [1], Nam [2], NVisionIP [3], VISUAL [4], VisFlow-Connect [5], NetViewer [6], Wisconsin Netpy [7], PortAll [8], IDS Rainstrom [9], IDGraphs [10], TNV [11], RUMINT [12], AfterGlow [13], InetVis [15], FlowTag [16], Flamingo [17], NfSen [18], PRISIMA [19], LogView [20], HMAT [21], FloVis [22], NAV [24], VIAssist [25], ScanViewer [26], CCScanViewer [27], InfoVis (EIC process Model) [28-30], IVTK [31], VizSec based on PE and RE [32] has been discussed to understand their working w.r.t. OSI layer (like layer 3rd, 4th or 7th), media portability and time (real time or offline or both. Further, these tools were investigated for having their own capturer, parser, threat detector and visualize reporting tool. Different visualization schemes and their utility are elaborated, so that a reader can take an inference on how these schemes can be developed for future VizSec. A table was prepared to compare the discussed data visualization schemes. Further literature related to usage of UML to validate the security system has been discussed. Lastly literature of usage of fault tree analysis and fuzzy faulty tree is discussed for calculating the reliability of a system.

Chapter 3

Integrated Network Traffic Visualization System (INTVS): - A Proposed Framework

This chapter formulates problem based on gap analysis as per literature review carried out. In this chapter, a desired list of attributes which must be reflected by a visualization based security system (VizSec) are highlighted. Objective of this research work have been defined. Further, this chapter presents research methodology, INTVS framework and composition. Finally this chapter presents INTVS designing and validation through unified modeling language (UML) by virtue of structural, behavioral and architecture modeling.

3.1 Gap analysis and problem formulation

From the previous chapter, it is learnt that all the discussed tools/frameworks/ approaches/ schemes for VizSec, are having some limitations in respect to be a complete VizSec.

The problems with present tools discussed in Chapter 2 is as follows:

In Flodar [1], lack of information about the OSI layer, media and data source. Wisconsin Netpy [7], IDS Rainstrom [9], IDGraphs [10], Rumint [12], NfSen [18], VIAssist [25] are capable to visualize only third layer data of OSI model.

NVisionIP [3], VISUAL [4], VisFlow Connect-IP [5], SIFT [5], NetViewer [6], PortAll [8], TNV [11], InetVis [15], FlowTag [16], Flamingo [17], FloVis [22], NAV [24], ScanViewer[[26], CCScanViewer [27] and InfoVis [28][29][30], all these tools are limited to data visualization of third layer and fourth layer of OSI model.

Although Nam [2] is capable till seventh layer data visualization of OSI model, there are flaws with respect to traceability of lost packets and real time visualization, AfterGlow [13] is capable to perform forensic analysis of OSI layer 7, but is dependent upon a parsing tool, which parses the log file to CSV format. Log View [20] and IVTK [31], both are able to visualize the 7th layer data, but do not have their own capturing module.

Based on review of these tools, it is clear that there is not even a single tool, which is capable to produce interactive visualization of network traffic of large network in seamless manner. These visualization systems lack in one way or other, in order to capture packets, tokenize, parse and then visualize them in an integrated interactive manner until Layer 7, in real-time for guided and unguided media. Proposed work INTVS - (***a system, which will work in both guided and unguided media, captures network traffic, tokenizes, parses and visualizes network traffic up-to layer 7 in visual reporting form, to detect the threats and to gives the status of resource utilization in real time)*** offers, solutions to the said gaps and offers integrated framework.

Desired list of attributes of a VizSec: A visualization based security system (VizSec) must possess following desire list of attributes:

a) *Media Support* - It must support, both guided and unguided media, to capture the network raw traffic.

b) *Interoperable* - A VizSec must be compatible with variety of operating environments.

c) *Real Time and Forensic Analysis* - A VizSec must work in real time & help in carrying out forensic analysis of network traffic.

d) *Security Analysis of Application Layer Traffic* - VizSec must facilitate visualization of application layer data such as http, ftp and email traffic load per machine, per network, w.r.t. whole campus area network. type of web service and all related stats like volume of data uploaded and downloaded, minimum, maximum, average, above average, below average, number of total networks, total number of machines live in CAN and machine under attacks.

e) *Visual Information* -INTVS must be a smart and intelligent. It must learn from its traffic patterns and generate alerts according to the customized settings. It must follows the principles of Shneiderman [35] such as:

Overview: getting an overview of the entire network (Network).

Zoom: zooming to select an item of interest

Filter: filtering of non-required items.

Details-on-demand: option of selecting an item or group for detail as when required.

Relate: must offer the facility to relate the items and give view based on relationships among items.

History: must maintain the history of actions to support undo, replay, and progressive refinement.

Extract: allow drill down to extract a sub-collections according to query parameters. .

f) *Scalable* - INTVS must have the capability to scale with going volume of traffic.

g) *Intelligence* - VizSec must understand the traffic patterns and take necessary action in self-defense in the absence of the manual intervention.

h) *Reporting* - VizSec must report network health in self-explained way. It must respond to all the queries such as – how many VLANs are detected, how many hosts are live, data

volume downloads and uploads by the single host, protocol wise bandwidth utilization, machine underutilization, machine over utilization, machine under attack, machine with network violating Network policy etc. The system must help network security analysts to view intended information and piping it to visual reports like network bandwidth consumed by particular application layer traffic with reference to total traffic as well as should able to report attack vectors.

3.2 Objectives

a) To review & analyze existing network traffic visualization models.

b) To propose & design a model for threat detection, analysis and reporting for HTTP/FTP/SMTP data

c) Implement and Validate the model on live – distribution Network Traffic Visualization System (NTVS).

3.3 Research Methodology

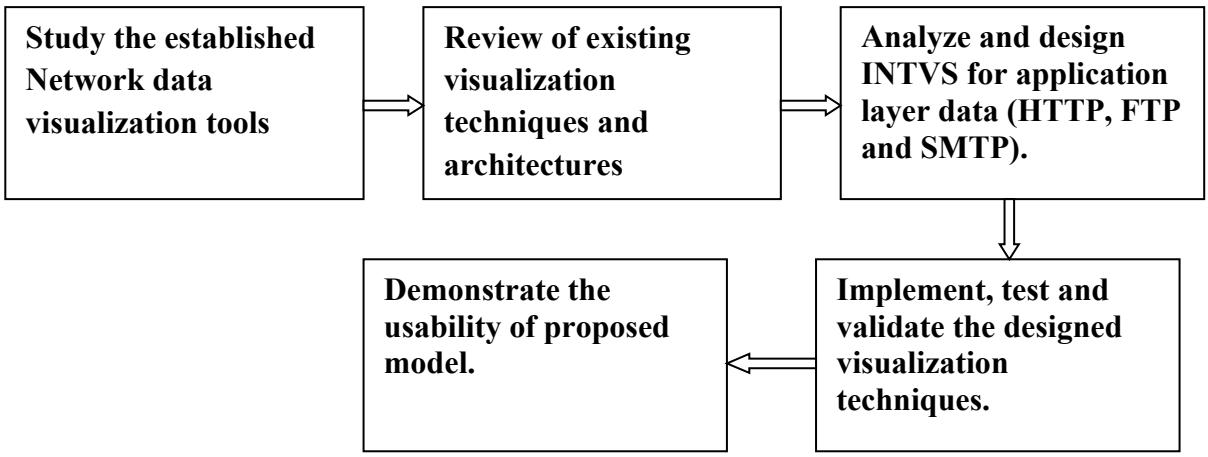


Figure 3.1: Research Methodology

To achieve the objectives mentioned in Section 3.2, a research methodology as shown in Figure-3.1 is adopted:

a) Study the established network data visualization tools - as discussed in Chapter 2.
b) Review of existing visualization techniques - as discussed in section 3.1.
c) Analysis and design of INTVS for application layer data (HTTP, FTP and SMTP) - is discussed in Section 3.5.
d) Implement, test and validate the designed visualization techniques - is discussed in Chapter 4 and Chapter 5.
e) Demonstrate the usability of proposed framework - is discussed in Chapter4 and Chapter5.

3.4 INTVS Framework

This subsection deals with INTVS framework which is having six main functional modules as shown in Figure 3.2. These modules are classified as follows: *First module* is a network traffic capturer (which helps to capture network data). *Second module* is tokenization module, which performs the tokenization of the data according to network traffic properties. *Third module* is parsing module and parses the network traffic to use it as an input for visualization engine, which is the *Fourth module*. *Fifth module* is dealing with real time data analysis with the help of data mining techniques and *Sixth module* is helpful to perform forensic analysis of the network data. It also uses data mining techniques. INTVS framework programmed using multi-threads to handle these modules effectively. For INTVS, to handle a big data set, in experimentation framework, memory (RAM) and speed (Processor) are of prime concern.

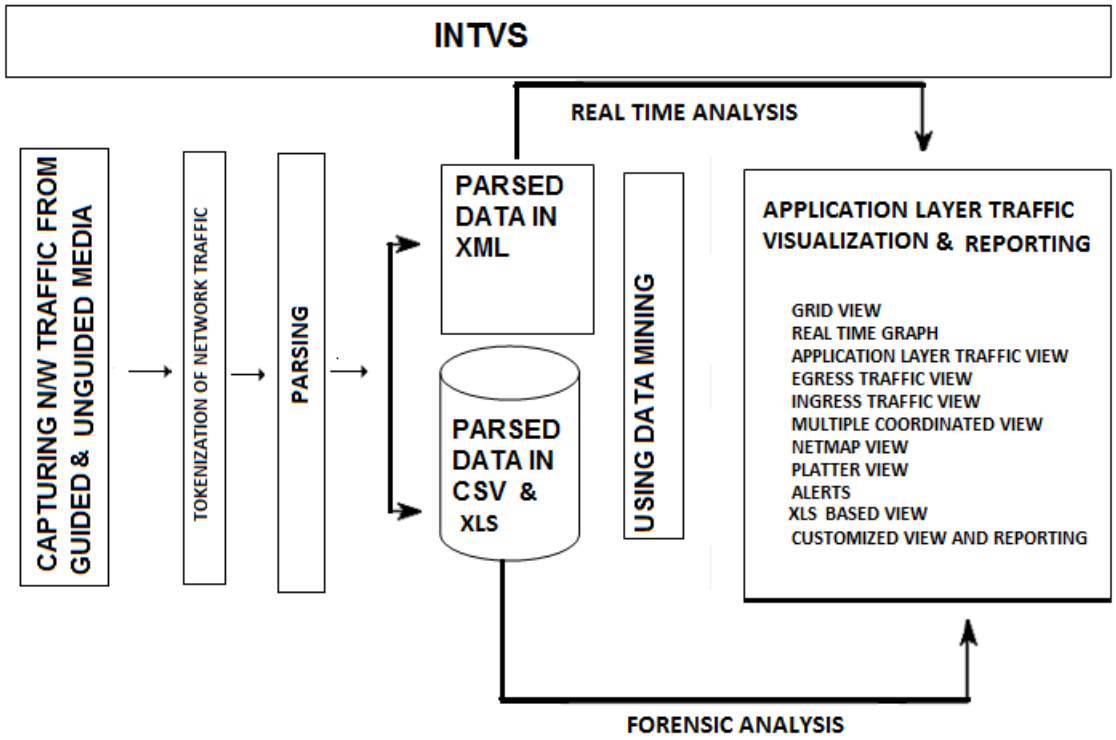


Figure: 3.2 INTVS Framework

3.4.1 INTVS framework supporting methodology

In the study, a methodology followed to support the INTVS framework consists of three prime layers as shown in Figure-3.3; (i) input layer, (ii) processing layer and (iii) output layer. In the present research, a hybrid approach of data mining is used to analyze network traffic patterns effectively. To manage the network resources, a supervised data mining technique is applied to cater the situation. Network policy is considered as an input for this processing layer. On the other side, un-supervised data mining technique is applied to learn data pattern through clustering and association methods. Suitable statistical analysis of these data patterns is helpful in detecting and reporting threats.

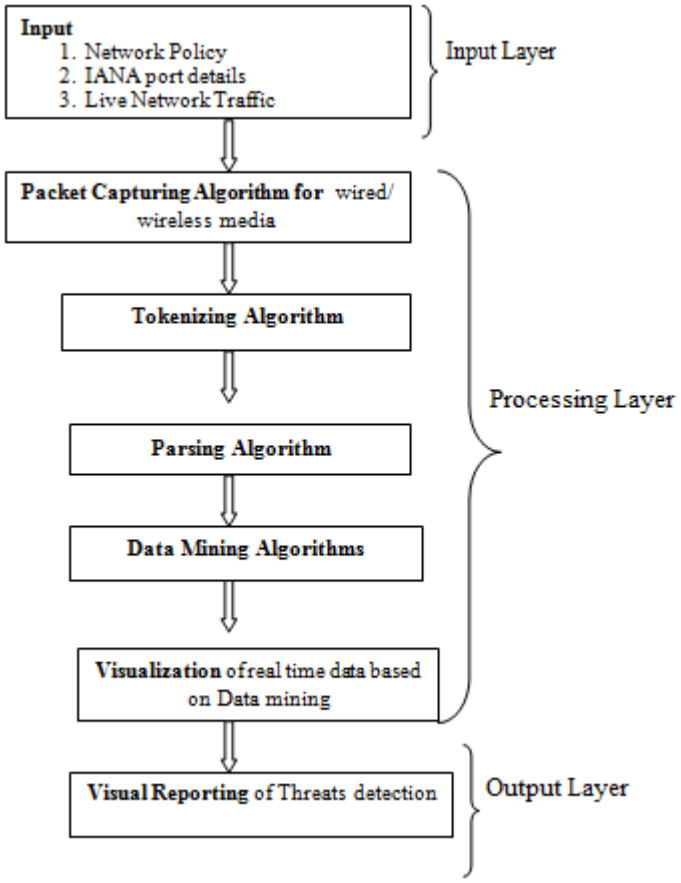


Figure: 3.3 INTVS Framework Supporting Methodology

3.4.1.1 Input layer

Input layer perform following tasks:

a) Live packet- capture from guided/unguided media.

b) Network policy- includes the network traffic rules for ingress and egress traffic, bandwidth allocation rules for all user, Network monitoring rules to understand the state of network, VLANs, nodes whether normal or under attack mentioned through set theory in Chapter 4, section 4.1.4.

c) Standard port numbers [140] - Internet Assigned Numbers Authority (IANA) numbers list used for classification of network traffic.

3.4.1.2 Processing layer

Processing layer is as certainly with following tasks:

a) Capturing module captures raw data (packets) from live media in real time with help of Winpcap/ Libpcap, JAVA and JRE.

b) Data cleansing performed by analyzing and tokenization of attributes of captured packets, all this happed with help of tokenization module, which also using Winpcap/ Libpcap, JAVA and JRE

c) Parsing of cleansed data performed for real time and forensic analysis - This is done with help of parsing module, which takes the tokenized data as input and parsed file as an output.

d) Further in this step, data piped to real time analysis stub as well as forwarded to visualization engine. XML formatted generated parsed file used for real time analysis and CSV/Excel file used for forensics analysis.

e) Next, data mining rule based classification algorithm is used to classify packets based on Internet protocol (IP) address (source IP and destination IP), port number (source machine port, destination machine port), transmission control protocol TCP) packet, user datagram protocol (UDP), virtual local area network (VLAN), campus area network CAN, wide area network WAN, ingress/egress, session wise, packet length, window size and application layer protocols. This classified data is also forwarded to visualization engine. Rule based classification of intruding machines.

f) Data mining rule based clustering of ports, machines and networks for below average, above average and normal w.r.t. TCP/UDP, ingress/egress and upload/

download traffic. Application layer protocols based cluster of machines and network of w.r.t. above average, below average and normal traffic. Rules based clustering of machines and network who are breaks the Network policy rules.

g) Data mining association rules are applied to establish relationship among key attributes of traffic pattern to identify the teardrop attack and sync flood attacks. Under this scheme packets are analyzed based on sessions and sizes, if a large number of packets of tiny size crossing under same session or different session over a machines, concluded as teardrop attacks and if packet size is very large then it is concluded as ping of death attack.

h) Finally, data mining rule based threat detection and visualization take place.

3.4.1.3 Output layer

Output layer generate the following reports

Visualization engine displays demanded and customized views, to analyze, understand and respond various conditions during attacks. Following visualization are the outcomes of the output layer along with mouse-over and drill down facility:

a) Grid view

b) Platter view

c) Listmap view

d) Real time line graph

e) Alert log file

f) Two dimensional network and machine level view

g) Parallel coordinated visualization

3.4.2 Composition of INTVS

Composition of INTVS is shown in Figure 3.4, consists of three layers:

a) Composition layer

b) Process layer and

c) Output layer.

At composition layer, all the basic development software (tools, plugins, library, operating system, system devices drivers) are used. Winpcap/ Libpcap, JAVA and JRE are used for capturing module well as for tokenizing module. JAVA and JRE are used for parsing module. JCommon-1.0.17, JfreeChart-1.0.14, jgrapht-jdk1.6 libraries, JAVA and JRE are used for visualization module.

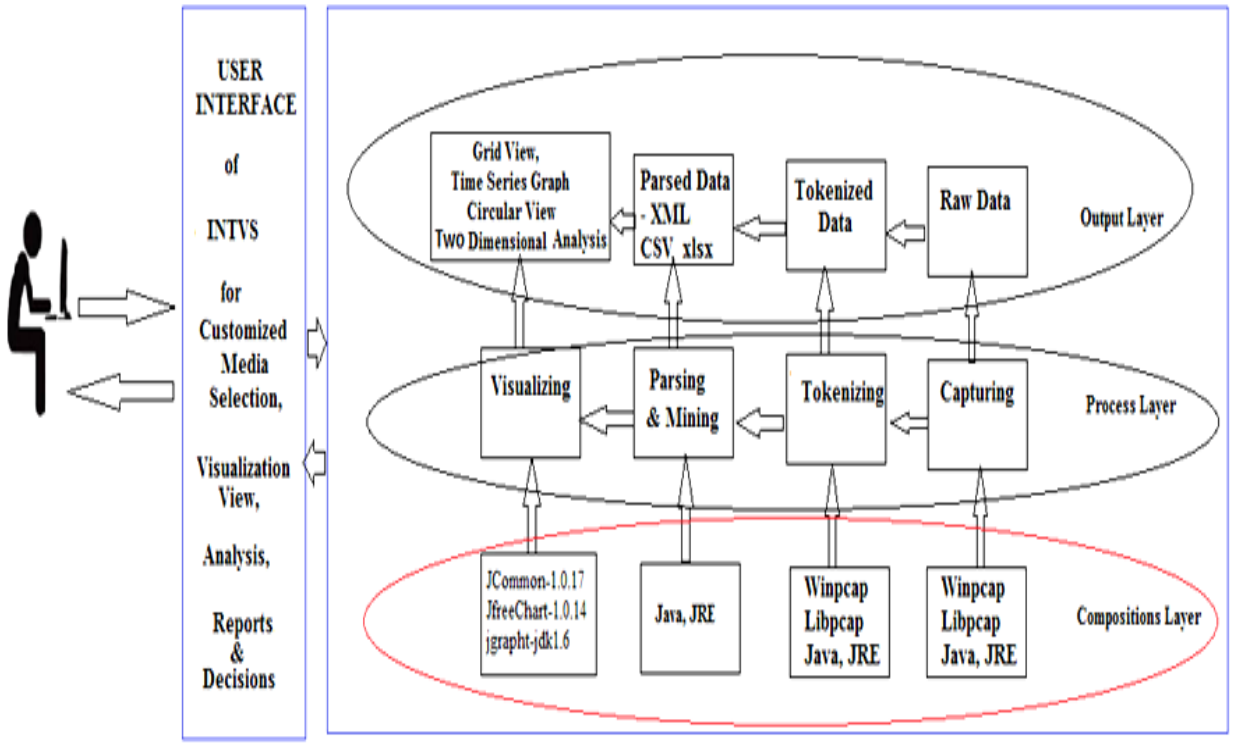


Figure 3.4: Composition Diagram of INTVS

Process layer: Capturing module takes input from composition layer and capture the raw data and gives its results (lives captured data) to output layer. Further, it gives the captured raw packets to tokenizing module, which further tokenize the captured packet and shows the tokenized data at output layer. Also, tokenize the raw packets and give it to parsing module. Parsing module generates parsed file in the form of XML file and comma separated value (CSV) format as an output. Mining module mined the network traffic data based on classification, clustering and association rule for results. The visualizing module at process layer takes the input from parsing module and use JCommon-1.0.17, JfreeChart-1.0.14, jgrapht-jdk1.6 libraries JAVA and JRE to generate the visual report as per the available choice to the user.

Output layer: The output of various module can be viewed such as live raw network traffic, tokenized data, parsed data, different visualization schemes: Grid view, Platter view, Listmap view, real time line graph, alert log file, two dimensional network and machine level view, Parallel coordinated visualization.

3.5 Validating INTVS design through UML diagrams

The use of UML in analyzing, designing and validating IT based solutions discussed in Section 2.2 of Chapter 2, motivated this work to analyze, design and to validate the INTVS framework.

This sub - section discusses validation of INTVS through three forms of UML modeling, step by step by, following the fundamental rules of Booch et al. [42], Jacobson et al. [43] and Rumbaugh et al. [52].

3.5.1 Use case of INTVS

Use case diagrams are used to structure the requirements of system analysts and network administrator after understanding their requirements. The purpose of use case diagram is to understand the dynamic aspect of INTVS including internal and external influence of INTVS, so that the functionalities and actors can be easily identified. In Figure 3.5, external actors are *Network Admin* and *Users.* Internal actors are *INTVS* main module while interacting with actor *Network Database*. The functionalities required by *Network Admin* are defining network policy, which includes the traffic rules for ingress and egress traffic, bandwidth allocation rules, network policy, network monitoring to understand the state of network, VLANs, nodes whether normal or under attack. Further Network Admin would like to control network traffic while taking some corrective decisions to protect the network resources from malicious activities and to ensure the effective utilization of bandwidth. Another external actor are network users, who want to understand whether his or her node/ machine is under attack or not, for which one can select the media, view the capturing of raw data, see various network information in a visual form, and take decision with help of functionality offered by the INTVS. For internal actor INTVS, INTVS database accepts the inputs and returns the results as a black box. The INTVS use case diagram specifies the above said events of INTVS and their flow in a specific manner and produces the high-level view of the INTVS for its users.

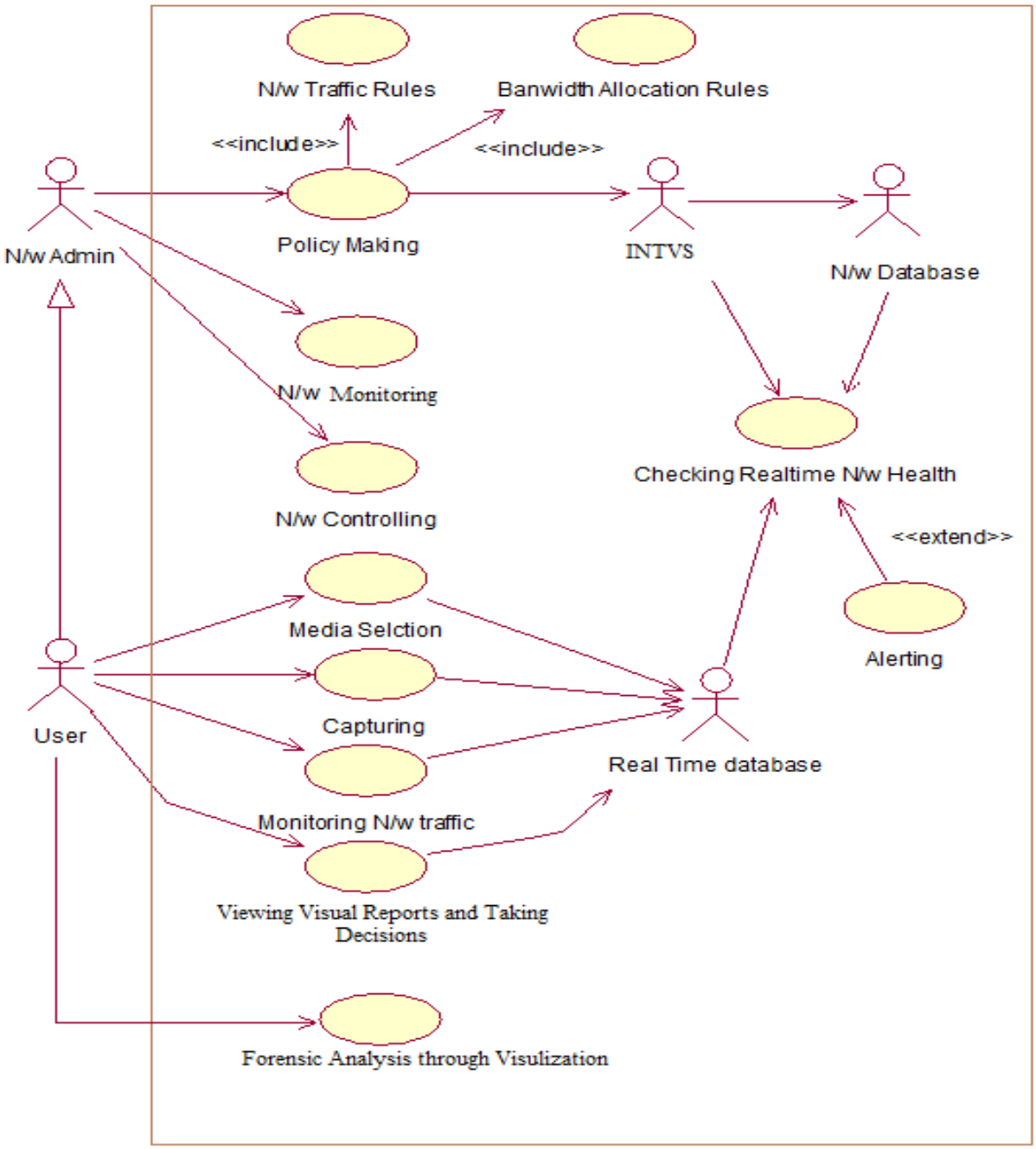


Figure 3.5: INTVS - Use Case Diagram

3.5.2 Activity diagram of INTVS

In Figure 3.6, INTVS - activity diagram demonstrates the dynamic behavior of the INTVS. An activity is defined as a function carried out by the system. After identifying the activities, it is necessary to understand how these activities are associated with constraints and conditions. In Figure 3.6, ADMIN, USER, INTVS and DATABASE system are identified as swim lane. The activities performed by the ADMIN are updating Network Policy (In this activity Network Administrator updating Network policy according to need of time), Creating Network policy Database (Here admin is creating and managing Network policy database), Selecting of Media (Selecting a media from the available options-wireless or wired), monitoring visualization (viewing different visuals and taking inference for decisions), and decision making (taking appropriate decisions).

The user group performs various activities such as media selection, monitoring visualization (viewing different visuals and taking inference for decisions), and decision making (taking appropriate decisions). Swim lane INTVS performs various activities such as: taking inputs (from ADMIN and USER), capturing live data (after selecting media, INTVS capture the network traffic), tokenizing and parsing raw network traffic in usable form, understanding the data patterns (taking inference from data and its patterns), checking validity of traffic (here INTVS checks the malicious data and generating alerts), using data mining (in this activity INTVS used the data mining schemes to understand the network traffic in effective manner, Visualizing Network traffic data (in this activity INTVS produces different visuals of network traffic), exporting data for forensic analysis, forensic analysis through visualization (here INTVS helps in Forensic analysis of network traffic for future usage).

he fourth swim lane DATABASE has the following activities: Network policy database, maintaining real time network data, maintaining historical data, and updating database. INTVS activity diagram can also be used to construct the executable system by using forward and reverse engineering techniques.

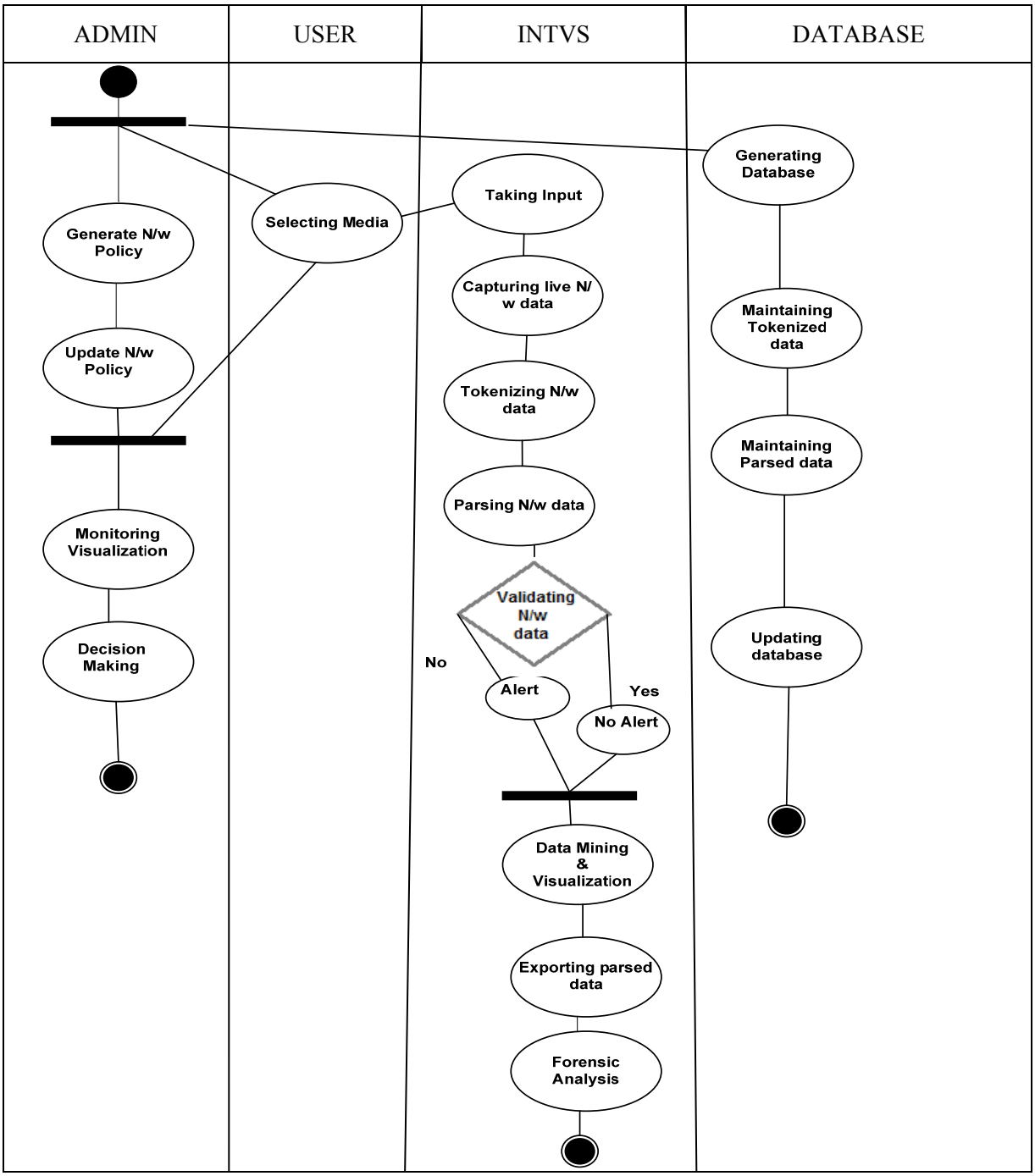


Figure 3.6: INTVS Activity Diagram

3.5.3 Class Diagrams of INTVS

INTVS class diagrams are used to describe the static view of the INTVS, and to show the collaboration among the various elements. INTVS class diagram also describes the functionalities performed by various sub-systems. Further INTVS class diagram leads to the construction of INTVS network security application using JAVA. INTVS Class diagram is also considered as the base for component and deployment diagrams of INTVS. INTVS Class diagrams are also used to construct the executable code for forward and reverse engineering of INTVS.

INTVS class diagram can be used for construction purpose as it facilitates to map with the static view directly with object oriented languages like Java, C++ etc. In Figure 3.7, INTVS - Class diagram, there are five classes mentioned namely as User, INTVS, Media, Network Policy and Decisions.

Further, User is generalized as Network User and Network Admin sub-class. Network Admin and User sub-class possess the same attributes, but Network Admin is having more operations as mentioned in Figure 3.7. INTVS class defined various operations: taking input ( ), capturing ( ) - both operation tokenizing ( ), parsing ( ), data mining ( ), visualizing ( ). Network policy class possess three attributes, rule_id, rule_name and rule_type; and two operations, i.e., Policy formulation ( ) and Policy update ( ). Class Decision is having attributes, that are, decision_id, decision_name; and having three operations allow ( ), drop ( ), Update ( ). Class Media possess the type_of_media attribute. Class Network details is subclass of INTVS having following attributes srno_of_pkt, SIP, DIP, src_port, dst_port, pkt_length, pkt_window_size, pkt_session, pkt_timestamp, pkt_protocol. subclass visualizing possess the graph_id and graph_name attributes; and have display view ( ) and display customized View ( ).

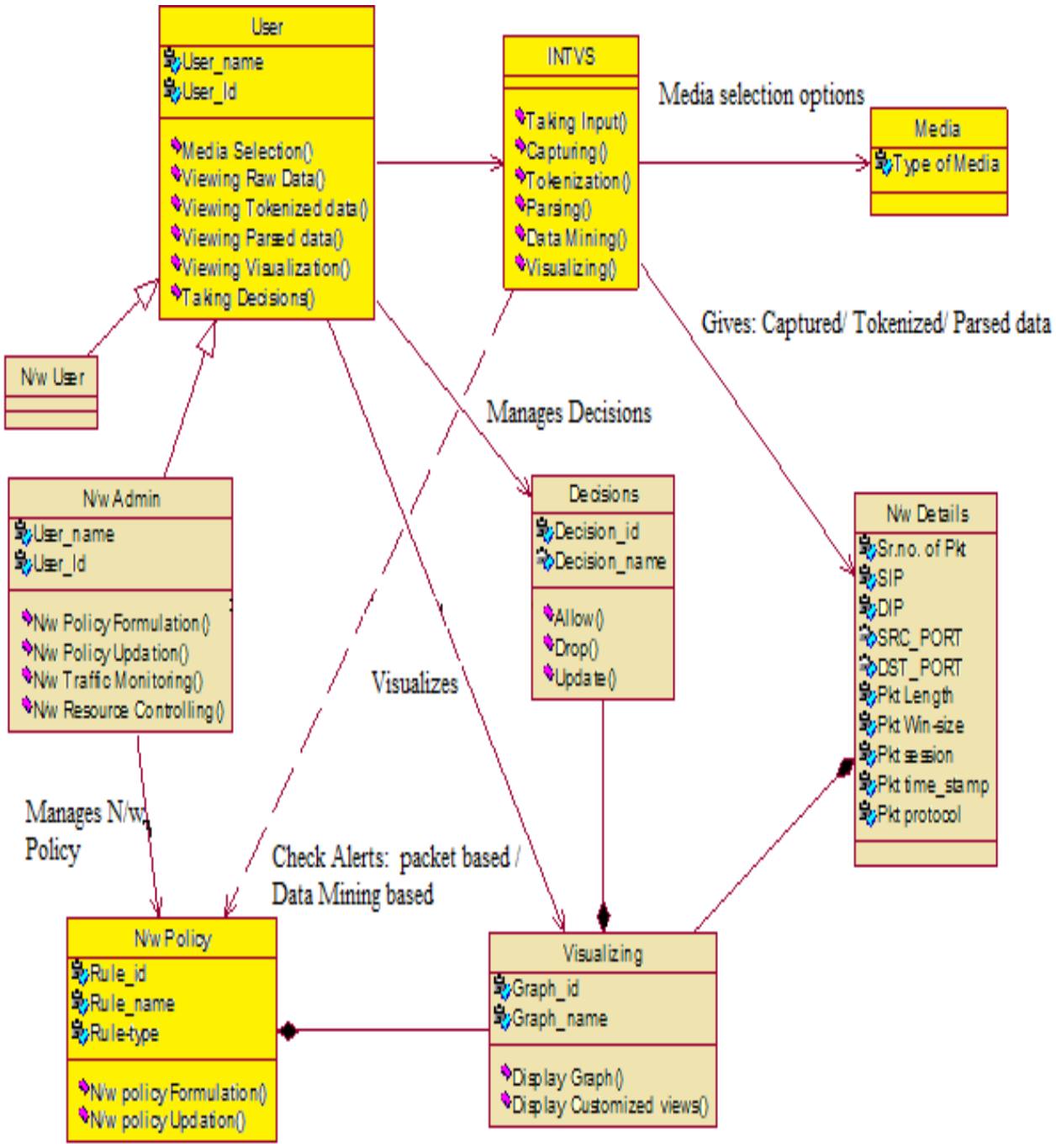


Figure 3.7: INTVS - Class Diagram

3.5.4 Sequence Diagram of INTVS

Sequence diagram are also known as interaction diagram which are used to show the time sequences of messages flowing from one object to another. In Figure 3.8, sequence diagram is used to visualize the sequence of messages generated in INTVS to perform specific functionalities. For example, object *:Network admin* defines the network rules while sending the message "Define_Rules" to Object *:INTVS*, then *:INTVS* object send these details about the network resources and their utilization rules to *:Network Policy Database* object and so on a functionality of defining network rules and network policy is accomplished. To update the rules as per the need of time another functionality is initiated while sending a message "Update_Rules" from *:Network Admin* object to *:INTVS* object.

Then same is communicated by *:INTVS* object to *:Network Policy Database* object through "Get_updates" message. Another functionality is selecting a media initiated by *:Network Admin* object and send a message "Select_Media" to *:INTVS* object. Then *:INTVS* object take this call and start to capture network live data from selected media, while sending a self-message "Capturing_Network_live_data". On similar way the functionality of tokenizing and parsing are accomplished by *:INTVS* object. The functionality of detecting any malicious activity and generating alert is done by *:INTVS* object after receiving the call from *:Network Policy Database* object.

Then *:INTVS* object sequence the next functionality of sending parsed data to database for real time and forensic analysis. The functionality of generating the various customized view of network traffic. Then *:Network admin* leads the following sequences to take a decision like sending a call to *:INTVS* object "Visualize_graph" then object *:INTVS* gives the choice for any specific view. A view is selected by *:Network_Admin* object, then a selected view is displayed by *:INTVS* object. Based upon to see a particular view, a decision is taken by *:Network_Admin* object, and so the functionality of how a decision supposed to be taken by the network administrator/user is accomplished. These interactions among the components of INTVS are very important for implementation and execution perspective.

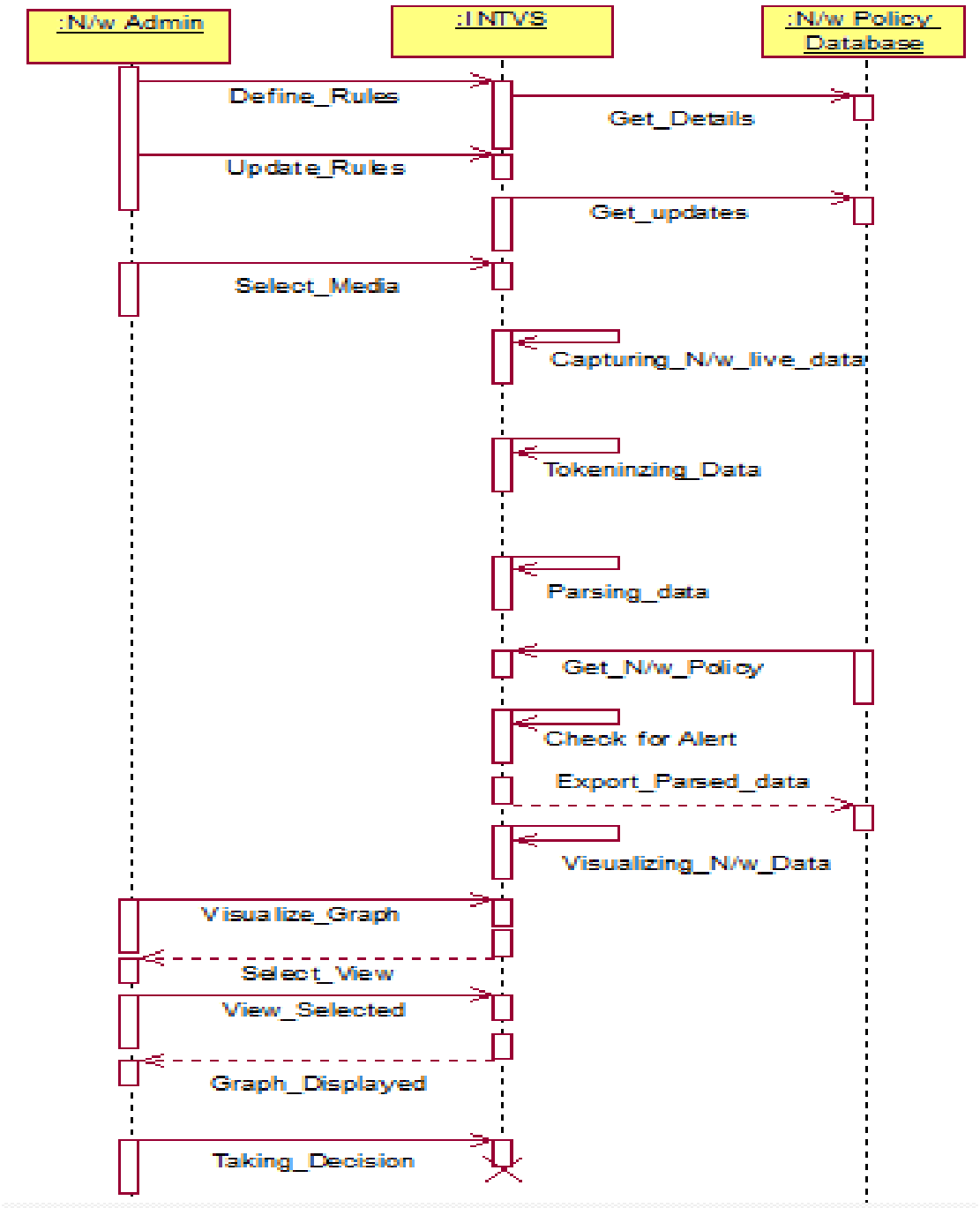


Figure 3.8: INTVS - Sequence Diagram

3.5.5 Collaboration Diagram of INTVS

The second interaction diagram is collaboration diagram which shows object organization of INTVS as shown in Figure 3.9. In collaboration diagram the method call sequences are numbered and these numbers indicates how the methods are called one after another.

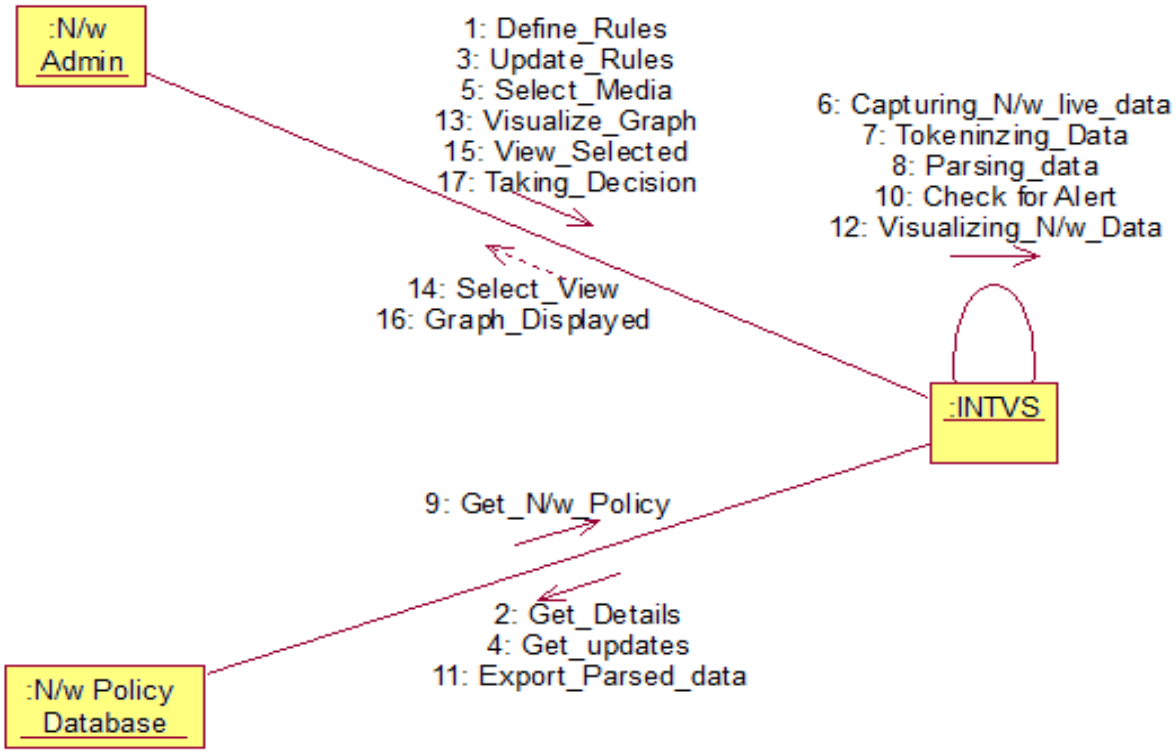


Figure 3.9: INTVS - Collaboration Diagram

Here the method calls/ messages are similar to that of a sequence diagram but the difference is that the sequence diagram does not describe the object organization as shown in Figure 3.8, where the collaboration diagram shows the object organization. When time sequence is important then sequence diagram is used and if organization is required then collaboration diagram is used.

3.5.6 State-chart Diagram of INTVS

State-chart diagram of INTVS defines different states of an object during its lifetime. The state-chart diagram is useful to model reactive systems, INTVS is to respond to external or internal events. In this diagram, the flow of control from one state to another state is described. Here state means defining conditions in which an object exists and changes when some event triggered. The main objective of state chart diagram is to model the life time of an object from creation to termination. Identifying the important objects is the primary job in state chart diagram and in case of INTVS network traffic/ data is identified as an object, which moves from one state to another like raw captured data to tokenized data, then tokenized data to parsed data, then parsed data to visual report form as shown in Figure 3.10. Meanwhile some events are triggered which causes the changes in states of network data. These include INTVS selecting media, capturing, tokenizing, parsing and alerting, data mining, visualizing and decision making.

When selecting any media, an event is triggering and then next event is captured, the data which is in signal form get converted in raw data states. As soon as raw data comes, then another event, i.e., tokenizing takes place, and converts raw data into tokenized data and changes the state of object from raw data to tokenized data. As soon as INTVS find the tokenized data, another event is triggered named as parsing, which causes the changes of tokenized data in to parsed data state. Generating alert based on altering event, analyzing data according to network policy, and if not valid, generating an alert, another state of data in signal form. Then parsed data is mined, based on data mining event, and parsed data converted to mined parsed data, and it parsed data is generated in INTVS, the visualizing event triggered and parsed data converts to visual report state and then decision takes place.

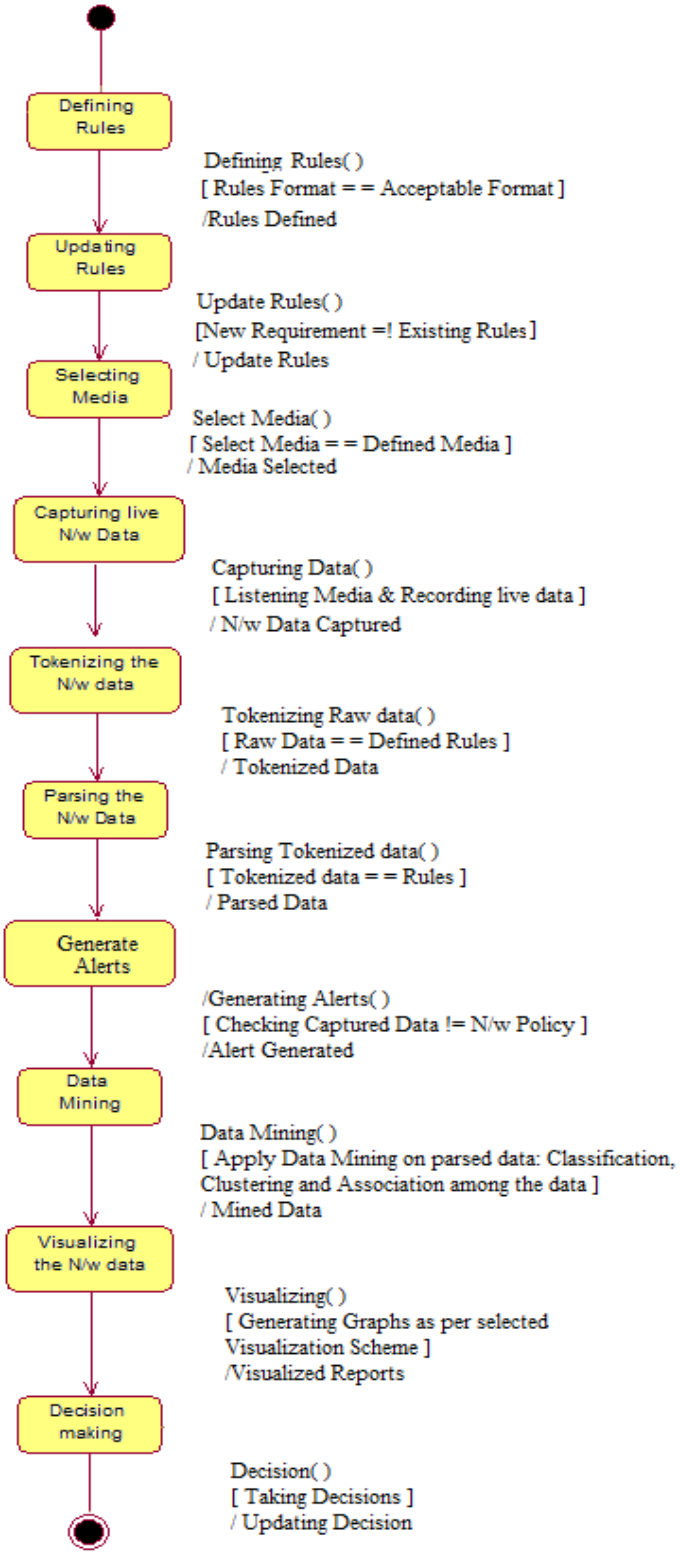


Figure 3.10: INTVS State-chart Diagram

3.5.7 Component diagram of INTVS

Component diagrams are used to model physical aspects of INTVS as shown in Figure 3.11. The INTVS Component diagrams are used to visualize the organization and relationships among components of its system. Component diagram are helpful in implementing the INTVS application effectively and helpful in improving its performance and maintenance. These diagrams are also used to make executable systems through forward engineering. The artifacts identified in INTVS are: XML File/ Database (CSV, xls), JRE, JAVA, Libraries (JAVA libraries are JCommon-1.0.17, JfreeChart-1.0.14, Jgrapht-jdk1.6), Libpcap, Winpcap, INTVS console, INTVS GUI.

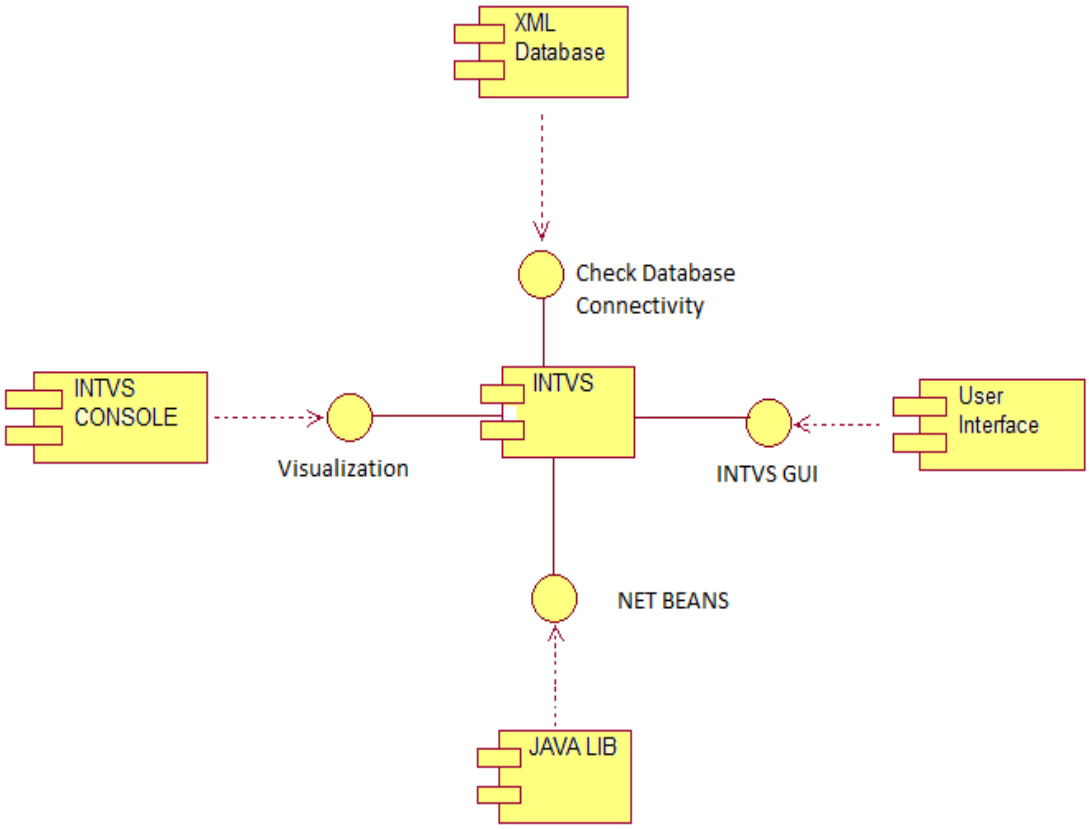


Figure 3.11: INTVS - Component Diagram

3.5.8 Deployment diagram of INTVS

To control the performance, maintainability, scalability and portability of INTVS, there is a need to understand the topology of the physical components of INTVS and where the INTVS components are deployed. The static deployment view of INTVS is shown in Figure 3.12. The purpose of INTVS component diagrams is to describe the components and deployment diagrams shows how they are deployed in hardware as shown in Figure 3.12: The INTVS deployment diagram is showing its hardware topology, describing the hardware components used for software component for deployment, while describing the runtime processing nodes of INTVS.

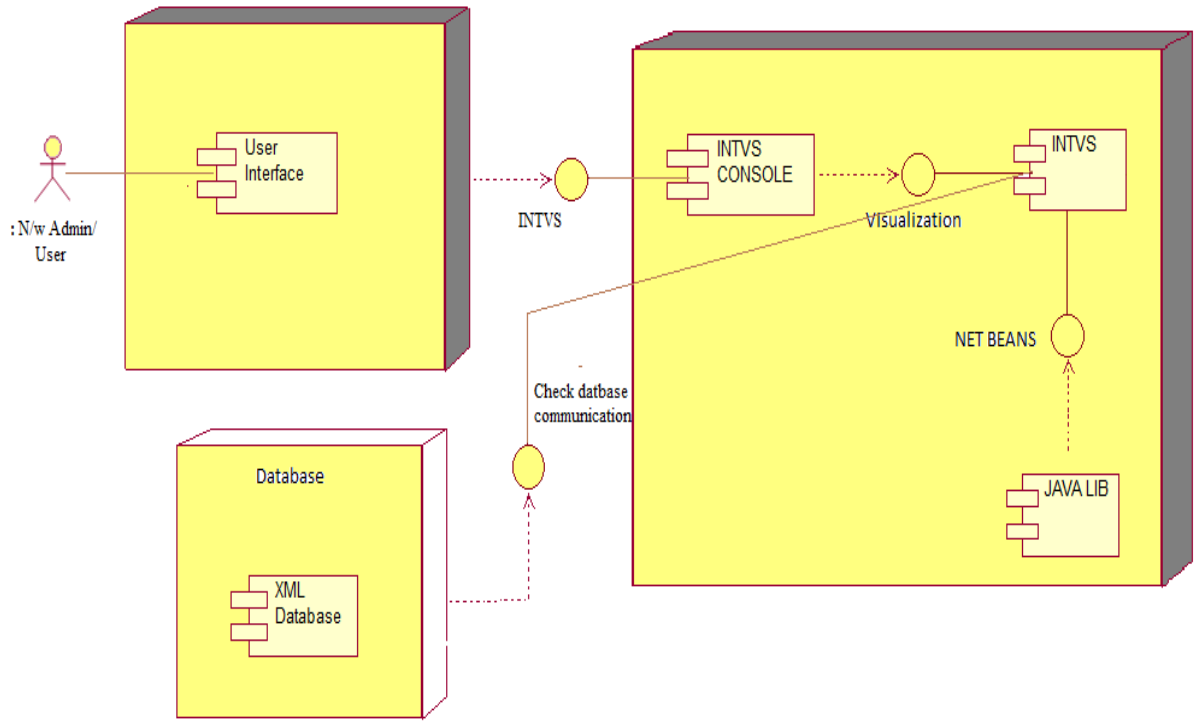


Figure 3.12: INTVS - Deployment Diagram

Conclusions:

This Chapter discussed problem formulation based on gap analysis. A desired list of features, which must be possessed by a VizSec was also prepared. Based on gap analysis the objectives of this study was frozen. A research methodology was proposed to achieve the defined objectives. INTVS framework has been proposed and definition of integrated network traffic visualization system has also been given. Three layer (Input, process and output) based INTVS working methodology was discussed, followed by composition of INTVS. Based on composition diagram, requirements of tools, plugins, library, operating system and system device drivers for the development of INTVS were listed. Lastly modeling has been done to validate visualization framework, using UML for structural, behavioral and architecture modeling of an Integrated Network Traffic Visualization System (INTVS).

Chapter- 4

INTVS Framework Implementation

This chapter gives implementation details of INTVS according to attribute list, and framework design proposed in previous chapter. INTVS has been implemented using Java SDK, Jpcap, Winpcap, JCommon, JfreeChart and jgrapht-jdk libraries. This chapter is divided into three main sections:

- Implementation of INTVS as a complete VizSec solution.
- Working of INTVS in real time environment.
- Working of INTVS in offline mode.

4.1 Implementation of INTVS as a complete VizSec solution

This section deals with the implementation of INTVS as per salient features discussed in Chapter 3 to make INTVS as complete VizSec (a *system, which will work with both wired and wireless media, captures network traffic, tokenizes, parses and visualizes network traffic up-to application layer in visual reporting form, to detect the threats and to gives the status of resource utilization in real time).*

Algorithms for capturing, tokenizing, parsing, visualizing and threat detection modules are developed and then coded in JAVA language as per composition diagram of INTVS shown in Figure 3.4.

The operation of INTVS on windows requires Winpcap and JRE where as in the case of Linux and its variants, user would need to have libpcap and JRE. INTVS uses JCommon-1.0.17, JfreeChart-1.0.14 and jgrapht-jdk1.6 libraries, which helps visualization engine to produce various visualization schemes viz. Grid view, time series graph, Listmap and circular visualization. Parsed file generated in XML format after tokenization are used for real time traffic analysis.

4.1.1 Packet Capture module

Packet capture module interacts with operating system with help of java libraries, which further interacts with underlying hardware to provide interface details. This module then sets the current devices to the one selected by user.

NetworkInterface[] devices = jpcap.JpcapCaptor.getDeviceList();

// jpcap is used for interacting with operating system (OS) to get information related to network devices. This function interacts with OS to get the list of all available network devices and provide this list to the user for network interface device choice.

For i is 0, i is less than device.length, i increments by 1

// iterate through list for each interface

If position *i* in devices equals selected device position

break;

EndIf

EndFor

set captor to jpcap.JpcapCaptor.openDevice with position i in devices, 65535, true.

```
        // initializes the selected device for current session
catch IOException e
        call method System.out.println with e.toString
        // catches and prints exception if any
EndTry
        call method captor.loopPacket with -1
// starts capturing of packets for the selected device
```

Implemented results of INTVS capturing module is shown in Figure 4.1, which shows device list where user can choose detected wireless/ wired media for capturing network data.

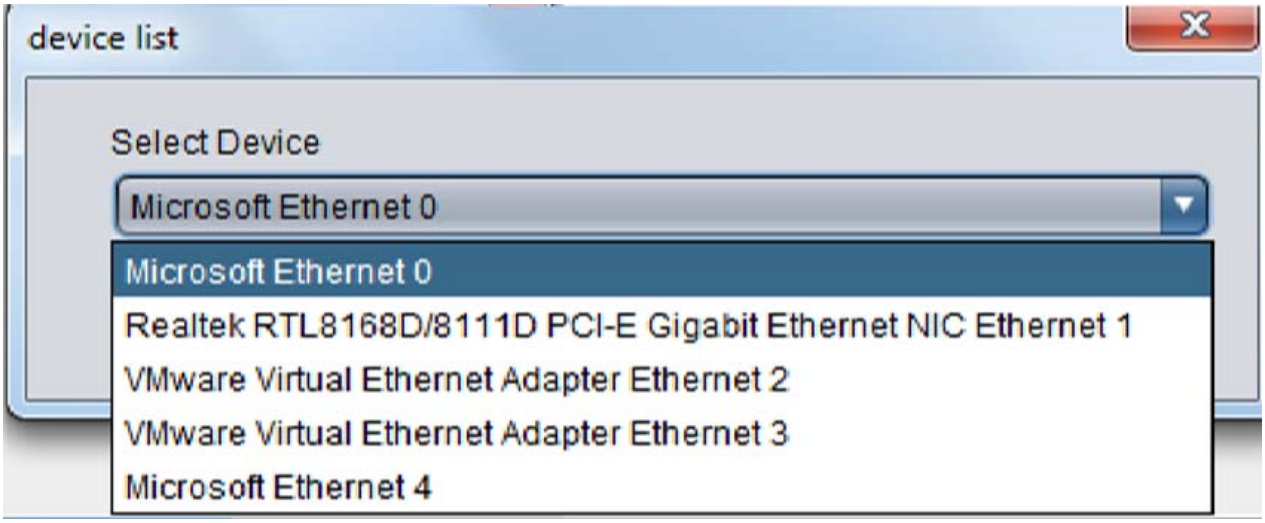


Figure 4.1: Selection of Media

INTVS can capture network data in real time environment as shown in Figure 4.2, which is more or less similar to Wireshark captures.

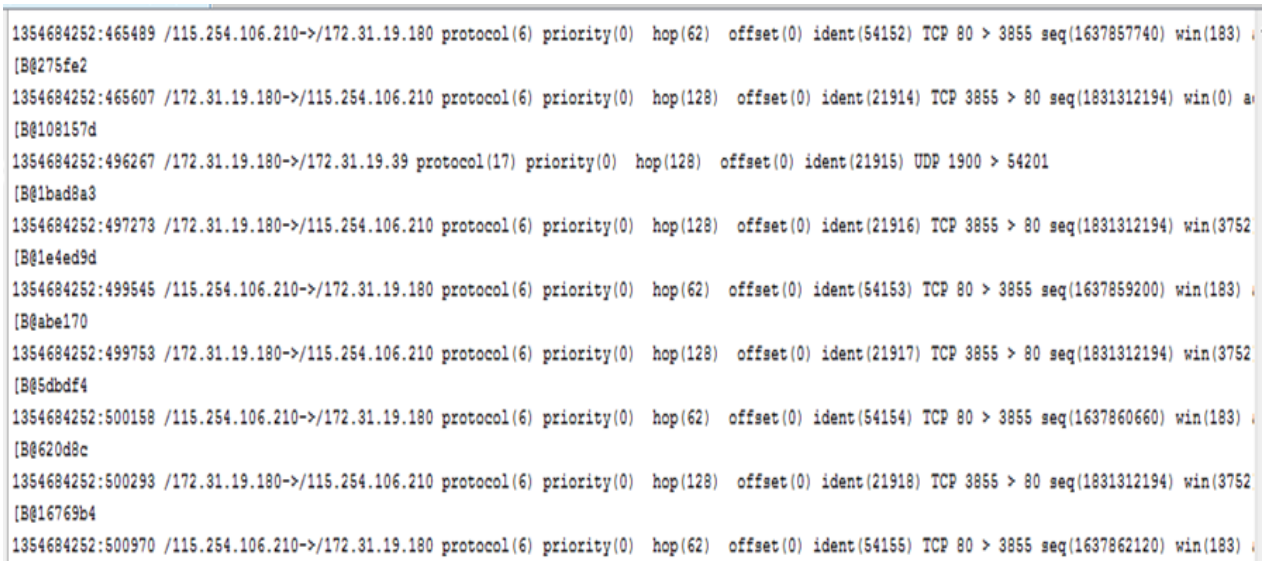


Figure 4.2: Real Time Capturing

Captured data is available with following attributes : time, source machine IP address, destination machine IP address, protocol (number), priority (number), hop(number), offset(number), ident (number), TCP/ UDP source port number, destination port number, seq(number), win(number) as shown in Figure 4.2.

4.1.2 Tokenized Module

INTVS can tokenize captured data seamlessly and use it for parsing in real time environment. Multithread programming concept is used to accomplish required job in parallel. Dedicated multithreads are involved through in tokenizing and parsing process. Output of capturing module is used as input to tokenization module.

// Tokenization module

// Following algorithm traverses through received packet character-by-character, and, then extracts desired information from it for tokenization

character[][] **// character[][] is a 2D char array in which character[i] holds the raw captured packet, character[i][j] holds a character of raw packet**

token[][]

// token[][] is a 2D array in which tokenized data is stored after extracting packets from token [i], where token [i] represent a captured packet number, and token[i][j] represents the intended attributes which are extracted from character[][] array.

Repeat through label start for each character[i] packet in character [][]

// loop for iteration of each of the received packet

If character[i][0]='A' set token[i][5]='ARP' otherwise go to label start1

// ARP packet starts with letter 'A', so it sets the protocol to 'ARP' if it finds the same

start1: While character[i][j] != ':' increment j

// as the first required information in packet is SIP, which is after first occurrence of ':', so the counter is incremented till it reaches ':'

Set token[i][1] = read from character[i][j+1] till character[i][k] != '/' **// SIP is read and stored into token array**

EndWhile

While character[i][l] != '/' increment l

set token[i][2]= read from character[i][l+1] till character[i][m]!=' ' **// DIP is extracted from packet string**

EndWhile

While character[i][m] != '(' increment m

set token[i][5] = read from character[i][m+1] till character[i][n]!=')'

// protocol name is read and stored into token array

EndWhile

While character[i][n] != 'P' increment n

EndWhile

set token[i][3] = read from character[i][n+1] till character[i][o]!='>'

// source port is read and stored into token array

While character[i][o] != '>' increment o

EndWhile

set token[i][4] = read from character[i][o+1] till character[i][p]!=' '

// destination port is read and stored into token array

While character[i][p] != 'seq(' increment p

EndWhile

set token[i][7] = read from character[i][p+1] till character[i][q]!=') '

// sequence no is read and stored into token array

While character[i][q] != 'win(' increment q

EndWhile

set token[i][8] = read from character[i][q+1] till character[i][r]!=')

// window size is read and stored into token array

While character[i][q] != ack ' increment r

EndWhile

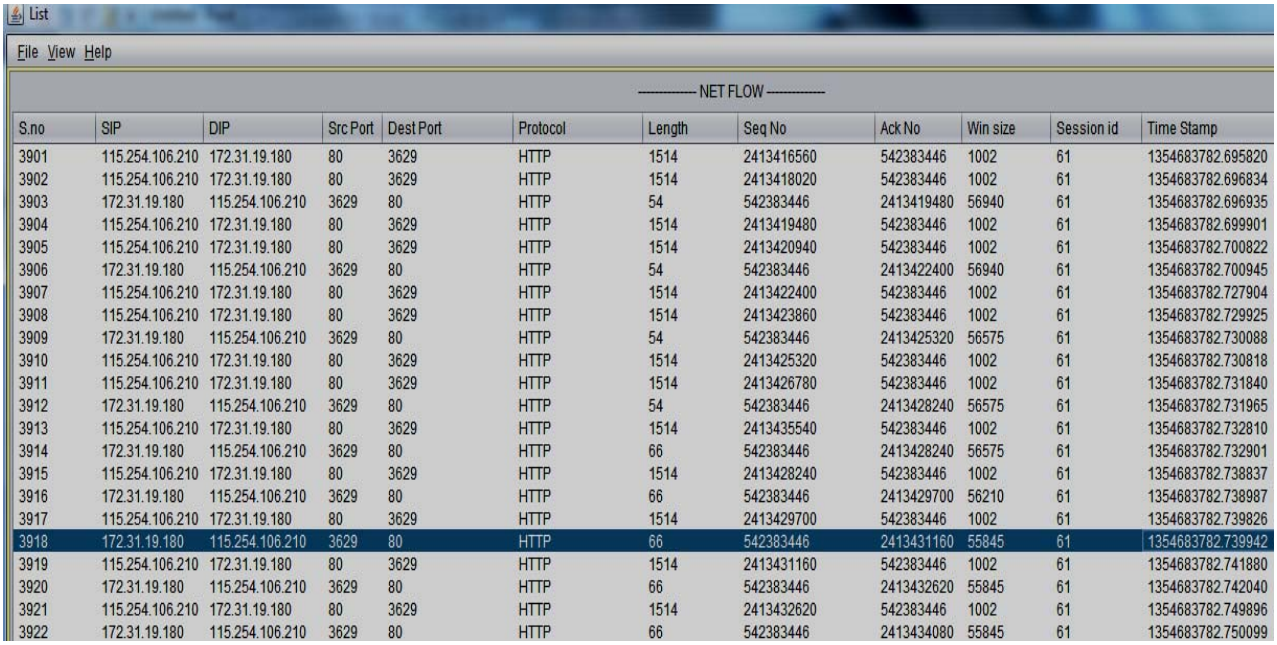
set token[i][8] = read from character[i][q+1] till character[i][s]!=' '

// acknowledgement no is read and stored into token array

EndWhile

Endif

Similarly others fields are read and stored as part of tokenization. Figure 4.3 is shows outcome of tokenization module. Tokenized data is further parsed in two types of formats, *XML for real time* analysis and spreadsheet format for forensic analysis.



List

File View Help

---------- NET FLOW ----------

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
3901	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413416560	542383446	1002	61	1354683782.695820
3902	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413418020	542383446	1002	61	1354683782.696834
3903	172.31.19.180	115.254.106.210	3629	80	HTTP	54	542383446	2413419480	56940	61	1354683782.696935
3904	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413419480	542383446	1002	61	1354683782.699901
3905	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413420940	542383446	1002	61	1354683782.700822
3906	172.31.19.180	115.254.106.210	3629	80	HTTP	54	542383446	2413422400	56940	61	1354683782.700945
3907	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413422400	542383446	1002	61	1354683782.727904
3908	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413423860	542383446	1002	61	1354683782.729925
3909	172.31.19.180	115.254.106.210	3629	80	HTTP	54	542383446	2413425320	56575	61	1354683782.730088
3910	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413425320	542383446	1002	61	1354683782.730818
3911	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413426780	542383446	1002	61	1354683782.731840
3912	172.31.19.180	115.254.106.210	3629	80	HTTP	54	542383446	2413428240	56575	61	1354683782.731965
3913	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413435540	542383446	1002	61	1354683782.732810
3914	172.31.19.180	115.254.106.210	3629	80	HTTP	66	542383446	2413428240	56575	61	1354683782.732901
3915	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413428240	542383446	1002	61	1354683782.738837
3916	172.31.19.180	115.254.106.210	3629	80	HTTP	66	542383446	2413429700	56210	61	1354683782.738987
3917	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413429700	542383446	1002	61	1354683782.739826
3918	172.31.19.180	115.254.106.210	3629	80	HTTP	66	542383446	2413431160	55845	61	1354683782.739942
3919	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413431160	542383446	1002	61	1354683782.741880
3920	172.31.19.180	115.254.106.210	3629	80	HTTP	66	542383446	2413432620	55845	61	1354683782.742040
3921	115.254.106.210	172.31.19.180	80	3629	HTTP	1514	2413432620	542383446	1002	61	1354683782.749896
3922	172.31.19.180	115.254.106.210	3629	80	HTTP	66	542383446	2413434080	55845	61	1354683782.750099

Figure 4.3: Tokenized filtered HTTP traffic

4.1.3 Reporting Visualization module

Data mining module helps INTVS in classification, association, and linking of network traffic. This helps in monitoring network resources as per network policy.

Following algorithm classifies tokenized packet in accordance to protocols.

BEGIN

token[][] **//token[][] is a 2D array that stores the tokenized packet where token[i] represents a particular token**

// token [i][j] represents information about that token and it attributes such as

//token[i][0] represents the source machine IP address,

//token[i][1] represents destination machine IP address,

// token[i][2] represents source port,

// token[i][3] represents destination port,

// token[i][4] represents sequence number,

// token[i][5] represents acknowledgement number,

// token[i][6] represents windows size,

// token[i][7] represents session id,

// token[i][8] represents timestamp,

PORT_DESC[][] // **PORT_DESC[][] is a constant 2- dimensional array, as per IANA standards** [140]

//PORT_DESC[i] represents a port,

// PORT_DESC[i][j] represents its attributes port no, port description

classified_packets[][][]

//classified_packets[][][] is a 3D array, which stores classified packets,

//classified_packets[i], represents a particular protocol (HTTP, FTP, SMTP, OTHERS),

//classified_packets[i][j] represents a token of type classified_packets[i], and

//classified_packets[i][j][k] stores the details of packet classified_packets[i][j]

//others[][] is as 2D array to store unclassified token

convert PORT_DESC[][] into a hashmap PORT_HASH (Key - Value pair)

For i =0, till i less than token.length, increment i // **for traversing each received packet**

Set desc_temp := PORT_HASH.get(token[i][2])

// get the type of protocol on basis of source port

If desc_temp not equals "others" set token[i][9] := desc_temp

Set token[i][9] := PORT_HASH.get(token[i][2]) // **sets the type of protocol if it is classified**

Set classified_packets[token[i][9]][k] := token[i]

// adds the classified packet in accordance to its protocol

Else

token[i][9] := 'OTHERS' // **sets the type of protocol to others**

Set others[k] := token[i]; // **add the token to others data structure**

Endif

EndFor

Data mining algorithm for traffic classification in INTVS is implemented and the results are shown in Figure 4.4 through Grid view visualization scheme. Entire network traffic is classified as follows: Netflow, Internet, Intranet, TCP, UDP and HTTP traffic grids along with different color backgrounds.

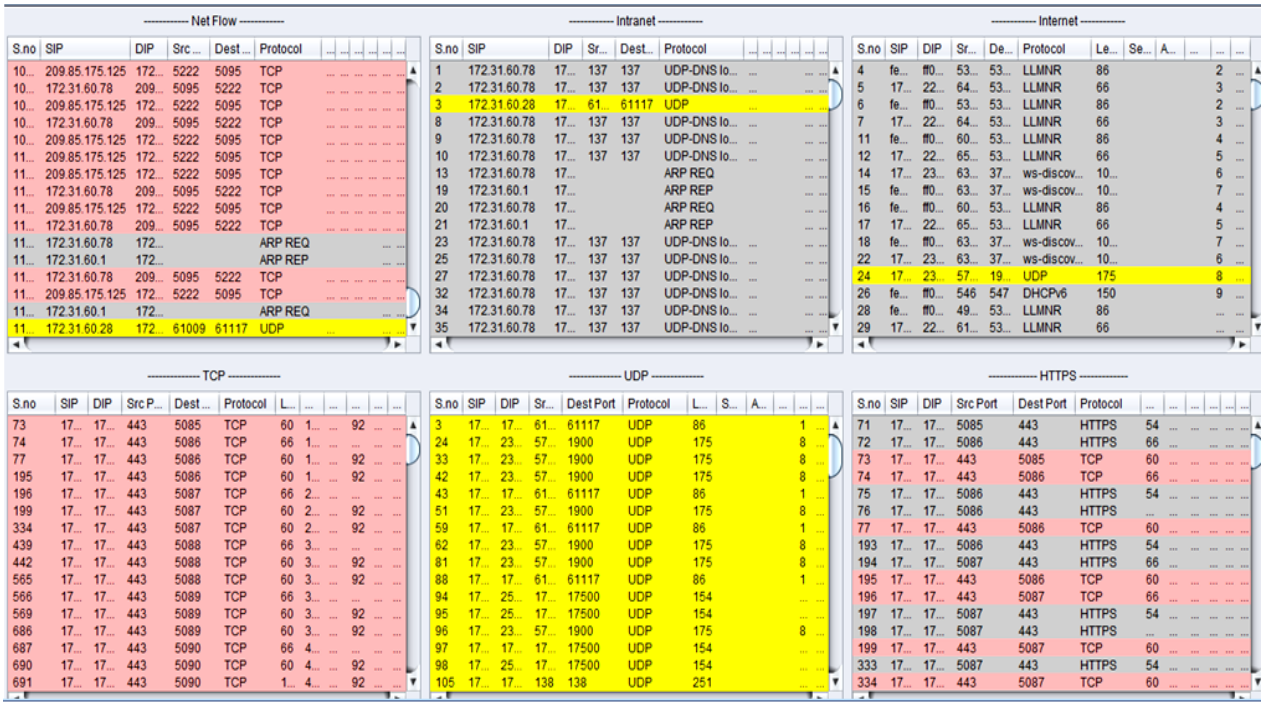


Figure 4.4: Grid View

Similarly, INTVS facilitates customization of Grid view component instead of using Netflow, Intranet, Internet, TCP, UDP and HTTP traffic view; wherein user can select Grid view of Netflow, Intranet, Internet, Email, FTP and HTTP traffic view as shown in Figure 4.5.

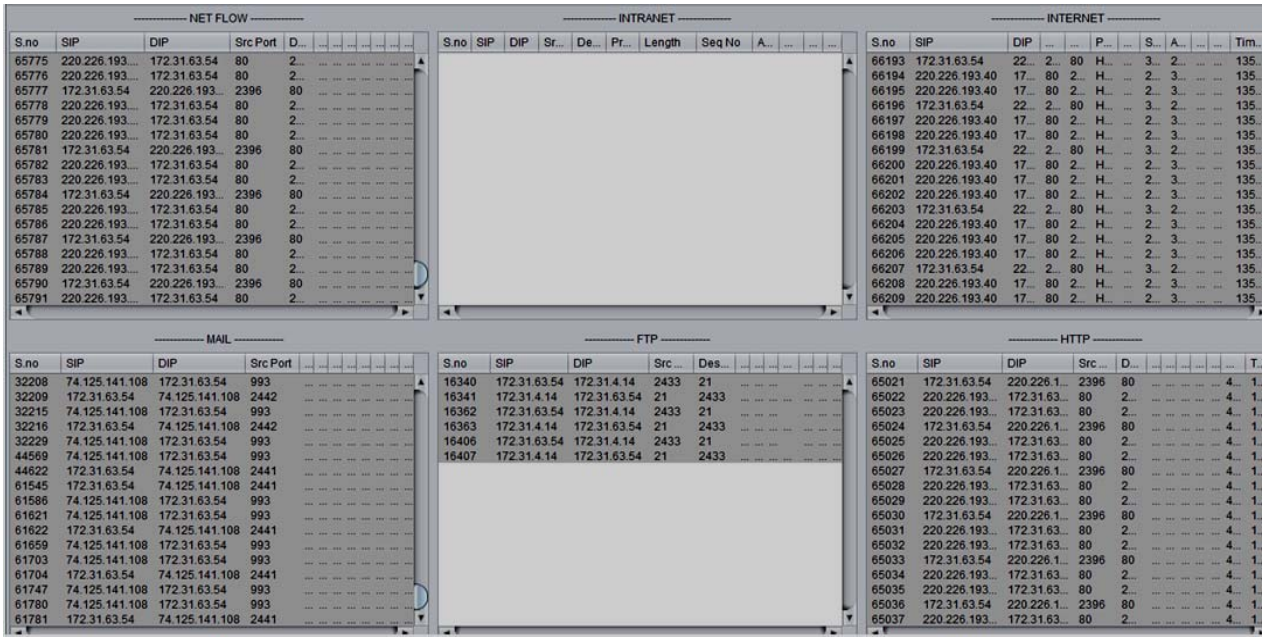


Figure 4.5 Grid View customized by user

Figure 4.6: View of HTTP/HTTPS Traffic

In addition INTVS also provides extended facility of selecting and filtering a particular traffic view as shown in Figure 4.6, where HTTP/HTTPS traffic view is shown. User can also view FTP, Email and other application layer traffic in same fashion.

INTVS also maintains listmap visualization scheme for network traffic and continuously updates it in real time. Simple mouse drag option is very useful, when hovered over listmap, it shows the traffic statistics of a particular machine, like uploading/ downloading data size, TCP data, UDP data and application layer data. Listmap can be easily viewed as shown in Figure 4.6 left-bottom section where active machines are detected and listed in view. Figure 4.7 shows clear view of listmap visualization scheme, which is giving entire view of the network on a single screen, while mouse hover brings statistics related to a particular machine.

4.1.4 Listmap module

// this algorithm will generate a Listmap based on the data retrieved from xml

Set data to obj.getLogPacketData **// it will fetch data from xml**

For i is 0, i is less than data.size, i increments by 1 **// for traversing each node**

set temp to Double.parseDouble with ((String))

// converts transferred data from string to double for arithmetic operations

If temp is greater than check **// checks if transferred data is more than threshold limit**

avgUpData += temp **// adds data into upload data variable**

++avgUpCount **// increments counter to maintain no. of machines**

EndIf

set temp to Double.parseDouble with ((String))

// **converts transferred data from string to double for arithmetic operations for download data**

If temp is greater than check

// **checks if transferred data is more than threshold limit for download data**

avgDownData += temp // **adds data into download data variable**

++avgDownCount // **increments counter to maintain no. of machines in download data**

EndIf

set temp to 0

EndFor

avgUpData/=avgUpCount // **computes average upload data**

avgDownData/=avgDownCount // **computes average download data**

For i is 0, i is less than data.size, i increments by 1 // **for traversing each node in captured data**

set i of button to new JButton with ((String))

button[i].setText() // **set the text of button to IP**

If Double.parseDouble with ((String)) is greater than avgUpData or Double.parseDouble with ((String)) is greater than avgDownData

set color of button to red // **sets the color to red**

Else

set color of button to green // **sets the color to green**

EndIf

call method add with position i in button

EndFor // **End listmap**

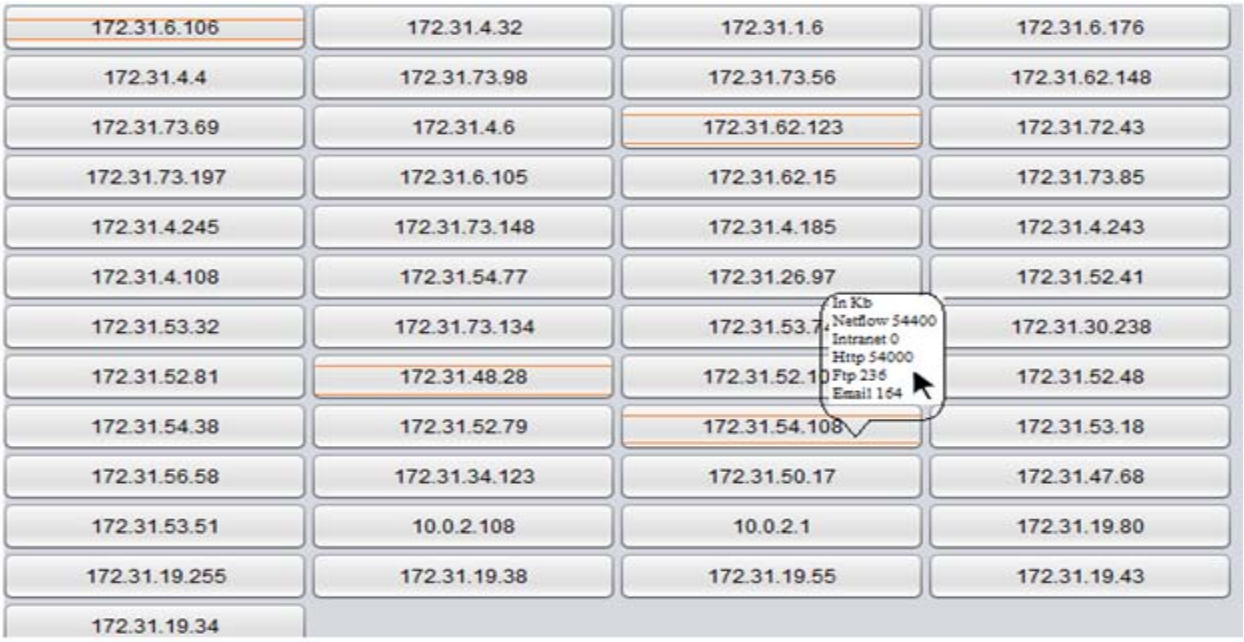


Figure 4.7: Listmap view of network traffic

4.1.5 Line Graph module

Real time series graph is another way to present the netflow of network. This helps the user to easily understand the data patterns in the network traffic . In Figure 4.6, right bottom as well as in Figure 4.8, yellow, purple, red, blue, green and torquies line curves are representing HTTP, Email, TCP, UDP, FTP and other traffic respectively.

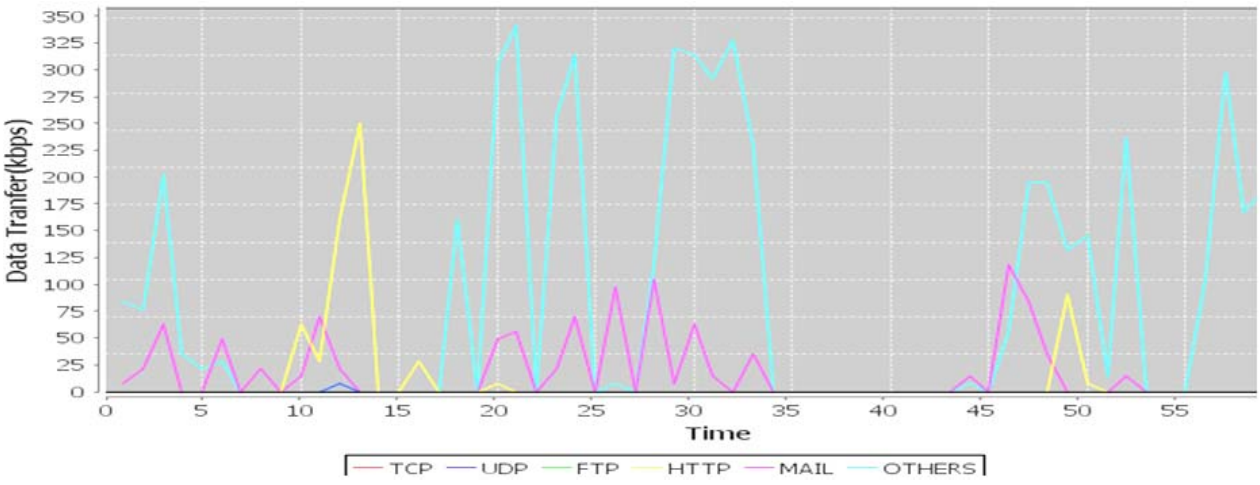


Figure 4.8: Real time line graph of network traffic

4.1.6 Abnormal activity detection module

This module helps in detecting abnormal activity of a machine from the network. University laid down network policy is chosen as basis for this detection.

BEGIN

data[][][][][]

// data[][][][][] is a 5- dimensional array that stores the parsed data of the netflow where,

//data[i] represents a subnet,

//data[i][j] represents a machine,

//data[i][j][k] represents an application layer protocol,

//data[i][j][k][l] represents a day

//data[i][j][k][0….6] represents Monday to Sunday and data[i][j][k][7] stores the average of the week

//data[i][j][k][l][m] stores the total upload(0) and download(1) data in lth day of kth protocol of jth machine of ith subnet.

subnet_data[]

// subnet_data[] is a 1 dimensional array to store total data transferred in each subnet

machine[]

// machine is a 1 dimensional array to store total data transferred of machine[i]

day[]

// day[] is a 1 dimensional array to store total data transferred of the CAN day wise

protocol[]

// protocol[] is a 1 dimensional array that will store total data transferred as per each protocol

machine_protocol[][]

// machine_protocol[][] is a 2D array to store total mined data transferred of a specific protocol

no_machine **// counter to stores count of current machines**

A[][]

// A[][] represents set of IP addresses and ports opened for legitimate user, from inside of the network

B[][]

// B[][] represents set of IP addresses and ports opened for authorized user from extranet

N[][]

// Array N is representing Netflow of traffic, N[i] is netflow packet number, N[j] represents the attributes of packet like SIP, SRC_port, Dst_Port, DIP, Session_id, ack_num, win_size, pkt_length, timestamp

Malicious traffic[][]

// Malicious traffic[][] represents an array, which keeps records of malicious packet traffic

If (A[]∪B[])=N[] // **Packet is clean**

Else Book as Malicious Packet[]

R[] // **R[] represents total resources**

R_i [] // **R_i [] represents resources for legitimate user from inside**

R_o[] // **R_o[] represents resources for legitimate user from outside**

R_{ri} [] // **R_{ri} [] represent restricted resource to inside user**

R_{ro}[] // **R_{ro} represents Restricted Resource to outside user**

R_{ri_policy} [] // **R_{ri_policy} [] represents policy for Restricted Resource to inside user= R_{ri_policy}**

R_{ro_policy} [] // **R_{ro_policy} [] represents policy for Restricted Resource to Outside user= R_{ro_policy}**

Policy for Total Restricted Resource is represented by RR= R_{ri_policy} []∪ R_{ro_policy} []

Total available resources represented by R= (R_i[]∪R_o[]) ∪ (R_{ri}[] ∪ R_{ro}[])

// **Defining Sets for various user**

Total Eligible users represented by U_rt[]

Legitimate users from inside = U_{ri}[]

Super users from inside = U_rS_i[]

Legitimate User from outside for extranet resources = U_{ro}[]

Guest user from public domain for web services only = U_r^g[]

Eligible users = U_rt[]

U_rt[] = (U_{ri}[]∪ U_rS_i[]) ∪ (U_{ro}[] ∪ U_r^g[])

Total visitors of network = U^e[]

Un-authorized visitor denoted by $U_{Un-Auth}$ [] = U^e[] ∩ U_rt[]

Relationship between available resources and users R ∈ U^e[]

Authorized but Malicious user denoted by U_{auth_mal} [] = (U_{ri}[] R_{ri_policy} []) ∪ (U_r o[] R_{ro_policy} [])

Total Malicious user $U_M[] = U_{Un-Auth}[] \cup U_{auth\_mal}[]$

C[] // **C[] is representing set of color**

$C_{Mal}[]$ // **C_{Mal} []representing set of color for malicious user**

$C_{Auth}[]$ // **C_{Auth} []set of color for Non-Malicious user**

$C_{Un-Auth}[]$ // **$C_{Un-Auth}$ [] set of color for Un-authorized user**

$C_{UE}[]$ represents colors for each visitor of the network

$C[] = C_{Mal}[] \cup C_{Un-Auth}[]$

If user $U^e i[] \in U_M[]$ then

$C_{UE} i[]$ =Get $C_{Mal}[]$ for $U_M i[]$

Else

$C_{UE} i[]$ =Get $C_{Auth}[]$ for $U^e i[]$

$M_{vis}[]$ will now plot $U^e i[]$ using $C_{UE} i[]$

EndIf

END

Based on these network set theory algorithms, infringement to policy rules is averted. Once unauthorized visitor tries to join the network, an alert will be generated as defined in the network security policy. As a result, packets and alert details with time, IP address, port are logged and can be flashed on dashboard as shown in Figure 4.9 as well as shown through red dots in platter view Figure 5.9 (Chapter 5). As per network policy at Thapar University, Patiala, only VLAN 172.31.1.0, 172.31.4.0 are allowed to have access of a server farm.

Packets	Alerts				
Sr. No.	**Time**	**IP address**	**Port**	**From**	**Alert Type**
1	28/10/2014 14:26	172.31.4.14	3306	172.31.72.16	Unauthorized DB access
2	28/10/2014 14:40	172.31.4.16	25	172.31.72.17	Unauthorized email access
3	28/10/2014 14:46	172.31.4.16	25	172.31.72.17	Unauthorized email access
4	28/10/2014 15:20	172.31.4.20	3306	172.31.72.16	Unauthorized DB access

Figure 4.9: Alert Analysis

4.1.7 Implementation of Platter View

Following algorithm generates a platter view, each platter denoting a subnet and within each platter it generates rings representing a machine within that subnet

BEGIN

data[][][] **// data[][][] is a 3D array in which, data[i] represents a sub-net, data[i]j] represents a machine n in data[i] subnet, data[i][j][0] represents IP address, data[i][j][1] represents total uploaded data from machine[i][j], data[i][j][2] represents total downloaded data from machine[i][j], total_network_data is used to hold total data transferred in the network**

subnet_data[]

//subnet_data[] is 1D array in which data[i] stores the total data transferred in subnet data[i]

screen[]

// **screen[] holds the size of screen, in which screen[0] is height and screen[1] is width**

origin[] **// origin[] holds the center of screen**

```
sortNetwork(data, total_network_data, subnet_data)  // this function will sort data list subnet wise
step_size:= (screen[0]-100)/(2 * length(data)) // to calculate the distance between each platter
i:=0 // initialize counter for traversing each subnet
For i<length(data) do
        outer_circles[radius]:= step_size*(i+1) // calculates the radius of platter
        j:=0 // initialize counter for traversing each machine
        For j<length(data[i]) do
                inner_circles[center_x] := center[x] + (outer_circle[i] * cos((ip.split('.')[3]*1.4)))
                // calculates the x coordinate for the center point of the inner circles, with
                // respect to their IP
                inner_circles[center_y] := center[y] + (outer_circle[i] * cos((ip.split('.')[3]*1.4)))
                // calculates the y coordinate for the center point of the inner circles
                inner_circles[radius] := step_size/2
                 // calculates the radius for the center point of the inner circles
        j++
        EndFor
i++
EndFor
drawCircles (outer_circles, inner_circles, total_network_data, subnet_data, data)
```

// this function will plot all the circles according to the calculated coordinates

End

4.1.7.1 Sorting and swapping of circles:

Following algorithm sorts data in order of subnet and the transferred data itself

BEGIN

SET i:=0 **// initialize the counter for calculating total and subnet wise data**

For i<length(data) do

j:= 0 **// initialize the counter for traversing subnet data**

For j<length(data[i]) do

total_network_data:= total_network_data + data[i][j][1] + data[i][j][2] **// add upload, as well as, download data of current machine to total_network_data**

subnet_data[i]: = subnet_data[i] + data[i][j][1] + data[i][j][2] **// add upload, as well as, download data of current machine to current subnet_data**

j++

EndFor

i++

EndFor

i:=length(data) **// initialize counter for sorting the list**

For i>0 do

SET j=1 **// initialize the counter for comparing current element with entire list**

```
        For j<i do
            If (subnet_data[j]<subnet_data[j-1]) then
            swap(data[j],data[j-1])
            // swap the positions of subnet if data transferred by first is less than
            // second
            Endif
        j++
    EndFor
i++
EndFor
End
```

4.1.7.2 Screen size responsive visualization:

Following algorithm plots the calculated coordinates onto the screen

```
BEGIN
SET i:=0 // counter for traversing each platter
For i<length(outer_circles) do
    setcolor(abs(118-(3*i),75,146)) // calculates unique color code for each platter
    drawcircle(outer_circles[i]) // plots the platter with calculated color code
    i++
EndFor
```

SET counter:=0 // **counter for plotting each inner circle**

SET prev_count:=0 // **counter for maintaining previously plotted circle**

SET i:=0 // **counter for traversing entire list**

For i<length(inner_circles) do

If(i<length(data[counter])) then

temp:=data[counter][i-prev_count][1]+data[counter][i-prev_count][2]

// **calculates the data transferred by current machine**

Else

counter++

prev_count=i

temp:= data[counter][i- prev_count][1]+data[counter][i-prev_count][2]

// **calculates the data transferred by current machine**

Endif

If ((temp > total_network_data/length(data)) then

setcolor(red) // **sets red color if data transferred by current machine is greater than average transfer data of each subnet in concurrence with network policy**

Else If(temp > avg(subnet_data)) then

setcolor(orange**) // sets orange color if data transferred by current machine is greater than average transfer data of each machine in current subnet**

Else setcolor(green)

// sets green color is color if data transferred by current machine is less than average transfer data of each machine in current subnet and average data transfer of entire subnet

drawcircle(inner_circle[i]) // plots the circle with selected color

EndIf

EndFor

The outcome after implementing these algorithms gives us global view of campus area network on a single screen with the help of platter view visualization as shown in the Figure 4.10, Figure 4.11 and Figure 4.12. In Platter view analysis, circles are used to represent networks and rings are used to represent the machines. The smallest inner most circle in Platter view is representing the smallest network and outer most biggest circle is representing the biggest network of a campus.

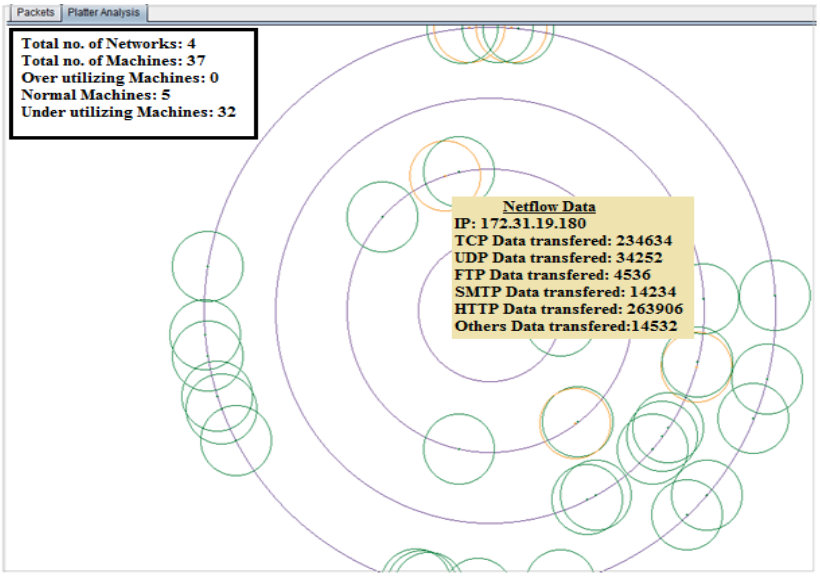


Figure 4.10: Global view of CAN through Platter View

Number of networks are detected as well their respective machines are detected from captured data. These machines in a particular circle, are represented by dots instead of rings as shown in the Figure 4.10. The Platter view analysis is capable to display as large as class B networks. Mouse hovering over the circle shows IP address of the network as well as number of live machines in that network as shown in Figure 4.11 and when mouse is hovered over a dot, it shows detail load (TCP, UDP, HTTP, FTP, downloading, uploading) on that machine as shown in Figure 4.12.

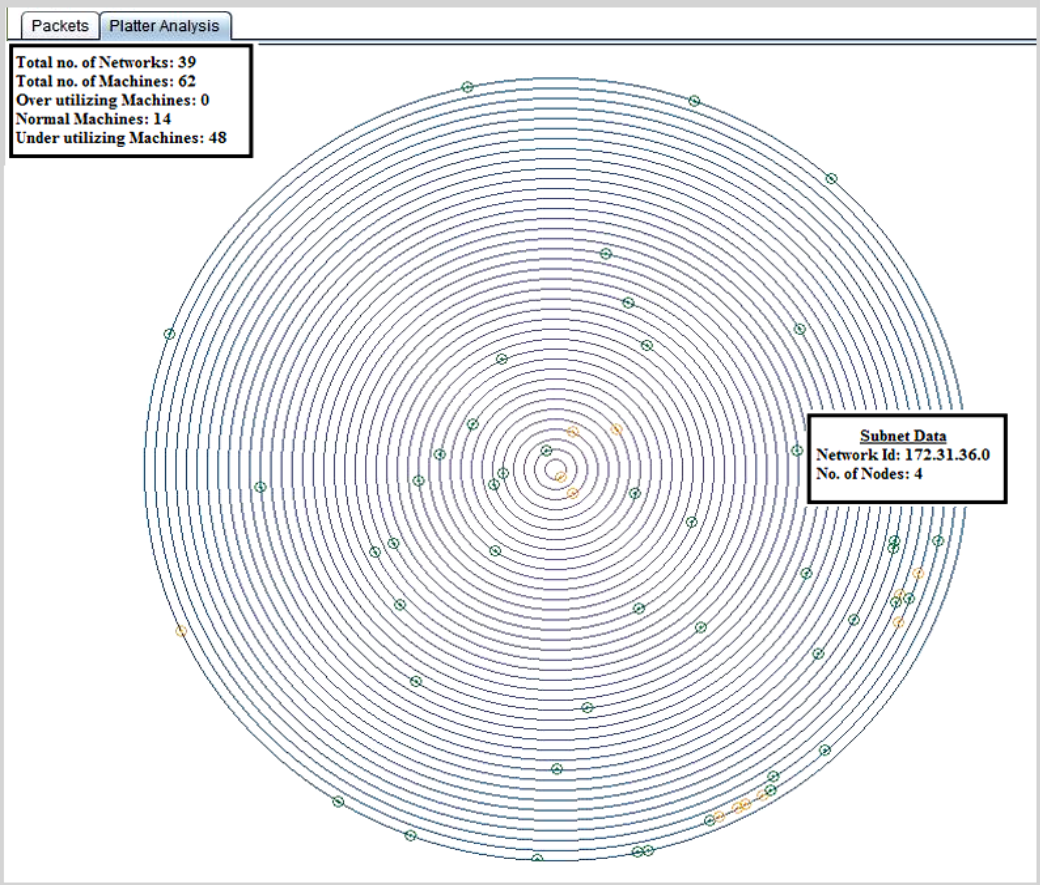


Figure 4.11: Global View of CAN with Subnet Data Details

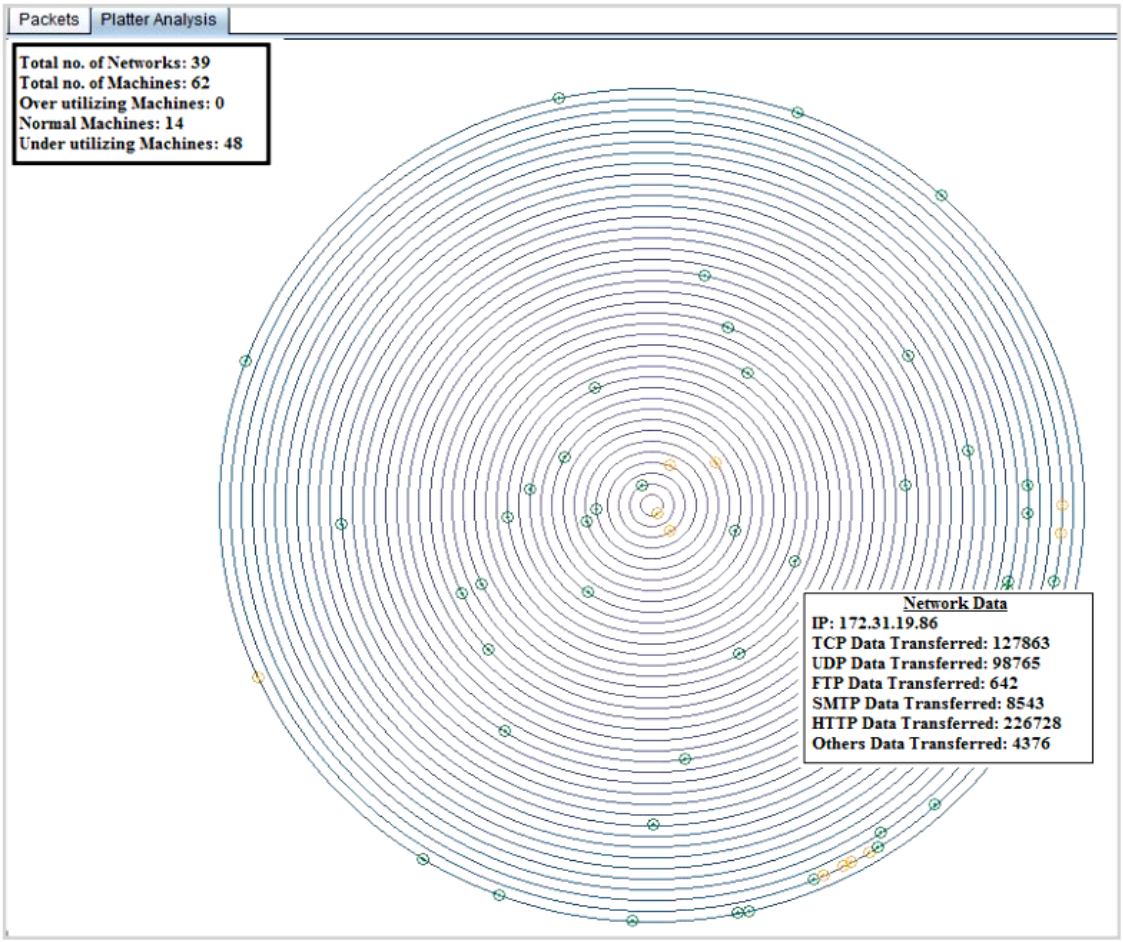


Figure 4.12: Global View of CAN with Particular Machine Data Details

Red dot in a circle depicts that the data transfer from/to this machine is above average of the network bandwidth and blue dot depicts a normal bandwidth usage by that machine in a particular network. On similar lines color of circle also changes if the bandwidth consumed by the network is more and/or less than the average bandwidth allocated by the administrator in concurrence with the network policy.

4.1.8 File exporter module

This module exports data in xml format and excel formats.

```
try
initialize filename to "./Data/export/deviceData-" plus new Date // path for storing file with data
```

create new HSSFWorkbook

initialize sheet to hwb.createSheet with "new sheet" // **initialize java library that supports excel worksheet creation**

initialize rowhead to sheet.createRow // **creates a new row for header**

For i is 0, i is less than headers.length, i increments by 1 // **for traversing each header name**

call method rowhead.createCell

EndFor

For i is 0, i is less than data.size, i increments by 1 // **for traversing each row in data**

initialize row to sheet.createRow with (short) // **creates a new row for detail**

For j is 0, j is less than headers.length, j increments by 1 // **for traversing each column in data**

call method row.createCell with (short) // **for writing data into the row cell**

EndFor

EndFor

create new FileOutputStream // **java library object that supports file handling**

call method hwb.write with fileOut // **java library for writing the created workbook into an excel file**

call method fileOut.close // **closing all opened files for releasing locks**

call method JOptionPane.showMessageDialog

catch Exception ex

call method System.out.println with ex

call method JOptionPane.showMessageDialog

EndTry

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
460	192.168.30.203	124.124.252.43	9230	80	HTTP	585	199051048	280312055	4380	55	1347371541.662668
461	124.124.252.43	192.168.30.203	80	9185	HTTP	54	177512474	255342449	705	5	1347371541.662923
462	192.168.30.203	124.124.252.43	9242	80	HTTP	66	2602065606		8192	82	1347371541.663297
463	124.124.252.43	192.168.30.203	80	9188	HTTP	54	181039109	565375161	311	49	1347371541.664477
464	192.168.30.203	124.124.252.43	9184	80	HTTP	949	293920680	175435639	4380	46	1347371541.690392
465	124.124.252.43	192.168.30.203	80	9229	HTTP	54	282204887	198650062	130	54	1347371541.699505
466	124.124.252.43	192.168.30.203	80	9228	HTTP	691	280134715	831660190	325	52	1347371541.700554
467	124.124.252.43	192.168.30.203	80	9242	HTTP	66	285260993	260206560	5840	82	1347371541.705267
468	192.168.30.203	124.124.252.43	9242	80	HTTP	54	260206560	285260993	4380	82	1347371541.705348
469	124.124.252.43	192.168.30.203	80	9230	HTTP	54	280312055	199051579	125	55	1347371541.705950
470	124.124.252.43	192.168.30.203	80	9184	HTTP	54	175435639	293920769	618	46	1347371541.708764
471	124.124.252.43	192.168.30.203	80	9185	HTTP	1514	177512474	255342449	705	5	1347371541.709990
472	124.124.252.43	192.168.30.203	80	9185	HTTP	400	177512620	255342449	705	5	1347371541.710997
473	192.168.30.203	124.124.252.43	9185	80	HTTP	54	255342449	177512654	4380	5	1347371541.711048
474	124.124.252.43	192.168.30.203	80	9188	HTTP	248	181039109	565375161	311	49	1347371541.711173
475	124.124.252.43	192.168.30.203	80	9187	HTTP	687	176870078	183443972	734	4	1347371541.722376
476	124.124.252.43	192.168.30.203	80	9186	HTTP	681	176781872	490036175	764	19	1347371541.724569
477	192.168.30.203	124.124.252.43	9228	80	HTTP	895	831660190	280134779	4380	52	1347371541.732719
478	124.124.252.43	192.168.30.203	80	9229	HTTP	1514	282204887	198650062	130	54	1347371541.744526
479	192.168.30.203	124.124.252.43	9185	80	HTTP	896	255342449	177512654	4380	5	1347371541.753545
480	124.124.252.43	192.168.30.203	80	9229	HTTP	1514	282205033	198650062	130	54	1347371541.764076
481	192.168.30.203	124.124.252.43	9229	80	HTTP	54	198650062	282205179	4380	54	1347371541.764135
482	124.124.252.43	192.168.30.203	80	9229	HTTP	1174	282205179	198650062	130	54	1347371541.771836
483	124.124.252.43	192.168.30.203	80	9230	HTTP	1514	280312055	199051579	125	55	1347371541.777519

Figure 4.13: Parsed file generated by Exporter Module

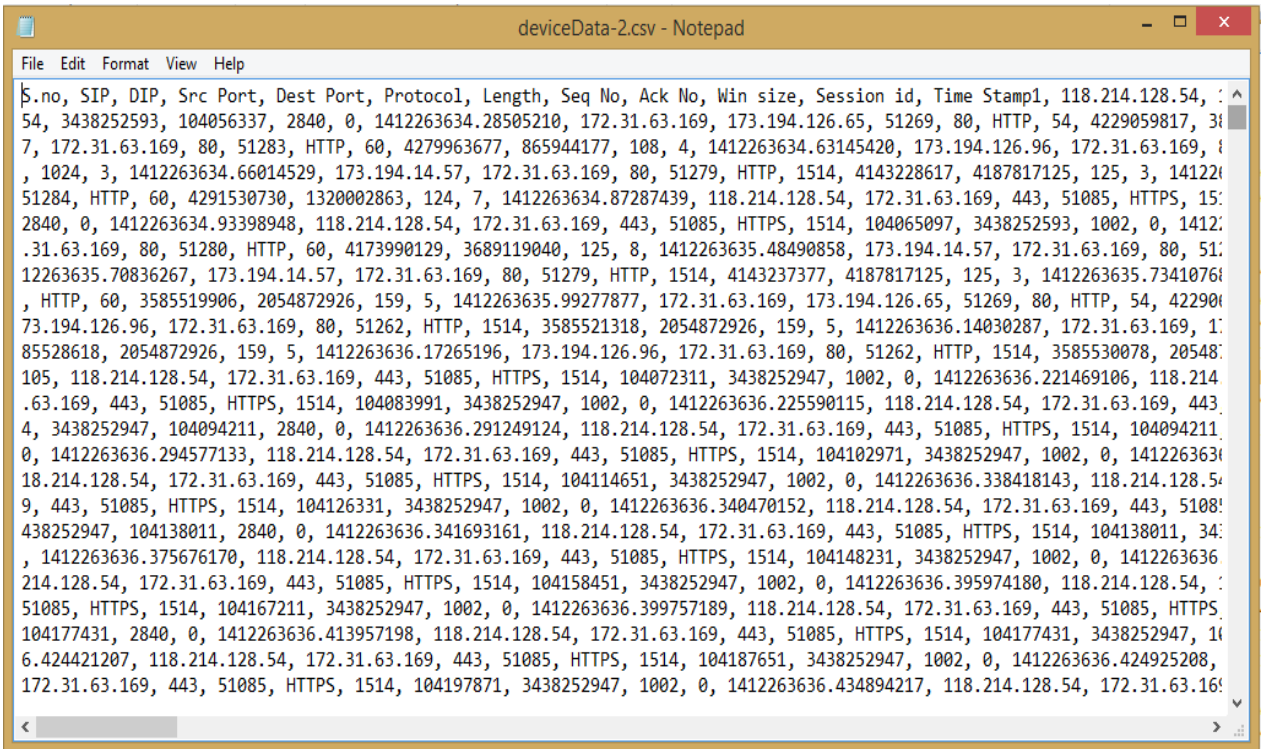


Figure 4.14: CSV file generated by Exported Module

4.1.9 CSV exporter module

This module exports data to Comma Separated Values files.

initialize dataNew to "" // **a temporary string variable that will be written into the file**

initialize len to headers.length minus 1 // **len will hold the no of headers**

For j is 0, j is less than len, j increments by 1 // **for traversing each header name**

dataNew+= headers[j]+", " // **will write each of the headers**

EndFor

dataNew+= headers[len]+\\n // **will add next line delimiter**

For i is 0, i is less than data.size, i increments by 1 // **for traversing each row in data**

dataNew+= (i+1) + ", " // **for adding serial no**

For j is 1, j is less than len, j increments by 1 // **for traversing each column in data**

dataNew+= (((String[])data.get(i))[j])+", "

// **for writing data into the string followed by a comma**

EndFor

dataNew+= (((String[])data.get(i))[len])+\\n // **for adding next line delimiter**

EndFor

try

initialize filename to "./Data/export/deviceData-" plus new Date // **path for storing file with data**

create new FileWriter // **java library object that supports file handling**

create new BufferedWriter // **java library object that supports buffering of string while writing from main memory to secondary memory**

call method out.write with dataNew // **java library object that supports file writing**

call method out.close // **closing all opened files for releasing locks**

EndTry

4.1.10 Two dimensional analysis module

Campus area network traffic analysis is achieved through three level drill down facility based on data mining. Implemented results are shown in Figure 4.15 & Figure 4.16.

BEGIN

data[][][][][][] // **data[][][][][][] is a 5- dimensional array that stores the parsed data of the netflow where, data[i] represents a subnet, data[i][j] represents a machine, data[i][j][k] represents an application layer protocol, data[i][j][k][l] represents a day, data[i][j][k][0….6] represents Monday to Sunday and data[i][j][k][7] stores the average of the week**

//data[i][j][k][l][m] stores the total upload(0) and download(1) data in lth day of kth protocol of jth machine of ith subnet.

subnet_data[] // **subnet_data[] is a 1 dimensional array to store total mined data transferred in each subnet**

machine[]

// machine is a 1 dimensional array to store total mined data transferred of machine[i]

day[]

//day[] is a 1 dimensional array to store total mined data transferred of the CAN day wise

protocol[] // **protocol[] is a 1 dimensional array that will store mined total data transferred of each protocol**

machine_protocol[][] **//machine_protocol[][] is a 2D array to store total mined data transferred of a specific protocol**

// no_machine is a counter that stores counter of current machines

Set no_machine = 0 **// it stores the current machine counter**

Repeat through label start: for each network data[i] **// loop for iteration of each subnet**

Repeat through label start: for each machine data[i][j]

// loop for iteration of machine within each subnet

Increment no_machine by 1 // **as machine is traversed so the counter is incremented by one**

Repeat through label start: for each protocol data[i][j][k]

// loop for iteration of each protocol within each machine

Repeat through label start: for each day data[i][j][k][l]

// loop for traversing data transfer for each day

Set subnet_data[i] += data[i][j][k][l][0] + data[i][j][k][l][1]

// add data to current subnet for subnet analysis

Set machine[no_machine] += data[i][j][k][l][0] + data[i][j][k][l][1]

// add data to current machine for machine level drill-down support within a subnet

Set day[l] += data[i][j][k][l][0] + data[i][j][k][l][1]

// add data to current day for day wise analysis

Set protocol[k] += data[i][j][k][l][0] + data[i][j][k][l][1]

// add data to current protocol for protocol wise analysis

start: set machine_protocol[no_machine][k] += data[i][j][k][l][0] + data[i][j][k][l][1]

// add data to current machine's current protocol for protocol level drill-down support within a machine

End

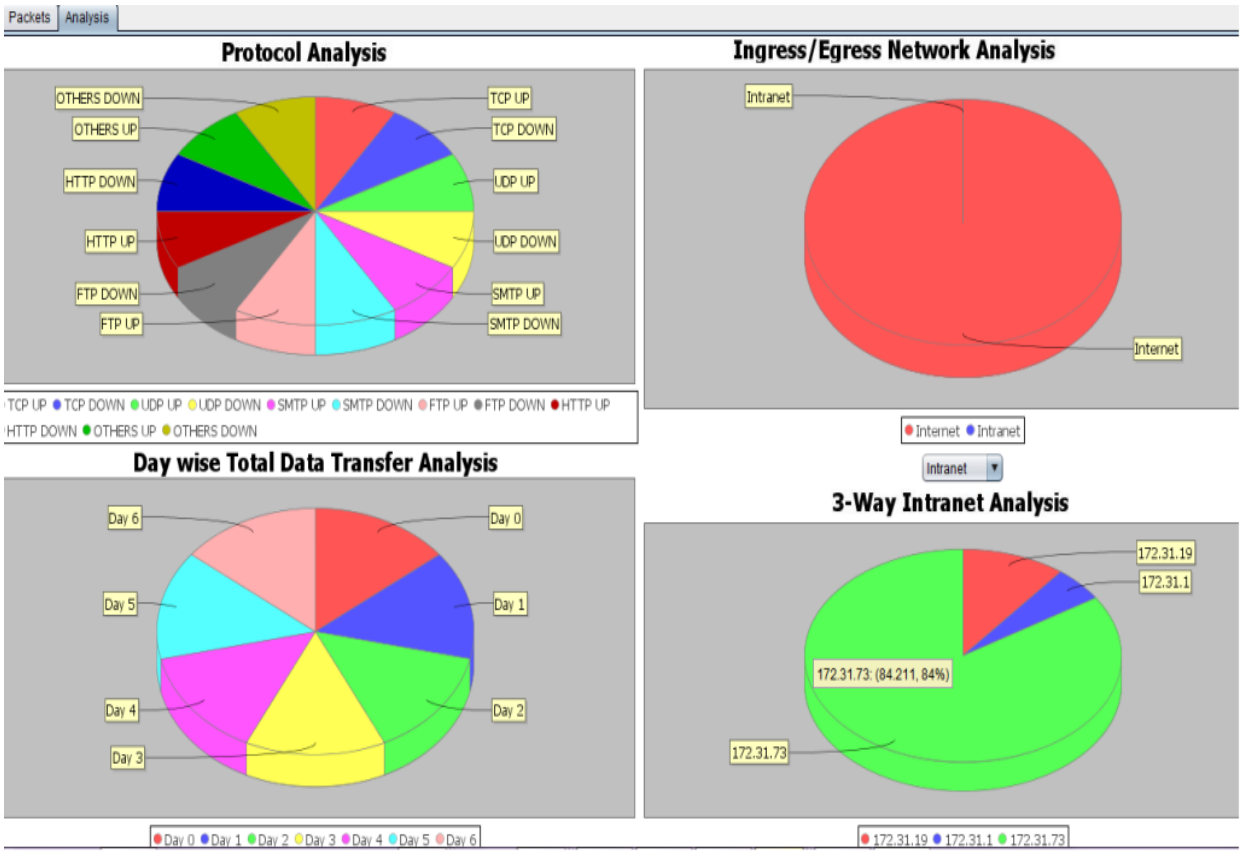


Figure 4.15: Two dimensional global view of CAN traffic day wise

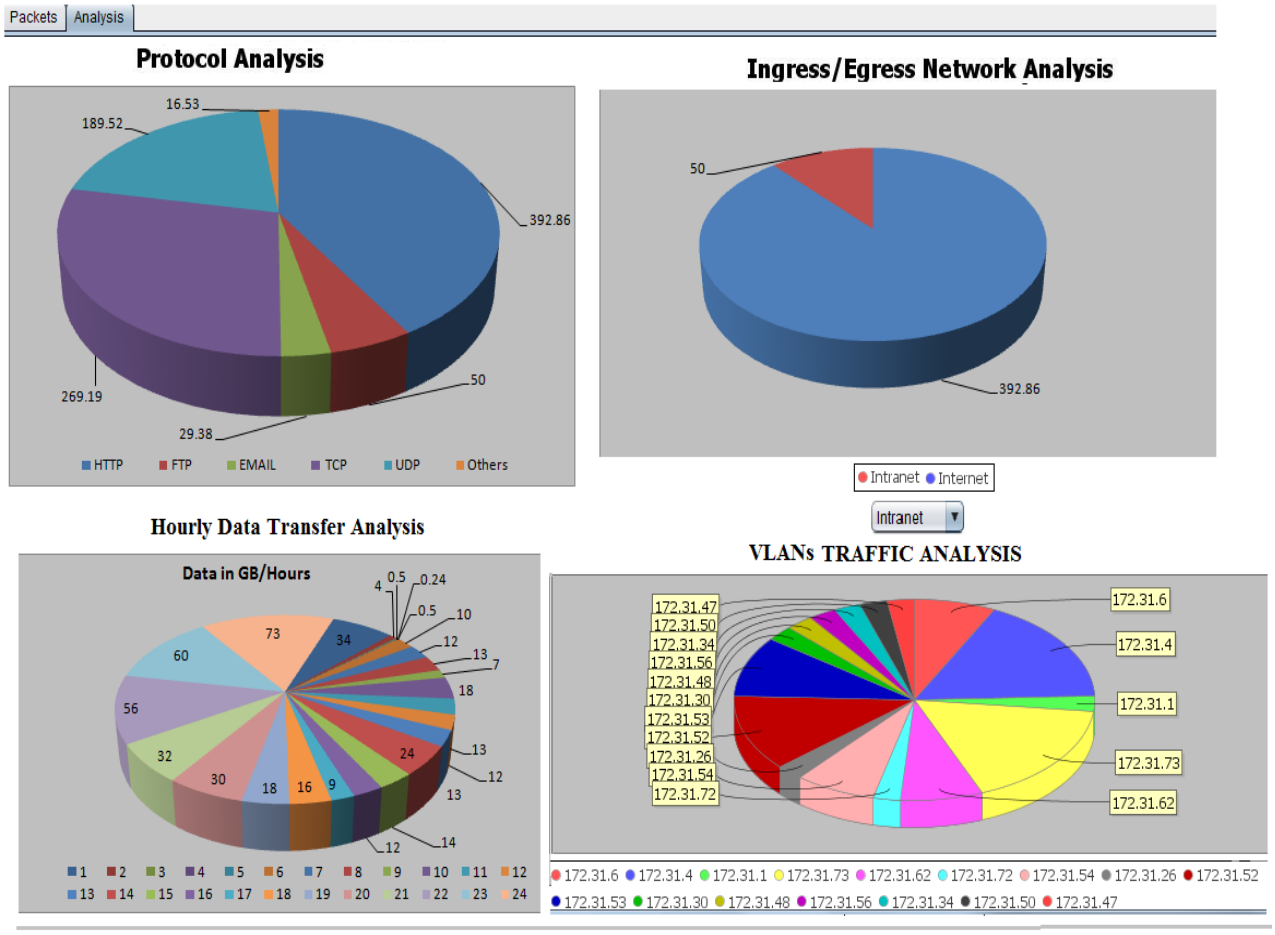


Figure 4.16: Two dimensional global view of CAN traffic hourly

4.2 INTVS in real time environment

This section deals with live distribution of INTVS. INTVS is compatible with wired/ wireless media and can work in real time mode as shown Figure 4.1. INTVS can capture network traffic data in real time mode as shown in Figure 4.2. Further INTVS can tokenize captured network traffic data in real time mode as shown in Figure 4.3. Further INTVS is using this tokenized data, parsed in XML format (shown in Figure 4.17) for further real time actions like generating alerts through malicious traffic detector module and to generate alarms in real time mode as shown in Figure 4.9.

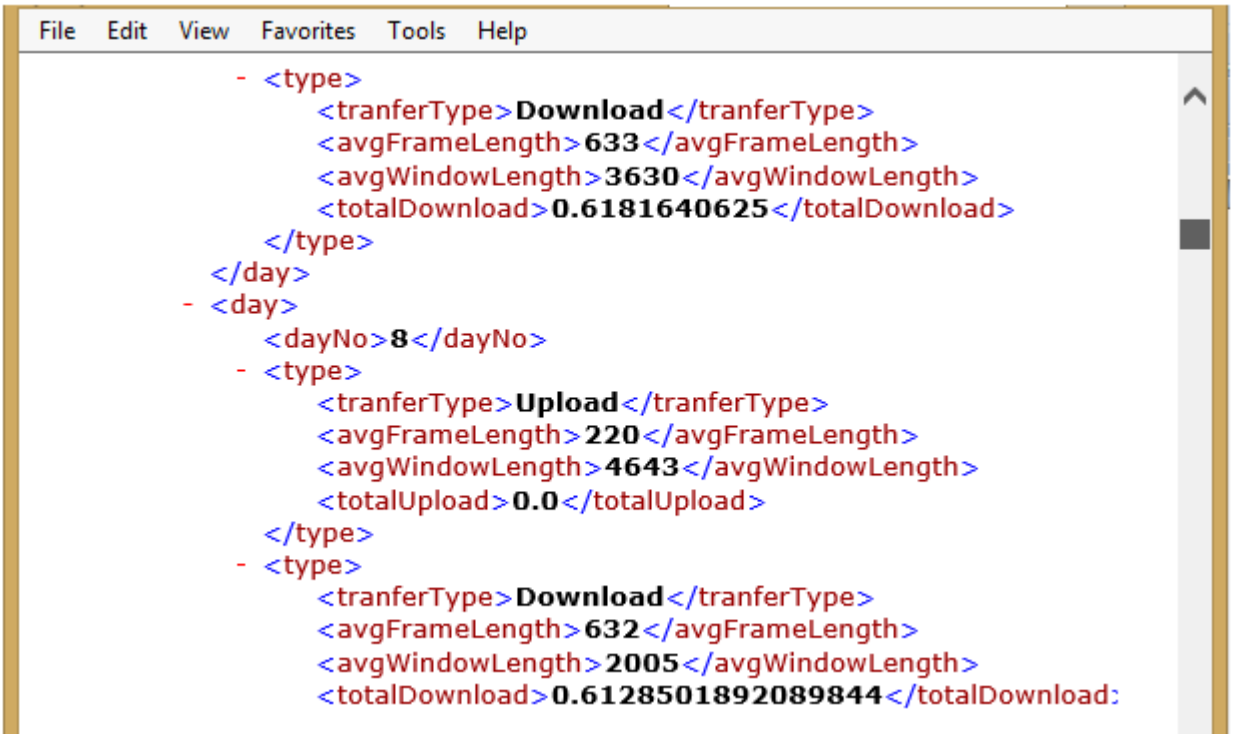


Figure 4.17: xml Parsed file

Generated XML file is used as input to visualization generating module, and visualization engine generates various type of visualizations, which are showing real time network traffic in pictorial form as shown in Figure 4.5 & 4.6 (Grid view showing classified network traffic flow).

With the help of Listmap scheme, whole network data flow statistics can be seen on single screen as shown in Figure 4.7. Mouse hovering gives network data statistics of respective machine in real time mode as shown in Figure 4.7 (for a machine, having IP address 172.31.54.108). Real time series graph gives view of different types of application layer traffic for last 60 seconds as shown in Figure 4.8, here in Figure 4.8 HTTP packets are consuming more bandwidth as compare to other application layer protocol in the network. In this figure lines showcase real time heartbeat of the network.

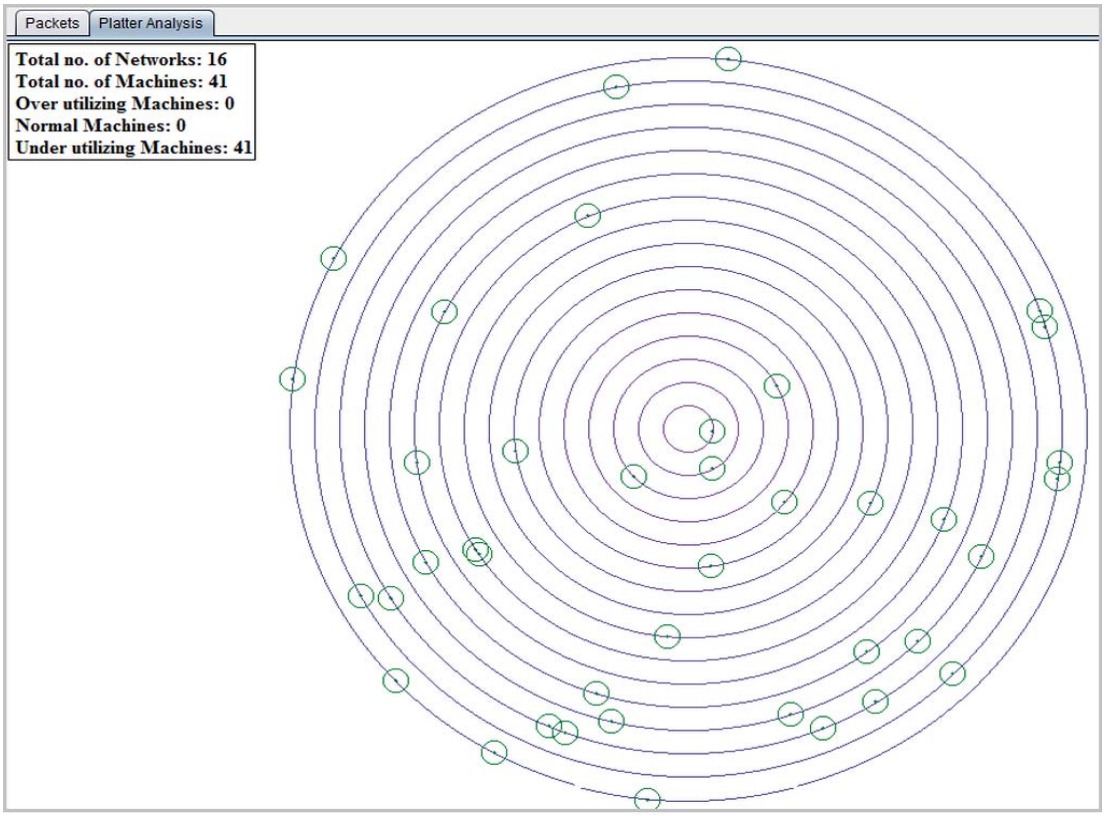


Figure 4.18: Platter view at time T1

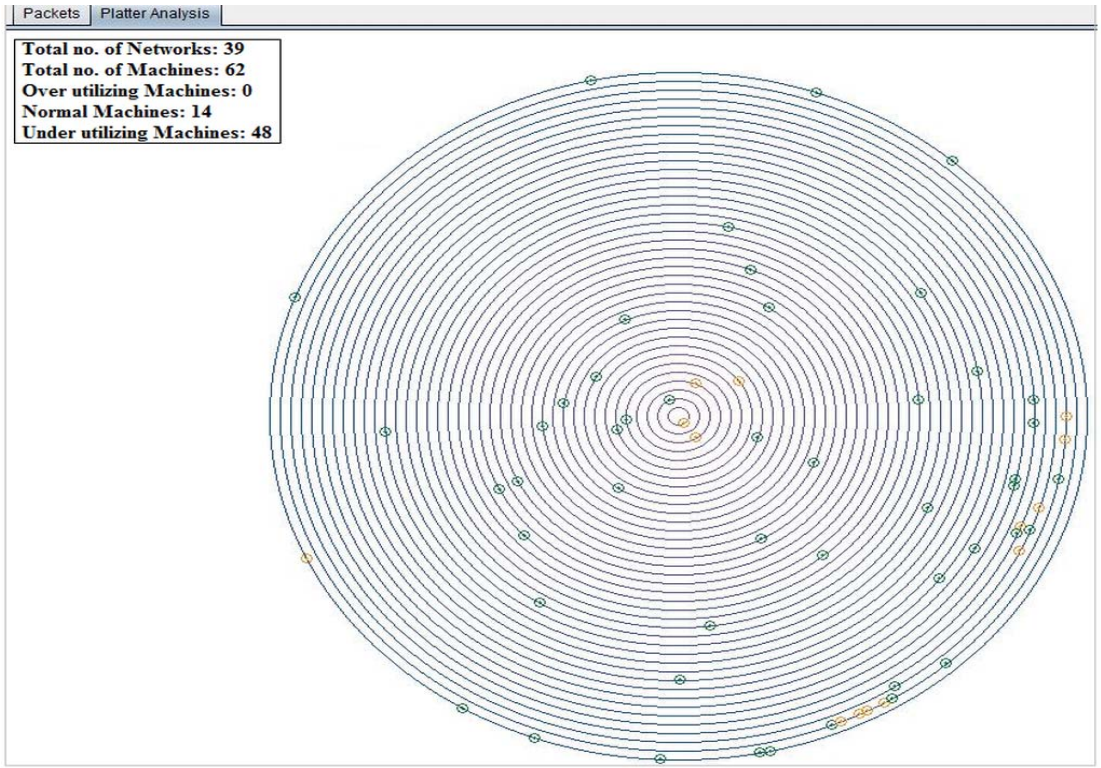


Figure 4.19: Platter view at time T2

Further parsed files are used in real time by Platter view visualization scheme (shown in Figure 4.14 to 4.15) to give global view of CAN, including the last packets captured. After every 10 seconds it refreshes itself and generates an updated Platter view and gives latest view of complete network showing in Figure 4.18 at time T1 there are 16 Networks and 41 machines and at time T2 there are 39 Networks and 62 machines as shown in Figure 4.19. Further mouse over facility gives details of machines in any network and application layer statistics of a particular node. Nodes with red color are analyzed through two dimensional analysis w.r.t. network traffic and application layer protocol.

Two - dimensional analysis of network traffic - INTVS facilitates to analyze whole network traffic through two - dimensional analysis to have a detailed understanding of network, like which VLAN is consuming maximum bandwidth, which particular device of a particular VLAN is consuming maximum bandwidth and which particular application layer based traffic is occupying a device. All this happens in INTVS with drill down facility. Figure 4.20 (Top left quadrant) is showing that HTTP traffic has consumed maximum bandwidth of the CAN. Figure 4.20 (Top right quadrant) is showing Intranet and Internet traffic usage. It gives an inference that user are also using intranet based service very well.

INTVS also maintains the records of CAN traffic at hourly basis as mentioned in Figure 4.20 (bottom left quadrant) is showing that mid night hours are most busy network time for the university. Details of VLANs, machines and traffic are available and viewed using data filtering option as outlined in graphical representation in Figure 4.20 (bottom left quadrant). This also gives information about bandwidth consumed by a particular machine in concurrence with Network bandwidth policy. In continuing with two-dimensional network analysis, for machine level view in Figure 4.22 (bottom right quadrant) displays protocol wise bandwidth on a particular

machine linked to the specified network. Figure 4.20 (bottom left quadrant) is showing VLAN 172.31.73.0 is consuming maximum bandwidth.

Network level view - Further with help of drill down facility all devices in a VLAN 172.31.73.0 are shown in Figure 4.21 (bottom right quadrant). The consumption of maximum and minimum bandwidth by a machine is shown in Figure 4.21 for network 172.31.73.0.

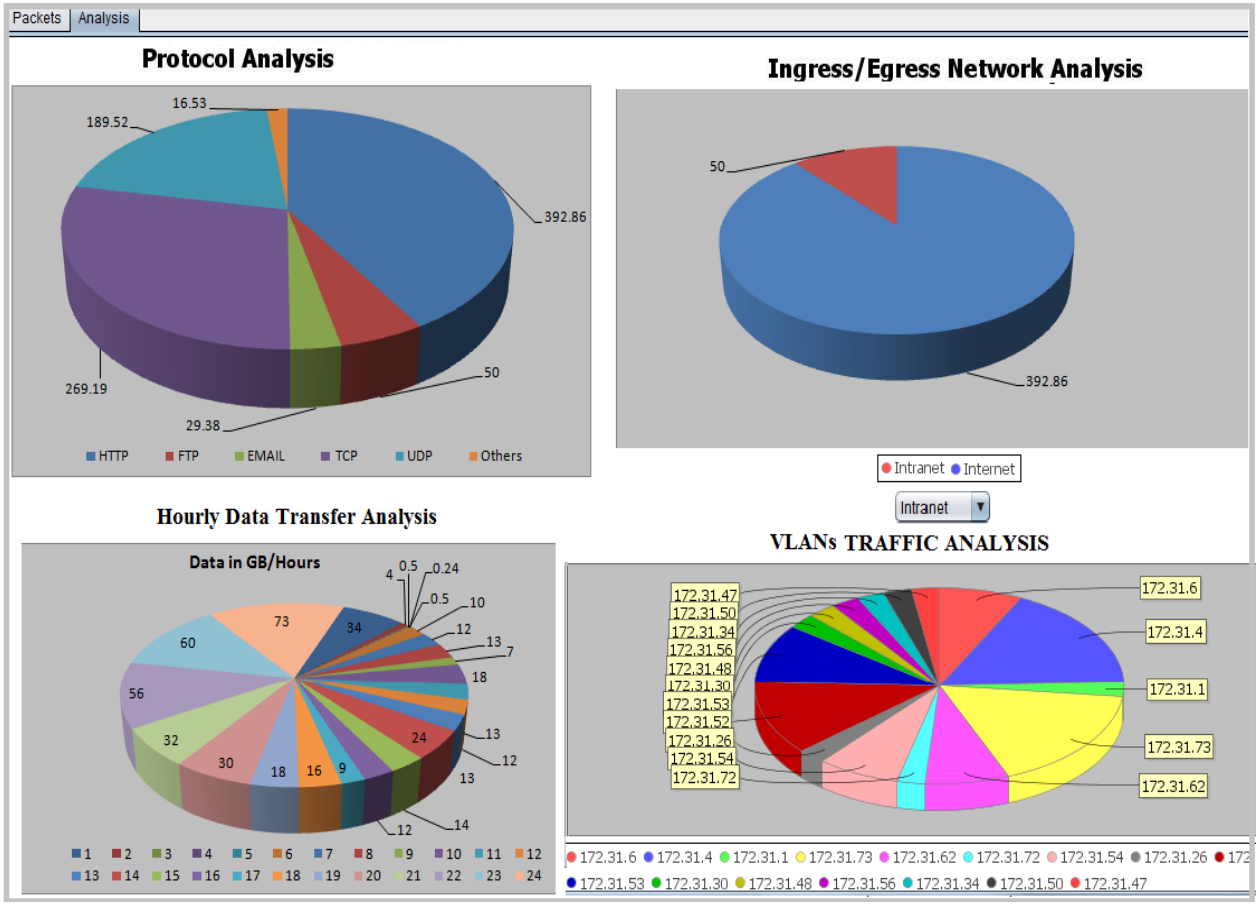


Figure 4.20: Two dimensional Analysis of CAN

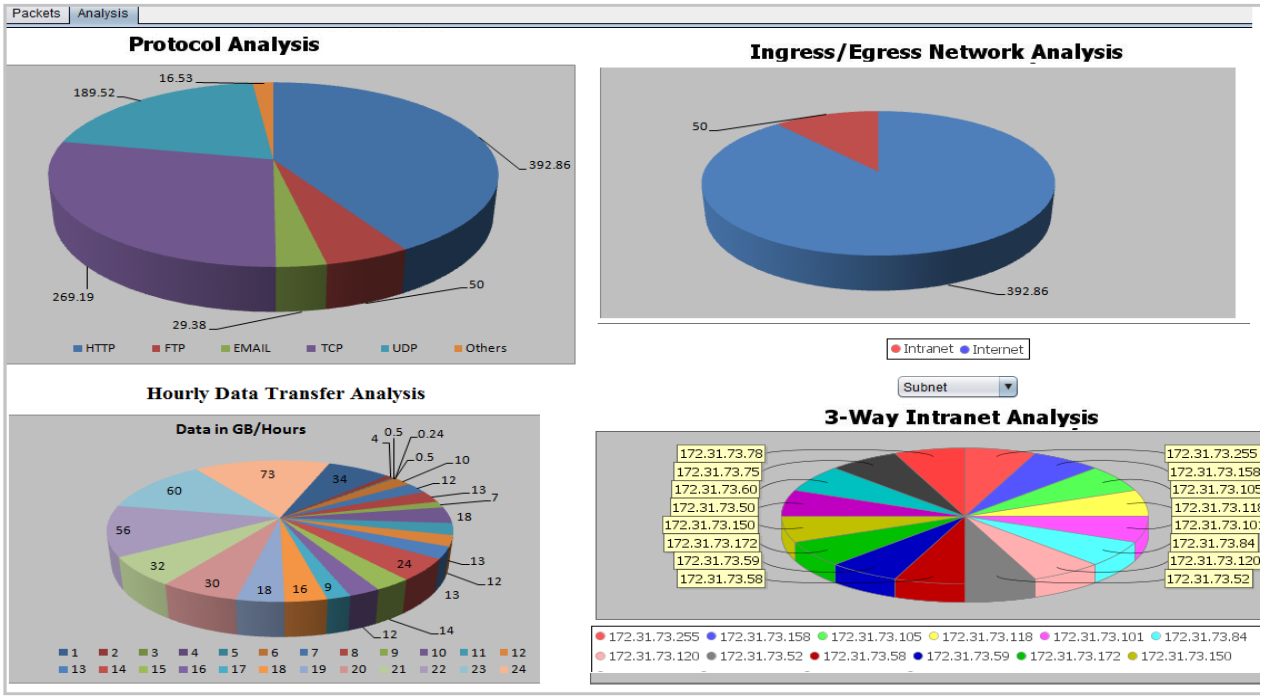


Figure 4.21: Two dimensional Network level view

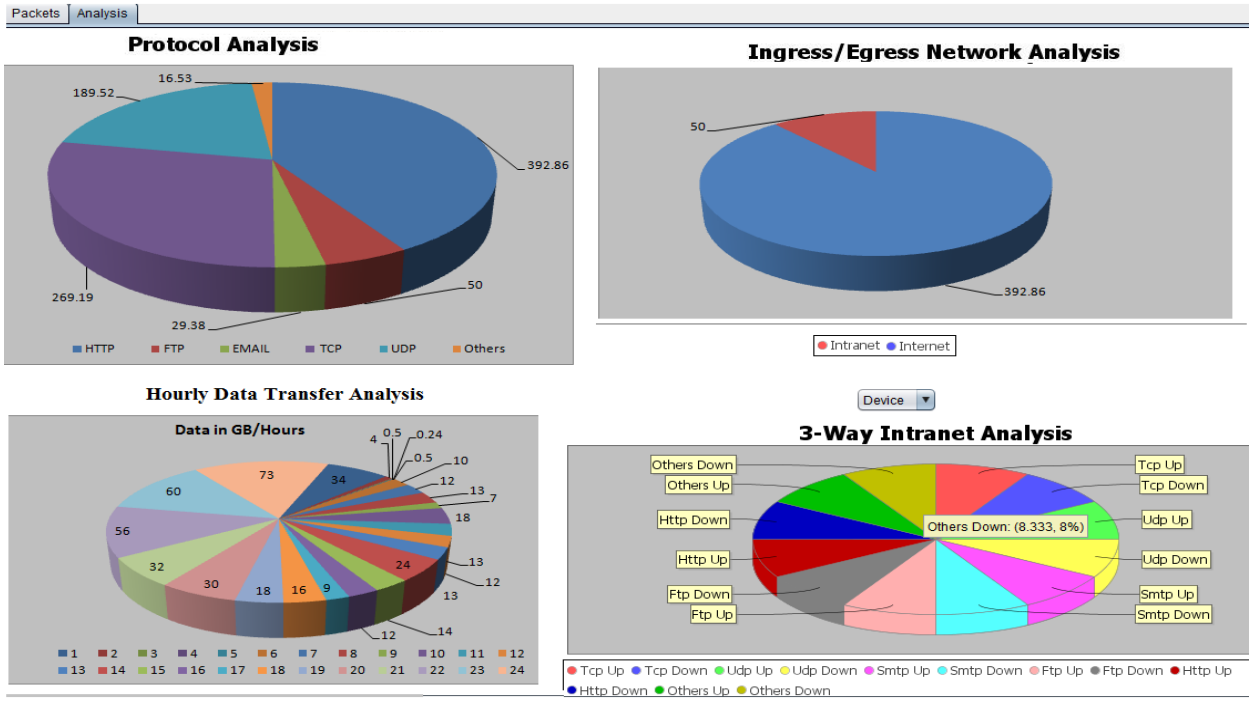


Figure 4.22: Two dimensional Machine level view

Machine level view - Continuing with two dimensional network analysis, admin can select a particular machine IP through mouse click option as shown in the Figure 4.22 (bottom right quadrant), which highlights protocol wise consumptions.

Network administrative can take the advantage of INTVS in real time mode to identify machines which is over utilizing the resources and then take appropriate action. In the absence of manual action INTVS can be configured to take action against these machines according to network policy by inserting dynamic rules.

Parallel Coordinated Analysis: In real time mode where an administrator finds an overloaded machine and wishes to know its connectivity patterns, parallel coordinated analysis based visualization implemented in INTVS helps in generating connectivity between machines of campus area network and Internet machines through proxy server as shown in Figure 4.23.

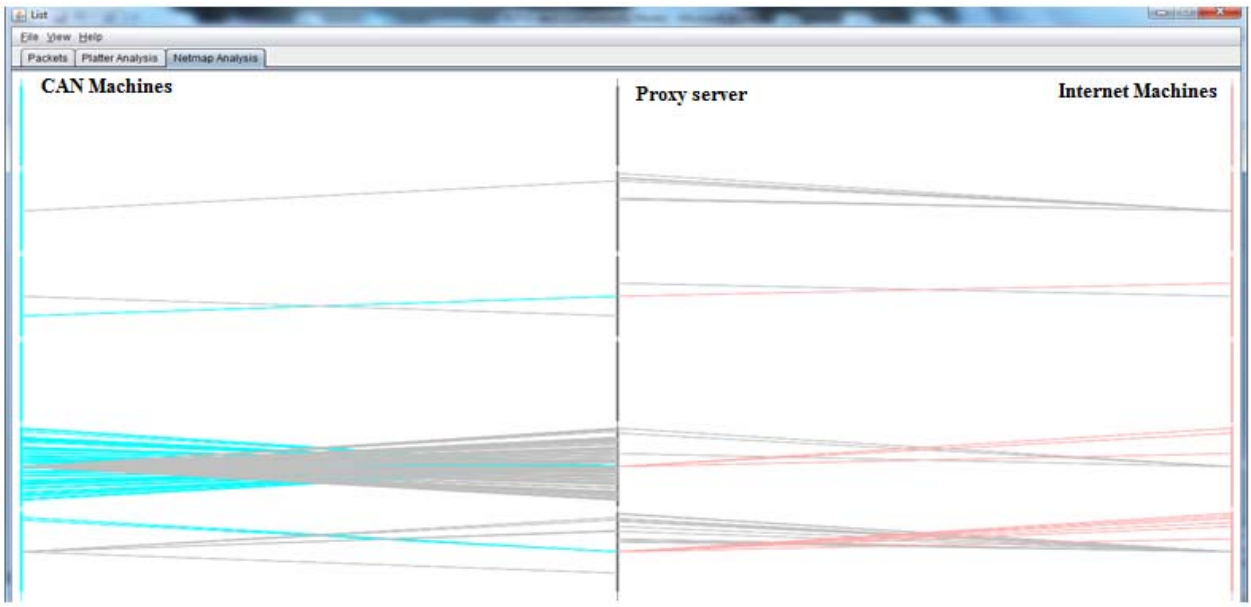


Figure 4.23: Parallel coordinated view of CAN traffic

4.3 Working of INTVS in offline mode

During implementation we categorized the attacks in three major category w.r.t. network, transport and application layer. At network layer each packet is analyzed for its source Internet protocol (SIP), destination Internet protocol (DIP) address and according to network policy for ingress and egress traffic. There are certain nodes and VLAN (virtual local area network) such as server farm/nodes (email server, Web kiosk server, database server) which are particularly accessible by particular nodes purposely, rest are prohibited. If any such node, which is not privileged to access server farm node by virtue of ingress and egress connections, then an attack is identified and classified under network layer attack . At transport layer, transmission control protocol (TCP) and user datagram protocol (UDP) are used to establish process to process connection for data transmission.

At server side, there are certain ports open according to network policy for its various users, if anybody tries to violate these rules, then transport layer port violation is tagged and alert is generated. At application layer, many attacks are experienced w.r.t. email service (SMTP Mail Flooding, spamming), HTTP server (HTTP-based attacks spanning multiple packets, HTTP header spoofing attacks), FTP service (FTP bounce attack, passive FTP attacks, client and server bounce attacks, FTP port injection attacks) etc. These are identified based on various features of packet such SIP, DIP, S_port, D_Port, Pkt_length, session and timestamp.

INTVS was experimented by deploying it in an isolated environment for forensic analysis to take corrective measures through a data mining scheme. Different attack were launched to validate INTVS framework for network layer attacks, transport layer attacks as well as application layer attacks.

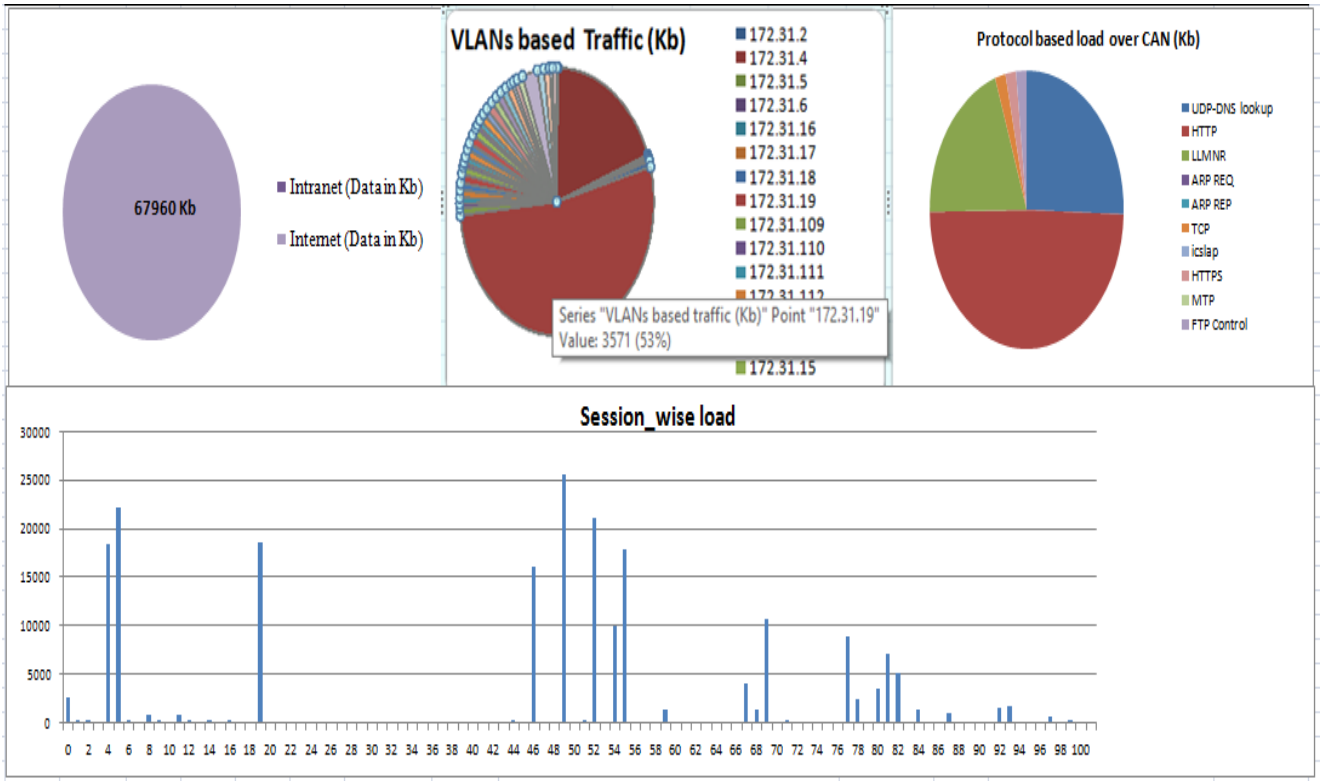
Forensic analysis of network traffic as done by INTVS is reported in a dashboard form (Figure 4.24). It explains current scenario of the network and helps to formulate future network policy decisions. In Figure 4.24, the dashboard presents total time, total number of machines, total number of VLANs, total data transmitted, total intranet load, total internet load, VLANs based traffic, particular VLAN based load over machines, particular machine load, reporting load, protocol-wise w.r.t. Campus area network, VLAN, machines, and different types of attacks.

There were 62 nodes and 39 VLANs in operation. In this time span, total data transmitted is 67960 Kb under 92 data sessions 18 network layer attacks and 11 application layers attacks detected, in concurrence with network policy. Out of 39 VLANs VLAN *172.31.19.0* is highly overloaded. In CAN there were 10 different network services observed and it was found that HTTP usage is maximum used. Further it was found that maximum resource consumption was by 172.31.19.184 machine.

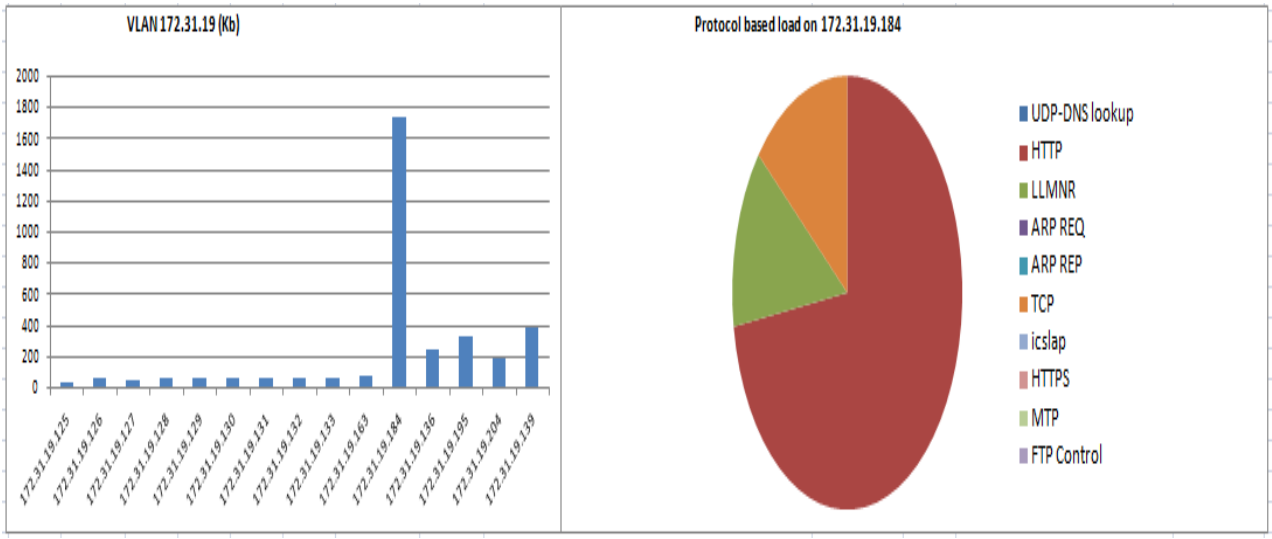
Sr.no.	Pkt_no.	SIP	DIP	S_port	D_port	Pkt_length	Session	Attack Type	Frequency
1								N/w layer	18
2								Transport layer	0
3								Transport layer (Deceptive)	0
4								Application layer attack based on HTTP	11
5								Application layer attack based on EMAIL	0
6								Application layer attack based on FTP	0
7								Application layer attack based on Telnet	0
8								No threats	13600
								Total Packet	13629

Toatl Machines	62
VLANs	39
Total Data Transmitted	67960
Interanet Data in Kb	0
Internet Data in Kb	67960
Total Session	984
Total packet	13629

(a)



(b)



(c)

Figure 4.24: Forensic Analysis

4.4 Salient Feature of the INTVS

a) ***Media Support*** – INTVS is capable of capturing network data from both wired and wireless media as shown in Figure 4.25, a feature that is offered by limited tools such as Nam, Rumint, AfterGlow and FloVis.

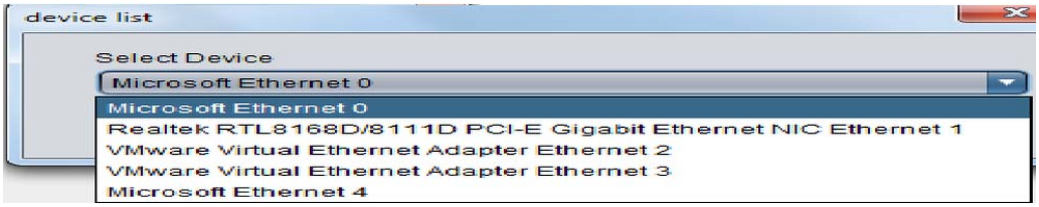


Figure 4.25 Snapshot of media selection

b) ***Interoperable*** – INTVS is compatible with many operating systems such as Windows, Linux and its variants as it is developed using Java. Figure 4.26 is shows INTVS running on Ubuntu.

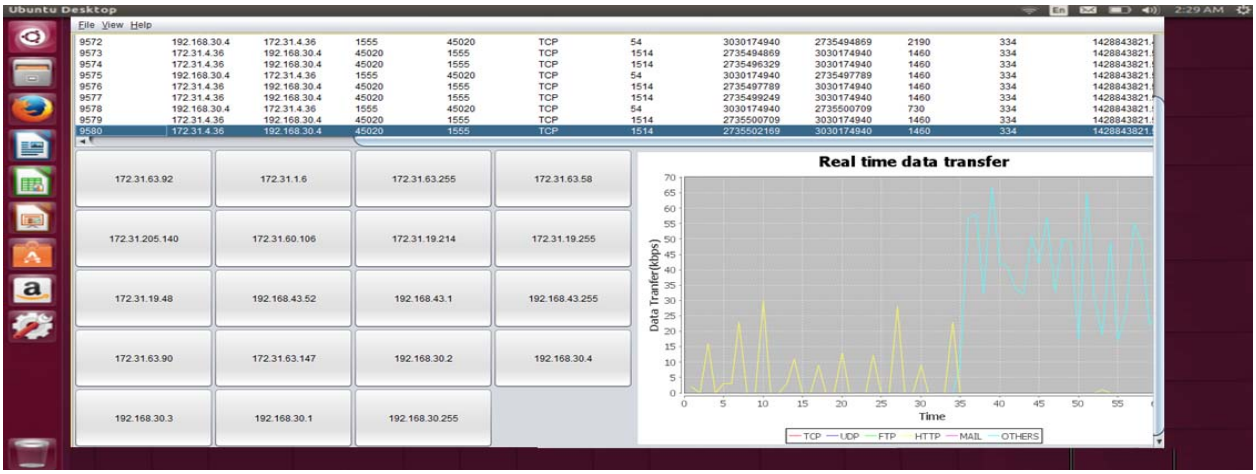


Figure 4.26: Grid View of INTVS on Ubuntu

c) ***Security Analysis of Application Layer Traffic*** – As discussed in *chapter-2*, only Nam and AfterGlow are capable to analyze application layer data that too with limitations. However, INTVS can facilitate visualization of application layer data as shown in Figure 4.27 about

HTTP traffic view, email, and FTP traffic. In addition to tracking VLANs, network, machine and type of web service and all related statistics like volume of data uploaded and downloaded, Listmap view (Figure 4.7), Platter view (Figure 4.12) and two - dimensional analysis of CAN traffic (Figure 4.21) visualization scheme are useful in security analysis. Further based on the traffic patterns, INTVS can detect threats.

d) ***Visual Information*** – INTVS learns its traffic pattern and generates alerts according to laid down network policy. INTVS follows principles of Shneiderman [35] advocating visualization benefits. In Grid view and Platter visualization, INTVS facilitate to customize data filtrations and drill down to reach the source of the problem. In addition, the system can also select a particular node for analysis. Two - dimensional analysis is helpful in application layer traffic analysis, which contributed towards novelty in network traffic visualization.

File View Help

---------- HTTP ----------

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
3700	115.254.106.208	172.31.30.56	80	58313	HTTP	1514	133927962	3923507462	1002	141	1367921853
3701	172.31.26.84	69.171.237.20	56329	80	HTTP	1514	2882534416	1627679389	254	517	1367921853
3702	172.31.25.115	115.254.106.207	51273	80	HTTP	60	3252637462	104547085	256	58	1367921853
3703	173.194.36.41	172.31.55.90	443	49358	HTTPS	1514	3420319581	2432902319	1107	8	1367921853
3705	68.232.45.96	172.31.33.84	80	49643	HTTP	1514	3763790741	1755211740	175	30	1367921853
3706	115.254.106.207	172.31.25.115	80	51273	HTTP	1418	104547085	3252637462	822	58	1367921853
3707	23.21.195.112	172.31.34.59	443	50432	HTTPS	1514	278878949	3600835839	142	32	1367921853
3709	115.254.106.210	172.31.25.21	80	54992	HTTP	1514	916675831	885697178	58400	57	1367921853
3710	69.171.248.16	172.31.36.63	443	54095	HTTPS	1428	689636195	2521865369	1002	541	1367921853
3711	172.31.26.180	103.246.40.42	54741	80	HTTP	60	1825033707	1549484790	16073	291	1367921853
3712	103.246.40.42	172.31.26.180	80	54741	HTTP	1466	1549484790	1825033707	110	291	1367921853
3713	173.194.52.112	172.31.48.78	80	50417	HTTP	1514	1848029327	3713081041	183	39	1367921853
3714	173.194.52.112	172.31.48.78	80	50417	HTTP	1418	1848030787	3713081041	183	39	1367921853
3715	173.194.14.42	172.31.46.55	80	49584	HTTP	1514	1586817363	3929591816	114	352	1367921853
3716	115.254.106.210	172.31.46.51	80	1694	HTTP	1514	3199653741	2960485177	115	68	1367921853
3718	124.124.40.9	172.31.26.96	80	49758	HTTP	1514	3438492732	3250457430	118	22	1367921853
3719	173.194.52.112	172.31.48.78	80	50417	HTTP	1514	1848032151	3713081041	183	39	1367921853
3720	74.125.128.117	172.31.19.134	443	56019	HTTPS	60	3330724368	1667618412	1169	513	1367921853
3721	74.125.128.117	172.31.19.134	443	56019	HTTPS	60	3330724368	1667618445	1169	513	1367921853
3722	109.163.229.149	172.31.35.43	80	51781	HTTP	1514	1847909553	4190727190	123	88	1367921853
3723	173.194.52.9	172.31.62.14	80	52436	HTTP	1514	1908799960	3024835239	183	118	1367921853
3724	180.149.59.12	172.31.45.37	80	50430	HTTP	1514	331687685	2479122055	183	488	1367921853

HTTP traffic

---------- FTP ----------

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
926	192.168.30.4	172.31.4.36	1402	21	FTP Control	66	4138879534	2378432353	4360	62	1428843475
927	172.31.4.36	192.168.30.4	21	1402	FTP Control	82	2378432353	4138879548	1460	62	1428843475
928	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879548	2378432381	4353	62	1428843475
929	192.168.30.4	172.31.4.36	1402	21	FTP Control	60	4138879548	2378432381	4353	62	1428843475
930	172.31.4.36	192.168.30.4	21	1402	FTP Control	73	2378432381	4138879554	1460	62	1428843475
931	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879554	2378432400	4349	62	1428843475
932	192.168.30.4	172.31.4.36	1402	21	FTP Control	65	4138879554	2378432400	4349	62	1428843475
933	172.31.4.36	192.168.30.4	21	1402	FTP Control	76	2378432400	4138879565	1460	62	1428843475
934	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879565	2378432422	4343	62	1428843475
935	192.168.30.4	172.31.4.36	1402	21	FTP Control	59	4138879565	2378432422	4343	62	1428843475
936	172.31.4.36	192.168.30.4	21	1402	FTP Control	77	2378432422	4138879570	1460	62	1428843475
937	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879570	2378432445	4337	62	1428843475
938	192.168.30.4	172.31.4.36	1402	21	FTP Control	62	4138879570	2378432445	4337	62	1428843475
939	172.31.4.36	192.168.30.4	21	1402	FTP Control	84	2378432445	4138879578	1460	62	1428843475
940	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879578	2378432475	4330	62	1428843475
941	192.168.30.4	172.31.4.36	1402	21	FTP Control	60	4138879578	2378432475	4330	62	1428843475
942	172.31.4.36	192.168.30.4	21	1402	FTP Control	103	2378432475	4138879584	1460	62	1428843475
943	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879584	2378432524	4318	62	1428843475
947	192.168.30.4	172.31.4.36	1402	21	FTP Control	60	4138879584	2378432524	4318	62	1428843475
948	172.31.4.36	192.168.30.4	21	1402	FTP Control	93	2378432524	4138879590	1460	62	1428843475
949	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879590	2378432563	4308	62	1428843475
953	172.31.4.36	192.168.30.4	21	1402	FTP Control	78	2378432563	4138879590	1460	62	1428843475
955	192.168.30.4	172.31.4.36	1402	21	FTP Control	54	4138879590	2378432587	4302	62	1428843475

FTP traffic

---------- MAIL ----------

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
45	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527325502	1408842911	1002	35	1367921971
149	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527328302	8104	35	1367921971
938	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527328302	1408842911	1002	35	1367921971
974	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527329702	8192	35	1367921971
1830	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527329702	1408842911	1002	35	1367921971
2726	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527332502	8104	35	1367921971
3175	74.125.25.109	172.31.51.93	993	62921	IMAPS	1466	242053956	1276105120	1002	588	1367921971
3176	74.125.25.109	172.31.51.93	993	62921	IMAPS	1466	242055356	1276105120	1002	588	1367921971
3295	74.125.25.109	172.31.51.93	993	62921	IMAPS	1466	242056756	1276105120	1002	588	1367921971
3298	74.125.25.109	172.31.51.93	993	62921	IMAPS	1466	242058156	1276105120	1002	588	1367921971
3665	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527336702	8104	35	1367921971
3792	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527338102	1408842911	1002	35	1367921971
3871	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527340902	8104	35	1367921971
3921	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527340902	1408842911	1002	35	1367921971
3935	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527342302	8192	35	1367921971
3999	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527342302	1408842911	1002	35	1367921971
4062	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527343702	1408842911	1002	35	1367921971
4083	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527345102	8104	35	1367921971
4278	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527349302	8104	35	1367921971
4331	74.125.129.109	172.31.25.24	993	59215	IMAPS	1466	1527349302	1408842911	1002	35	1367921971
4349	172.31.25.24	74.125.129.109	59215	993	IMAPS	66	1408842911	1527350702	8192	35	1367921971

Email traffic

Figure 4.27: Application layer traffic view by INTVS

e) ***Scalable*** – INTVS framework supports real time as well as off line data analysis in addition to supporting multi format data. System has capability to capture and visualize traffic from a small network of hundreds of nodes to thousands of nodes.

f) ***Intelligence*** – It is necessary to understand the traffic patterns and take necessary action in self-defense through corrective measures. INTVS framework is capable to manage huge visualization data on a single screen, thus facilitating the administrator to identify and display compromised machines in a network through colors schemes of dot/ring and circle in a platter visualization mode.

g) ***Reporting*** – INTVS produces data patterns reporting via platter visualization and two - dimensional traffic analysis. It can respond to many queries such as – how many VLANs are present, how many hosts are live, the number of downloads and uploads by single host, port association, protocol wise bandwidth utilization etc. This system also helps network security analysts to view intended information and piping it to visual reports like network bandwidth consumed by particular application layer traffic with reference to total traffic. INTVS is portable, scalable and facilitate its users to export captured data for future references in CSV and .xls files for forensic analysis.

h) ***Real Time and Forensic Analysis*** – In real time domain, there are very few tools that offer real time analysis. INTVS is parsing captured traffic data in XML format used for real time analysis mode as discussed above in Section 4.2. INTVS is also capable in handling forensic analysis of network traffic while exporting the data files which can be stored in a relational database management system (RDBMS) for offline reporting and analysis as discussed above in Section 4.3.

Conclusions:

This chapter detailed implementation of INTVS, its features, design, development, deployment and experimentations. Chapter was divided into three main sections, which talked about implementation of INTVS as a complete VizSec solution, working of INTVS in real time environment and working of INTVS in offline mode. INTVS demonstrated capturing, tokenizing, parsing and visualization of campus area network traffic through Grid view, Listmap, Platter view and line graphs schemes in real time. Two - dimensional analysis view demonstrated root cause of a network problem. Working of INTVS in offline mode talked about the forensic analysis of netflow, Internet traffic, Intranet traffic, VLANs detail, application layer traffic details, attacks details, and machine level details.

Chapter 5

INTVS Results, Reports and Reliability Validation

This chapter presents various set of results and reports collected while running INTVS using various network hierarchies. Focus was to test whether INTVS is able to capture the live data, tokenized the captured data, parsed the tokenized data, and visualized the parsed data in real time as well as in offline. Following queries are answered and discussed: Does INTVS is able to report network health in visual form properly and precisely? Does INTVS is able to detect and display attack vectors? Does INTVS is able to display the under-utilized and over-utilized network resources? The results of these queries are discussed in this chapter. Further, using fuzzy fault tree analysis approach (FFTA) reliability of INTVS is established.

5.1 INTVS Results

5.1.1 Capturing of live network traffic

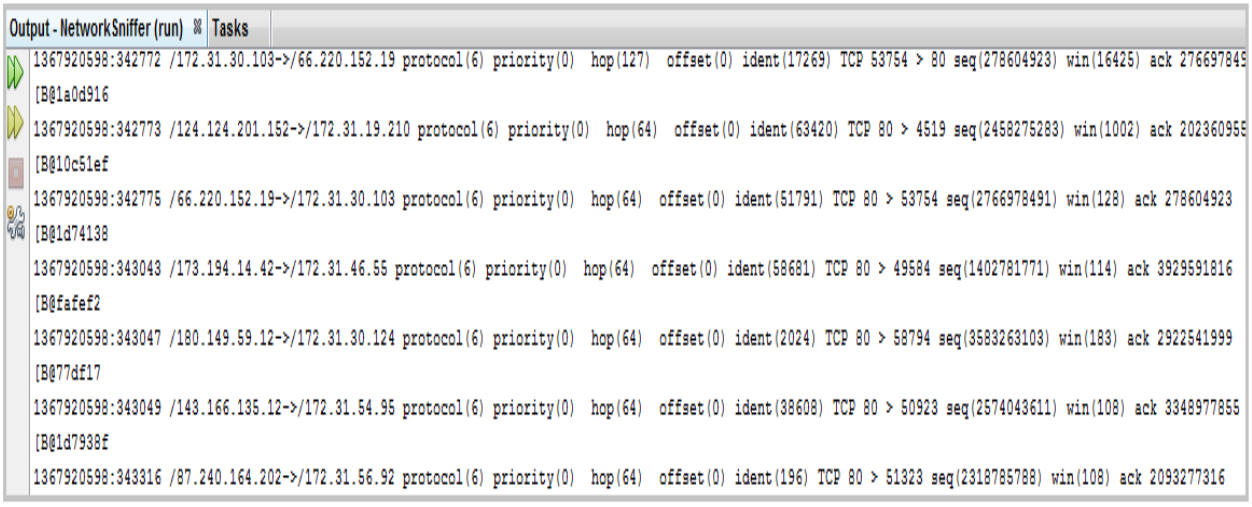
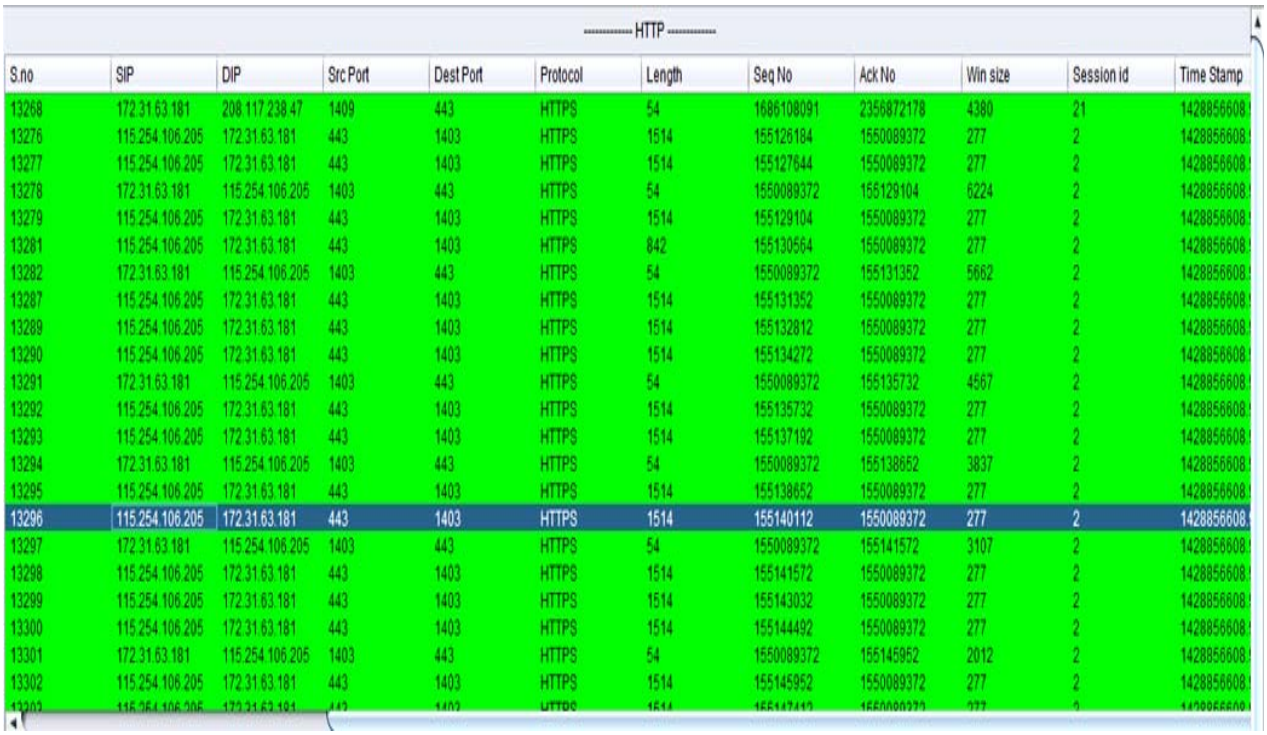


Figure 5.1: Real Time Capturing

Figure 5.1 is showing real time capturing of network traffic. Hence, it validates that INTVS can capture the network traffic in real time like Wireshark sniffing tool.

5.1.2 Tokenizing of captured network traffic

INTVS can tokenize data and can use it for parsing in real time environment. For this, multi-threading is used to accomplish the required tasks in parallel mode. Dedicated multi-threads have been invoked for tokenizing and parsing. Output of capturing module is used as input to tokenization module. Outcomes of tokenization module of INTVS is shown in Figure 5.2. The tokenized data is further used to parse in, two types of formats XML for real time analysis and XLS/CSV format for offline analysis.



---------- HTTP ----------

S.no	SIP	DIP	Src Port	Dest Port	Protocol	Length	Seq No	Ack No	Win size	Session id	Time Stamp
13268	172.31.63.181	208.117.238.47	1409	443	HTTPS	54	1686108091	2356872178	4380	21	1428856608
13276	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155126184	1550089372	277	2	1428856608
13277	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155127644	1550089372	277	2	1428856608
13278	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155129104	6224	2	1428856608
13279	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155129104	1550089372	277	2	1428856608
13281	115.254.106.205	172.31.63.181	443	1403	HTTPS	842	155130564	1550089372	277	2	1428856608
13282	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155131352	5662	2	1428856608
13287	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155131352	1550089372	277	2	1428856608
13289	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155132812	1550089372	277	2	1428856608
13290	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155134272	1550089372	277	2	1428856608
13291	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155135732	4567	2	1428856608
13292	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155135732	1550089372	277	2	1428856608
13293	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155137192	1550089372	277	2	1428856608
13294	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155138652	3837	2	1428856608
13295	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155138652	1550089372	277	2	1428856608
13296	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155140112	1550089372	277	2	1428856608
13297	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155141572	3107	2	1428856608
13298	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155141572	1550089372	277	2	1428856608
13299	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155143032	1550089372	277	2	1428856608
13300	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155144492	1550089372	277	2	1428856608
13301	172.31.63.181	115.254.106.205	1403	443	HTTPS	54	1550089372	155145952	2012	2	1428856608
13302	115.254.106.205	172.31.63.181	443	1403	HTTPS	1514	155145952	1550089372	277	2	1428856608

Figure 5.2: Tokenized Data in Real Time

5.1.3 Parsing the tokenized network traffic

INTVS can parse the tokenized network traffic data into either XML file (shown in Figure 5.3) for real time analysis or MS Excel/ CSV file for offline analysis.

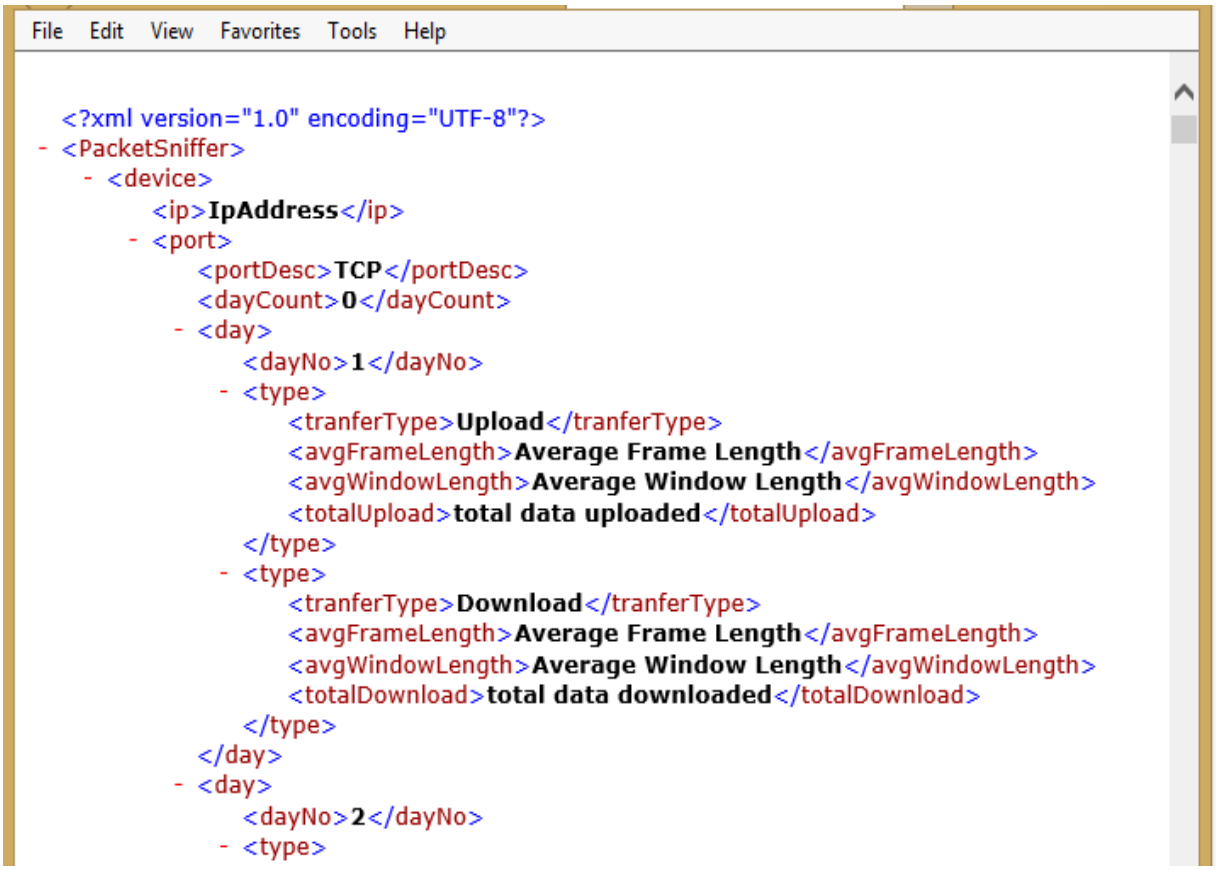


Figure 5.3: Parsed Data (XML format) in Real Time

5.1.4 Visualizing the network traffic in real time

INTVS can visualize the network traffic as Grid view, Listmap view and Platter view as shown in Figure 5.4, Figure 5.5 and Figure 5.6 respectively. Listmap view and Platter view gives a holistic view of complete CAN. Whereas line graph gives a view of bandwidth consumption in real time as shown in Figure 5.4 (bottom right).

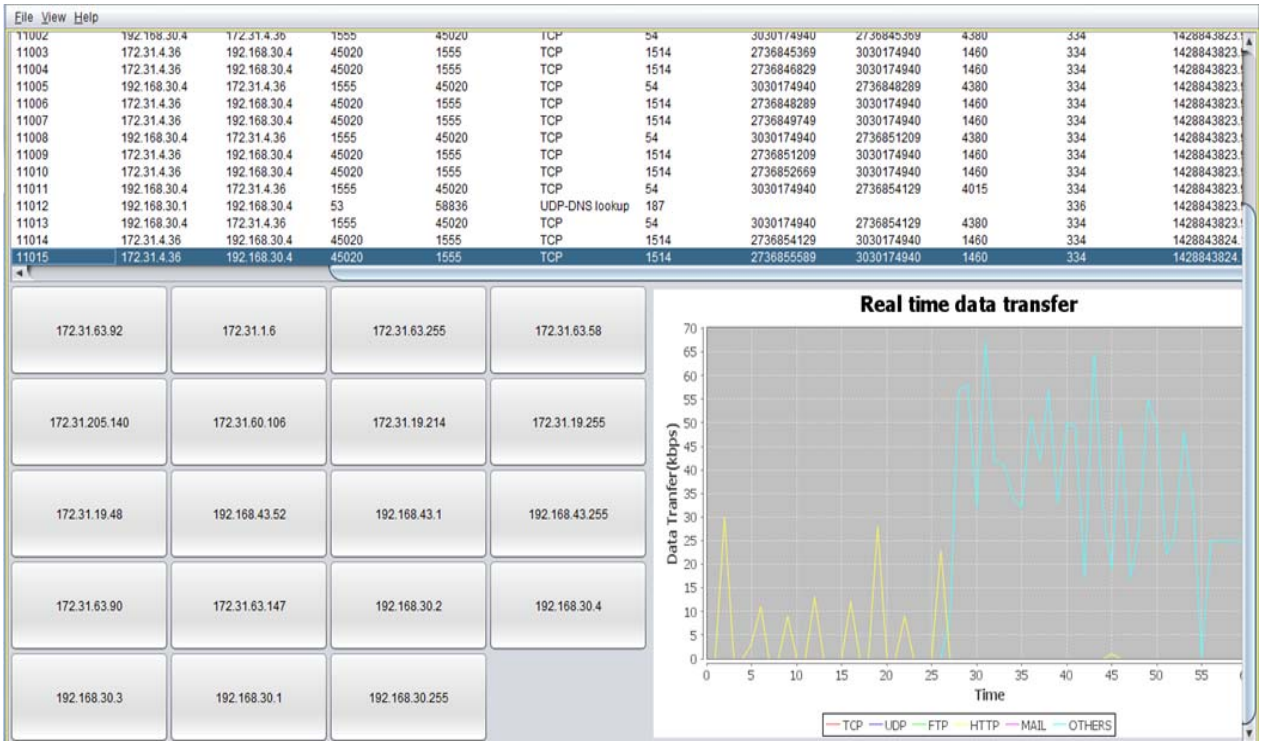


Figure 5.4: Grid view

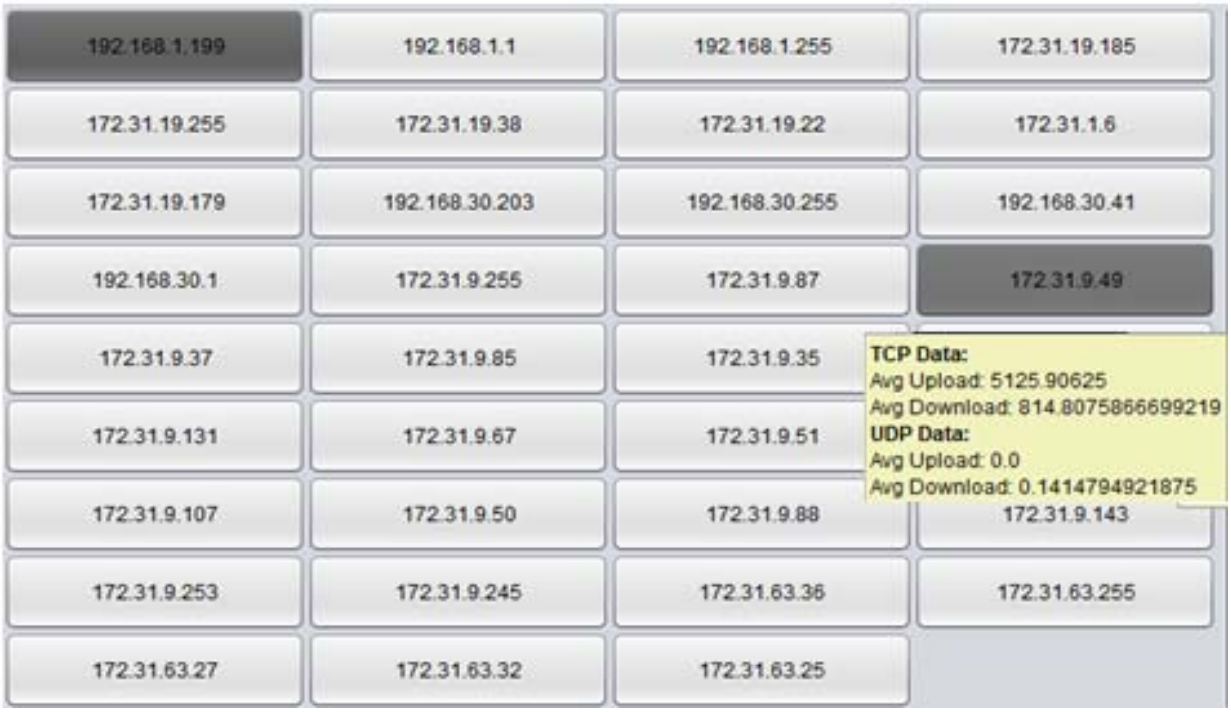


Figure 5.5: Listmap view

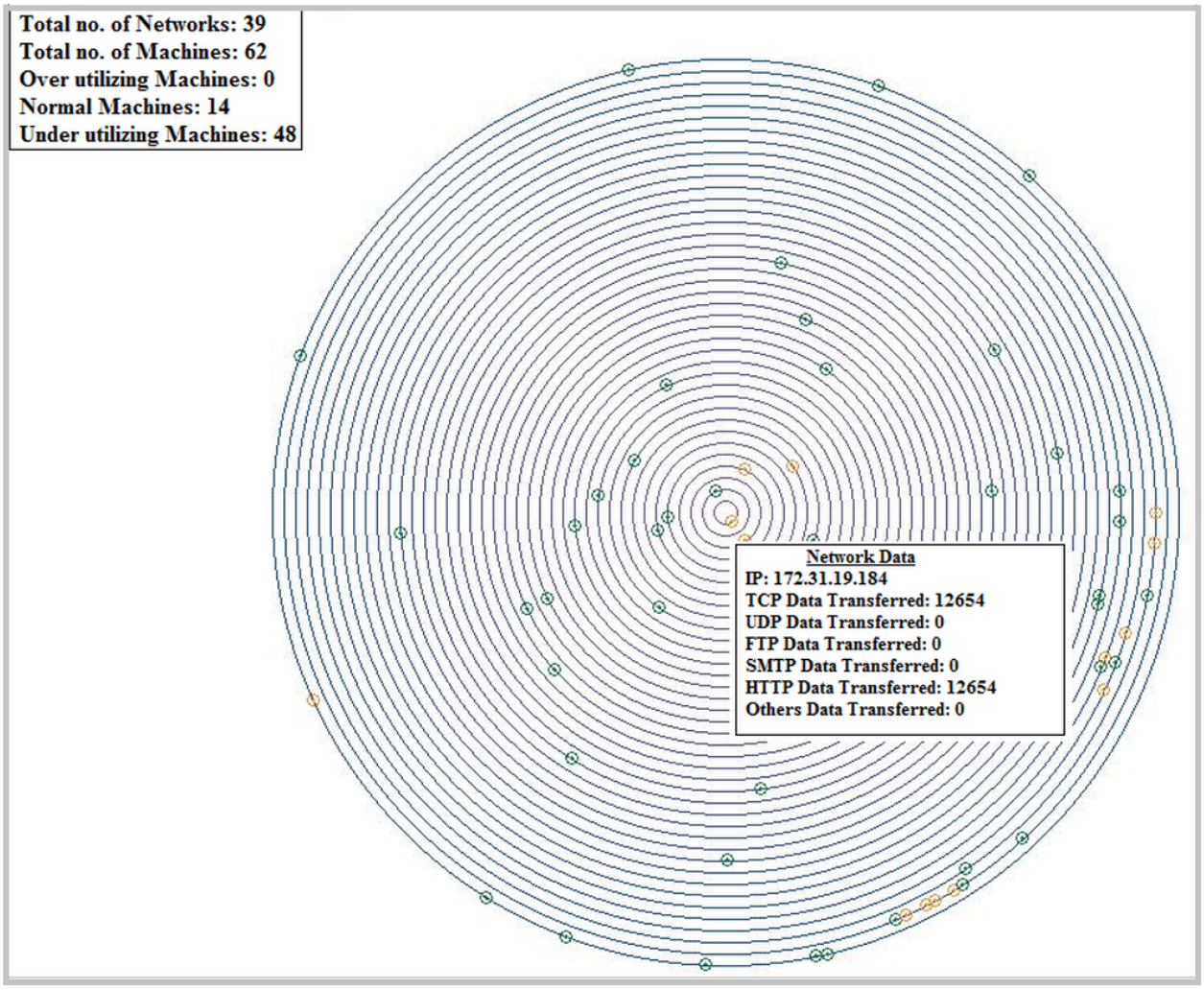


Figure 5.6: Platter view of Real Time traffic of CAN

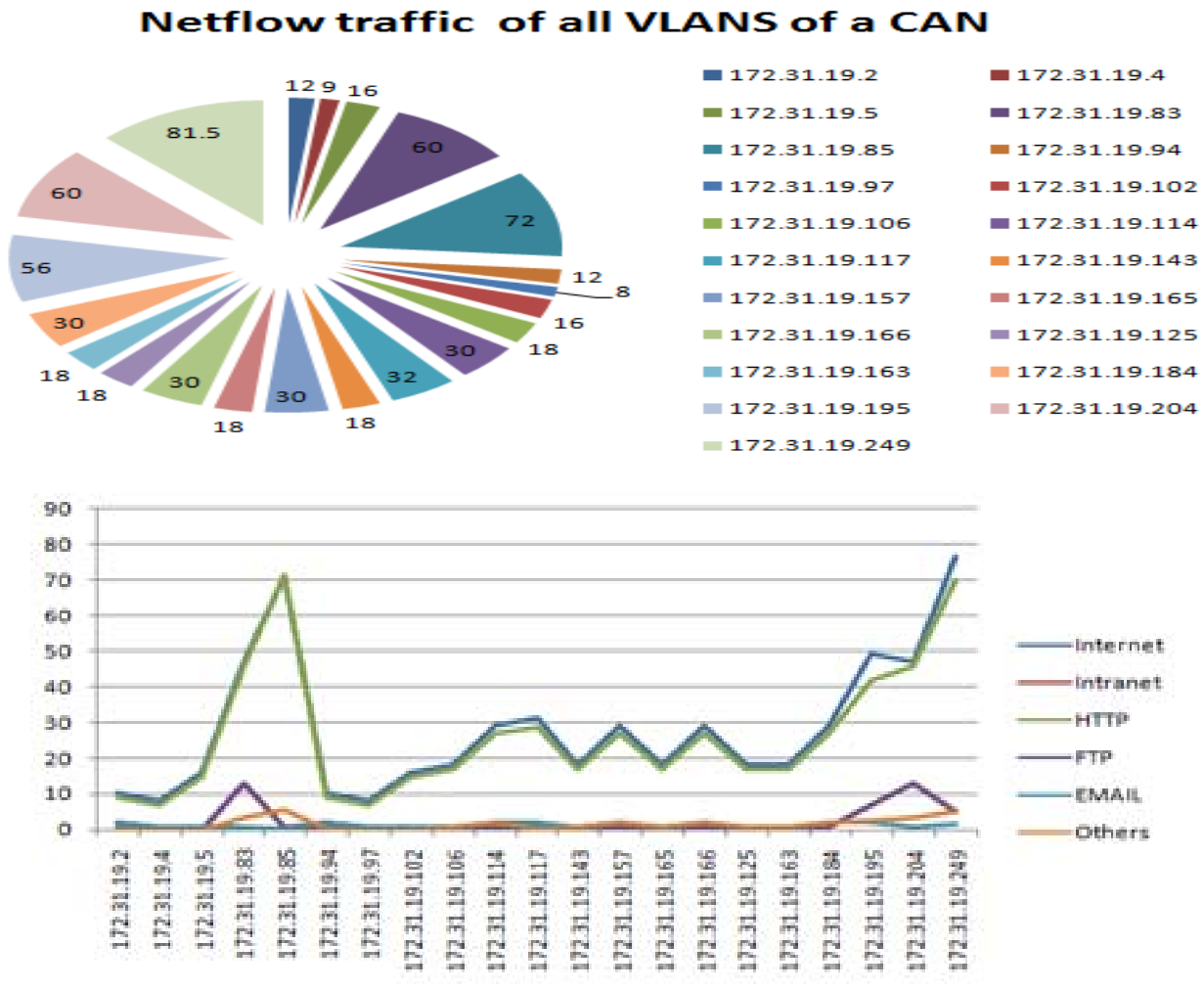
5.1.5 Visualizing network traffic in off line mode

INTVS also helps in offline analysis of CAN traffic as shown in Figure 5.7(a), Figure 5.7(b) and Figure 5.7(c). Figure 5.7(a) is showing a detailed report of campus area network traffic. It shows that 114 live machines, 39 networks and 98543 packets have been detected. It also reports application layer attacks. Figure 5.7(b) shows the load of netflow over all the VLANs of a CAN through pie chart along with application layer traffic load through line chart.

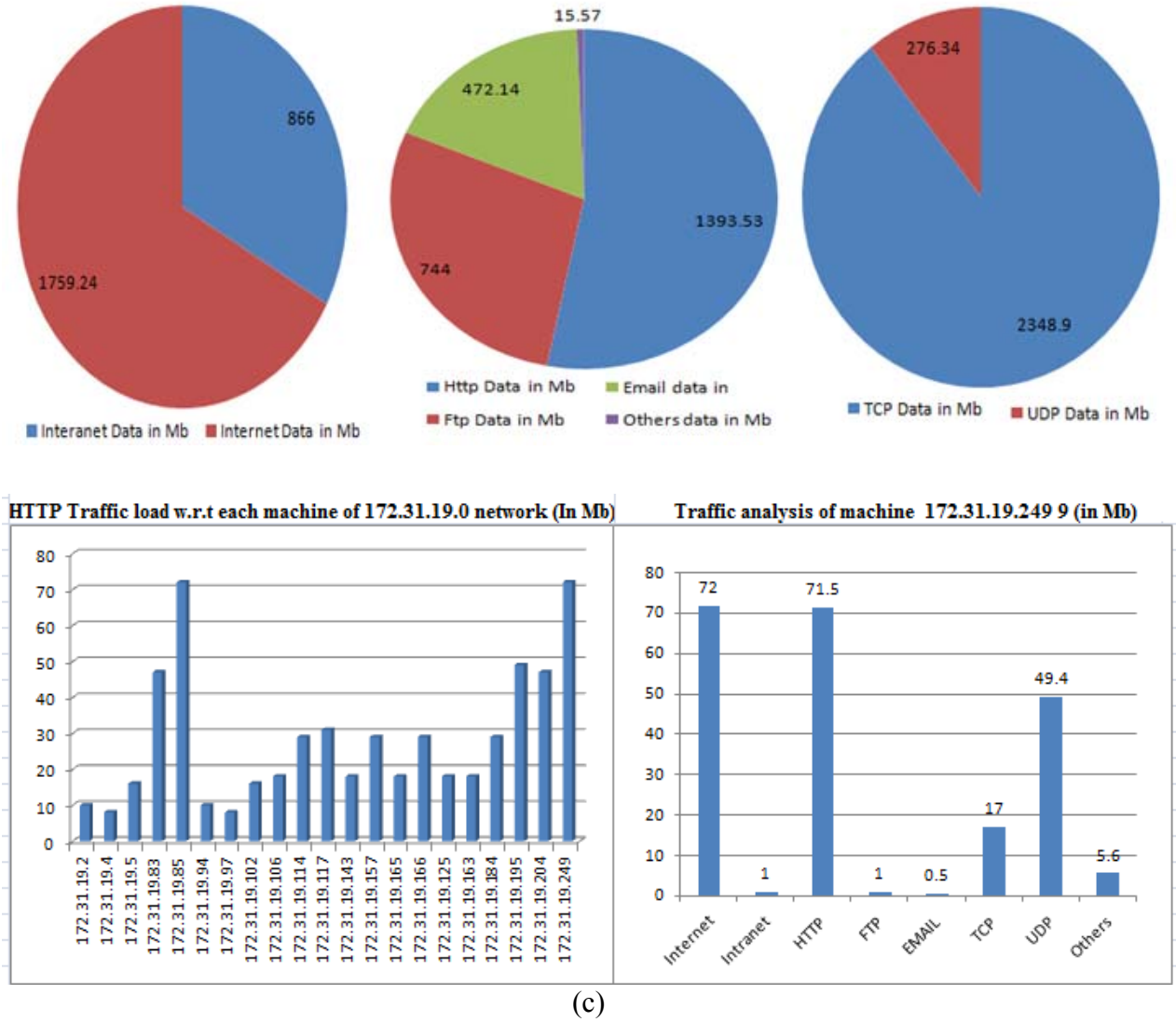
Sr.no.	Pkt_no.	SIP	DIP	S_port	D_port	Pkt_lengh	Sessipn	Attack Type	Frequency
1								N/w layer	18
2								Transport layer	0
3								Transport layer (Deceptive)	0
4								Application layer attack based on HTTP	11
5								Application layer attack based on EMAIL	0
6								Application layer attack based on FTP	0
7								Application layer attack based on Telnet	0
8								No threats	98514
								Total Packet	98543

Toatl Machines	114
VLANs	39
Total Data Transmitted	2625.24
Interanet Data in Mb	866
Internet Data in Mb	1759.24
Http Data in Mb	1393.53
Ftp Data in Mb	744
Email data in	472.14
Others data in Mb	15.57
Total Session	984
Total Time in seconds	600.125
TCP Data in Mb	2348.9
UDP Data in Mb	276.34

(a)



(b)



(c)

Figure 5.7: Forensic Analysis

Figure 5.7(c) gives a dashboard view of CAN traffic. It shows the view of Internet versus Intranet usage. It shows application layer protocol usage. Also shows the TCP versus UDP traffic over the CAN. Whereas column chart is showing machine number 172.31.191.185 is using maximum http traffic in VLAN 172.31.19.0.

5.1.6 Visualizing network health INTVS

INTVS can give a precise and right view of complete network through Platter view scheme as shown in Figure 5.8. Figure shows that there are 14 machines which are using network in normal manner, whereas 48 machines are using network in below average manner. This figure also indicated that there is no threat at this point of time.

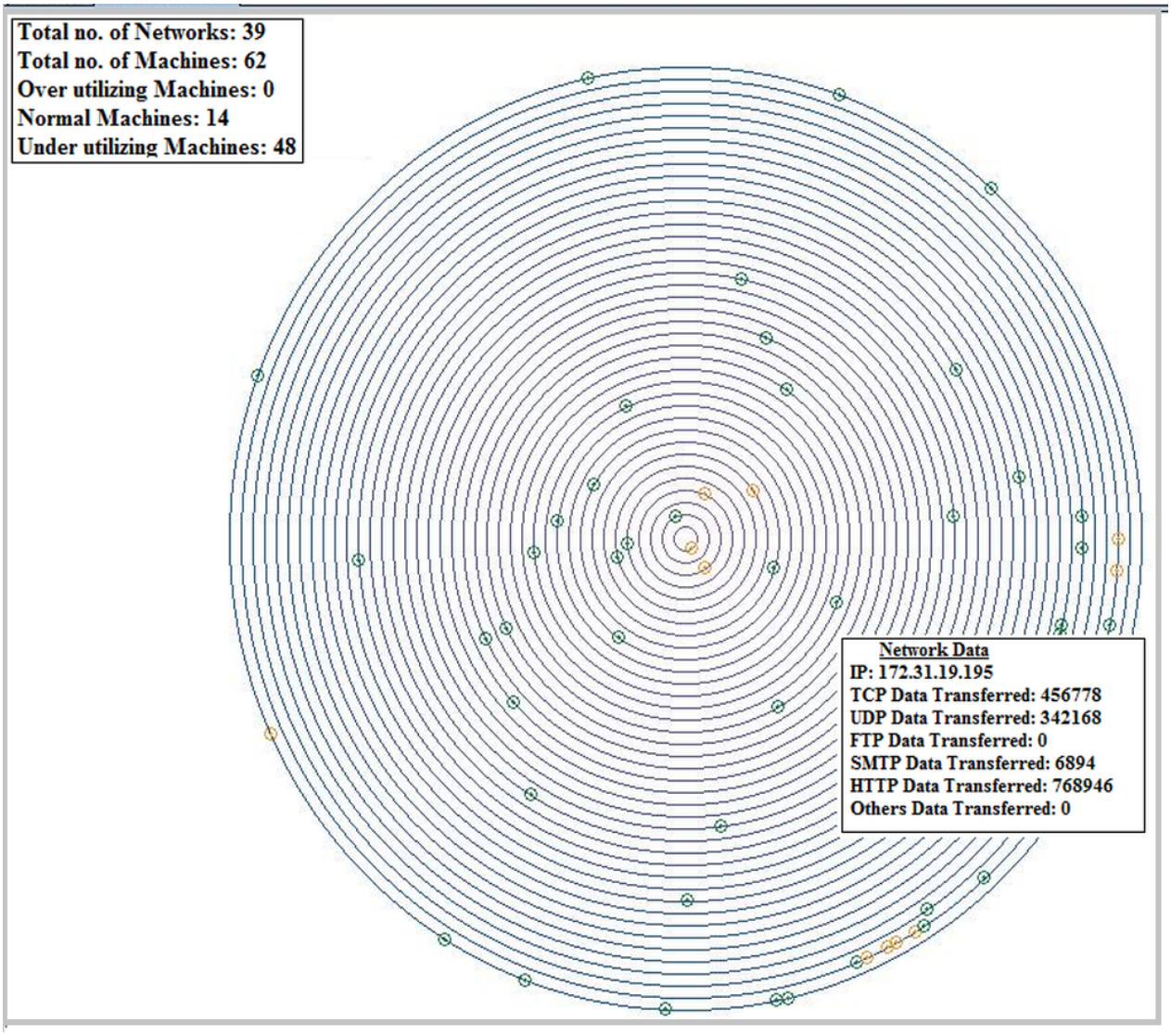


Figure 5.8: Platter view

5.1.7 Visualizing the attack vector through INTVS

INTVS is having the capability of detecting and visualizing under attack machine through Platter view. A machine which is violating the network policy and went beyond its privileges, is marked red dot. In Figure 5.9, 39 networks and 114 machines are displayed at the time of experiment. In which 29 machines (shown as orange dots) are over utilizing the network resources, but there are 7 machines which have broken network policy rules are shown as red dots. Mouse over facility gives details about that node, such as machine 172.31.19.184 is over utilizing as well as is under attack as shown in Figure 5.9.

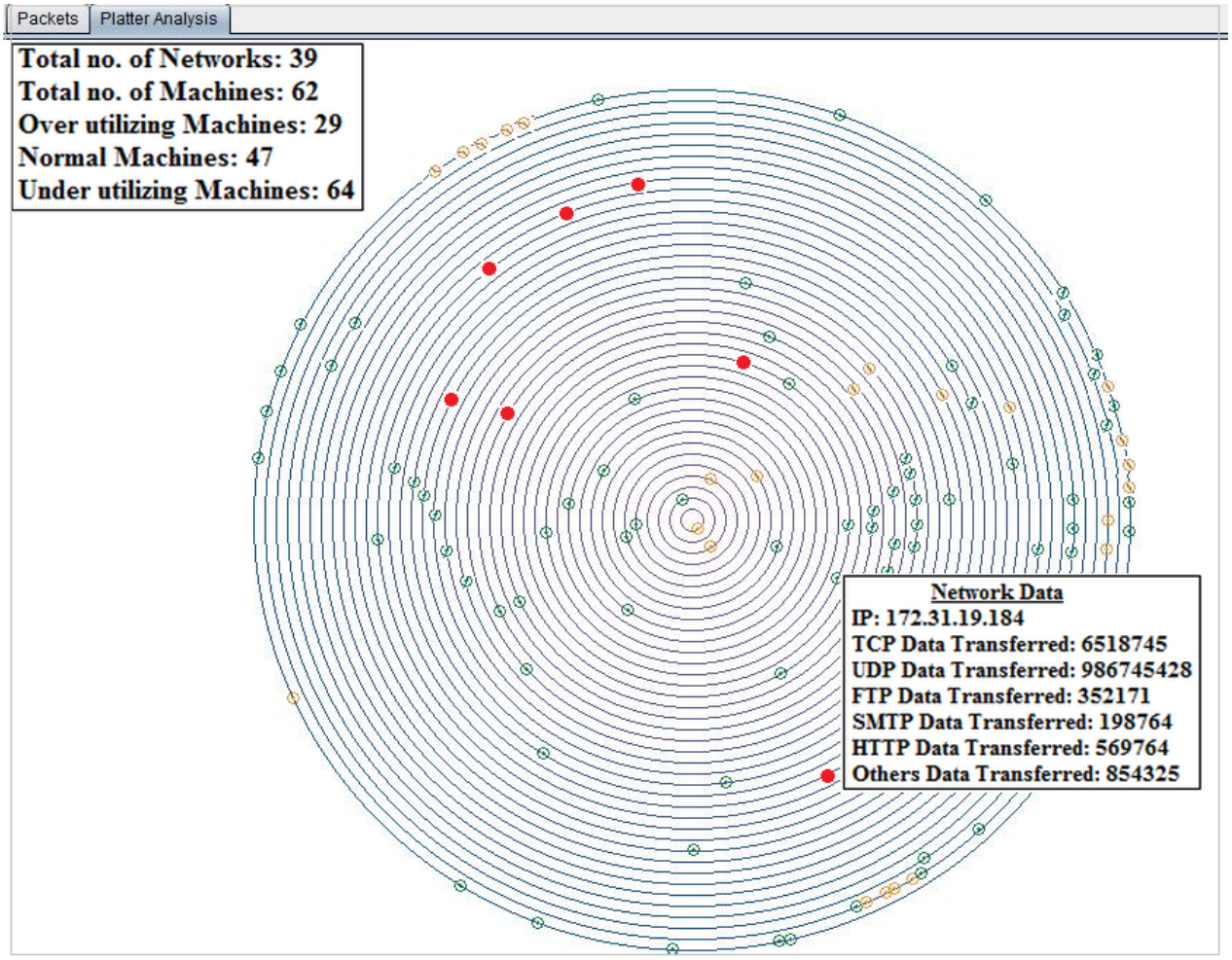


Figure 5.9 Platter view

Parallel Coordinated Analysis: Present section dealt with a particular condition in which an administrator finds an overloaded machine and can easily view related connectivity of a machine, with other network machines with the help of parallel coordinated netmap analysis. In addition, detail of its connectivity with other connected devices/machines can also be viewed through mouse over facility, as shown in the Figure 5.10.

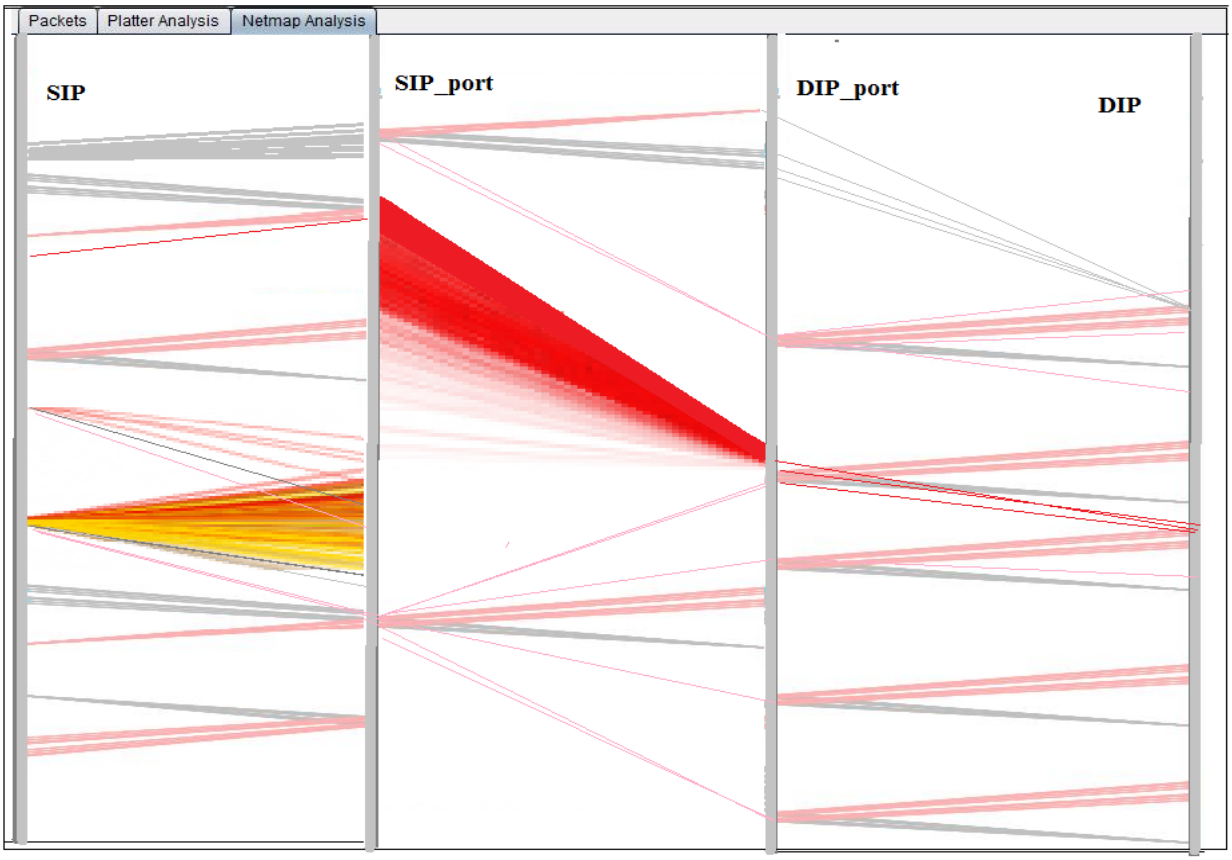


Figure 5.10: Netmap Analysis through parallel coordinated view

5.1.8 Visualizing the under/over utilized network resources

INTVS can visualize the underutilized and over utilized machines as shown in Figure 5.9, there are 29 over utilized machines and 64 underutilization in the CAN.

5.2 Reliability testing of INTVS using Fault tree analysis

Any system can be used only if it is reliable. There are different ways to calculate the reliability of a system. Fault tree analysis is one of the widely used approach for evaluating the reliability of the system. To evaluate the reliability of a system, the reliability of all the subsystems should be known. However, there may always exist uncertainties about the reliability of the subsystems. In the literature [76], [77], [84], [85], [86] references, to dealt with such uncertainties and several researchers have represented the reliability of subsystems by fuzzy numbers. In this section, the fault tree of INTVS is prepared. With the help of this fault tree and fuzzy reliability of the sub components of INTVS, the fuzzy reliability of INTVS is calculated.

In the absence of crisp data set related to reliability analysis of sub components of INTVS fuzzy data can be used to determine the reliability, this work used the same approach of fault tree analysis (FTA) and fuzzy fault tree analysis approach (FFTA). Under FTA the minimal cut set (MCS) approach is used to determine the detail cause of failure of an INTVS. The trapezoidal fuzzy numbers are used to calculate the reliability of INTVS.

Many researchers incorporate system reliability analysis approach to design the network security tools like designing of Intrusion detection system [76], Intrusion prevention system, honeypot, honeynet and firewalls as discussed in Chapter 2. However, the designing of these security tools is dependent on their respective components, therefore, the system reliability analysis is important to be measured.

Allenotor et al. [126] used the fuzzy logic framework to standardize grid resources user-QoS that deal with the ambiguity in linguistic definitions of the Service Level Agreements (SLA) document and to achieve the higher QoS.

5.2.1 Preliminaries

In this section, some basic definitions and arithmetic operations of fuzzy numbers are reviewed. Further, fault tree analysis's basic concepts have also been discussed.

5.2.1.1 Basic definitions

Definition 5.2.1.1.1 [56] [90] the characteristic function μ_A of a crisp set $A \subseteq X$ assigns a value either 0 or 1 to each member in X. This function can be generalized to a function $\mu_{\tilde{A}}$ such that the value assigned to the element of the universal set X fall within a specified range $[0,1]$ i.e. $\mu_{\tilde{A}} : X \to [0,1]$. The assigned values indicate the membership grade of the element in the set A. The function $\mu_{\tilde{A}}$ is called the membership function and the set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)); x \in X\}$ defined by $\mu_{\tilde{A}}$ for each $x \in X$ is called a fuzzy set.

Definition 5.2.1.1.2 [56][90] A fuzzy set $\tilde{A}$, defined on the universal set of real numbers, is said to be a fuzzy number if its membership function has the following characteristics:

(i) $\mu_{\tilde{A}} : \mathbf{R} \to [0,1]$ is continuous

(ii) $\mu_{\tilde{A}}(x) = 0$ for all $x \in (-\infty, a] \cup [d, \infty)$

(iii) $\mu_{\tilde{A}}(x)$ is strictly increasing on $[a,b]$ and strictly decreasing on $[c,d]$

(iv) $\mu_{\tilde{A}}(x) = 1$ for all $x \in [b,c]$, where $a,b,c,d \in \mathbf{R}$

Definition 5.2.1.1.3 a fuzzy number $\tilde{A} = (a, b, c, d)$ is said to be a trapezoidal fuzzy number if its membership function is given by

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-a}{b-a}, & a < x \le b, \\ 1, & b < x \le c, \\ \dfrac{d-x}{d-c}, & c \le x < d, \\ 0, & \text{otherwise} \end{cases},$$

Definition 5.2.1.1.4 the $\alpha-$cut A_α of the fuzzy set $\tilde{A}$ in the universe of discourse X is defined by

$$A_\alpha = \left\{ x_i \middle| \mu_{\tilde{A}}(x_i) \ge \alpha, x_i \in X \right\}$$

Where $\alpha \in [0,1]$.

For example, Figure 5.11 shows a fuzzy number with α-cuts, where $A_\alpha = \left[a_1^{(\alpha)}, a_2^{(\alpha)} \right]$.

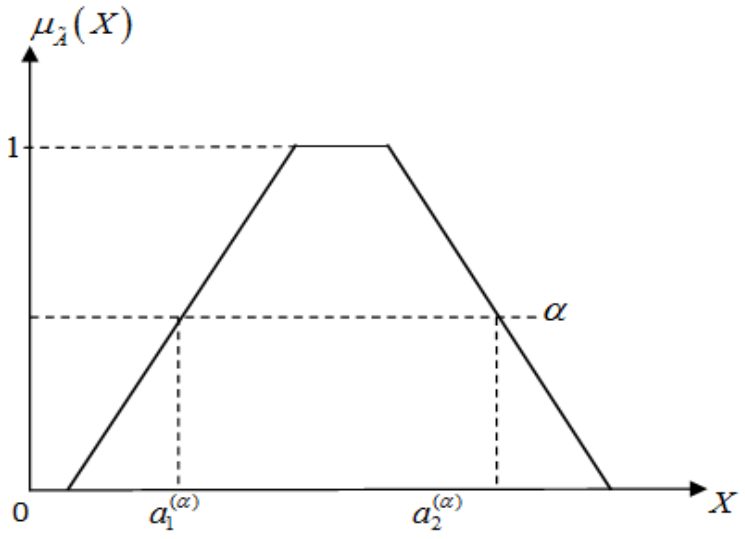


Figure 5.11: A trapezoidal fuzzy number $\tilde{A}$ with α-cuts

5.2.1.2 Arithmetic operations - Arithmetic operations between two trapezoidal fuzzy numbers:

Let $\widetilde{A}_1 = (a,b,c,d)$ and $\widetilde{A}_2 = (e,f,g,h)$ be two trapezoidal fuzzy numbers, then

(i) $\widetilde{A}_1 \oplus \widetilde{A}_2 = (a,b,c,d) \oplus (e,f,g,h) = (a+e, b+f, c+g, d+h)$ (1)

(ii) $\widetilde{A}_1 \Theta \widetilde{A}_2 = (a,b,c,d)\ \Theta\ (e,f,g,h) = (a-h, b-g, c-f, d-e)$ (2)

(iii) $\widetilde{A}_1 \otimes \widetilde{A}_2 \approx (x,y,z,w)$ (3)

Where $x = \min(ae, ah, de, dh), y = \min(bf, bg, cf, cg),$

$z = \max(bf, bg, cf, cg), w = \max(ae, ah, de, dh)$

(iv) $\widetilde{A}_1 \oslash \widetilde{A}_2 = (a/h, b/g, c/f, d/e)$ (4)

In fuzzy number division, in order to avoid 0, we must limit $0 \notin [e,h]$.

Let $A_{1\alpha} = \left[a_{11}^{(\alpha)}, a_{12}^{(\alpha)}\right]$ and $A_{2\alpha} = \left[a_{21}^{(\alpha)}, a_{22}^{(\alpha)}\right]$ be the α-cut of the fuzzy numbers $\widetilde{A}_1$ and $\tilde{A}_2$ respectively, where $\alpha \in [0,1]$, then

(v) $\widetilde{A}_1 \textcircled{\wedge} \tilde{A}_2 = \int_0^1 \alpha \left[a_{11}^{(\alpha)} \wedge a_{21}^{(\alpha)}, a_{12}^{(\alpha)} \wedge a_{22}^{(\alpha)}\right]$ (5)

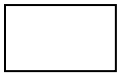
(vi) $\widetilde{A}_1 \textcircled{\vee} \tilde{A}_2 = \int_0^1 \alpha \left[a_{11}^{(\alpha)} \vee a_{21}^{(\alpha)}, a_{12}^{(\alpha)} \vee a_{22}^{(\alpha)}\right]$ (6)

Where '$\textcircled{\wedge}$' and '$\textcircled{\vee}$'are the fuzzy "AND" and "OR" operators of the fuzzy numbers respectively.

5.2.1.3 FTA basic concepts

Fault tree analysis (FTA) is a top down, deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events.

Fault tree symbols (Ebeling, 2000) [33].

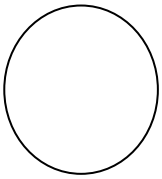


Top Event:

An undesired state of a system caused by an events occurring within the system. Top event is represented by rectangle.

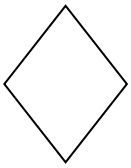
Intermediate Event:

An intermediate event is a fault event, which occurs from a combination of other events via logic gates. Intermediate event is also represented by rectangle.



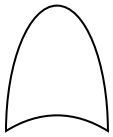
Basic Event:

The circle describes a basic initiating fault event that requires no further development. In other words, it signifies that the appropriate limit of resolution have been reached.



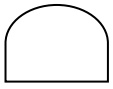
Undeveloped Event:

An event, which is not further developed either because it is of insufficient consequence or because information is unavailable.



OR Gate:

Output fault occurs if at least one of the input faults occurs.



AND Gate:

Output fault occurs if all of the input faults occur.

5.2.2 Minimal cut set approach

Minimal cut sets (mcss) constitute the simplified fault tree through the Boolean operation. To determine the mcss of a fault tree, the tree is first transformed to its equivalent Boolean equations and then either the "top-down" or "bottom-up" substitution method is used. Any fault tree consists of a finite number of mcss, which are unique for the top event. The minimal cut set expression for the top event can be written in the general form,

$$Top = \sum_{i=1}^{k} MCS_i \tag{7}$$

where Top is the top event and MCS_i is the i^{th} minimal cut set and k is the number of MCS_s. Each minimal cut set consists of combination of specific basic events and hence in general n-component minimal cut set can be expressed as

$$MCS_i = \prod_{i=1}^{n} X_i \tag{8}$$

Where X_i is the i^{th} basic event and n is the number of basic events in a minimal cut set.

5.2.3 Composition diagram of INTVS framework

Figure 5.12 is showing the composition of INTVS components, in which computer machine is at the bottom. Above that, there is operating system (Windows, Linux or Ubuntu) along with network device (wired and wireless) drivers. Above that, is JAVA and JRE platform for platform independency? Open sources Winpcap / Libpcap is used to support the capturing module and tokenizing module. Above that, open source libraries are used viz. JCommon-1.0.17, JfreeChart-1.0.14 and jgrapht-JDK 1.6 for visualization module. Above that there is capturing module, followed by tokenizing module, parsing module, and visualization module.

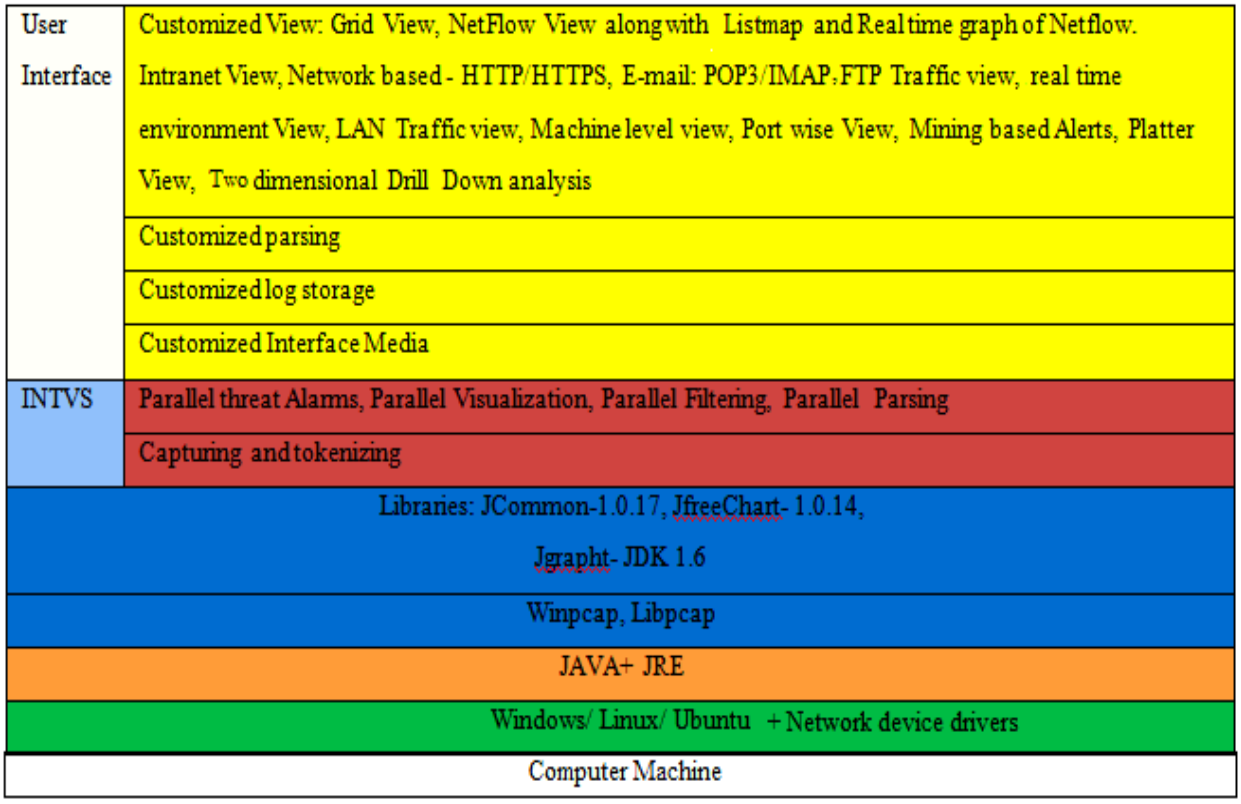


Figure 5.12: Composition Diagram of INTVS

5.2.4 Fuzzy reliability analysis of INTVS

The reliable result of INTVS is dependent upon the file generated by parsing tool and interpretation done by the visualization module. Further, the reliable working of visualization module is dependent upon the visualization middleware, library file, platform, policy and its configuration and availability of hardware resources like processor and RAM.

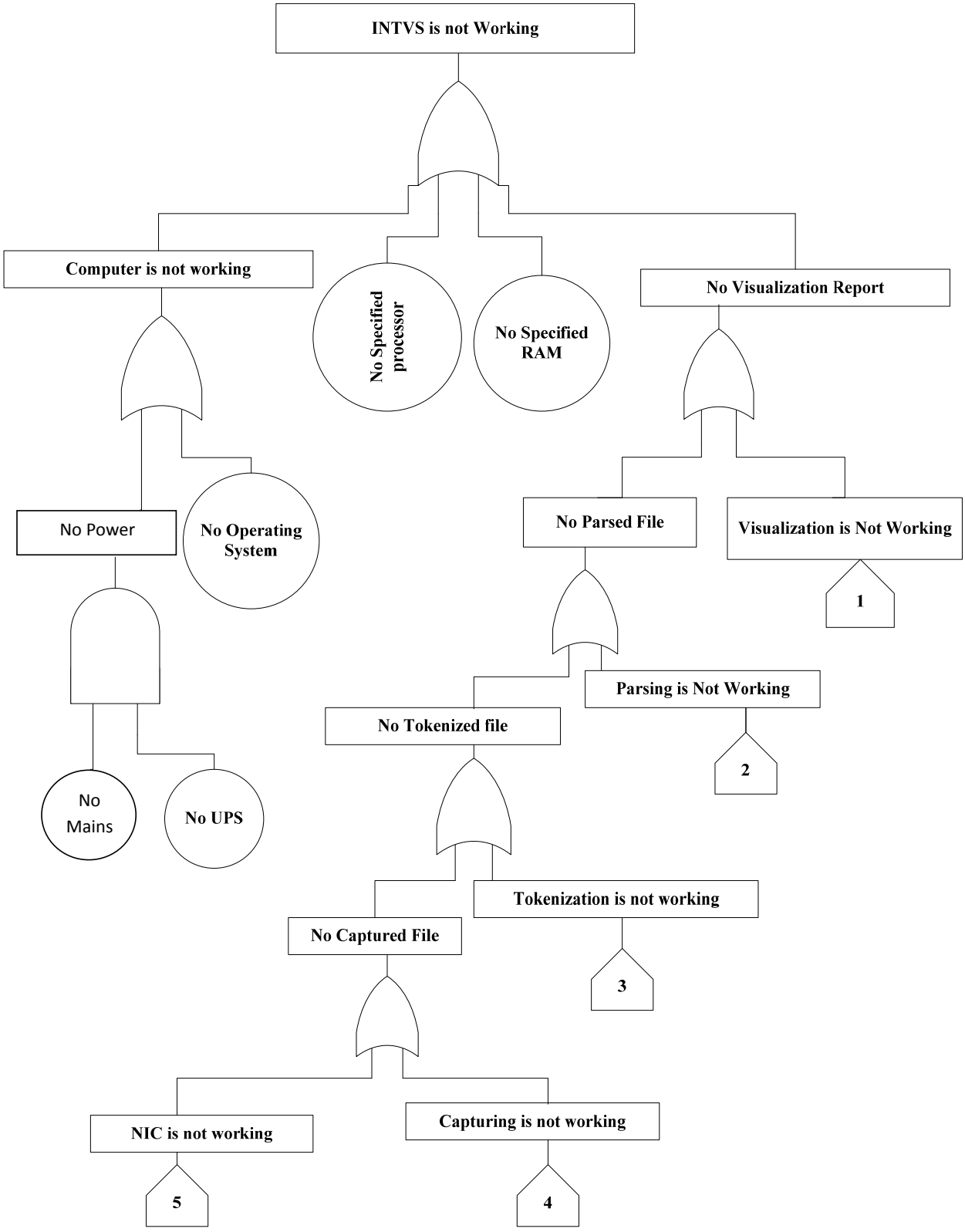
The reliable parsing is dependent upon the reliable tokenized data and reliable parsing module. Further, the working of parsing module is dependent upon the parsing middleware, library file, platform, policy and its configuration and also on the availability of specified processor and RAM.

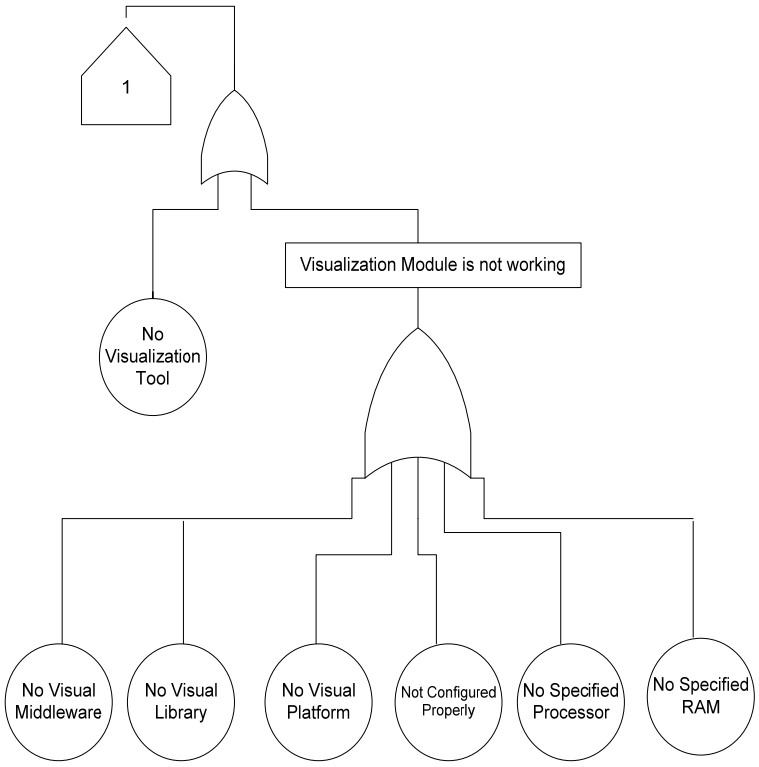
The tokenization is dependent upon the captured file, handed over by the capturing tool. Further, the working of tokenization module is dependent upon the tokenization middleware, library file, platform, policy and its configuration and also on availability of specified processor and RAM.

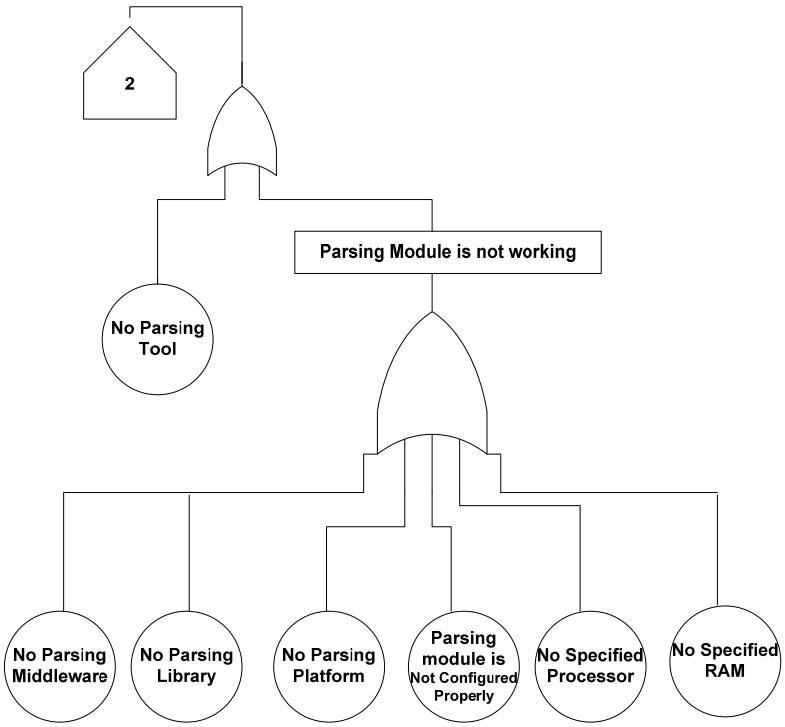
The capturing is dependent upon working of network interface card (NIC) and capturing module. Capturing module is dependent upon the capturing middleware, capturing library file, platform, policy and its configuration and also on availability of specified processor and RAM.

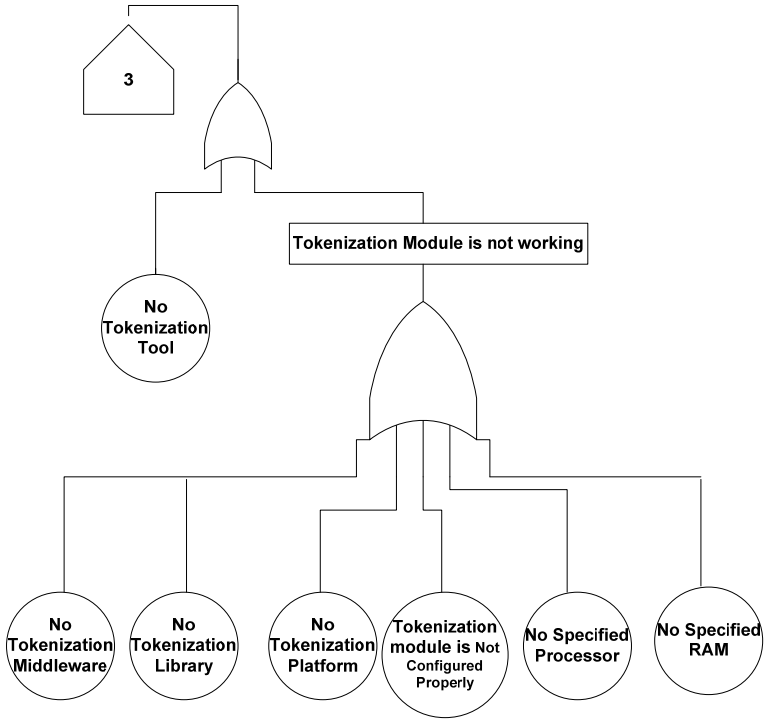
The INTVS is platform independent VizSec (Visualization based security solution) compatible with Windows and Linux like operating systems. To run INTVS on windows, user must have Winpcap, and JRE on their machine and in case of Linux, like systems, user must have libpcap and JRE. To develop new version of INTVS user must have some library files for windows platform, i.e. Jpcap jar file and Jpcap.gz library for Linux. JCommon-10.17, JfreeChart-1.0.14 and jgrapht-jdk1.6 libraries, are also used to develop visualization engine of INTVS.

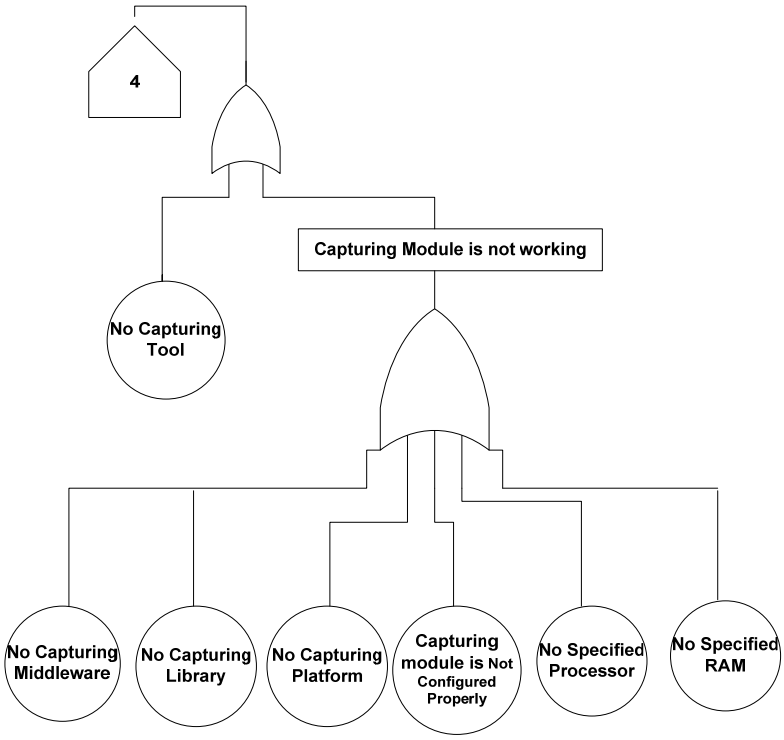
Overall, working of INTVS is depending upon working of computer, working of visualization reporting module and specified RAM and processor. Working of computer is dependent upon operating system and power supply.


INTVS is not Working
Computer is not working
No Specified processor
No Specified RAM
No Visualization Report
No Power
No Operating System
No Parsed File
Visualization is Not Working
1
Parsing is Not Working
No Tokenized file
2
No Mains
No UPS
Tokenization is not working
No Captured File
3
NIC is not working
Capturing is not working
5
4


1
Visualization Module is not working
No Visualization Tool
No Visual Middleware
No Visual Library
No Visual Platform
Not Configured Properly
No Specified Processor
No Specified RAM


2
Parsing Module is not working
No Parsing Tool
No Parsing Middleware
No Parsing Library
No Parsing Platform
Parsing module is Not Configured Properly
No Specified Processor
No Specified RAM


3
Tokenization Module is not working
No Tokenization Tool
No Tokenization Middleware
No Tokenization Library
No Tokenization Platform
Tokenization module is Not Configured Properly
No Specified Processor
No Specified RAM


4
Capturing Module is not working
No Capturing Tool
No Capturing Middleware
No Capturing Library
No Capturing Platform
Capturing module is Not Configured Properly
No Specified Processor
No Specified RAM

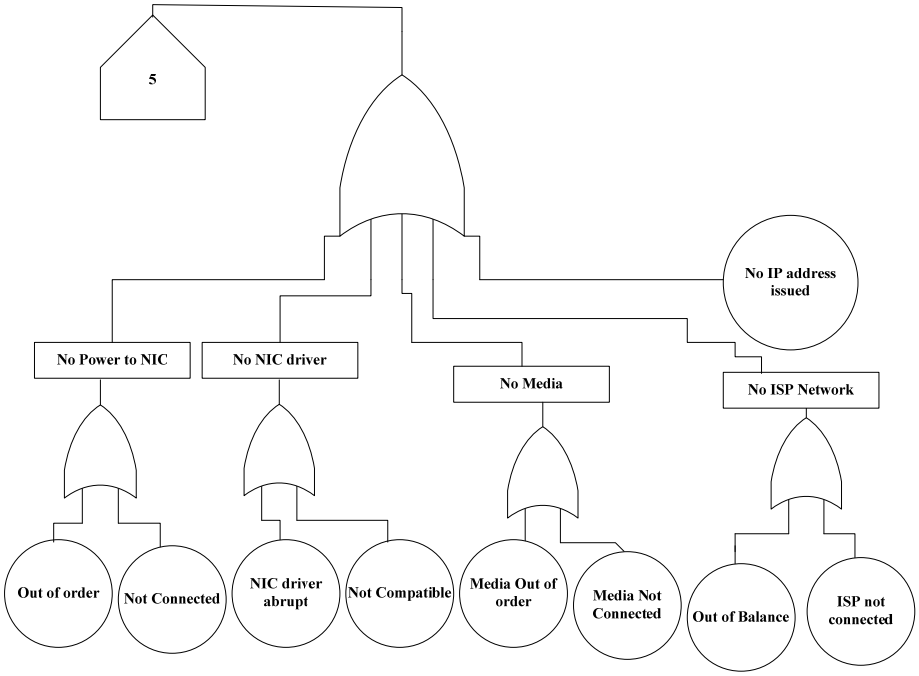


Figure 5.13: Fault tree of INTVS

Table 5.1

Main and Middle events of INTVS

Code	*Fault*	*Code*	*Fault*
F	Failure of INTVS	F_{10}	Failure of media
F_1	Computer is not working	F_{11}	Failure of network /ISP
F_2	No power	F_{12}	Failure of capturing
F_3	Failure of visualized file built	F_{13}	Failure of capturing module
F_4	Failure of parsed file built	F_{14}	Failure of tokenizing

F_5	Failure of tokenized file built	F_{15}	Failure of tokenizing module
F_6	Failure of captured file built	F_{16}	Failure of parsing
F_7	NIC module is not working	F_{17}	Failure of parsing module
F_8	Failure of power to NIC	F_{18}	Failure of visualization
F_9	Failure of NIC driver	F_{19}	Failure of visualization module

Table 5.2

Basic events of INTVS

Code	*Fault*	*Code*	*Fault*
V_1	No mains	V_{18}	No platform for capturing
V_2	No UPS	V_{19}	No configuration properly of capturing
V_3	No Specified OS	V_{20}	No Tokenization tool
V_4	No Specified processor	V_{21}	No tokenization middleware
V_5	No Specified RAM	V_{22}	No lib for tokenization
V_6	NIC out of order	V_{23}	No platform for tokenization
V_7	NIC not connected	V_{24}	No configuration properly of tokenization
V_8	NIC driver abrupt	V_{25}	No parsing tool

V_9	NIC driver is not compatible	V_{26}	No parsing middleware
V_{10}	Media is out of order	V_{27}	No lib for parsing
V_{11}	Media is not connected	V_{28}	No platform for parsing
V_{12}	ISP not connected due out of balance	V_{29}	No configuration properly of parsing
V_{13}	ISP is not connected	V_{30}	No Visualization tool
V_{14}	No IP address	V_{31}	No Visualization middleware
V_{15}	No capturing tool	V_{32}	No lib for Visualization
V_{16}	No capturing middleware	V_{33}	No platform for Visualization
V_{17}	No lib for capturing	V_{34}	No configuration properly of Visualization

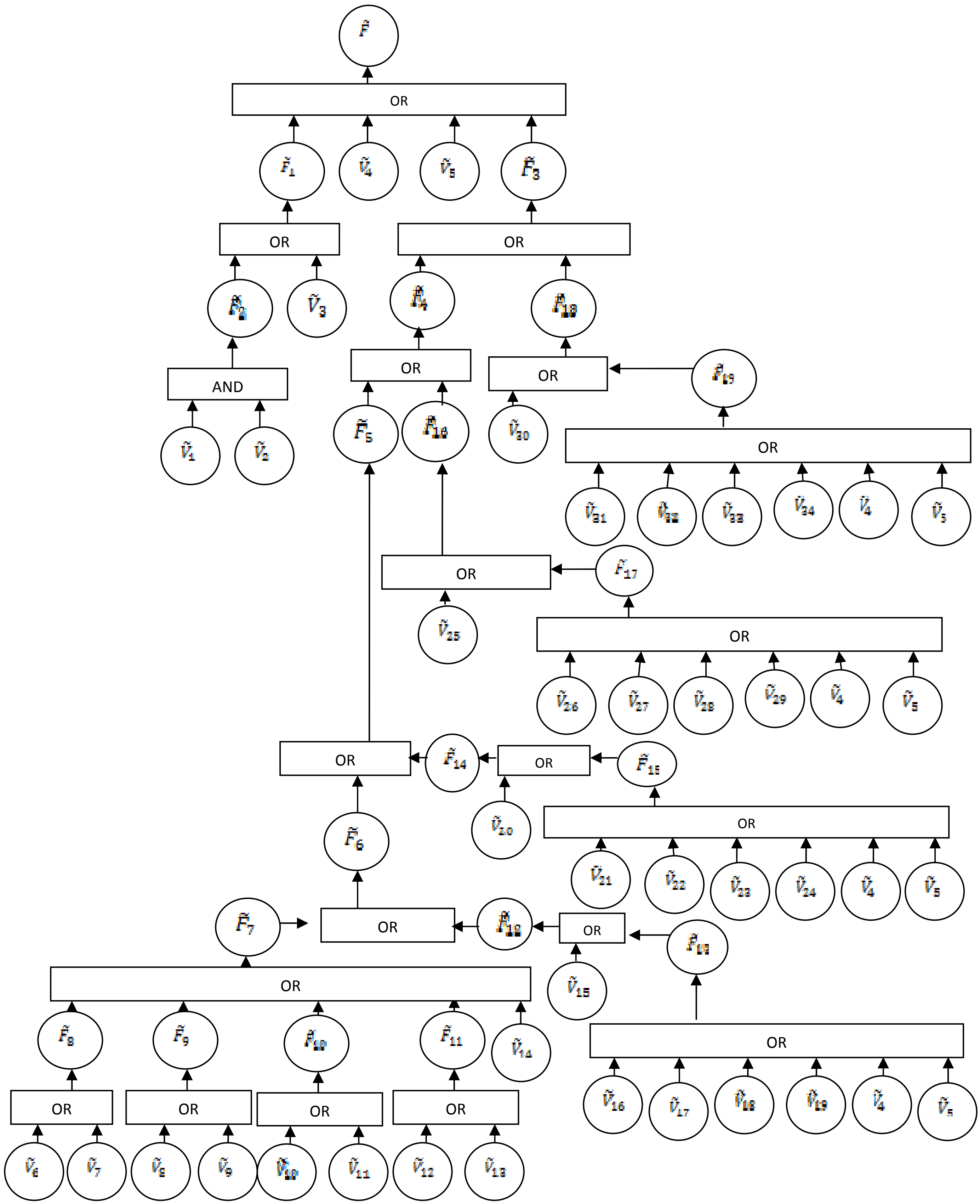


Figure 5.14: Fuzzy fault tree of INTVS

Evaluation of fuzzy reliability of INTVS:

(i) First, top event of INTVS (i.e. INTVS failure) in Boolean function using the fault tree shown in Figure 5.13 to Figure 5.14 and algebraic properties of sets as follows:

$$F = \left\{ \begin{array}{l} V_1 \cap V_2 \cup V_3 \cup V_4 \cup V_5 \cup V_{25} \cup V_{30} \cup V_{31} \cup V_{32} \\ \cup V_{33} \cup V_{34} \cup V_4 \cup V_5 \cup V_{20} \cup V_{26} \cup V_{27} \cup V_{28} \\ \cup V_{29} \cup V_4 \cup V_5 \cup V_{14} \cup V_6 \cup V_7 \cup V_8 \cup V_9 \cup \\ V_{10} \cup V_{11} \cup V_{12} \cup V_{13} \cup V_{15} \cup V_{16} \cup V_{17} \cup V_{18} \\ \cup V_{19} \cup V_4 \cup V_5 \cup V_{21} \cup V_{22} \cup V_{23} \cup V_{24} \cup V_4 \cup V_5 \end{array} \right\}$$

(ii) Using existing top event (i.e. INTVS failure) may be expressed in the term of minimal cut sets as follows:

$$F = \left\{ \begin{array}{l} V_1 \cap V_2 \cup V_3 \cup V_4 \cup V_5 \cup V_6 \cup V_7 \cup V_8 \cup V_9 \cup \\ V_{10} \cup V_{11} \cup V_{12} \cup V_{13} \cup V_{15} \cup V_{16} \cup V_{17} \cup V_{18} \\ \cup V_{19} \cup V_{20} \cup V_5 \cup V_{21} \cup V_{22} \cup V_{23} \cup V_{24} \cup V_{25} \\ \cup V_{26} \cup V_{27} \cup V_{28} \cup V_{29} \cup V_{30} \cup V_{31} \cup V_{32} \cup \\ V_{33} \cup V_{34} \end{array} \right\}$$

(iii) Now, fuzzy reliability, $\tilde{R}$, of the INTVS can be obtained by using $\tilde{R} = 1 \Theta \tilde{R}_F$, where

$$\tilde{R}_F = (1\Theta((1\Theta(\tilde{R}_{V_1} \otimes \tilde{R}_{V_2})) \otimes (1\Theta\tilde{R}_{V_3}) \otimes (1\Theta\tilde{R}_{V_4}) \otimes (1\Theta\tilde{R}_{V_5}) \otimes (1\Theta\tilde{R}_{V_6}) \otimes (1\Theta\tilde{R}_{V_7}) \otimes$$

$$(1\Theta\tilde{R}_{V_8}) \otimes (1\Theta\tilde{R}_{V_9}) \otimes (1\Theta\tilde{R}_{V_{10}}) \otimes (1\Theta\tilde{R}_{V_{11}}) \otimes (1\Theta\tilde{R}_{V_{12}}) \otimes (1\Theta\tilde{R}_{V_{13}}) \otimes (1\Theta\tilde{R}_{V_{14}}) \otimes$$

$$(1\Theta\tilde{R}_{V_{15}}) \otimes (1\Theta\tilde{R}_{V_{16}}) \otimes (1\Theta\tilde{R}_{V_{17}}) \otimes (1\Theta\tilde{R}_{V_{18}}) \otimes (1\Theta\tilde{R}_{V_{19}}) \otimes (1\Theta\tilde{R}_{V_{20}}) \otimes (1\Theta\tilde{R}_{V_{21}}) \otimes$$

$$(1\Theta\tilde{R}_{V_{22}}) \otimes (1\Theta\tilde{R}_{V_{23}}) \otimes (1\Theta\tilde{R}_{V_{24}}) \otimes (1\Theta\tilde{R}_{V_{25}}) \otimes (1\Theta\tilde{R}_{V_{26}}) \otimes (1\Theta\tilde{R}_{V_{27}}) \otimes (1\Theta\tilde{R}_{V_{28}}) \otimes$$

$$(1\Theta\tilde{R}_{V_{29}}) \otimes (1\Theta\tilde{R}_{V_{30}}) \otimes (1\Theta\tilde{R}_{V_{31}}) \otimes (1\Theta\tilde{R}_{V_{32}}) \otimes (1\Theta\tilde{R}_{V_{33}}) \otimes (1\Theta\tilde{R}_{V_{34}})) \qquad (1)$$

5.2.4.1 *Data*: After burning INTVS on live distribution media, circulated among 1000 Internet users in the campus, after a month's experience with INTVS live distribution a survey was conducted. Response of first 100 respondents recorded w.r.t. each event failure, an average of data values taken and used as given shown in Table 5.3.

Table 5.3: The possible range of bottom event failure

Failure possibility	a_i	b_i	c_i	d_i
$\tilde{V}_1$	0.040	0.07	0.08	0.10
$\tilde{V}_2$	0.010	0.030	0.040	0.050
$\tilde{V}_3$	0.07	0.085	0.0925	0.1
$\tilde{V}_4$	0.2	0.35	0.425	0.5
$\tilde{V}_5$	0.5	0.65	0.725	0.8
$\tilde{V}_6$	0.001	0.0015	0.00175	0.002
$\tilde{V}_7$	0.001	0.0015	0.00175	0.002
$\tilde{V}_8$	0.001	0.0015	0.00175	0.002
$\tilde{V}_9$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{10}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{11}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{12}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{13}$	0.2	0.35	0.425	0.5
$\tilde{V}_{14}$	0.5	0.65	0.725	0.8

$\tilde{V}_{15}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{16}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{17}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{18}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{19}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{20}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{21}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{22}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{23}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{24}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{25}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{26}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{27}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{28}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{29}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{30}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{31}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{32}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{33}$	0.001	0.0015	0.00175	0.002
$\tilde{V}_{34}$	0.001	0.0015	0.00175	0.002

5.2.4.2 Fuzzy reliability evaluation of INTVS

The fault tree of INTVS, shown in Figure 5.13, is then converted into simplified fuzzy fault tree of INTVS as shown in Figure 5.14. Using arithmetic operation of trapezoidal fuzzy numbers, data presented in Table 5.4 and minimal cut sets.

So, reliability of the INTVS is '$\widetilde{R}$' = $1 - P(\widetilde{F})$

$$\tilde{R} = (0.8623, 0.9589, 0.9809, 0.9927)$$

5.2.4.3 Results and discussion

The obtained results can be explained as follows:

(i) Reliability of INTVS lies between **0.8623 and 0.9927**. The most possible value of reliability of INTVS lies between **0.9589 and 0.9809** with the degree **1** of membership; the pessimistic values are **0.8623 and 0.9927** with the degree **0** of membership.

(ii) Values of $\mu_{\tilde{R}}(r)$ corresponding to different values of crisp reliabilities r, $r \in [0.8623 \text{ and } 0.9927]$, can be evaluated as follows:

$$\mu_{\tilde{R}}(r) = \begin{cases} (r - 0.8623) + 0.0966, & 0.8623 \leq r < 0.9589 \\ 1, & 0.9589 \leq r \leq 0.9809 \\ (0.9927 - r) + 0.0118, & 0.9809 < r \leq 0.9927 \\ 0, & \text{otherwise} \end{cases}$$

(iii) The results obtained by using fuzzy reliability and crisp reliability (using the data values $(b_i + c_i)/2$ of Table 5.4) approaches for different membership values are shown in Table 5.4.

The membership values representing the fuzzy reliability of an INTVS are shown in Table 5.4. The top event of INTVS reliability evaluated by using the fault tree is same for all the presumption level and a single real value while using fuzzy fault tree are intervals. It is clear that the use of fuzzy reliability approach over crisp reliability approach has a great advantage in assessing the top event of INTVS reliability when uncertainty exists in dealing with the ambiguity of failure data and the uncertainties associated with the INTVS system modeling.

Table 5.4: Results

Results of the existing methods corresponding to different membership values

α	Fuzzy reliability				Crisp reliability
	a	*b*	*c*	*d*	
1	0.9589	0.9699	0.9699	0.9809	0.9699
0.9	0.94924	0.9699	0.9699	0.98208	0.9699
0.8	0.93958	0.9699	0.9699	0.98326	0.9699
0.7	0.92992	0.9699	0.9699	0.98444	0.9699
0.6	0.92026	0.9699	0.9699	0.98562	0.9699
0.5	0.9106	0.9699	0.9699	0.9868	0.9699
0.4	0.90094	0.9699	0.9699	0.98798	0.9699
0.3	0.89128	0.9699	0.9699	0.98916	0.9699
0.2	0.88162	0.9699	0.9699	0.99034	0.9699
0.1	0.87196	0.9699	0.9699	0.99152	0.9699
0	0.8623	0.9699	0.9699	0.9927	0.9699

Conclusions

In this chapter, reports, results and reliability of INTVS have been discussed. We answered almost all pertinent queries in this chapter like: Is INTVS is able to work in online, offline mode, able to detect abnormal behaviour of a machine/node, able to visually report machines that deviate from network policy and generate reports/files for forensic analysis of data. Further fuzzy fault tree of INTVS was constructed and using it, fuzzy reliability of INTVS was evaluated.

Chapter 6

Conclusions and Future Scope

This chapter presents conclusions and future scope. It concludes the thesis by highlighting major contributions of this work. To enhance this work further, research directions are discussed in future scope.

6.1 Contributions of the research work

This thesis contributed in the following ways:

a) A systematic review of the existing approaches towards visualization based network traffic analysis have been discussed and compared.

b) Need for Integrated network traffic visualization system (INTVS) framework for network traffic threat detection and reporting in real time till seventh layer of OSI model have been identified.

c) A visualization framework INTVS has been proposed, designed, implemented and test results reported.

d) INTVS is designed using component based architecture, in which a user can easily add or remove new components according to the requirements.

e) INTVS support real time as well as offline network traffic analysis.

f) INTVS can capture data from wired as well as wireless media.

g) INTVS is portable, it works on Windows (Winpcap and Jpcap) as well as Linux (libpcap and Jpcap.gz library) and its variants.

h) INTVS is having its own capturing, tokenizing, parsing, visualizing and abnormal activity detection modules, helpful in detecting and reporting threats.

i) INTVS can visualize the network though Grid view based on data classification.

j) INTVS can summarize the network health with the help of Listmap view and Platter view visualizations schemes. Both Listmap view and Platter view visualization schemes are the outcomes of INTVS framework.

k) Listmap view can summarize usage of statistics of network resources along with bandwidth consumption by a machine w.r.t. application layer protocol and Netflow.

l) Platter visualization can highlight the network which may be under attack based on resources consumptions.

m) Platter visualization can highlight the machines which are violating the network policy.

n) Two dimensional analysis facility (interactive) can help user to understand reasons behind the malicious node(s).

o) Two dimensional analysis facility also helps to know, which application layer protocol (HTTP/SMTP/FTP) is consuming maximum bandwidth.

p) Parallel coordinated visualization can helps to know linkage between ingress and egress flow of a network traffic.

q) INTVS is economically viable as it is developed using open source tools and technologies.

6.2 Future scope

INTVS can be extended to monitor more than one local area network operating under different Internet service providers at different locations. Client agents can be developed and deployed at various locations, which send traffic anomalies to server for further correlation analysis and report generation in visual form. Also these client agents would help individual machine user to get reports about his/her machine being part of a Botnet or not.

INTVS can be extended to predict attack vectors based on traffic learning scenarios and can make dynamic insertion of rules in the network policy. Further, INTVS can be enhanced to produce three dimensional graphics visualization with help of Platter visualization, instead of parallel coordinated graphs.

As more and more dependence is increasing on mobile devices, INTVS interface for android and iOS can be developed to make administrator aware about traffic anomalies and network health reports via mobile devices.

References:

1) Swing E (1998) Flodar: Flow Visualization of Network Traffic. Computer Graphics and Applications, IEEE, 18(5):6–8.

2) Estrin D, Handley M, Heidermann J, McCanne S, Xu Y, Yu H (2000) Network visualization with Nam, The VINT network administrator. IEEE, Computer Society.

3) Lakkaraju K, Yurcik W, Lee A J (2004) NVisionIP: Netflow visualizations of system state for security situational awareness. ACM Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC ’04, 65–72.

4) Ball R, Fink G A, North C (2004) Home-centric visualization of network traffic for security administration. ACM, Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC ’04, 55–64.

5) Yin X, Yurcik W, Treaster M (2004) VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness. ACM, Workshop on Visualization and Data Mining for Computer Security, VizSEC/DMSEC'04, DOI: 1581139748/04/0010.

6) Kim S S, Reddy A L N (2005) NetViewer: A Network Traffic and Analysis Tool. USENIX, 19th Large Installation System Administration Conference (LISA ’05), (19): 185-196.

7) Estan C, Magin G (2005) Interactive Traffic Analysis and Visualization with Wisconsin Netpy. USENIX, 19th Large Installation System Administration Conference, LISA 05(19):177-184.

8) Fink G A, Muessig P, North C (2005) Visual Correlation of Host Processes and Network Traffic. IEEE, Workshop on Visualization for Computer Security, VizSEC 05, 11-19.

9) Abdullah K, Lee C P, Conti G, Copeland J A, Stasko J (2005) IDS RainStorm: Visualizing IDS Alarms. IEEE, Workshop on Visualization for Computer Security, VizSEC 05, 1-10.

10) Ren P, Gao Y, Li Z, & Watson B (2005) IDGraphs: Intrusion Detection and Analysis Using Histographs. IEEE, Workshop Visualization for Computer Security (39-46).

11) Goodall J, Lutters W, Rheingans P and Komlodi A (2005) Preserving the Big Picture: Visual Network Traffic Analysis with TNV. IEEE, Workshop Visualization for Computer Security (VizSEC ’05), 47-54.

12) http://www.rumint.org. Accessed on 20 January 2013.

13) http://afterglow.sourceforge.net/. Accessed on 20 January 2013.

14) http://www.secviz.org/node/89. Accessed on 20 January 2013.

15) Reil J P V, Irwin B (2006) InetVis, A Visual Tool for Network Telescope Traffic Analysis. ACM, International conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH 2006, 85-89.

16) Lee C P and Copeland J A (2006) Flowtag: A Collaborative Attack-Analysis, Reporting, and Sharing Tool for Security Researchers. ACM, workshop on Visualization for Computer Security (VizSEC), ACM Press, 2006, 103–108.

17) Oberheide J, Goff M, Karir M (2006) Flamingo: Visualizing internet traffic. IEEE, In Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium, 150– 161.

18) http://nfsen.sourceforge.net/#mozTocId201388. Accessed on 20 January 2013.

19) Godinho I, Meiguins B, Gonçalves A, Carmo C, Garcia M, Almeida L, Lourenço R (2007) PRISMA – A Multidimensional Information Visualization Tool Using Multiple Coordinated Views. IEEE, 11th International Conference on Information Visualization (IV '07), 23-32.

20) Makanju A, Brooks S, Zincir-Heywood A N, and Milios E (2008) LogView: Visualizing Event Log Clusters. Sixth Annual Conference on Privacy, Security and Trust IEEE, Computer Society, 99-108.

21) Frei A and Rennhard M (2008) Histogram Matrix: Log File Visualization for Anomaly Detection. The Third International Conference on Availability, Reliability and Security IEEE, Computer Society, 610-617.

22) Taylor T, Paterson D, Glanfield J, Gates C, Brooks S, McHugh J (2009) FloVis: Flow Visualization System. IEEE, Cybersecurity Applications & Technology Conference for Homeland Security, 186-198.

23) Jiawan Z, Peng Y, Liangfu L and Lei C (2009) NetViewer: A Visualization Tool for Network Security Events. International Conference on Networks Security, Wireless Communications and Trusted Computing, IEEE, Computer Society 434-437.

24) Allen M, McLachlan P (2009) NAV- Network Analysis Visualization. University of British Columbia, [Online, 29 May 2009].

25) Goodall J R, Sowul M (2009) VIAssist: Visual Analytics for Cyber Defense. IEEE, Technologies for Homeland Security, HST '09, 143-150.

26) Jiawan Z, Liang L, Liangfu L, Ning Z (2008) A Novel Visualization Approach for Efficient Network Scans Detection, IEEE, International Conference on Security Technology, SECTECH '08, 23-26.

27) Lu L F, Zhang J W, Huang M L and Fu L (2010) A new concentric-circle visualization of multi-dimensional data and its application in network security. Journal of Visual Languages and Computing (21), 194–208.

28) Osborne G, Turnbull B and Slay J (2010) The 'Explore, Investigate and Correlate' (EIC) conceptual framework for digital forensics Information Visualisation, IEEE, International Conference on Availability, Reliability, and Security, ARES '10, 629-634.

29) Osborne G and Slay J (2011) Digital Forensics Infovis-An implementation of a process for visualisation of digital evidence. Sixth International Conference on Availability, Reliability and Security, IEEE Computer Society, 196-201.

30) Osborne G, Turnbull B and Slay J (2012) Development of InfoVis Software for Digital Forensics. IEEE, 36th International Conference on Computer Software and Applications Workshops, IEEE Computer Society, 213-217.

31) Shurkhovetskyy G, Bahey A, and Ghoniem M (2012) Visual Analytics for Network Security. IEEE, Symposium on Visual Analytics Science and Technology, Seattle, WA, USA: IEEE, 301-302.

32) Donahue J, Paturi A and Mukkamala S (2013) Visualization Techniques for Efficient Malware Detection, ISI 2013. Seattle, Washington, USA: IEEE, 289-291.

33) Ebeling C E (2000) an Introduction to Reliability and Maintainability Engineering, Tata McGraw-Hill Publishing Company Limited, New Delhi.

34) Becker R A, Eick S G and Wilks A R (1995) Visualizing network data. IEEE Trans. Vis. Comput. Graph., 1, pp. 16–21

35) Shneiderman B (1996) The eyes have it: A task by data type taxonomy of information visualizations. IEEE, Symposium on Visual Languages, 336-343.

36) Singh M P, Subramanian N and Rajamenakshi. (2009) Visualization of Flow Data Based on Clustering Technique for Identifying Network Anomalies. IEEE, Symposium on Industrial Electronics and Applications, ISIEA 2009, 973-978.

37) Ahmad I, Abdullah A B and Alghamdi A S (2009) Application of Artificial Neural Network in Detection of Probing Attacks. IEEE, Symposium on Industrial Electronics and Applications ISIEA 2009, 557-562.

38) Kuper P and Stanley M (2005) The State of Security. Published by the IEEE Computer Society 1540-7993/05, 2005 IEEE, IEEE Security & Privacy.

39) Golnabi K, Min R K, Khan L and Al-Shaer E (2006) Analysis of Firewall Policy Rules Using Data Mining Techniques. IEEE, 10th IEEE/IFIP, Network Operations and Management Symposium, NOMS'2006, 305 – 315.

40) Vaarandi R (2009) Real-time classification of IDS alerts with data mining techniques. IEEE, Military Communications Conference, MILCOM 2009, 1-7.

41) Creese S, Goldsmith M, Moffat N, Happa J and Agrafiotis I (2013) CyberVis: Visualizing the Potential Impact of Cyber Attacks on the Wider Enterprise. IEEE, International Conference on Technologies for Homeland Security, HST'2013, 73-79.

42) Booch G, Rumbaugh J and Jacobson I (1999) Unified Modeling Language User Guide. Addison Wesley.

43) Jacobson I, Booch G and Rumbaugh J (1999) The Unified Software Development Process, Pearson Education.

44) Devamalar P M, Bai V T, Murali N and Srivatsa S K (2008) Visualization and Construction of Real Time Web Centric Intelligent Health Care Diagnostic System Using UML. CCECE/CCGEI Niagara Falls. Canada: IEEE 501-506.

45) Dwyer T, Marriott K, Schreiber F, Stuckey P J, Woodward M and Wybrow M (2008). Exploration of Networks Using Overview + Detail with Constraint-based Cooperative Layout. IEEE Transactions on Visualization and Computer Graphics, 14, 1293-1300.

46) Hu S X and Shan T C (2010) Designing Resource Oriented Architecture in UML - A Case Study on Smart Grid Home Area Network (HAN). IEEE, 6th World Congress on Services (154-155).

47) Jakimi A and Koutbi M E (2009) An Object-Oriented Approach to UML Scenarios Engineering and Code Generation. International Journal of Computer Theory and Engineering, 1 (1), 35-41.

48) Jong G D (2002) A UML- Based Design Methodology for Real- Time and Embedded Systems. Design, Automation and Test in Europe Conference and Exhibition, IEEE Computer Society.

49) Ming H, Hong W and Luoming M (2003) Solution and Architecture for Integrated Network Management, Proceedings of ICCT2003, 1650-1654.

50) Kukkala P, Helminen V, Hannikainen M and Hamalainen T D (2004) UML 2.0 Implementation of an Embedded WLAN protocol, IEEE, 1158-1162.

51) Ray H T, Vemuri R and Kantubhukta H R (2005) Toward an Automated Attack Model for Red Teams. IEEE, Security and Privacy, 18-25.

52) Rumbaugh J, Jacobson I and Booch G (2005) Unified Modeling Language Reference Manual (Vol. 2nd Edition), Pearson Education.

53) Wenhui S, Feng L, Gang D and Jinyu Z (2007) Formal Analysis of the VPN Service Management System. Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE Computer Society, 493-497.

54) Lamm S E and Reed D A (1996) Real-time geographic visualization of World Wide Web traffic. Fifth Int. World Wide Web Conf., 1996.

55) Lipsinky Z (2009) UML-based reliability modeling of network services, a UDP Echo service case study. Fourth International Conference on Dependability of Computer Systems, IEEE, Computer Society, 50-57.

56) Zadeh L A (1965) Fuzzy sets. Information and Control, Vol. 8, No. 3, 338–353.

57) Eick S G and Wills G J (1993) Navigating large networks with hierarchies. Proc. IEEE, Conference on Visualization, pp. 204–210.

58) Singer D (1990) A fuzzy set approach to fault tree and reliability analysis. Fuzzy Sets and Systems, Vol. 34, No. 2, 145-155.

59) Cox K C and Eick S G (1995) 3D displays of internet traffic. Information Visualization Symposium.

60) Cox K C, Eick S G and He T (1996). 3D geographic network displays. Sigmod Rec., 25, (4), pp. 50–54.

61) Eick S G (1996) Aspects of network visualization. IEEE, Comput. Graph. App., 16, pp. 69–72.

62) Melissargos G and Pu P (1999) Conceptualizing bandwidth allocation in network management. Proceeding, Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM'99), (ACM Conference on Information and Knowledge Management (CIKM'99), pp. 62–69.

63) Liang G S and Wang M J J (1993) Fuzzy fault tree analysis using failure possibility. Microelectronics Reliability, Vol. 33, No. 4, 583-597.

64) Chen S M (1994) Fuzzy system reliability analysis using fuzzy number arithmetic operations. Fuzzy Sets and Systems, Vol. 64, No. 1, 31-38.

65) Huffaker B, Nemeth E and Claffy K (1999) Otter: a general-purpose network visualization tool. Proc. 9th Annual Conf. of the Internet Society (INET'99).

66) Lin C T and Wang M J (1997) Hybrid fault tree analysis using fuzzy sets. Reliability Engineering and System Safety, Vol. 58, 205-213.

67) Pan H S and Yun W Y (1997) Fault tree analysis with fuzzy gates. Computers and Industrial Engineering, Vol. 33, No. 3-4, 569-572.

68) Chanda R S and Bhattacharjee P K (1998) A reliability approach to transmission expansion planning using fuzzy fault-tree model. Electric Power Systems Research, Vol. 45, No. 2, 101-108.

69) Cheswich B, Burch H and Branigan S (2000) Mapping and visualizing the Internet. Proc. USENIX Technical Conf., pp. 1–12.

70) Chang S Y, Lin C R and Chang C T (2002) A fuzzy diagnosis approach using dynamic fault trees. Chemical Engineering Science, Vol. 57, No. 15, 2971-2985.

71) Chen S J and Chen S M (2003) Fuzzy risk analysis based on similarity measure of generalized fuzzy numbers. IEEE Transactions on Fuzzy Systems, Vol. 11, No.1, 45-56.

72) Bai X and Asgarpoor S (2004) A Fuzzy-Based Approach to Substation Reliability Evaluation. Electric Power Systems Research, Vol. 69, No. 2-3, 197-204.

73) Conti G (2005) Countering Network Level Denial of Information Attacks Using Information Visualization. A Dissertation Presented to The Academic Faculty by Gregory John, Conti Georgia Institute of Technology.

74) Inselberg A (1997) Multidimensional Detective. IEEE Proceedings of Information Visualization '97, 100-107.

75) Wegman E (1990) Hyper dimensional Data Analysis Using Parallel Coordinates. Journal of the American Statistical Association, 85:411, 664-675.

76) Wang G, Hao J, Ma J and Huang L (2010) A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Systems with Applications, Vol.37, 6225–6232.

77) Al-Kuwaiti M, Kyriakopoulos N and Hussein S (2006) Network Dependability, Fault-tolerance, Reliability, Security, Survivability: A Framework for Comparative Analysis. 1-4244-0272-7/06/$20.00 C2006 IEEE 282-287.

78) Verma A K, Srividya A and Kumar R H M (2002) A framework using uncertainties in the composite power system reliability evaluation. Electric Power Components and Systems, Vol. 30, 679-691.

79) Volkanovski A, Cepin M and Mavko B (2009) Application of the fault tree analysis for assessment of power system reliability. Reliability Engineering and System Safety, Vol. 94, No. 6, 1116-1127.

80) Li Y and Song K (2011) Application of quantitative fault tree analysis based on fuzzy synthetic evaluation to software development for space camera. IEEE, 978-1-4577-0536-6/11, 2821-2824.

81) Rong W and Xin D (2010) Application of Fuzzy Fault Tree Analysis of Burning and Blasting of LPG Tank. IEEE, 1093-1096.

82) Goldring T (2004) Scatter (and other) Plots for visualizing user profiling data and network traffic. VizSEC/DMSEC '04.

83) Oetiker T (1998) MRTG – the multi router traffic grapher. Conf. Proc. USENIX LISA'98.

84) Kumar M, Yadav S P and Kumar S (2012) Reliability Analysis of Computer Sceuroty System based on Intuitionistic Fuzzy Fault Tree. Advanced Materials Research, 3495-3502.

85) Kumar A, Yadav S P and Kumar S (2006) Fuzzy reliability of a marine power plant using interval valued vague sets. International Journal of Applied Science and Engineering 4 (1), 71-82.

86) Kumar A, Yadav S P and Kumar S (2008) Fuzzy system reliability using different types of vague sets. International Journal of Applied Science and Engineering 6 (1), 71-83.

87) Mahmood Y A, Ahmadi A, Verma A K, Srividiya A and Kumar U (2013) Fuzzy fault tree analysis: a review of concept and application. International Journal of System Assurance Engineering and Management 03/2013; 4(1), 19-32.

88) Chris P Lee (2005) Visual Firewall: Real time network security monitor. Workshop on visualization for computer society October 26, 2005 IEEE.

89) Abdullah K, Lee C, Conti G and Copeland J (2005) Visualizing network data for intrusion detection. IEEE, information Assurance Workshop (IAW), 2005.

90) Zimmermann H Z (1996) Fuzzy Set Theory and Its Applications, 3rd edn., Kluwer Nijhoff, Boston.

91) https://securelist.com/analysis/kaspersky-security-bulletin/68010/kaspersky-security-bulletin-2014-overall-statistics-for-2014/ Access on 30th November 2014.

92) Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G and Vázquez E (2009) Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1), 18-28.

93) Gómez J, Gil C, Padilla N, Baños R and Jiménez C (2009) Design of a snort-based hybrid intrusion detection system. Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, Springer Berlin Heidelberg, 515-522.

94) Grégio A, Santos R and Montes A (2007) Evaluation of data mining techniques for suspicious network activity classification using honeypots data. Proc. SPIE 6570, Data Mining, Intrusion Detection, Information Assurance and Data Networks Security.

95) Lee W, Stolfo S J and Mok K W (1998) Mining Audit Data to Build Intrusion Detection Models, in KDD, 66-72.

96) Lee W, Stolfo S J and Mok K W (1999) A data mining framework for building intrusion detection models. IEEE Symposium on Security and Privacy, IEEE, 120-132.

97) Muda Z, Yassin W, Sulaiman M N and Udzir N I (2011) Intrusion detection based on K-Means clustering and Naïve Bayes classification. 7th International Conference on Information Technology in Asia (CITA 11), 1-6.

98) Oreku G S and Mtenzi F J (2009) Intrusion Detection Based on Data Mining. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC'09, IEEE 696-701.

99) Plonka D (2000) Flowscan: a network traffic flow reporting and visualization tool. LISA 2000 Conf., 2000, pp. 305–317.

100) Paxson V (1999) Bro: A System for Detecting Network Intruders in Real-Time. Computer Networks, 31(23-24), 2435-2463.

101) Ilgun K (1993) USTAT: A real-time intrusion detection system for UNIX. IEEE Computer Society Symposium on Research in Security and Privacy, 16-28.

102) Joshi A, Lal R, Finin T and Joshi A (2013) Extracting cyber security related linked data from text. IEEE Seventh International Conference on Semantic Computing, IEEE, 252-259.

103) Pongsiri J, Parikh M, Raspopovic M and Chandra K (1999) Visualization of Internet traffic features. 12th International Conference of Scientific Computing and Mathematical Modeling.

104) Kim I, Kim D, Choi Y, Kang K, Oh J and Jang J (2009) Validation Methods of Suspicious Network Flows for Unknown Attack Detection. International Journal of Computers, 3(1), 104-114.

105) Kohlrausch J (2009) Experiences with the NoAH Honeynet Testbed to Detect new Internet Worms. Fifth International Conference on IT Security Incident Management and IT Forensics, Stuttgart, Germany, 13-26.

106) Lazarevic A, Kumar V and Srivastava J (2005) Intrusion detection: A survey. Managing Cyber Threats, Springer, 19-78.

107) Lee S M, Kim D S and Park J S (2007) A Hybrid Approach for Real-Time Network Intrusion Detection Systems. International Conference on Computational Intelligence and Security, IEEE, 712-715.

108) Erbacher R F and Frincke D (2000) Visualization in detection of intrusions and misuse in large scale networks. Proc. International Conference, On Information Visualization 2000, July 2000, pp. 294–299.

109) Erbacher R F and Frincke D (2001) Visual behavior characterization for intrusion and misuse in large scale networks. Proc. SPIE 2001 Conference on Visual Data Exploration and Analysis VIII, January 2001, pp. 210– 218.

110) Chandra, S K and Gnanamurthy R K (2009) Executable Specification and Prototyping of Network Protocols Using UML and Java. pp-1-6 International Conference on Current Trends in Information Technology (CTIT 2009) IEEE, PP: 1-6.

111) Parish D J, Pagonis A, Barnett D M, Sandford J M and Phillips I W (2004) Presentation of real-time communication network measurement. IEE Proc., Sci. Meas. Technol., 151, pp. 319–326.

112) Luo H, Fang B and Yun X (2006) Anomaly detection in SMTP traffic. Third International Conference on Information Technology: New Generations (ITNG 2006), IEEE, 408-413.

113) Zou C C, Towsley D and Gong W (2004) Email worm modeling and defense. 13th International Conference on Computer Communications and Networks (ICCCN 2004), IEEE, 409-414.

114) Still M and Mc Creath E C (2011) DDoS protections for SMTP servers. International Journal of Computer Science and Security (IJCSS), 4(6), 537.

115) Saroliya A, Mishra U and Rana A (2012) A pragmatic analysis of peer to peer networks and protocols for security and confidentiality. International Journal of Computing and Corporate Research, ISSN2249054X-V2I6M7-112012.

116) Koutsofios E, North S C and Keim D A (1999) Visualizing large telecommunication data sets. IEEE Comput. Graph. Appl., 1999.

117) Koutsofios E E, North S C, Truscott R and Keim D A (1999) Visualizing large-scale telecommunication networks and services (case study). Proc. Conf. on Visualization '99.

118) Koutsofios E and Truscott R (2001) Large scale network visualization with 3D-graphics. Inf. Des. J., 2001, 10, (3), pp. 230–236.

119) Snehi J and Dhir R (2013) Web Client and Web Server approaches to Prevent XSS Attacks. International Journal of Computers & Technology, 345-352.

120) Sachdeva M, Singh G, Kumar K and Singh K (2010) DDoS Incidents and their Impact: A Review. International Arab Journal of Information Technology, 7(1), 14-20.

121) Thuraisingham B, Khan L, Awad M and Wang L (2010) Design and implementation of data mining tools. CRC Press.

122) Schiavone S L, Garg L and Summers K (2014) Ontology of Information Security in Enterprises. The Electronic Journal Information Systems Evaluation, Vol. 17, Issue 1, 071-087.

123) Zhang J, Zulkernine M and Haque A (2008) Random-forests-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38(5), 649-659.

124) Usmani K, Mohapatra A K and Prakash N (2013) An Improved Framework for Incident Handling. Information Security Journal: A Global Perspective, 1-9.

125) Munzner T and Burchard P (1995) Visualizing the structure of the world wide web in 3D hyperbolic space. VRML '95: Proc. first Symp., On Virtual Reality Modelling Language, 1995, pp. 33–38.

126) Allenotor D and Thulasiram R K (2009) A Fuzzy Grid-QoS Framework for Obtaining Higher Grid Resources Availability. Inder Science, Intl. J. High Performance Computing and Networking.

127) Takada T and Koike H (2002) Tudumi: Information visualization system for monitoring and auditing computer logs. Proc. 6th Int. Conf. On Information Visualization, 2002

128) Eick S G, Nelson M C and Schmidt J D (1994) Graphical analysis of computer log files. Commun., ACM, 1994, 37, (12), pp. 50–56.

129) http://www.yourdictionary.com/cyberpornography Access on 30th November 2014.

130) http://www.internetlivestats.com/total-number-of-websites/ Accessed on 23 Dec 2014.

131) http://www.internetworldstats.com/stats.htm Accessed on 23 Dec 2014.

132) http://www.internetworldstats.com/stats3.htm Accessed on 23 Dec 2014.

133) http://www.netindex.com/download/# Access on 23 Dec 2014.

134) http://www.netindex.com/value/2,11/India/ Access on 23 Dec 2014.

135) http://explorer.netindex.com/maps?country=India Access on 23 Dec 2014.

136) http://www.ebizmba.com/articles/business-websites Access on 23rd December 2014.

137) http://www.ebizmba.com/articles/ebusiness-websites Access on 23 December 2014.

138) http://www.ebizmba.com/articles/social-networking-websites Access on 23 Dec, 2014.

139) http://hackmageddon.com/2014/11/10/october-2014-cyber-attacks-statistics/ Access on 30th November 2014.

140) http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml Access on 23 Dec, 2014.