CHAPTER 1

INTRODUCTION

1.1 Introduction

Every organization follows a networking technology through which it connects all of its devices and communicates over the channels. There exist a lot of technologies that can be used to establish a network and they can vary depending on needs and costs of one's organization.

However, every organization's network structure is vulnerable to network attacks which attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of the information being transferred on the network.

The project focuses on implementing a software that protects the network infrastructure by developing intrusion detection and traffic monitoring tools and integrating them in a software that provides a bird's eye view to the entire network. An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces electronic reports to a management station. Network traffic monitoring is the process of sniffing, reviewing, analysing and managing network traffic for any abnormality or process that can affect network performance, availability and/ or security. For the purpose of demonstration, various types of attacks are fabricated. This will provide a better understanding of the system.

1.2 History

The IDS and Traffic Monitoring journey started thirty five years ago when increasing enterprise network access spawned a new challenge: the need for user access and user

monitoring. Levels of access to systems in a network and clear visibility into user activity was required to operate safely and securely.

The initial headway on IDS was made within the U.S. Air Force. In 1980, James P. Anderson, a pioneer in information security and member of the Defense Science Board Task Force on Computer Security at the U.S. Air Force, produced "Computer Security Threat Monitoring and Surveillance," a report that is often credited with introducing automated IDS. Soon after this report was released, the first model was built, born out of the same methods used by anti-virus applications: rule-based systems that constantly scanned and compared network traffic against a list of known threats.

During the late 1980's, with a growing number of shared networks, enterprise system administrators all over the world began adopting network traffic monitoring systems. In the 1990's, IDS technology improved to address the increasing number and sophistication of network attacks.

The advent of cloud computing, however, has brought new relevancy to IDS and monitoring systems, resulting in a surge in the market. An essential component of today's security best practices, IDS systems are designed to detect attacks that may occur, alongside preventative measures like monitoring systems. In fact, it is now one of the top selling security technologies, and predicted to continue to gain momentum.

1.3 Literature Survey

1.3.1 Basic Terminology

- Intrusion: An unauthorised entry into a network or a system, frequently synonymous with an information technology security incident.
- Network Traffic: Incoming and outgoing packets generating traffic.

- Attacks: Any method, process, or means used to maliciously attempt to compromise network security.

1.3.2 NIDS

Network based Intrusion Detection System (NIDS) is a system which monitors network intrusion. Intrusion may be detected by techniques like anomaly detection, signature pattern matching etc.

1.3.3 Network Traffic Monitoring

Network Traffic Monitoring Network Traffic Monitor is a network analytic tool that examines a network usage and provides a display of its statistics. The main purpose of the application is monitoring the IP traffic between the local area network and Internet. Network Monitor is a network diagnostic tool that monitors local area networks and provides a graphical display of network statistics. Network administrators can use these statistics to perform routine trouble- shooting tasks, such as locating a server that is down, or that is receiving a disproportionate number of work requests. The process by which Network Monitor collects this information is called capturing.

CHAPTER 2

ANALYSIS AND MODELLING

2.1 Introduction

Intrusion Detection Systems (IDSs) extract information from a computer or a network of computers, and attempt to detect the presence of intrusions from external sources, as well as abuses by authorized system users. The detection of system intrusions or abuses (both called attacks) presupposes the existence of a model. Modelling can take two forms.

In misuse detection, one models the known attack patterns through the construction of a library of attack signatures. Incoming patterns that match an element of the library are labelled as attacks. If only exact matching allowed, misuse detectors operate with no false alarms. By allowing some tolerance in matching attack, one runs the risk of false alarms, but Expects to be able to detect certain classes of unknown attacks, that "do not deviate much" from the attacks listed in the Library. We call such unknown attacks neighbouring attacks.

In anomaly detection, one models the normal behavior of the system. Incoming patterns that deviate substantially from normal behavior are labelled as attacks. Implicit in the utilization of abnormal patterns indicated to attacks is the premise that malicious activity is a subset of anomalous activity. One is tacitly accepting the presence of false alarms in this case, in exchange for the hope of detecting unknown attacks, which may be different from the neighbouring attacks.

2.2 Data Flow Diagram

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour.

DFD consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system DFD are used.

The network has a bunch of running servers. These servers may run facilities like telnet, FTP, web server and port/ socket. The network also has a Network Intrusion Detection System that will prevent the attacks from happening. The network is also connected to internet that allows users from around the web to access the services provided by the server. Unfortunately, this internet connection via a router or a switch also makes the network susceptible to attacks.

IDS along with the router/ switch will keep a tab on what are all the IP addresses and traffic generated by the clients that visit the network to user the services of the server. If an attacker tries to attack the servers, his activity will also be recorded by IDS. The attacks demonstrated can be categorized as follows

- Poisoning
- Flooding
- Brute Force Attack
- Spoofing

Poisoning attacks such as ARP spoofing, ARP cache poisoning, or ARP poison routing, is a technique by which an attacker sends (spoofed) Address Resolution Protocol (ARP) messages onto a local area network. Generally, the aim is to associate the attacker's MAC address with the IP address of another host, such as the default gateway, causing any traffic meant for that IP address to be sent to the attacker instead. They directly or indirectly affect the router/ switch to which the network is connected to the internet.

Flooding is a Denial of Service (DoS) attack that is designed to bring a network or service down by flooding it with large amounts of traffic. Flood attacks occur when a network or service becomes so weighed down with packets initiating incomplete connection requests that it can no longer process genuine connection requests. By flooding a server or host with

connections that cannot be completed, the flood attack eventually fills the host's memory buffer. Once this buffer is full no further connections can be made, and the result is a Denial of Service.

A brute force attack is a trial-and-error method used to obtain information such as a user password. In a brute force attack, automated software is used to generate a large number of consecutive guesses as to the value of the desired data. Brute force attacks may be used by criminals to crack encrypted data, or by security analysts to test an organization's network security.

A spoofing attack is when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware or bypass access controls. There are several different types of spoofing attacks that malicious parties can use to accomplish this. Some of the most common methods include IP address spoofing attacks, ARP spoofing attacks and DNS server spoofing attacks.

Now IDS is monitoring the entire network is creating the logs of everything that is going around the network and will generate alert if any malicious intent traffic is picked up by the Intrusion Detection System logic. The IDS will sniff the network for such traffic and logs and alert the administrator for the same.

The network traffic monitoring with attack and intrusion detection system will maintain logs of every aspect of the network infrastructure and logs can be manually analysed in the graphical format by the administrator for security purposes. The logs can be maintained, stored and read by the administrator in case of breach of security.

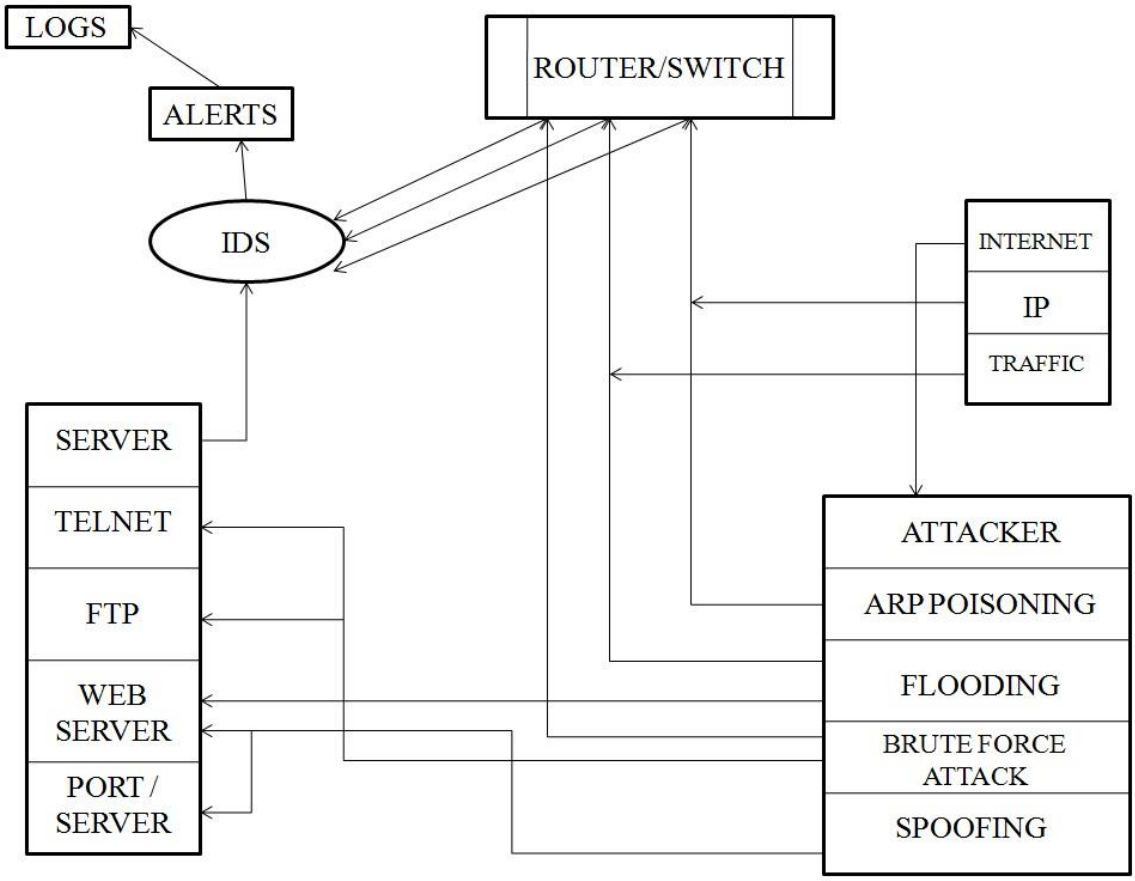

Figure 2.1. 0 Level Data Flow Diagram

2.3 IP Scanner Module

IP Scanner Module will run on server. Server will send ARP request to all the nodes in the network. These requests will be a broadcast. Each node will individually reply to the broadcast request. The reply will contain the MAC address of the node. The server receives the reply and will display the information on the console.

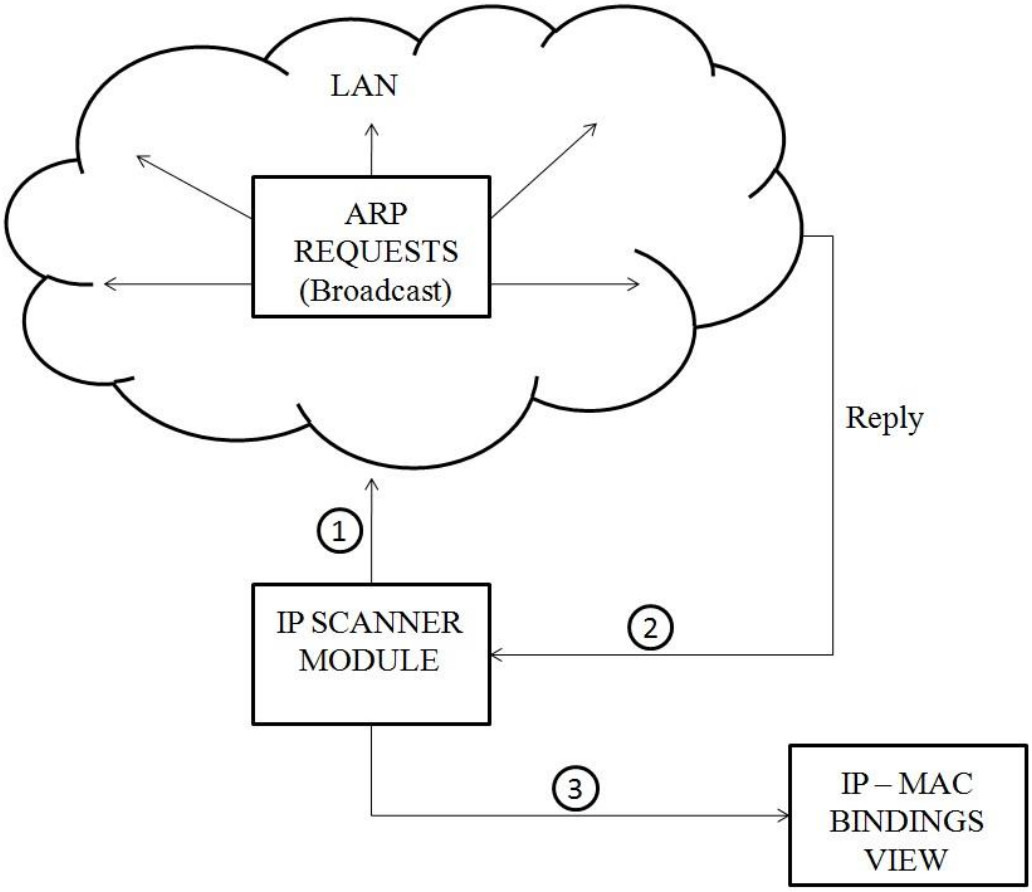

Figure 2.2. IP Scanner Module

2.4 System Query Module

System Query Module will run on the server. Server will request the agents running on the nodes of the network, asking them to send the information of the traffic on their device. This information is gathered by sniffing the network. The agent running on the nodes will reply to the server with the required information. When the server receives this information, it displays this information on the console.

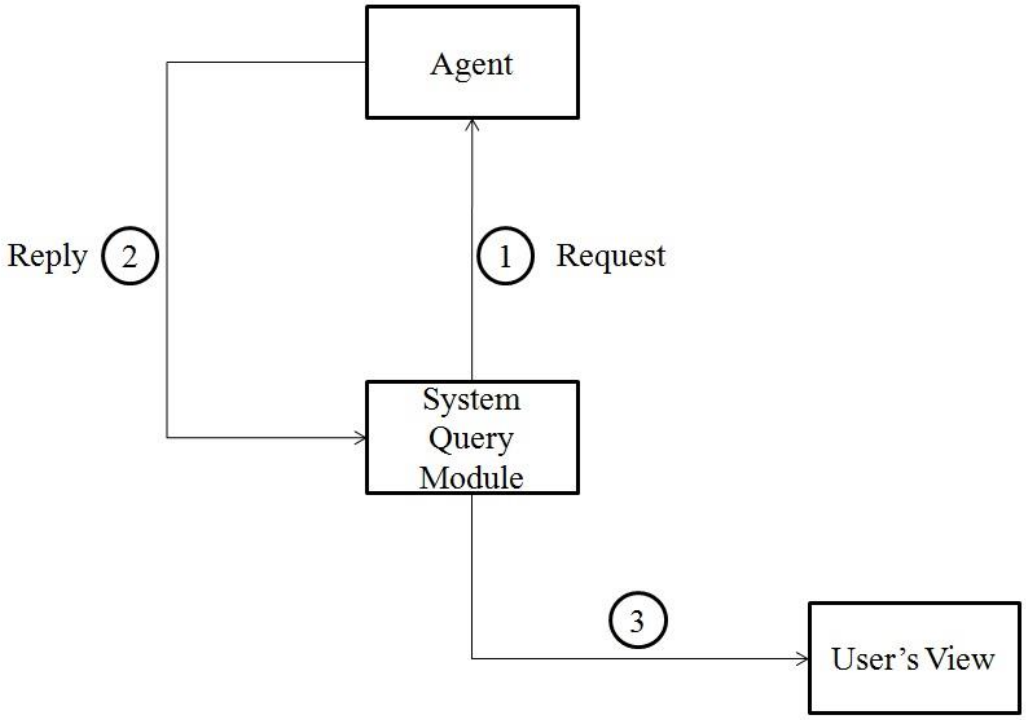

Figure 2.3. System Query Module

2.5 Network Traffic Monitoring and Intrusion Detection Module

The system users on the network are running agent of the monitoring program. The servers on the network can be accessed by all the nodes within the network and all the devices on the internet can also access the server. Every device connected in the network, internally or externally will generate some traffic. The agents will sniff the traffic and the sniffed traffic will be available for monitoring in the display console and attack detection module will apply its intrusion detection logic to analyse the intrusion and will generate alerts in the same case.

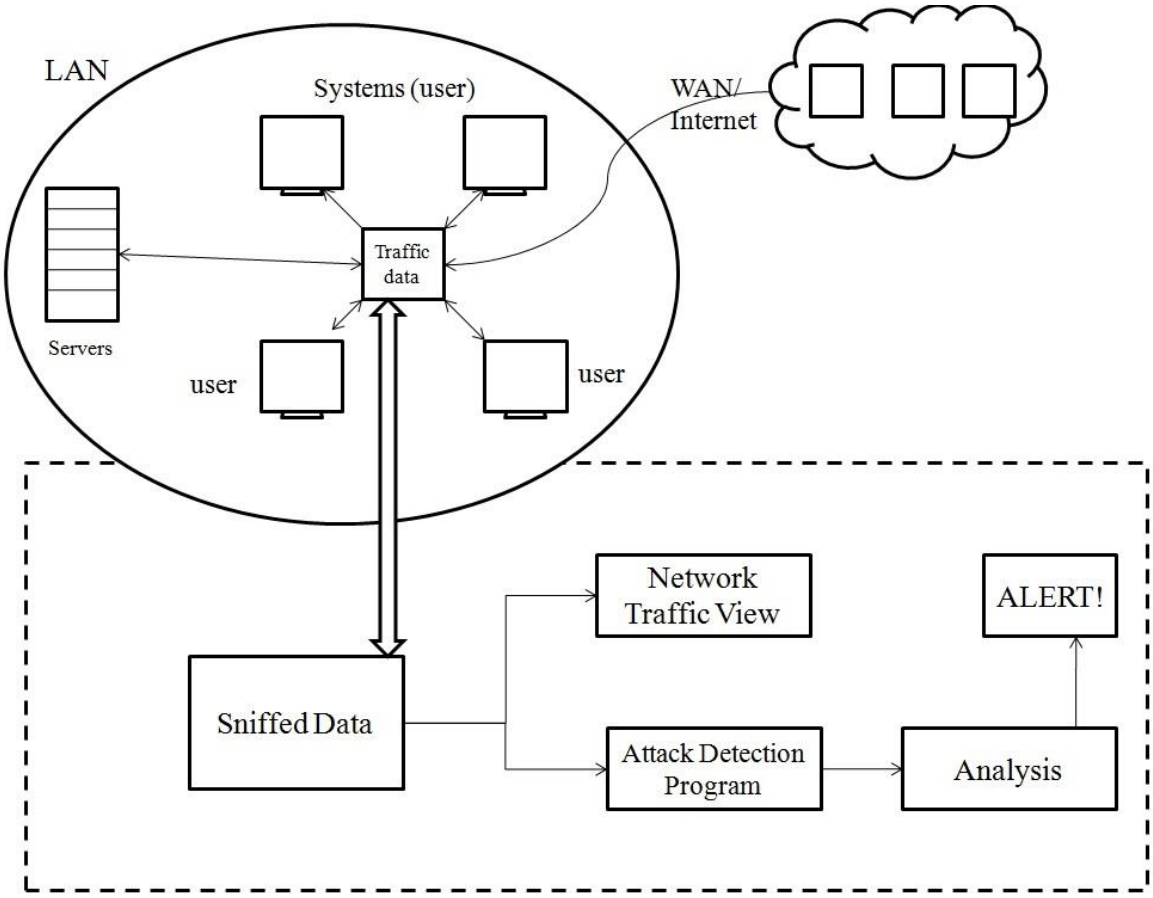

Figure 2.4. Network Traffic Monitoring and Intrusion Detection Module

CHAPTER 3

METHODOLOGY AND FUNCTIONALITY

3.1 Introduction

The implemented software is installed in the server in the network. The server is a centralized resource that administers and governs the network. All the other devices in the network will run a program called agent. This forms the backbone of the system. The server interacts with the agents and gains information regarding the data being transferred and the status of each machine. All the devices in the network are identified using their logical address. The server communicates to all the system using these addresses and vice versa.

On the server machine, a console provides a detailed description of the activity over the network. This includes 'what' activity is being performed and 'where' is it taking place. The administrator can view the network activity in real time. For example, if the agent is accessing a website through the network, the administrator will have the details of what website is being accessed, the location and the IP address of the system that is accessing the website. This kind of activity proves to be very useful in case of traffic monitoring. Detailed study of network traffic monitoring module is discussed further.

The console also provides a notification system, which informs the admin of the suspicious activities or attacks on the network. This activity is detected by the intrusion detection module of the system. It alerts the administrator when someone or something is trying to compromise information system through malicious activities or through security policy violations. The administrator will have the information about the address of the devices involved which allows the administrator to take appropriate actions whenever the attacks occur.

Apart from the above two major functions, the software provides other tools to facilitate the working of the system. The 'Action' menu consists of two fields. The first field, 'Update Users', shows a list of all the users in the network, along with what activity is being performed at each active terminal. The second field, 'Dummy', lists all the dumb terminals in the network. Dumb terminal is a device which does not have processing capability and is inactive but is currently a part of the network. This shows all the IP addresses in the network, irrespective of the current status – active or inactive.

Next menu on the console is ‘Send Message’. It consists of two fields. The ‘Broadcast’ field allows the administrator to send message as broadcast, which is delivered to all the devices in the network. The other option ‘To Specific User’, allows the admin to send message to a specific user in the network. The admin would be required to specify the IP address of the user he wants to send a message to. In both the cases, a dialog box will appear asking the admin to write the message and send.

The ‘Monitor’ menu is the home menu, and this is where the traffic monitoring and IDS modules are displayed.

The ‘Report’ menu shows a graphical representation of the traffic in the network. Histogram is a diagram consisting of rectangles whose area is proportional to the frequency of a variable and whose width is equal to the class interval. ARP, HTTP, ICMP, DNS Query, UDP, DHCP, FTP, TELNET and SSH are types of packets that can be analyzed at once in this menu.

Division of the software has been done into 4 modules and each module helped us with the division of labour and keeping a track of progress of the entire software. The proposed system was designed, plans were laid out concerning the programming, communications, and security issues.

The three modules are:

- Network traffic monitoring module
- Intrusion detection module
- Integration and user interface

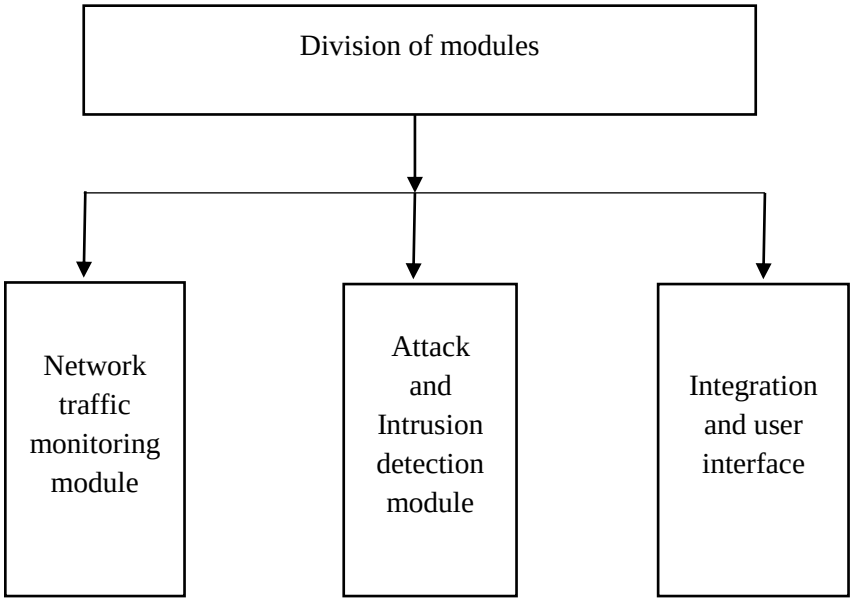

Figure 3.1. Division of Modules

- Under network traffic module, traffic has been monitored. Network monitoring is the use of a system that constantly monitors a computer network for creating logs of traffic and notifies the network administrator. It is part of network management.
- Under intrusion detection module, the administrator will be notified of any malicious traffic that can cause harm to the network in any way. A separate dialog box has been created to display such traffic. Various attacks like DoS attack, MAC flooding attack, SYN Flood attack, Brute force attack, DNS Spoofing attack have been tested out on the intrusion detection system.
- Under integration and user interface, the above mentioned have been unified under a single User Interface and tested as an individual unit. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates,

applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

3.2 Network Traffic Monitoring Module

Network traffic monitoring is the process of reviewing, analysing and managing network traffic for any abnormality or process that can affect network performance, availability and/or security. It is a network management process that uses various tools and techniques to study computer network based communication/data/packet traffic. The key objective behind network traffic monitoring is to ensure availability and smooth operations on a computer network. Network monitoring incorporates network sniffing and packet capturing techniques in monitoring a network. Network traffic monitoring generally requires reviewing each incoming and outgoing packet.

As company intranets continue to grow it is increasingly important that network administrators are aware of and have a handle on the different types of traffic that is traversing their networks. Traffic monitoring and analysis is essential in order to more effectively troubleshoot and resolve issues when they occur, so as to not bring network services to a stand still for extended periods of time. Numerous tools are available to help administrators with the monitoring and analysis of network traffic.

Network monitoring is a difficult and demanding task that is a vital part of a Network Administrators job. Network Administrators are constantly striving to maintain smooth operation of their networks. If a network were to be down even for a small period of time productivity within a company would decline, and in the case of public service departments the ability to provide essential services would be compromised. In order to be proactive rather than reactive, administrators need to monitor traffic movement and performance throughout the network and verify that security breaches do not occur within the network.

When choosing a particular tool to use for monitoring, an Admin must first decide if they would like to use a more proven system or a newer system. If the proven system is the

direction that feels more comfortable, network monitoring is beneficial tool to use since a data analysis package can be used in conjunction with it to present the data in a user friendly environment.

Network performance could be measured using either active or passive techniques. Active techniques are more intrusive but are arguably more accurate. Passive techniques are of less network overhead and hence can run in the background to be used to trigger network management actions.

Various software tools are available to measure network traffic. Some tools measure traffic by sniffing and others use SNMP, WMI or other local agents to measure bandwidth use on individual machines and routers. However, the latter generally do not detect the type of traffic, nor do they work for machines which are not running the necessary agent software, such as rogue machines on the network, or machines for which no compatible agent is available. In the latter case, inline appliances are preferred. These would generally 'sit' between the LAN and the LAN's exit point, generally the WAN or Internet router, and all packets leaving and entering the network would go through them. In most cases the appliance would operate as a bridge on the network so that it is undetectable by users.

Measurement tools generally have these functions and features:

- User interface (web, graphical, console)
- Real-time traffic graphs
- Network activity is often reported against pre-configured traffic matching rules to show:
  - Local IP address
  - Remote IP address
  - Port number or protocol
  - Logged in user name
- Bandwidth quotas
- Support for traffic shaping or rate limiting (overlapping with the network traffic control page)

- Support website blocking and content filtering
- Alarms to notify the administrator of excessive usage (by IP address or in total)

3.3 Attacks and Intrusion Detection Module

3.3.1 Attack Detection

In computer and computer networks an attack is any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset. Without security measures and controls in place, your data might be subjected to an attack. Some attacks are passive, meaning information is monitored; others are active, meaning the information is altered with intent to corrupt or destroy the data or the network itself. An attack can be perpetrated by an insider or from outside the organization.

An inside attack is an attack initiated by an entity inside the security perimeter (an insider), i.e., an entity that is authorized to access system resources but uses them in a way not approved by those who granted the authorization.

An outside attack is initiated from outside the perimeter, by an unauthorized or illegitimate user of the system (an outsider). In the Internet, potential outside attackers range from amateur pranksters to organized criminals, international terrorists, and hostile governments.

A resource (either physical or logical), called an asset, can have one or more vulnerabilities that can be exploited by a threat agent in a threat action. The result can potentially compromises the Confidentiality, Integrity or Availability properties of resources of the organization and others involved parties (customers, suppliers).

The so-called CIA triad is the basis of Information Security.

The attack can be active when it attempts to alter system resources or affect their operation: so it compromises Integrity or Availability. A passive attack attempts to learn or make use of

information from the system but does not affect system resources: so it compromises Confidentiality.

A Threat is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability. A threat, if intentional i.e. intelligent; e.g., an individual cracker or a criminal organization needs to be addressed as soon as possible. A natural threat, like earthquake or flood cannot be prepared for to an impressive extent.

An attack usually is perpetrated by someone with bad intentions: Black hatted attacks falls in this category, while other perform Penetration testing on an organization information system to find out if all foreseen controls are in place.

The attacks can be classified according to their origin: i.e. if it is conducted using one or more computers: in the last case is called a distributed attack. Botnet are used to conduct distributed attacks.

Following attacks are detected by the software:

- Denial-of-Service Attack
- SYN flood attack
- Brute force attack
- DNS Spoofing
- MAC Flooding

3.3.1.1 Denial-Of-Service Attack

In computing, a denial-of-service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet.

Symptoms of denial-of-service attacks to include:

- Unusually slow network performance (opening files or accessing web sites)
- Unavailability of a particular web site
- Inability to access any web site
- Dramatic increase in the number of spam emails received—(this type of DoS attack is considered an e-mail bomb)
- Disconnection of a wireless or wired internet connection
- Long term denial of access to the web or any internet services

If the attack is conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent by incorrectly configured or flimsy network infrastructure equipment.

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services.

The most serious attacks are distributed and in many or most cases involve forging of IP sender addresses (IP address spoofing) so that the location of the attacking machines cannot easily be identified, nor can filtering be done based on the source address.

3.3.1.2 SYN Flood Attack

Simple attacks such as SYN floods may appear with a wide range of source IP addresses, giving the appearance of a well distributed DoS. These flood attacks do not require completion of the TCP three way handshake and attempt to exhaust the destination SYN queue or the server bandwidth. Because the source IP addresses can be trivially spoofed, an attack could come from a limited set of sources, or may even originate from a single host. Stack enhancements such as syn cookies may be effective mitigation against SYN queue flooding, however complete bandwidth exhaustion may require involvement.

Normally when a client attempts to start a TCP connection to a server, the client and server exchange a series of messages which normally runs like this:

1. The client requests a connection by sending a SYN (synchronize) message to the server.
2. The server acknowledges this request by sending SYN-ACK back to the client.
3. The client responds with an ACK, and the connection is established.

This is called the TCP three-way handshake, and is the foundation for every connection established using the TCP protocol.

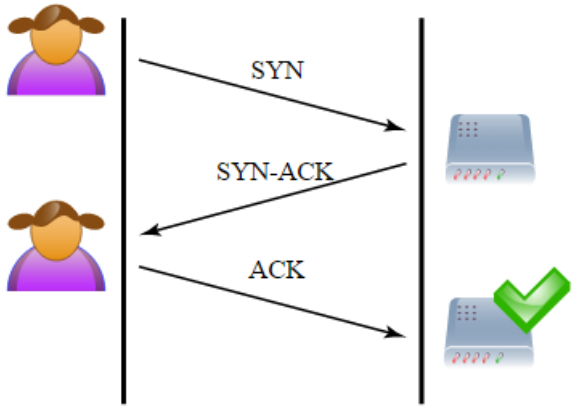

Figure 3.2. Three-Way Handshake Between the Client and the Server

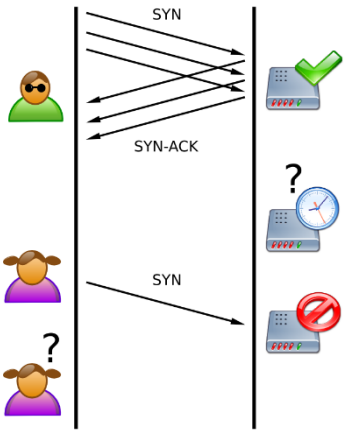

Figure 3.3. 0 SYN Flooding Blocks Other Users to Perform Three-Way Handshake With the Server

3.3.1.3 Brute Force Attack

In cryptography, a brute-force attack, or exhaustive key search, is a cryptanalytic attack that can, in theory, be used against any encrypted data (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier. It consists of systematically checking all possible keys or passwords until the correct one is found. In the worst case, this would involve traversing the entire search space.

When password guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because of the time a brute-force search takes.

When key guessing for modern cryptosystems, the key length used in the cipher determines the practical feasibility of performing a brute-force attack, with longer keys exponentially more difficult to crack than shorter ones.

Brute force attacks work by calculating every possible combination that could make up a password and testing it to see if it is the correct password. As the password's length increases, the amount of time, on average, to find the correct password increases exponentially. This means short passwords can usually be discovered quite quickly, but longer passwords may take decades.

Credential recycling refers to the hacking practice of re-using username and password combinations gathered in previous brute-force attacks. A special form of credential recycling is pass the hash, where unsalted hashed credentials are stolen and re-used without first being brute forced.

3.3.1.4 DNS Spoofing

DNS is a Domain Name System, which has all the websites names and its corresponding IP address in its database in the form of records and are placed in a hierarchal

manner in the Internet. Whenever a client access a particular website, say www.google.com, a request will be first sent to the Local DNS server for the IP address of that particular website. Then the DNS server checks for that particular IP address in its data base and once it finds it, then it immediately sends a response to the client browser regarding the IP information. If it does not find in its database, then it will forward the request to the top level DNS server in the hierarchy. In this way, the DNS server resolves name resolution requests coming from the clients.

When an entry in DNS server is modified in such a way that, a particular website to an ip address which is not the expected one. Then the clients, whose requests are being resloved by this DNS server, may be redirected to another website than the expected one.This kind of situation occurs, when the entry is not added correctly in the DNS server or any unauthorised user has modified the DNS entries. The process of modifying the DNS entries in an unauthorised manner is known as DNS Spoofing.

DNS spoofing is mainly achieved by using the following methods.

- DNS cache Poisioning and
- DNS ID spoofing.

The DNS Cache Poisioning method, can be explained with an example. Consider two DNS servers – one which is Local DNS server with domain name www.abc.com for your organisation and the other is a compromised DNS server with domain name www.attacker.com. The attacker adds some customized entries, which includes legitimate website names with his own relevant ip address in the compromised DNS server. After that he sends a name resolution request for the ip address information of the domain www.attacker.com to the DNS server of the domain www.abc.com. Since the DNS server, doesn't have the information in its data base, it sends response to the attacker after getting the information from the comprimised DNS Server. During this transaction period, the DNS server of www.abc.com not only recieves the IP address information of www.attacker.com but also the other records present in the DNS server in to its cache. This is normally referred to as cache poisioning. At this moment, if a legtimate user connects to

local DNS server for name resolution he will be misguided to other website than the expected one.

In case of DNS ID spoofing, when a name resolve request is generated by the client to send it to the DNS server, an ID will be generated along with the request. The client will accept the response for his request, if the ID of the response packet matches with the requested packet ID. But this way of name resolution is not secured. Because any unauthorized user can sniff the request and can create a response packet on the fly with the same ID and IP information contained in it is not the expected one. This kind of DNS attack is known as DNS ID Spoofing.

3.3.1.5 MAC Flood Attack

In computer networking, MAC flooding is a technique employed to compromise the security of network switches.

Switches maintain a MAC Table that maps individual MAC addresses on the network to the physical ports on the switch. This allows the switch to direct data out of the physical port where the recipient is located, as opposed to indiscriminately broadcasting the data out of all ports as a hub does. The advantage of this method is that data is bridged exclusively to the network segment containing the computer that the data is specifically destined for.

In a typical MAC flooding attack, a switch is fed many Ethernet frames, each containing different source MAC addresses, by the attacker. The intention is to consume the limited memory set aside in the switch to store the MAC address table.

The effect of this attack may vary across implementations, however the desired effect (by the attacker) is to force legitimate MAC addresses out of the MAC address table, causing significant quantities of incoming frames to be flooded out on all ports. It is from this flooding behavior that the MAC flooding attack gets its name.

After launching a successful MAC flooding attack, a malicious user could then use a packet analyzer to capture sensitive data being transmitted between other computers, which would not be accessible were the switch operating normally. The attacker may also follow up with an ARP spoofing attack which will allow them to retain access to privileged data after switches recover from the initial MAC flooding attack.

3.3.2 Intrusion Detection

Intrusion detection is a process that monitors network or system activities for malicious activities or policy violations and produces electronic reports to a management station.NIDS is a network security system focusing on the attacks that come from the inside of the network (authorized users). Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.

IDPSes typically record information related to observed events, notify security administrators of important observed events and produce reports. Many IDPSes can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content.

Network Intrusion Detection Systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet, and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. An example of an NIDS would be installing it on the

subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network.

NID Systems are also capable of comparing signatures for similar packets to link and drop harmful detected packets which have a signature matching the records in the NIDS. When we classify the designing of the NIDS according to the system interactivity property, there are two types: on-line and off-line NIDS. On-line NIDS deals with the network in real time. It analyses the Ethernet packets and applies some rules, to decide if it is an attack or not. Off-line NIDS deals with stored data and passes it through some processes to decide if it is an attack or not.

In a passive system, the intrusion detection system (IDS) sensor detects a potential security breach, logs the information and signals an alert on the console or owner. In a reactive system, also known as an intrusion prevention system (IPS), the IPS auto-responds to the suspicious activity by resetting the connection or by reprogramming the firewall to block network traffic from the suspected malicious source. The term IDPS is commonly used where this can happen automatically or at the command of an operator; systems that both "detect (alert)" and "prevent".

IDS should not be confused with firewalls. a firewall is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted, secure internal network and another outside network, such as the Internet, that is assumed not to be secure or trusted. Firewalls are often categorized as either network firewalls or host-based firewalls. Network firewalls are a software appliance running on general purpose hardware or hardware-based firewall computer appliances that filter traffic between two or more networks. Host-based firewalls provide a layer of software on one host that controls network traffic in and out of that single machine. Firewall appliances may also offer other functionality to the internal network they protect such as acting as a DHCP or VPN server for that network.

Though they both relate to network security, an intrusion detection system (IDS) differs from a firewall in that a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an application layer firewall.

3.4 Integration and User Interface

In this module, integration and user interface, the above mentioned have been unified under a single User Interface and tested as an individual unit. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea

is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatical complexity of the software and software architecture, reusability of modules and life-cycle / versioning management.

But before integration testing and integrating all the modules under a single User Interface, unit testing was done.Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

- Finds problems early
- Facilitates change
- Simplifies integration
- Documentation
- Design

For integrating the modules, we used tkinter python library to create the user interface. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk

GUI toolkit. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application. Tkinter is free software released under a Python license.

The user interface by default opens up monitoring tab where the interface is divided into 3 display screens. First row includes twoThe 'Action' menu consists of two fields. The first field, 'Update Users', shows a list of all the users in the network, along with what activity is being performed at each active terminal. The second field, 'Dummy', lists all the dumb terminals in the network. Dumb terminal is a device which does not have processing capability and is inactive but is currently a part of the network. This shows all the IP addresses in the network, irrespective of the current status – active or inactive.

Next menu on the console is 'Send Message'. It consists of two fields. The 'Broadcast' field allows the administrator to send message as broadcast, which is delivered to all the devices in the network. The other option 'To Specific User', allows the admin to send message to a specific user in the network. The admin would be required to specify the IP address of the user he wants to send a message to. In both the cases, a dialog box will appear asking the admin to write the message and send.

The 'Monitor' menu is the home menu, and this is where the traffic monitoring and IDS modules are displayed.

The 'Report' menu shows a graphical representation of the traffic in the network. Histogram is a diagram consisting of rectangles whose area is proportional to the frequency of a variable and whose width is equal to the class interval. ARP, HTTP, ICMP, DNS Query, UDP, DHCP, FTP, TELNET and SSH are types of packets that can be analyzed at once in this menu.

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 Purpose

The purpose of this project is to create a software that protects the network infrastructure by developing intrusion detection and traffic monitoring tools and integrating them in a software that provides a bird's eye view to the entire network.

4.2 Scope of the Project

The software developed in this project has the potential to be used in systems of the organizations which are connected to the network and contain data critical to the business processes. As every organization's network structure is vulnerable to network attacks such as DOS, DDOS, SYN flood attack, etc. which attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of the information being transferred on the network. This software will be monitoring all the traffic entering and leaving the systems in order to detect any attacks when happens and proper action could be taken.

4.3 Technologies Used

4.3.1 Ubuntu Operating System:

Ubuntu is a Debian based Linux operating system and distribution for personal computers, smart phones and network servers. It uses Unity as its default user interface. It is based on free software and named after the Southern African philosophy of ubuntu, which often is translated as 'humanity towards others' or 'the belief in a universal bond of sharing that connects all humanity'.

Development of Ubuntu is led by UK-based Canonical Ltd., a company owned by South African entrepreneur Mark Shuttleworth. Canonical generates revenue through the sale of technical support and other services related to Ubuntu. The Ubuntu project is publicly

committed to the principles of open-source software development; people are encouraged to use free software, study how it works, improve upon it, and distribute it.

4.3.2 VMWare Player

VMware Player, is a virtualization software package for x64 computers running Microsoft Windows or Linux, supplied free of charge by VMware, Inc., a company which was formerly a division of, and whose majority shareholder remains EMC Corporation. VMware Player can run existing virtual appliances and create its own virtual machines (which require an operating system to be installed to be functional). It uses the same virtualization core as VMware Workstation, a similar program with more features, but not free of charge. VMware Player is available for personal non-commercial use, or for distribution or other use by written agreement. VMware, Inc. does not formally support Player, but there is an active community website for discussing and resolving issues, and acknowledge base.

The free VMware Player was distinct from VMware Workstation until Player v7, Workstation v11. In 2015 the two packages were combined as VMware Workstation 12, with a free for non-commercial use Player version which, on purchase of a license code, became the higher-specification VMware Workstation Pro.

4.3.3 TKinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI, and is included with the standard Microsoft Windows and Mac OS X install of Python. The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.
Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of

Tkinter). There are several popular GUI library alternatives available, such as wxPython, PyQt (PySide), Pygame, Pyglet, and PyGTK. Tkinter is free software released under a Python license.

4.3.4 Scapy Library

Scapy is a packet manipulation tool for computer networks, written in Python by Philippe Biondi. It can forge or decode packets, send them on the wire, capture them, and match requests and replies. It can also handle tasks like scanning, trace routing, probing, unit tests, attacks, and network discovery.

Scapy provides a Python interface into libpcap, (WinPCap on Windows), in a similar way to that in which Wireshark provides a view and capture GUI. It can interface with a number of other programs to provide visualisation including Wireshark for decoding packets, GnuPlot for providing graphs, graphviz or VPython for visualisation, etc. Scapy works only with python 2.x.

4.3.5 Gedit

Gedit is the default text editor of the GNOME desktop environment and part of theGNOME Core Applications. Designed as a general purpose text editor, gedit emphasizes simplicity and ease of use. It includes tools for editing source code and structured text such as markup languages.

It is designed to have a clean, simple graphical user interface according to the philosophy of the GNOME project, and it is the default text editor for GNOME. In addition, it is also available for both Mac OS X and Microsoft Windows.

gedit is free and open-source software subject to the requirements of the GNU General Public License version 2 or later.

4.3.6 Psutil Library:

Psutil (python system and process utilities) is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disksand network) in Python. It is useful mainly for system monitoring, profiling and limiting process

resources and management of running processes. It implements many functionalities offered by command line tools such as: ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap. It currently supports Linux, Windows, OSX, Sun Solaris, FreeBSD, OpenBSD and NetBSD, both 32-bit and 64-bit architectures, with Python versions from 2.6 to 3.5 (users of Python 2.4 and 2.5 may use 2.1.3 version).

4.3.7 Language

Python 2.7 has been used for coding the methods and implementing the functionalities required by the target software.

4.4 System Requirements

4.4.1 Hardware Requirements

S.No.	Description	Alternative
1	PC with minimum 2 GB hard-disk and 256 MB RAM	Not applicable
2	Wi-Fi enabled router	Ethernet

Table 4.1. Hardware requirement

4.4.2 Software Requirements

S.No.	Description	Alternative
1	Python 2.7	Python 3.5
2	Tkinter	Turtle
3	VMWare	Oracle – Virtual Box
4	Linux Ubuntu OS	Windows

Figure 4.2. Software requirements

*Scapy library must be installed on the system.

4.5 Description of Main Functionality

A Local Area Network consists of a server, a machine running IDS and all the clients running agents. The Local Area Network is connected to internet via a router. Everyone can access the server on the Local Area Network through the internet connection. Every person accessing the network internally or via internet will be monitored. The agents running on every client will monitor the entire activities on their system along with other system details and it will report to the admin when queried.

CHAPTER 5

SCREENSHOTS

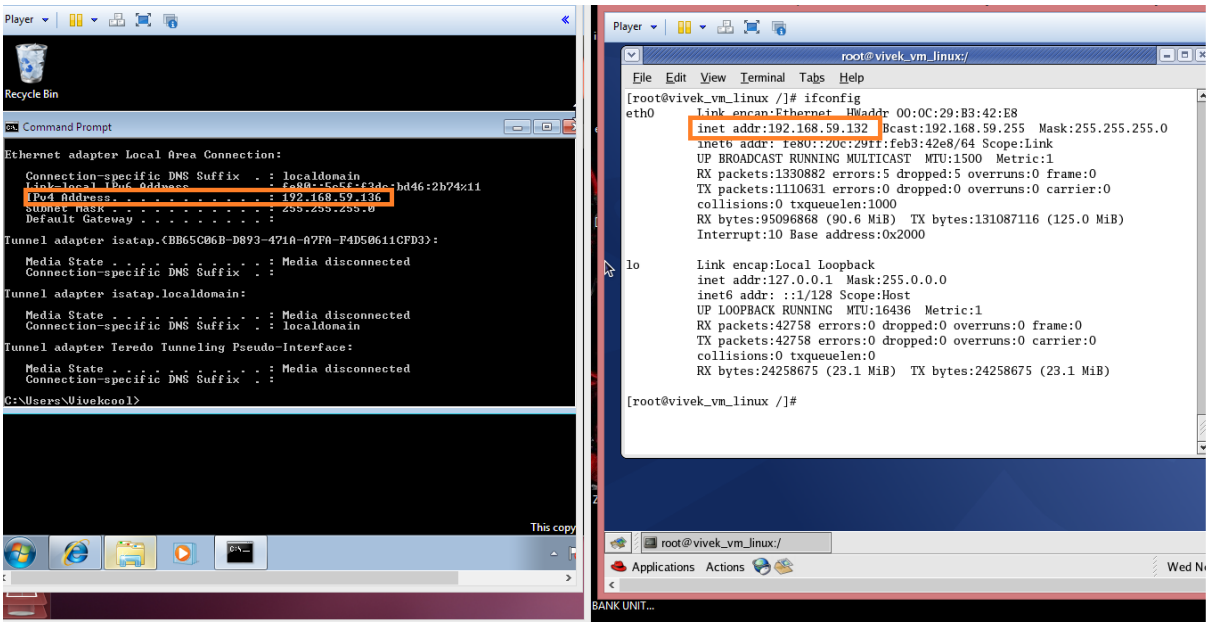

Figure 5.1. Server and User Address

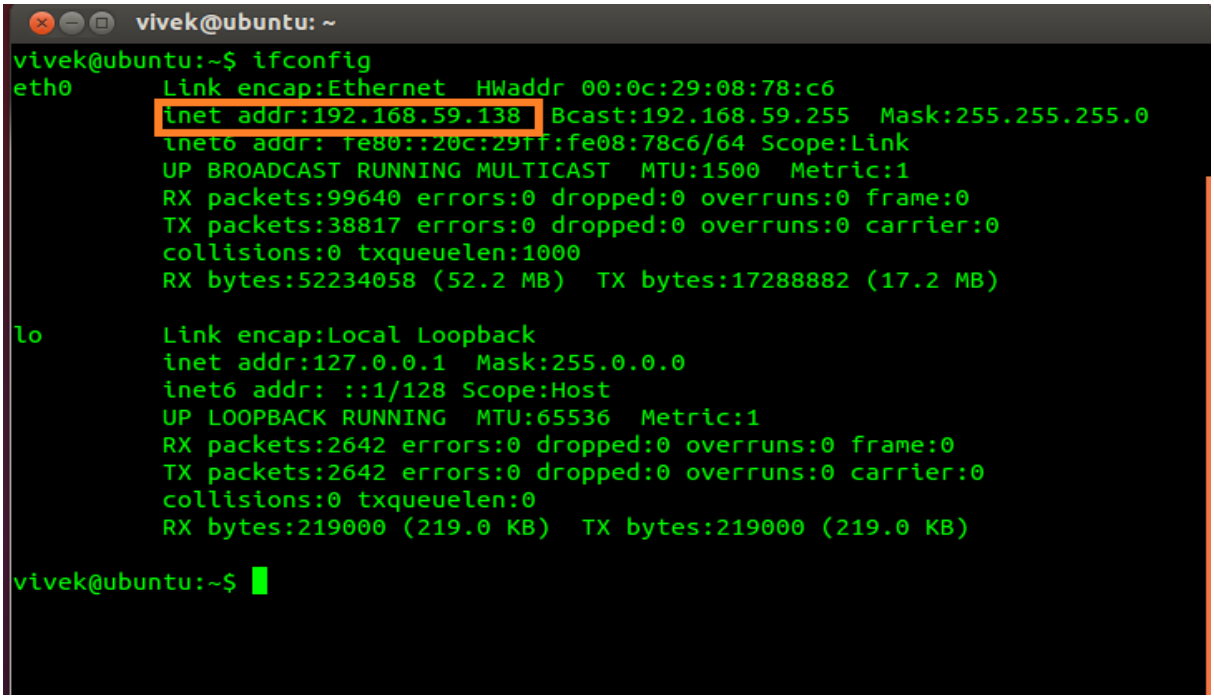

Figure 5.2. Zucknet Address

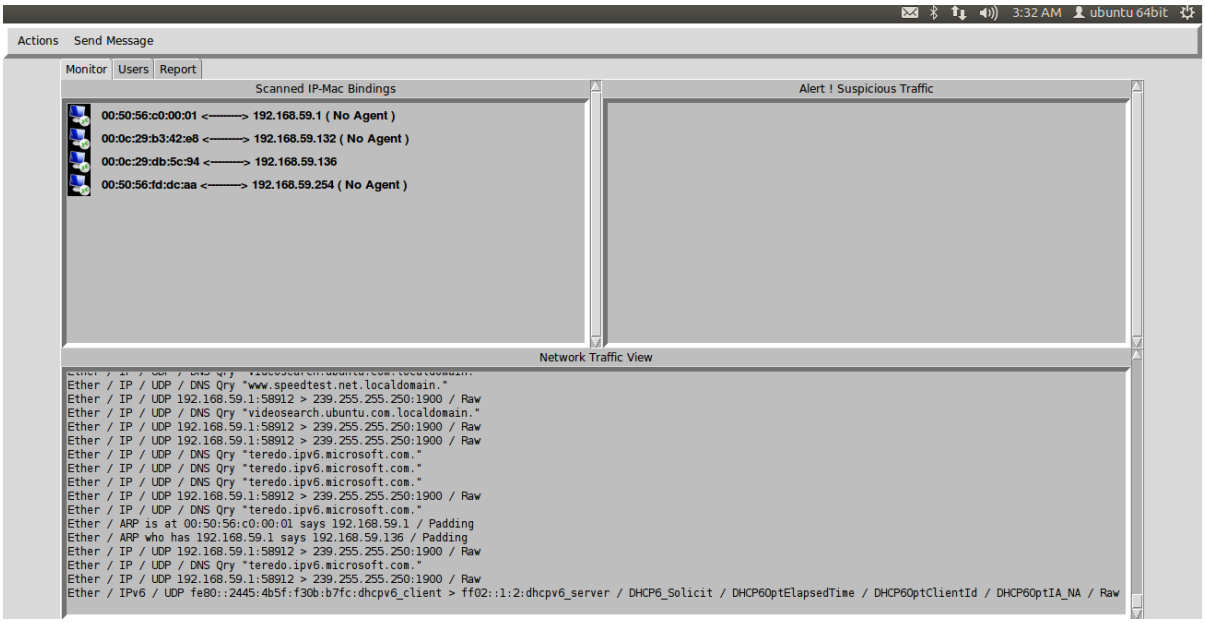

Figure 5.3. Startup Scan with MAC IP Address binding and Live Traffic Monitoring

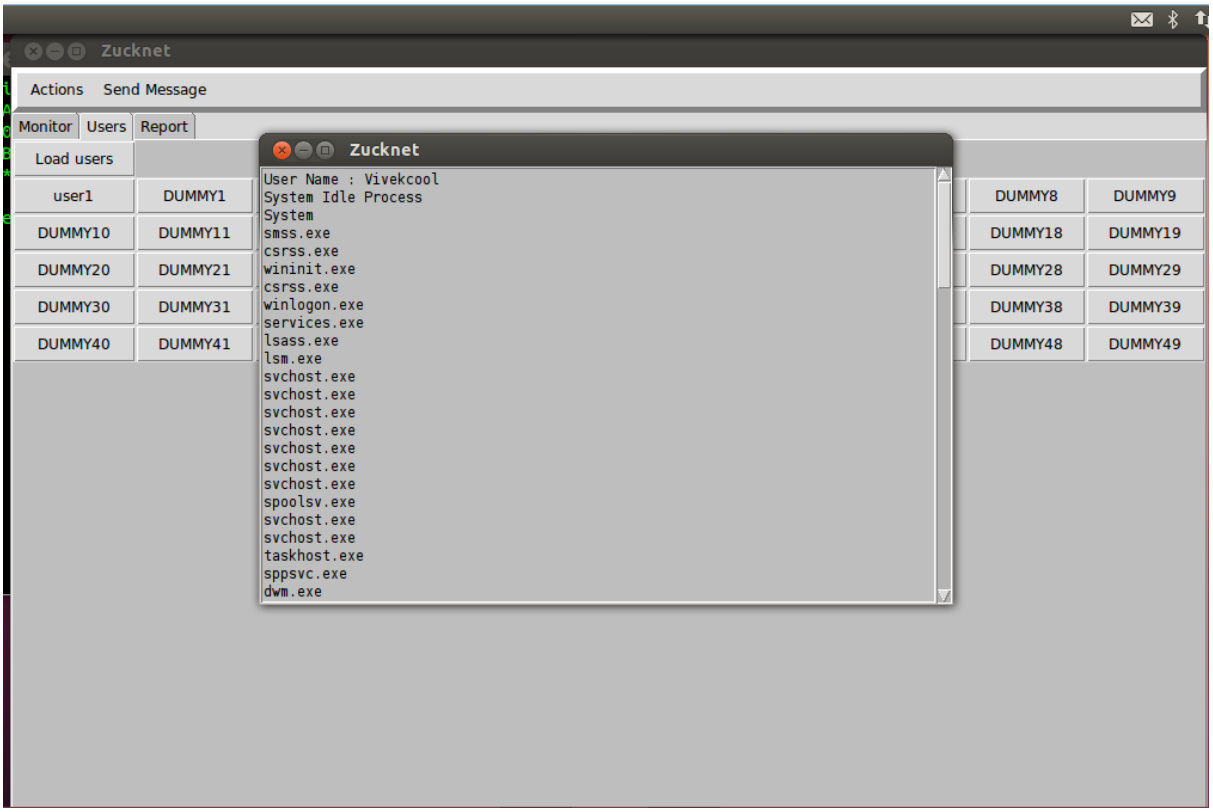

Figure 5.4. Users Tab

Figure 5.5. Report Histogram for Manual Analysis

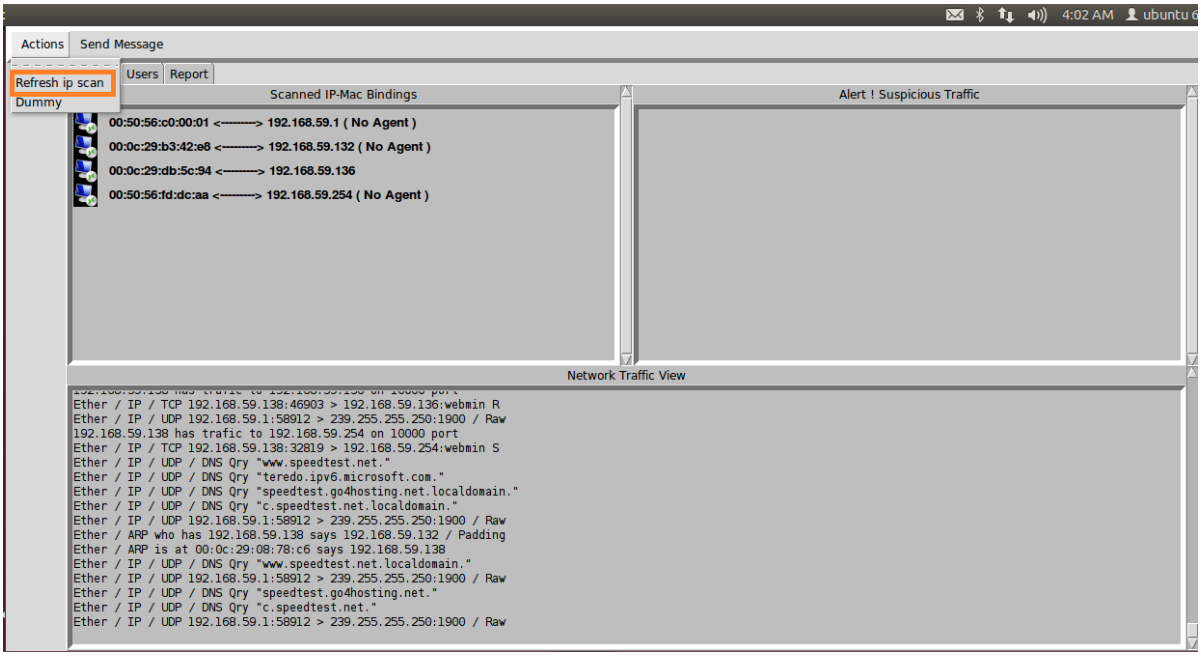

Figure 5.6. Refresh IP Scan

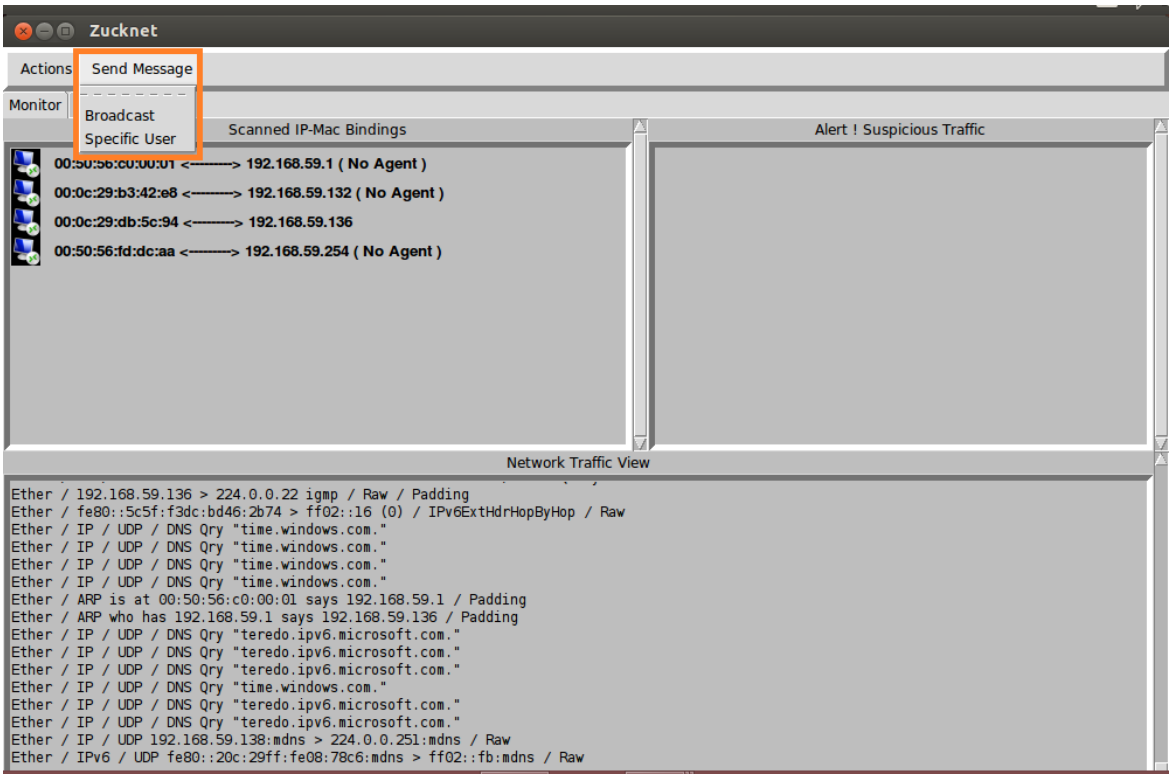

Figure 5.7. Send Message Tab

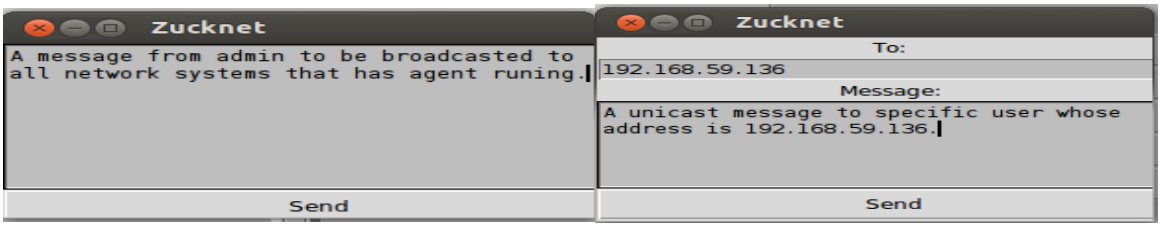

Figure 5.8. Broadcast and Unicast a Message

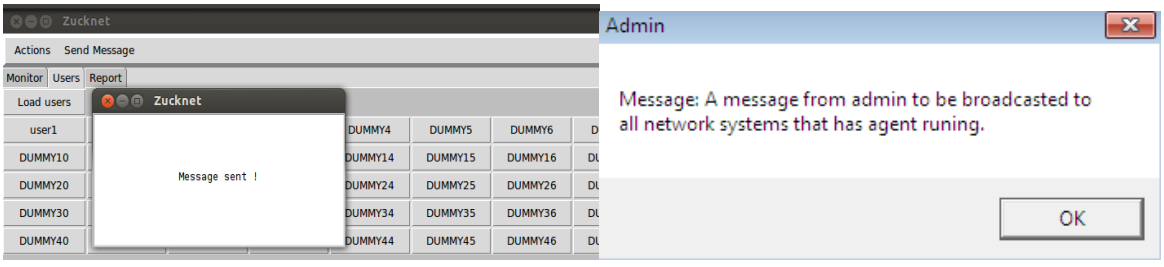

Figure 5.9. Broadcast Message Sent and Received

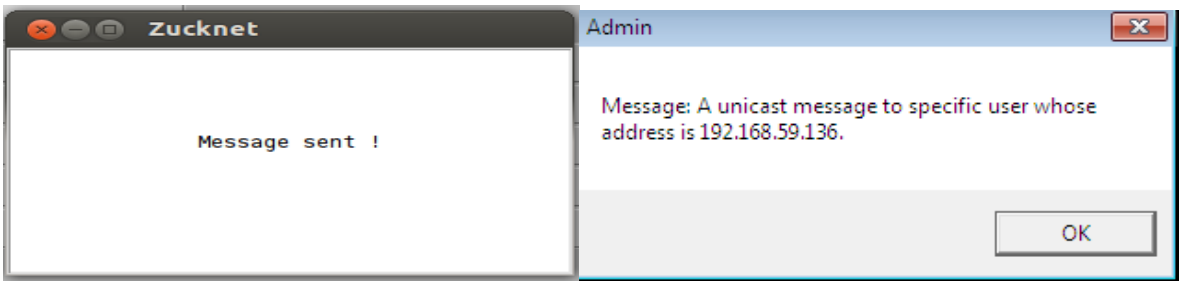

Figure 5.10. Unicast Message Sent and Received

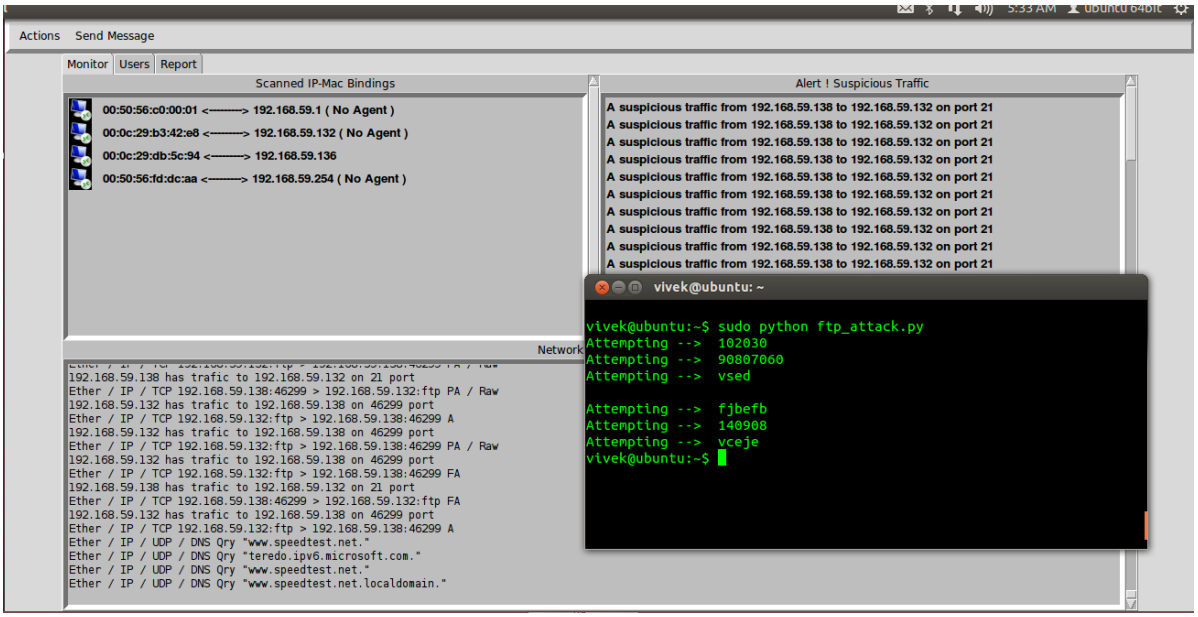

Figure 5.11. FTP Brute Force Attack Detection

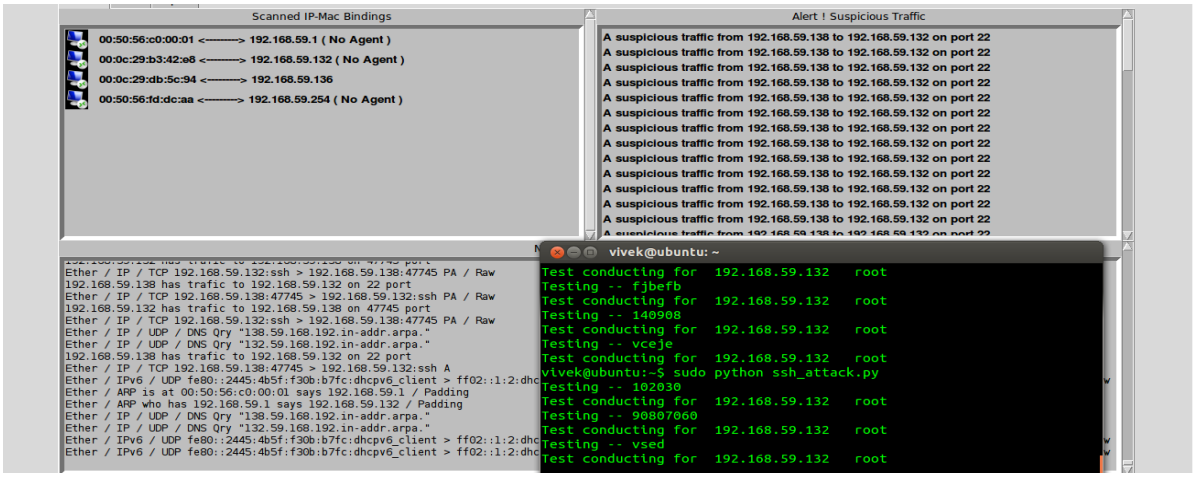

Figure 5.12. SSH Attack Detection

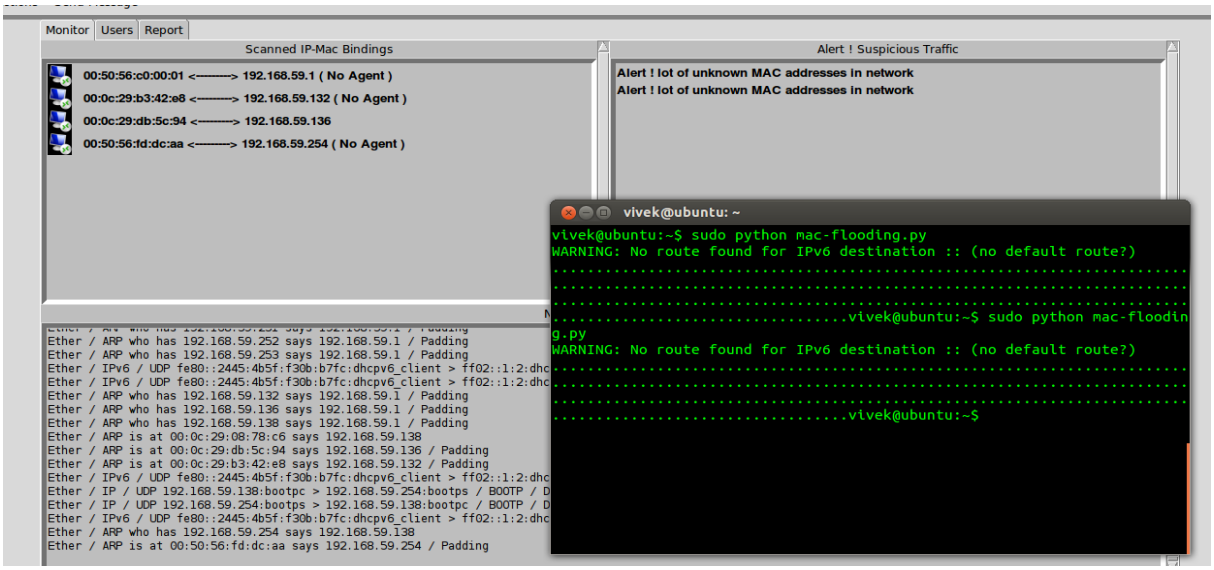

Figure 5.13. MAC Flood Detection

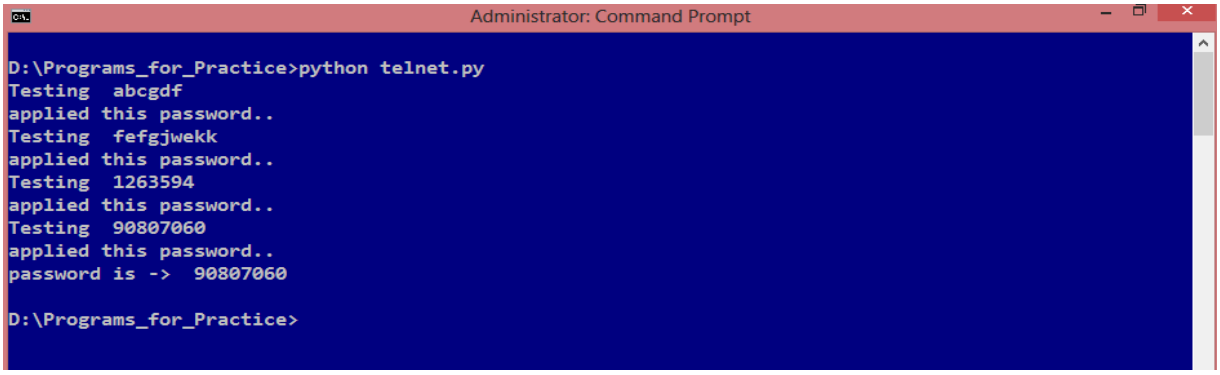

Figure 5.14. Telnet Brute Force Attack

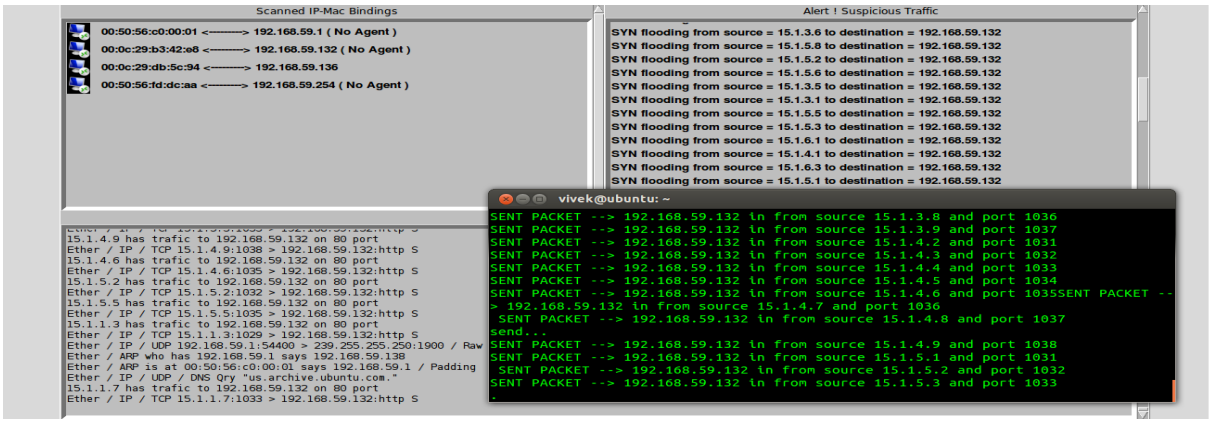

Figure 5.15. DDoS Attack Detection

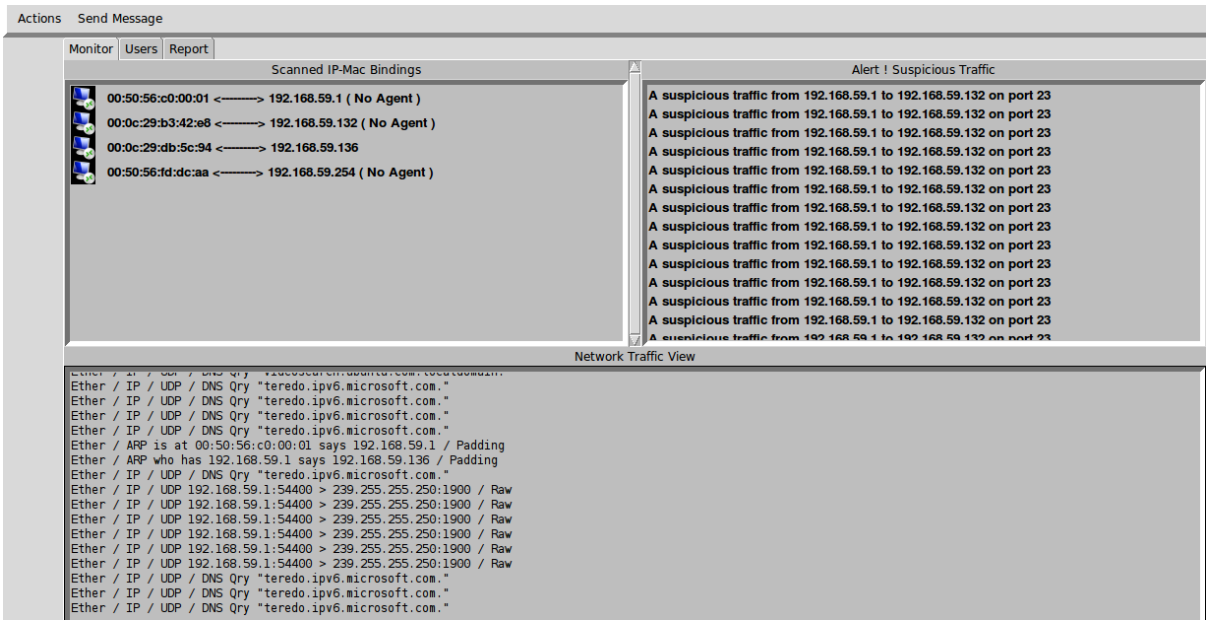

Figure 5.16. Telnet Brute Force Attack Detection

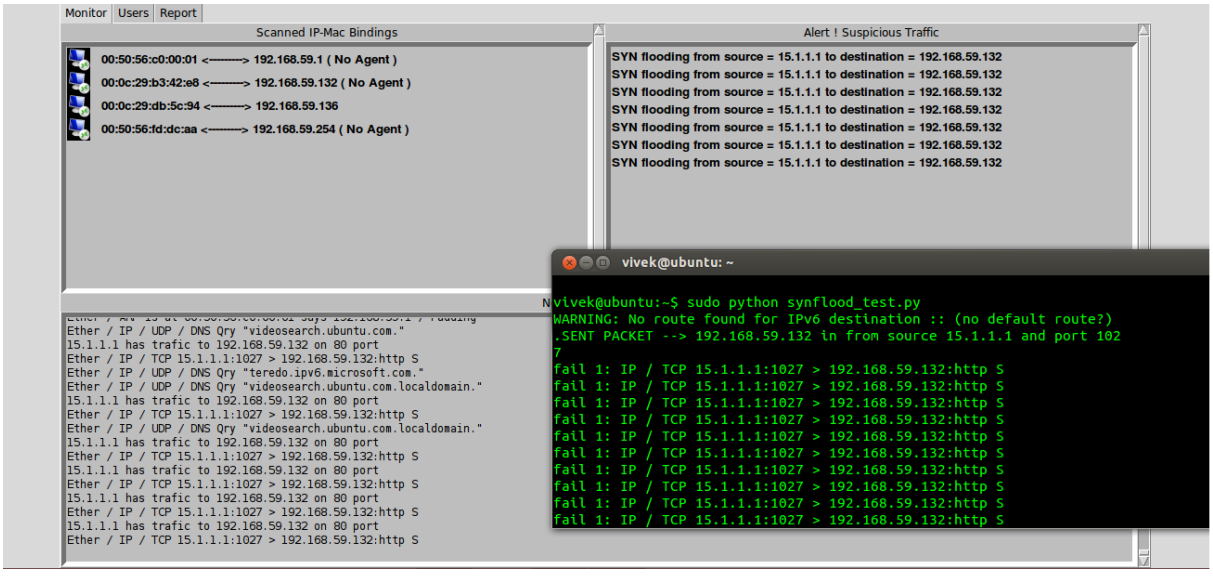

Figure 5.17. SYN Flood Attack Detection

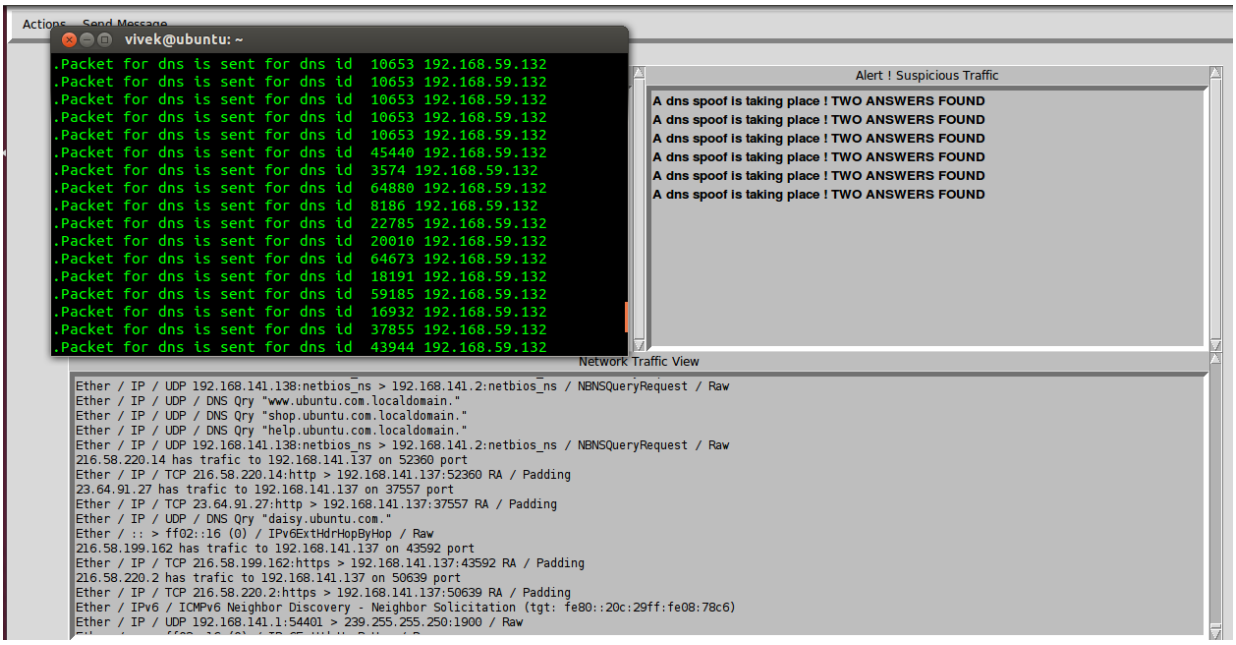

Figure 5.18. DNS Spoof Detection

CHAPTER 6

RESULT

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces electronic reports to a management station. It inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.To put it in simpler terms, an Intrusion detection system can be compared with a burglar alarm. For example, the lock system in a car protects the car from theft. But if somebody breaksthe lock system and tries to steal the car, it is the burglar alarm that detects that the lock has been broken and alerts the owner by raising an alarm. The Intrusion detection system in a similar way complements the firewall security. The firewall protects an organization from malicious attacks from the Internet and the Intrusion detection system detects if someone tries to break in through the firewall or manages to break in the firewall security and tries to have access on any system in the trusted side and alerts the system administrator in case there is a breach in security. Moreover, Firewalls do a very good job of filtering incoming traffic from the Internet; however, there are ways to circumvent the firewall. For example, external users can connect to the Intranet by dialing in through a modem installed in the private network of the organization. This kind of access would not be seen by the firewall. Therefore, an Intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyzes that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization.Advantages of Network based Intrusion Detection Systems: 1. Lower Cost of Ownership: Network based IDS can be deployed for each network segment. An IDS monitors network traffic destined for all the systems in a network segment. This nullifies the requirement of loading software at different hosts in the network segment. This reduces management overhead, as there is no need to maintain sensor software at the host level. 2. Easier to deploy: Network based IDS are easier to deploy as it does not affect existing systems or infrastructure. The network-based IDS systems are Operating system independent. A network based IDS sensor will listen for all the attacks on a network segment regardless of the type of the operating system the target host is running. 3. Detect network based attacks: Network based IDS sensors can detect attacks, which host-based sensors fail to detect. A network based IDS checks for all the packet headers for any malicious attack. Many IP-based denial of service attacks like TCP SYN attack, fragmented packet attack etc. can be identified only by looking at the packet headers as they travel across a network. A network based IDS sensor can quickly detect this type of attack by looking at the contents of the packets at the real time. 4. Retaining evidence: Network based IDS use live network traffic and does real time intrusion detection. Therefore, the attacker cannot remove evidence of attack. This data can be used for forensic analysis. On the other hand, a host-based sensor detects attacks by looking at the system log files. Lot of hackers are capable of making changes in the log files so as to remove any evidence of an attack. 5. Real Time detection and quick response: Network based IDS monitors traffic on a real time. So,

network based IDS can detect malicious activity as they occur. Based on how the sensor is configured, such attack can be stopped even before they can get to a host and compromise the system. On the other hand, host based systems detect attacks by looking at changes made to system files. By this time critical systems may have already been compromised. 6. Detection of failed attacks: A network based IDS sensor deployed outside the firewall can detect malicious attacks on resources behind the firewall, even though the firewall may be rejecting these attempts. This information can be very useful for forensic analysis.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

IDS are becoming the logical next step for many organizations after deploying firewall technology at the network perimeter. IDS can offer protection from external users and internal attackers, where traffic doesn't go past the firewall at all. However, the following points are very important to always keep in mind. If all of these points are not adhered to, an IDS implementation along with a firewall alone cannot make a highly secured infrastructure. 1. Strong identification and authentication: An IDS uses very good signature analysis mechanisms to detect intrusions or potential misuse; however, organizations must still ensure that they have strong user identification and authentication mechanism in place. 2. Intrusion Detection Systems are not a solution to all security concerns: IDS perform an excellent job of ensuring that intruder attempts are monitored and reported. In addition, companies must employ a process of employee education, system testing, and development of and adherence to a good security policy in order to minimize the risk of intrusions. 3. An IDS is not a substitute for a good security policy: As with other security and monitoring products, an IDS functions as one element of a corporate security policy. Successful intrusion detection requires that a well-defined policy must be followed to ensure that intrusions and vulnerabilities, virus outbreaks, etc. are handled according to corporate security policy guidelines. 4. Human intervention is required: The security administrator or network manager must investigate the attack once it is detected and reported, determine how it occurred, correct the problem and take necessary action to prevent the occurrence of the same attack in future. Lastly, tight integration between host and network based IDS is very much necessary. As shown in Picture1, it is advised to use network based IDS inside and outside the firewall or between each firewall in a multi-layered environment and host based IDS on all critical or key hosts. Also it is important although not always necessary to have an integrated deployment of host based and network based Intrusion Detection Systems. As security continues to move to the center stage, managers and network administrators alike are beginning to focus their attention on intrusion-detection technology. While modern-day IDSes are far from bulletproof, they can add significant value to established information-security programs. With vendors working on

eliminating the shortcomings of Intrusion Detection Systems, the future looks brighter for this technology.

7.2Future Work

A feature called host based intrusion detection system or HIDS which complements the working of intrusion detection system can be added. Host Intrusion Detection Systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their configurations.A host-based intrusion detection system (HIDS) is an intrusion detection system that monitors and analyzes the internals of a computing system as well as (in some cases) the network packets on its network interfaces (just like a network-based intrusion detection system (NIDS) would do). This was the first type of intrusion detection software to have been designed, with the original target system being the mainframe computer where outside interaction was infrequent.

Advantages of Host based Intrusion Detection Systems:

1. Verifies success or failure of an attack: Since a host based IDS uses system logs containing events that have actually occurred, they can determine whether an attack occurred or not with greater accuracy and fewer false positives than a network based system. Network based IDS sensors although quicker in response than host based IDS sensors, generate a lot of false positives because of the very fact that it detects malicious packets on the real time and some of these packets could be from a trusted host.

2. Monitors System Activities: A host based IDS sensor monitors user and file access activity including file accesses, changes to file permissions, attempts to install new executables etc. A host based IDS sensor can also monitor all user logon and logoff activity, user activities while connected to the network, file system changes, activities that are normally executed only by an administrator. Operating systems log any event where user accounts are added, deleted or modified. The host based IDS can detect an improper change as soon as it is executed. A network-based system cannot give so much detailed information about system activities.

3. Detects attacks that a network based IDS fail to detect: Host based systems can detect attacks that network based IDS sensors fail to detect. For example, if an unauthorized user makes changes to system files from the system console, this kind of attack goes unnoticed by the network sensors. So, host based sensors can be very useful in protecting hosts from malicious internal users in addition to protecting systems from external users.

4. Near real time detection and response: Although host based IDS does not offer true real-time response, it can come extremely close if implemented correctly. Unlike older systems, which use a process to check the status and content of log files at predefined intervals, many current host-based systems receive an interrupt from the operating system when there is a new log file entry. This new entry can be processed immediately, significantly reducing the time between attack recognition and response.

5. Does not require additional hardware: Host based Intrusion detection sensors reside on the host systems. So they do not require any additional hardware for deployment, thus reducing cost of deployment.

6. Lower entry cost: Host based IDS sensors are far cheaper than the network based IDS sensors.

Analysis of logs can be another feature to the software. Log analysis is one of the most overlooked aspects of intrusion detection. Nowadays every desktop comes with an antivirus, with multiple firewalls.

There are simple techniques like SQL queries to analyze the logs. Suppose an attacker, via the internet, tries an attack on the server, by generating unusual amount of traffic from a single source, the agents will monitor the source of the traffic and report it to the network administrator who in-turn will take the required action against the IP address of the attacker.

The general actions include kicking the IP off the network and blocking it from further activities on the network. This can be done firing a simple SQL query by the administrator where he can apply Boolean concepts like select user from logs where generated traffic is 20 MB under 10 seconds where an ideal user would generate only 1 MB of data.

REFERENCES

1. Joao B. D. Cabrerat B. Ravichandran and K. Raman Mehra*Statistical Traffic Modelingfor Network Intrusion Detection,* 2001

2. J. B. D. Cabrera, Handran B., and R. K. Mehra. *Data Classification and Data Clustering Algorithms for intrusion detection on Computer Networks*. Technical Report AFRL-IS-TR-2000-51, Air Force Research Laboratory, Information Directorate, Rome, NY, April 2000.

3. Emerald Project. *EMERALD TCP Statistical Analyser* 1998 Evaluation Results, 1999. Available at http://www.sdl.sri.com/emerald98-evalestat/index.html

4. S. Floyd and V. Paxson. *Why we do not know how to simulate the Internet*, October 1999. Available at http://www.aciri.org/floyd papers.html

5. Graf, R. Lippmann, R. Cunningham, D. Fried, K. Kendall, S. Webster, and M. Zissman. Results of J998 *Offine DARPA Intrusion Detection Evaluation.* December 1998. DARPA Pl Meeting slides available at http://videval.ll/mit.edu

6. K. Kendall. *A database of computer attacks for the evaluation of detection systems.* Master's thesis, Massachusetts Institute of Technology, June 18, 1999

7. Lee, S. J. Stolfo, and K. W Mok, *A data mining framework for building intrusion detection models.* In Proceedings of the IEEE Symposium on Security and Privacy, 1999

8. W Lee, S. J. Stolfo, and K. W. Mok. *Mining in a flow environment: Experience on network intrusion detection*. In Proceedings of the Conference on knowledge discovery and databases, 1999

9. Shaik Akbar Assoc. Profr, Dept. of C.S.E, SVIET, Nandamuru, Krishna Dist, Andhra Pradesh, India Dr.K.Nageswara Rao Prof & H.O.D, Dept. of C.S.E P.V.P.S.I.T, Vijayawada, Krishna Dist, Andhra Pradesh, India Dr.J.A.Chandulal Prof, Dept. of C.S.E GITAM University, Visakhapatnam, Andhra Pradesh, India, *Intrusion Detection System Methodologies*

Based on Data Analysis, International Journal of Computer Applications (0975 – 8887) Volume 5– No.2, August 2010

10. Honeywell International Inc. *Method and system for monitoring and evaluating user activity in computer systems*, WO2004049251A2

11. Alcatel Alsthom Compagnie Generale D'electricite, *Facility for detecting intruders and suspect callers in a computer installation and a security system including such a facility*, US5621889