Journal of Information Security, 2011, 2, 28-38
doi:10.4236/jis.2011.21003 Published Online January 2011 (http://www.SciRP.org/journal/jis)

A Comparative Study of Related Technologies of Intrusion Detection & Prevention Systems

Indraneel Mukhopadhyay[1], Mohuya Chakraborty[1], Satyajit Chakrabarti[2]
[1]*Department of Information Technology, Institute of Engineering & Management, Kolkata, India*
[2]*Institute of Engineering & Management, Kolkata, India*
E-mail: *imukhopadhyay@gmail.com, mohuyacb@yahoo.com, director@iemcal.com*
Received November 3, 2010; *revised December* 20, 2010; *accepted January* 10, 2011

Abstract

The rapid growth of computer networks has changed the prospect of network security. An easy accessibility condition causes computer networks to be vulnerable against numerous and potentially devastating threats from hackers. Up to the moment, researchers have developed Intrusion Detection Systems (IDS) capable of detecting attacks in several available environments. A boundlessness of methods for misuse detection as well as anomaly detection has been applied. Intrusion Prevention Systems (IPS) evolved after that to resolve ambiguities in passive network monitoring by placing detection systems on the line of attack. IPS in other words is IDS that are able to give prevention commands to firewalls and access control changes to routers. IPS can be seen as an improvement upon firewall technologies. It can make access control decisions based on application content, rather than IP address or ports as traditional firewalls do. The next innovation is the combination of IDS and IPS known as Intrusion Detection and Prevention Systems (IDPS) capable of detecting and preventing attacks from happening. This paper presents an overview of IDPS followed by their classifications and applications. A new signature based IDPS architecture named HawkEye Solutions has been proposed by the authors. Authors have presented the basic building blocks of the IDS, which include mechanisms for carrying out TCP port scans, Traceroute scan, ping scan and packet sniffing to monitor network health detect various types of attacks. Real time implementation results of the system have been presented. Finally a comparative analysis of various existing IDS/IPS solutions with HawkEye Solutions emphasizes its significance.

Keywords: Advances of Network Security, Intrusion Detection System, Intrusion Prevention System, HawkEye Solutions

1. Introduction

The Internet is a worldwide network of interconnected computers enabling users to share information along multiple channels. A computer connected to the Internet is able to access information from a vast array of available servers and other computers by moving information from them to former computer's local memory. Common uses of the Internet are Email, World Wide Web, remote access, collaboration, streaming media and file sharing. But nowadays malfunctions on the Web are increasing. There are computer investment frauds, cyber crimes, financial crimes, phishing scams, chatting (masquerading) and crimes associated which share trading on Web. Network Security consists of the provisions made in an underlined computer network infrastructure and policies adopted by the Network Administrator to protect the network and network accessible resources from unauthorized access, consistent and continuous monitoring and measurement of its effectiveness combined together.

In the last few years networking revolution has finally come of age due to changing nature of Internet computing. However complete prevention of breaches of security is unrealistic. Intrusion detection is the process of monitoring the events occurring in a computer system/ network and analyzing them for signs of possible attacks, which can lead to violations or imminent threats of violation of computer security policies, of the organization. An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents. IDS and IPS technologies offer many of the same capabilities, and administrators can usually disable prevention features in IPS products, causing them to function as IDSs. The combination of IDS and IPS known as Intrusion Detection and Prevention Systems (IDPS) is capable of detecting and preventing attacks from happening. This paper presents an overview of IDPS followed by their classifications and applications. A new signature based IDPS architecture named HawkEye Solutions has been proposed by the authors.

1.1. Meaning of IDS/IPS

IDS generally do not react against occurred attacks and usually have the state of informing administrator for occurrence of an intrusion and have several methods for detecting attacks. Monitoring and analyzing network activities, finding vulnerable parts in network and integrity testing of sensitive and important data are few examples of IDS operations for intrusions detection [1]. Incidents have many causes, malware, attackers gaining unauthorized access to systems, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. Many incidents are malicious in nature; many are not. IPS on the other hand is software that has all the capabilities of IDS and can attempts to stop possible incidents. Accordingly, for brevity the term Intrusion Detection and Prevention Systems (IDPS) is used throughout the rest of this article to refer to both IDS and IPS technologies.

1.2. IDPS Components

Typical components of IDPS and their functionalities are [2]:

- Sensor/Agent: Monitors and analyzes network activity. The term sensor is used for IDPS that monitor networks, including network-based, wireless, and network behavior analysis technologies. The term agent is used for host-based IDPS technologies.
- Database Server: Used as a repository for event information recorded by the sensors or agents processed by the management server.
- Management Server: Centralized device that receives; analyzes and manages event information from the sensors/agents. It identifies events that the sensors/agents cannot.
- Console: Provides an interface for the users and administrators. Console software is typically installed onto standard computers providing both administration and monitoring capabilities.

IDPS are differentiated by the types of events that they can recognize and the methodologies that they use to identify incidents. IDPS typically perform the following functions:

- Recording Information: Event information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management solutions, and enterprise management systems.
- Notifying Security Administrators: Alerts or alarms occur when any of the following like-e-mails, web pages, messages on the IDPS user interface, SNMP traps, syslog messages, and user-defined programs, are detected by the system. A simple notification message includes basic information regarding an event; administrators need to access the IDPS Console for additional information in order to neutralize them.
- Producing Reports: Summarized reports of the monitored events and/or action taken by the administrator based on the details of the particular events.

1.3. Types of IDPS

IDPS perform extensive logging of data that is related to detected events in the network. These data can then be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDPS and other logging sources [2].

- Host-Based: Monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Examples of the types of characteristics a host-based IDPS might monitor are network traffic, system logs, running processes, application activity, file access and modification, and system configuration changes. Host-based are deployed on critical hosts such as publicly accessible servers and servers containing sensitive information.
- Network-Based: Monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. It can identify many different types of events of interest. It is mostly deployed at a boundary between networks, virtual private network servers, remote access servers, and wireless networks.
- Hybrid: Both host-based as well as network-based IDPS may be used simultaneously.
- Network Behavior Analysis (NBA): Examines net-

work traffic to identify threats that generate unusual traffic flows, such as distributed denial of service attacks, certain forms of malware, and policy violations. NBA systems are most often deployed to monitor flows of the internal networks, and are also sometimes deployed where they can monitor flows between an organization's internal networks and external.

The organization of the paper is as follows. After the introduction in Section 1, different techniques of intrusion detection is discussed in Section 2. Section 3 deals with various types of analysis techniques performed by IDPS. Section 4 highlights the related works that act as a motivation for the proposed signature based IDPS architecture called HawkEye Solutions, whose architecture is shown in Section 5. Working principle and features of HawkEye Solutions are presented in Sections 6 and 7 respectively. Snapshots of real time implementation results are shown in Section 8. In Section 9 a comparative analysis of various existing IDS/IPS solutions is made with HawkEye Solutions that emphasizes its significance. Section 10 deals with issues and challenges faced by an IDPS environment. Finally the article is concluded in Section 11 with some highlights on future works.

2. Techniques of Intrusion Detection

Many of the techniques used in attempting to detect intrusion are reviewed here in this section. The most common ones are summarized below.

- Artificial Neural Networks (ANNs): Can be trained to recognize arbitrary patterns in input data, and associate such patterns with an outcome, which can be a binary indication of whether an intrusion has occurred [3].
- State Transition Tables: Describe a sequence of actions an intruder does in the form of a state transition diagram. When the behavior of the system matches those states, an intrusion is detected [4].
- Genetic Algorithms (GAs): Mimic the natural reproduction system in nature where only the fittest individuals in a generation will be reproduced in subsequent generations, after undergoing recombination and random change. The application of GAs in IDS research appeared as early as 1995, and involves evolving a signature that indicates intrusion [5]. A related technique is the Learning Classifier System (LCS), where binary rules are evolved, that collectively recognizes patterns of intrusion.
- Bayesian Network: A set of transition rules are represented as probabilistic interdependencies in a graphical model. Each node contains the state of random variable and a conditional probability table, which determine the probabilities of the node in a state, given a state of its parent [6]. An advantage of the approach is that it can deal with incomplete data.
- Fuzzy Logic: A set of concepts and approaches designed to handle vagueness and imprecision. A set of rules can be created to describe a relationship between the input variables and the output variables, which may indicate whether an intrusion has occurred. Fuzzy logic uses membership functions to evaluate the degree of truthfulness [7].

3. Types of Analysis Techniques

IDPS implementation uses a single technique or a combination of two techniques among the commonly used are:

- Code Analysis: Aims at identifying malicious activity by analyzing attempts to execute code. For example, code-behavior analysis can first execute code in a virtual environment and compare its behavior to profiles or rules; buffer overflow detection identifies typical sequences of instructions that attempt to perform stack and heap buffer overflows.
- Network Traffic Analysis and Filtering: Analyses network, transport and application layer protocols and include processing for common applications. Sensors/Agents often include a host-based firewall that can restrict incoming and outgoing traffic for the system.
- File System Monitoring: Includes a number of methods, such as file integrity checking, file attribute checking; these two methods can only determine after-the-fact if the file has been changed. Some sensors/agents typically those who use a small library the transparently intercepts, are able to monitor all attempts to access critical files and stop attempts that are suspicious. The current attempt is compared against a set of policies regarding file access and blocked if the type of access that has been requested (read-write-execute) contradicts a policy.
- Log Analysis: Some sensors/agents can identify malicious activity by monitoring and analyzing system and application logs, which contain information e.g., shutting down the system, starting a service, application startup and shutdown, failures, configuration changes.
- Network Configuration Monitoring: Sensors are able to monitor a host's current network configuration and detect changes to it. For example, network interfaces being placed in promiscuous mode, ad-

ditional TCP or UDP ports or unusual protocols being used could indicate that the host has already been compromised and is being configured for use in future attacks or for transferring data.

- Process Status Monitoring: Some host-based IDPSs can monitor the status of the processes and services running on a host; when they detect that one has stopped, they restart automatically. This provides protection against some forms of malware which can sometimes disable antivirus software and the like.
- Network Traffic Sanitization: This protection is usually implemented by appliance-based IDPSs. Sanitization of traffic may rebuild all requests and responses directed to the host or coming from it, thus neutralizing certain unusual activity, particularly in packet headers and application protocol headers. It can also reduce the amount of reconnaissance the attackers can perform on the host, by hiding OS fingerprints and application error messages.
- Signature Based: Based on pattern matching. A dictionary of known fingerprints is used and run across a set of input. This dictionary contains a list of known bad signatures, such as malicious network payloads or the file contents of a worm executable. This database of signatures is the key to the strength of the detection system, and its prowess is a direct result of its speed. It uses network payload signatures, as is used in network intrusion detection systems [8]. The detection methods used performs an evaluation of packet contents received from the network, typically using passive capture techniques. This can include matching signatures based on payload contents measured by string comparisons, application protocol analysis, or network characteristics. Lists of unacceptable patterns are compared against a list of network traffic and alerts are issued when a match is found. The biggest drawback to signature-based detection methods is that they are reactionary; they rarely can be used to detect a new worm.
- Anomaly Based: In this model, computer behavior is studied extensively under normal operating conditions [9]. On compromise by a worm, virus, or attacker, the system's behavior is expected to change. A monitoring system can detect these changes and respond accordingly [10]. In this way, the host is able to adapt to its normally changing behavior while remaining responsive to new threats. While such a system would prove to be nearly infinitely adaptive the biggest challenge is the long training time required to develop a reliable baseline of behavior. This assumes that no anomalies occur during this period.

4. Related Works

Easy accessibility condition in wireless networks causes more vulnerability against wired networks. The level of vulnerability has made it mandatory to adopt security policies in wireless networks more now than before. In centralized-IDPS, the analysis of data is performed at a fixed number of locations. But in distributed-IDPS the analysis of data is performed at a number of locations that is commensurate to number of available systems in the network. In ad-hoc-based wireless networks we are forced to use distributed-IDPS because we cannot set of fixed locations/hosts for using centralized IDS [11]. Recently, new methods appear in distributed-IDS categories known as Grid Intrusion Detection system, which uses Grid Computing to detect intrusion packets [12].

Distributed intrusion detection is an ideal approach to the detection of worm activities. As worms spread on the network from host to host, they will quickly cover a large network if left unchecked. As such, a disconnected set of network-IDS monitors will generate an increasing number of alerts. However, with no central infrastructure, the larger picture of a spreading worm will be difficult to gain at an early enough time to contain the spread of the worm [13].

Design of a robust security system should fulfill the objectives of security like authenticity, confidentiality, integrity, availability & non-repudiation. IDPS contains modules to detect intrusion, filtering intrusion, trace-back of intrusion origin, and prevention mechanism for theses intrusions. This security system needs the robust automated auditing and intelligent reporting mechanism and robust prevention techniques. The system should be divided into three sub-systems:

- Intrusion Detection System
- Backtracking of Intrusion Source
- Prevention Techniques

The components of the intrusion detection and prevention system are shown in **Figure 1**. The rule based intelligent intrusion detection and prevention model contains a scheduler to prepare schedule to check different logs for possible intrusions, and detectors to detect normal or abnormal activity. If activity is normal then standard alarming and reporting would be executed.

If abnormal activity is found then the rule engine checks the rule to detect intrusion point and type of intrusion. The model also contains an expert system to detect source of intrusion and suggests best possible prevention technique and suitable controls for different intrusions. This model also uses security audit as well as

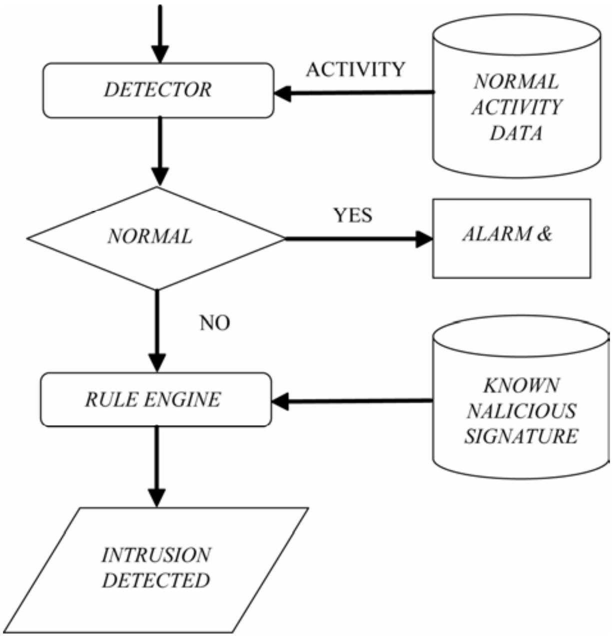


Figure 1. Components of IDPS.

alarming and reporting mechanisms. The malicious activity database is stored for future intrusion detection. To detect the source by tracking, backward chaining approach is used. The rules are defined and are stored in the Rule Engine of the system. Intrusion points & types are passed to the expert system. Expert system evaluates that data with known malicious activity database and detects the source using backward chaining approach. After detecting source, system suggests the different prevention techniques. For this robust security system the authors use intelligent models like expert system.

Expert systems are the most common form of Artificial Intelligence applied today in intrusion detection system. An expert system consists of a set of rules that encode the knowledge of a human "expert". These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Expert system permits the incorporation of an extensive amount of human experience into a computer application and then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack. Expert system detects intrusions by encoding intrusion scenarios as a set of rules. These rules replicate the partially ordered sequence of actions that include the intrusion scenario. Some rules may be applicable to more than one intrusion scenario. Rule-based programming is one of the most commonly used techniques for developing expert systems. Rule based analysis relies on sets of predefined rules that can be repeatedly applied to a collection of facts and that are provided by an administrator, automatically created by the system or both. Facts represent conditions that describe a certain situation in the audit records or directly from system activity monitoring & rules represent heuristics that define a set of actions to be executed in a given situation & describe known intrusion scenario(s) or generic techniques. The rule then *fires*. It may cause an alert to be raised for a system administrator. Alternatively, some automated response, such as terminating that user's session, block user's account will be taken. Normally, a rule firing will result in additional assertions being added to the fact base. They, in turn, may lead to additional rule-fact bindings. This process continues until there are no more rules to be fired. Consider the intrusion scenario in which two or more unsuccessful authentication attempts are made in a period of time shorter than it would take a human to present biometric info in the login information at biometric sensor. If the rule or rules for this scenario fire, then suspicion level of specific user can get increased. The system may raise an alarm or report 'freeze action' to the named user's account. Account freeze would be entered into the fact database.

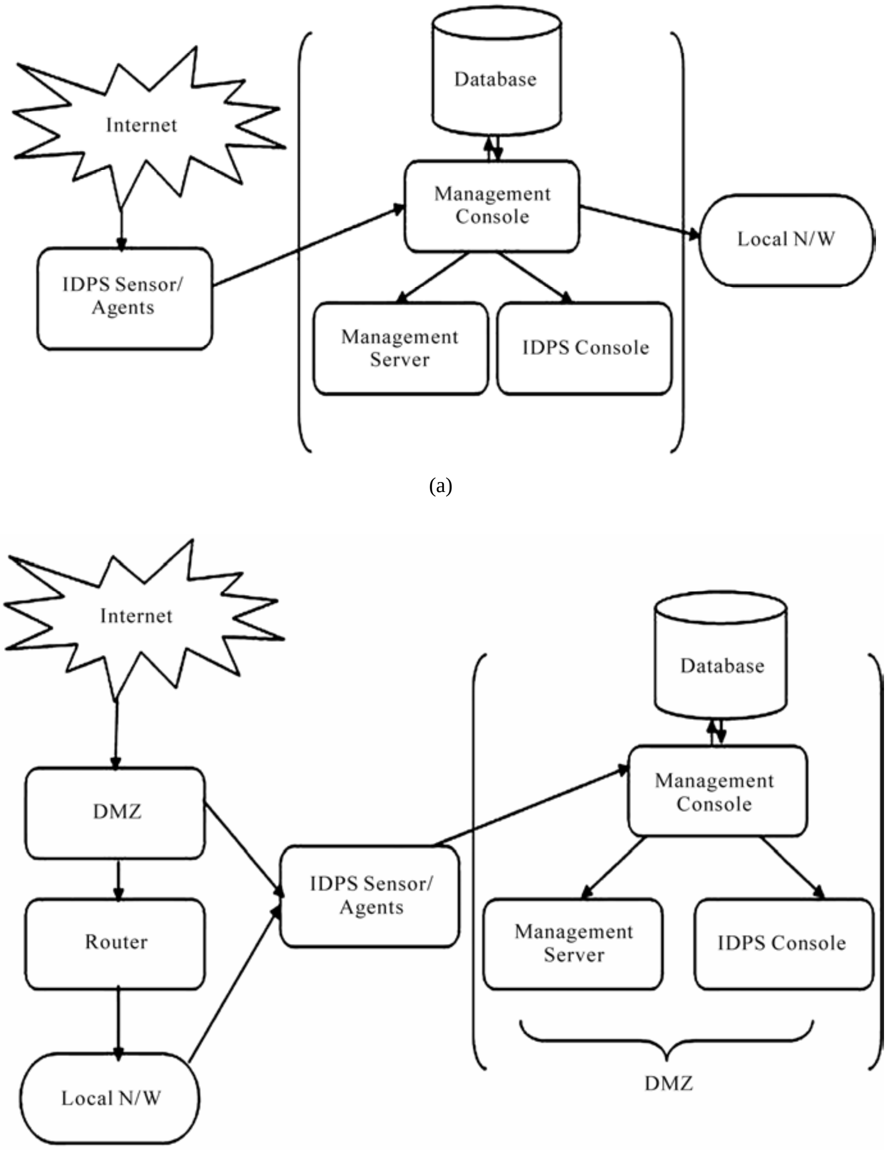
The model suggested in this paper is useful to detect the intrusion and also contains an expert system to detect source of intrusion and suggests best possible prevention technique and suitable controls for different intrusions. This model also uses security audit as well as alarming and reporting mechanisms. The malicious activity database is stored for future intrusion detection. To detect the source by tracking, backward chaining approach is used. The rules are defined and are stored in the Rule engine of the system. The intelligent model uses AI and expert system is backbone of this system.

5. Architecture of HawkEye Solutions

The architecture of HawkEye Solutions is focused on performance, simplicity, and flexibility. The architecture comparison between standard IDPS and HawkEye Solutions is shown in **Figure 2**. **Figure 2(a)** shows the standard IDPS Architecture and **Figure 2(b)** shows HawkEye Solutions Architecture.

The different components of HawkEye Solutions are:

- Sensors/agents monitor and analyze activities.
- Management server is a centralized device which receives and manages information from the sensors or agents.
- Database server is a repository for event information recorded by sensors, agents, and/or management servers.
- Console provides an interface for IDPS's users and administrators.
- Demilitarized Zone (DMZ) works as the primary filter, which has the normal security software's



(b)

Figure 2. Standard IDPS vs. HawkEye solutions architecture. (a) Standard IDPS architecture; (b) HawkEye solutions architecture.

loaded, but for a network it does not mean that the network is safe from attacks. So IDPS is implemented in both the DMZ and also in the network where the sensors/agents monitor attacks. In normal IDPS the DMZ is not available.

Till date research on IPS dealt with the level of threat-risk assessment on the attacked asset based via Hidden Markov Model (HMM) and Fuzzy Risk Assessment [14]. But work must be done to deal with real data with better HMM model. Kalman filter and its integration with agents/sensors could be a good option [15], in this direction the authors have simulated a DoS attack and then used a Kalman Filter to detect foreign intrusion in the network. The filter worked on the data provided by the network router. In the simulation it was seen that due to the use of Kalman Filter with the increase in the number of observations, higher was the estimation accuracy. Kalman filter showed a stabilized oscillation around a constant positive value. It proved that the illegitimate scan activities are mainly caused by a worm infection. If the illegitimate scan traffic is caused by non-worm noise, the traffic does-not grow exponentially, and the estimated value of infection rate would either fluctuate without any point or band of convergence, or it would oscillate around zero.

6. Working Principle of HawkEye Solutions

This section deals with the working principle of HawkEye Solutions. The various steps followed by HawkEye Solutions are as follows:

- An event record is created. This occurs when an action happens; such as packets of data transmitting in the network or even a file is opened or a program is executed like the text editor like Microsoft Word. The record is written into a file that is usually protected by the operating system trusted computing base.
- The target agent transmits the file to the command console. This happens at predetermined time intervals over a secure connection.
- The detection engine, configured to match patterns of misuse, processes the file.
- A log is created that becomes the data archive for all the raw data that will be used in prosecution.
- An alert is generated. When a predefined pattern is recognized, such as access to a mission critical file, an alert is forwarded to a number of various subsystems for notification, response, and storage.
- The security flag/message are sent *i.e.* notified.
- A response is generated. The response subsystem matches alerts to predefined responses or can take response commands from the security officer. Responses include reconfiguring the system, shutting down a target, logging off a user, or disabling an account.
- The alert is stored. The storage is usually in the form of a database. Some systems store statistical data as well as alerts.
- The raw data is transferred to a raw data archive. This archive is cleared periodically to reduce the amount of disk space used.
- Reports are generated. Reports can be a summary of the alert activity.
- Data forensics is used to locate long-term trends and behavior is analyzed using both the stored data in the database and the raw event log archive.

The flow diagram of the steps discussed above is shown in **Figure 3**. The lifecycle of an event recorded through the proposed architecture is advantageous as everything hap-pens in real-time. The disadvantage is that the end users suffer from system performance degradation.

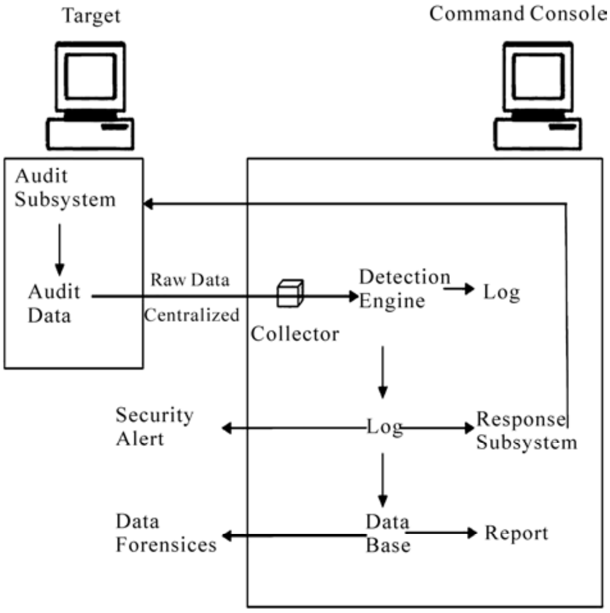


Figure 3. Flow diagram of working principle of HawkEye Solutions.

7. Features of HawkEye Solutions

This section describes the various features of HawkEye Solutions that has been developed viz., Ping Scan, Trace Route Scan, TCP Scan and Packet Sniffing.

7.1. Ping Scan

The Internet Ping command bounces a small packet off a domain or IP address to test network communications, and then tells how long the packet took to make the round trip [16]. The Ping command is one of the most commonly used utilities on the Internet by both people and automated programs for conducting the most basic network test, which is to test whether one computer can reach another computer on the network, and if so the time it takes. It works by sending a small packet of information containing an ICMP ECHO_REQUEST to a specified computer, which then sends an ECHO_REPLY packet in return [17].

7.2. Trace Route Scan

The Trace Route scan traces the network path of Internet routers that packets take as they are forwarded from your computer to a destination address. The “length” of the network connection is indicated by the number of Internet routers in the trace route path. Trace routes can be useful to diagnose slow network connections. For example, if one can usually reach an Internet site but it is slow today, then a trace route to that sites should show you one or more hops with either long times or marked with “*” indicating the time was really long.

7.3. TCP Scan

The process of scanning TCP ports involves probing each and every port for a specific domain name to check the status of the ports so as to determine which ports are open, closed or dropped. It will enable the network administrator to also view the services by which the concerned domain name is connected with the host computer [18,19].

7.4. Packet Sniffing

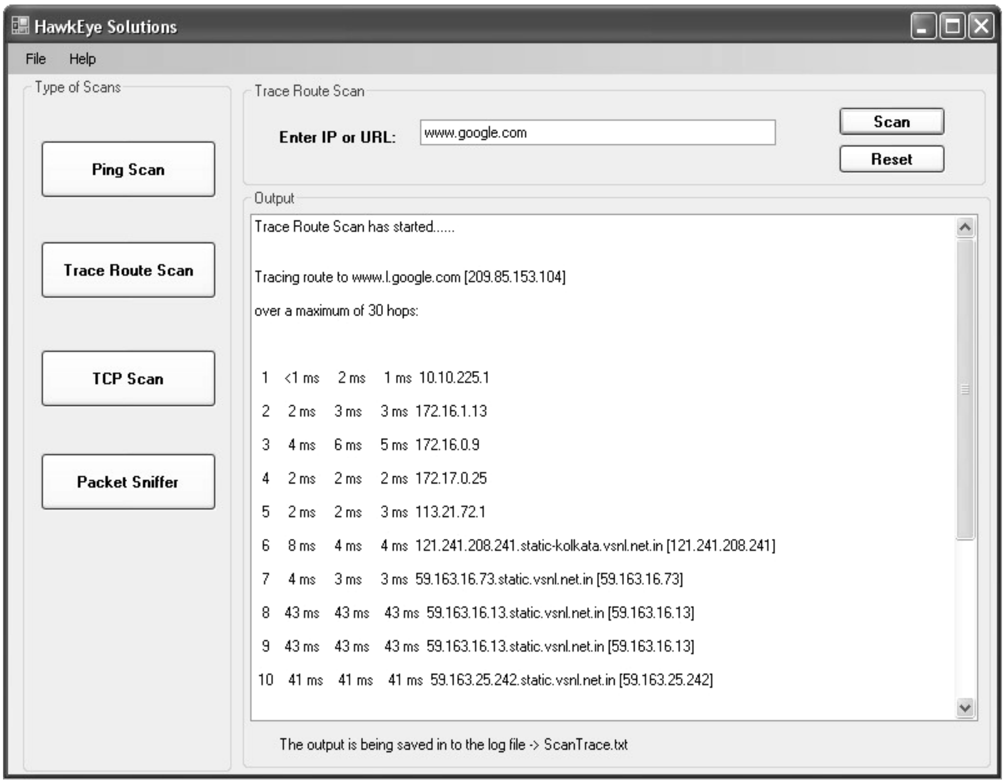
A Sniffer is a program that eavesdrops on the network traffic by grabbing information traveling over a network. A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used legitimately by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission.

In its simple form a packet sniffer simply captures all of the packets of data that pass through a given network interface. Typically, the packet sniffer would only capture packets that were intended for the machine in question. However, if placed into promiscuous mode, the packet sniffer is also capable of capturing packets traversing the network regardless of destination.

8. Implementation Results

This section provides the real time implementation results of HawkEye Solutions for trace route scan and abnormal packet detection through its packet sniffing utility.

Figure 4 shows the screenshot of trace route scan. On selecting the Trace Route Scan option, a textbox appears on the right hand panel that requests the user to enter the IP address or URL of the destination to be traced. The output consists of 3 columns corresponding to each router or hop. Each of the 3 columns is a response from the concerned router in terms of how long it took (each hop is tested 3 times). The result of the scan is shown in the output text box and is automatically saved into the log file ScanTrace.txt. **Figure 5** shows the screenshot of packet sniffing utility. On selecting the Packet Sniffer option and on clicking the Start button, the sniffing of packets starts with the packet details and data of each packet shown instantaneously. The information shown in the figure includes the details of Ethernet header, IP header and TCP/UDP header [20]. The packet sniffer also detects the abnormal packets (if any) and the cause for the abnormality for individual packets. The screenshot of the result is shown in **Figure 6**. These are displayed

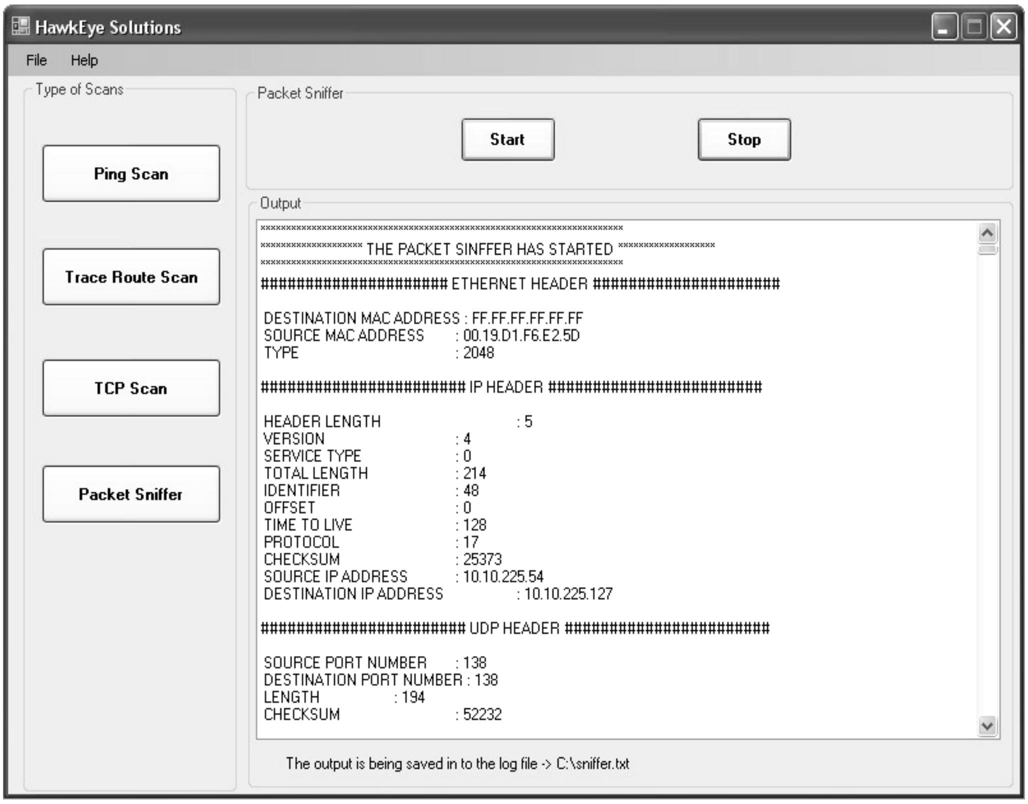


Enter IP or URL: www.google.com

```
Trace Route Scan has started......

Tracing route to www.l.google.com [209.85.153.104]
over a maximum of 30 hops:


 1   <1 ms    2 ms    1 ms  10.10.225.1
 2    2 ms    3 ms    3 ms  172.16.1.13
 3    4 ms    6 ms    5 ms  172.16.0.9
 4    2 ms    2 ms    2 ms  172.17.0.25
 5    2 ms    2 ms    3 ms  113.21.72.1
 6    8 ms    4 ms    4 ms  121.241.208.241.static-kolkata.vsnl.net.in [121.241.208.241]
 7    4 ms    3 ms    3 ms  59.163.16.73.static.vsnl.net.in [59.163.16.73]
 8   43 ms   43 ms   43 ms  59.163.16.13.static.vsnl.net.in [59.163.16.13]
 9   43 ms   43 ms   43 ms  59.163.16.13.static.vsnl.net.in [59.163.16.13]
10   41 ms   41 ms   41 ms  59.163.25.242.static.vsnl.net.in [59.163.25.242]
```

The output is being saved in to the log file -> ScanTrace.txt

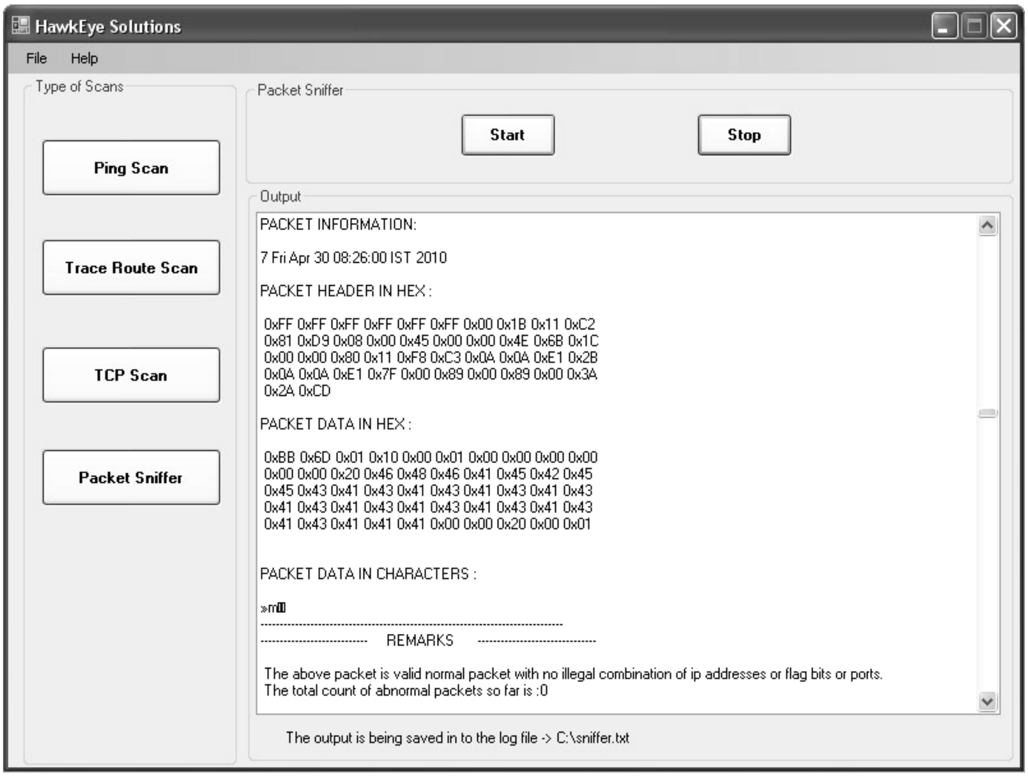
Figure 4. Screenshot of trace route scan of HawkEye solutions.



```
*************************************************************************
******************* THE PACKET SINFFER HAS STARTED *******************
*************************************************************************
###################### ETHERNET HEADER ######################

 DESTINATION MAC ADDRESS : FF.FF.FF.FF.FF.FF
 SOURCE MAC ADDRESS      : 00.19.D1.F6.E2.5D
 TYPE                    : 2048

######################## IP HEADER ########################

 HEADER LENGTH                     : 5
 VERSION                 : 4
 SERVICE TYPE            : 0
 TOTAL LENGTH            : 214
 IDENTIFIER              : 48
 OFFSET                  : 0
 TIME TO LIVE            : 128
 PROTOCOL                : 17
 CHECKSUM                : 25373
 SOURCE IP ADDRESS       : 10.10.225.54
 DESTINATION IP ADDRESS            : 10.10.225.127

######################## UDP HEADER ########################

 SOURCE PORT NUMBER      : 138
 DESTINATION PORT NUMBER : 138
 LENGTH           : 194
 CHECKSUM                : 52232
```

The output is being saved in to the log file -> C:\sniffer.txt

Figure 5. Screenshot of packet sniffing utility of HawkEye solutions.



```
PACKET INFORMATION:

7 Fri Apr 30 08:26:00 IST 2010

PACKET HEADER IN HEX :

 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x00 0x1B 0x11 0xC2
 0x81 0xD9 0x08 0x00 0x45 0x00 0x00 0x4E 0x6B 0x1C
 0x00 0x00 0x80 0x11 0xF8 0xC3 0x0A 0x0A 0xE1 0x2B
 0x0A 0x0A 0xE1 0x7F 0x00 0x89 0x00 0x89 0x00 0x3A
 0x2A 0xCD

PACKET DATA IN HEX :

 0xBB 0x6D 0x01 0x10 0x00 0x01 0x00 0x00 0x00 0x00
 0x00 0x00 0x20 0x46 0x48 0x46 0x41 0x45 0x42 0x45
 0x45 0x43 0x41 0x43 0x41 0x43 0x41 0x43 0x41 0x43
 0x41 0x43 0x41 0x43 0x41 0x43 0x41 0x43 0x41 0x43
 0x41 0x43 0x41 0x41 0x41 0x00 0x00 0x20 0x00 0x01


PACKET DATA IN CHARACTERS :

»m▯▯
--------------------------------------------------------------------------------
--------------------------    REMARKS    ------------------------------

 The above packet is valid normal packet with no illegal combination of ip addresses or flag bits or ports.
 The total count of abnormal packets so far is :0
```

The output is being saved in to the log file -> C:\sniffer.txt

Figure 6. Screenshot of abnormal packet detection by HawkEye solutions.

along with the total count of abnormal packets discovered up to that instant. The data contained in the packet is displayed in the hexadecimal and string format. The result of the scan is shown in the output text box and is automatically saved into the log file sniffer.txt.

9. Comparative Analysis

A comparative analysis of HawkEye Solutions with other signature-based IDS/IPS solutions like Snort Inline, Strataguard, IntruPro IPS, and Packet Alarm [21] is made. **Table 1** shows the comparative analysis chart vis-à-vis design parameters that include IDS as well as IPS. The table clearly indicates that HawkEye Solutions at it stands today is able to meet some of the design parameters that are not met by IntruPro IPS like personalized rule creation and vulnerabilities scanner.

Features of HawkEye Solutions which scores over other available IDS/IPS are:

- Capturing packets, organized by TCP or UDP threads.
- Passively monitoring network.
- Packet viewing and logging in Hex-format.
- Detection of abnormal packet on comparison with benchmark ones and stating cause of abnormality. In case of abnormality the Source IP address can be traced.

The Ping Scan and Packet Sniffing utility the user has a chance of detecting an IP Spoofing. Detected IP can be blocked.

10. Issues and Challenges

Majority of the past research employed analysis was based on data sourced from audit trails, system calls and network traffic. In the network traffic, most research studies looked at the packet header for analysis. Some other research analyzed the payload. Analyzing the packet header is prone to IP address spoofing, while analyzing the payload is prone to data encryption. Several papers also presented the kernel as a data source [22]. IDS assume that signatures of the malware would remain unchanged during the malware's lifetime at present. But if the malware code mutates then the detector (IDS/IDPS) cannot recognize the signature until the new signature has been integrated with its database [23].

11. Conclusions

It is not realistic to accept that IDPS should be capable of detecting all attacks and also prevent them from happening. Perfect detection and prevention is simply not an attainable goal given the complexity and rapid evolution in both attacks and systems. Nowadays even malware developers are creating self mutating worms, which are very hard to detect even for an IDPS. In this article a new type of signature based IDPS–HawkEye Solutions has been discussed which can detect abnormal packets, blocks

Table 1. Comparison of different IDS with HawkEye solutions vis-à-vis design parameters.

Design Parameters	Performance Analysis of various IDS/IPS				
	Snort Inline (IDS)	Strata Guard (IDS)	IntruPro (IPS)	HawkEye Solutions (IDS)	Packet Alarm (IDS)
Anomalies Detection.	√	√	√	√	√
Firewall Inclusion					
IP Tunnels Inspection					
IPv6 Support		√			
Protection against DoS Attack			√	√	√
Personalized Rule Creation	√			√	
Automatic Rules Actualization	√		√	√	
Vulnerabilities Scanner				√	√
Multi-sensor Management		√	√		√
Secure Management (SSH/HTTPS)		√			√
Remote Management		√	√	√	√
Reports Generation		√	√	√	

attacking IP addresses and generates reports. Much work is yet to be done on this solution that should fulfill monitoring of network traffic, creation of per-flow packet traces and adaptive learning of intrusion, inclusion of firewall. It should be able to capture a wide variety of hard-to-see protocol-bug-based attacks, SYN Flood, Land, Teardrop, Smurf and whatever has not been invented yet**.**

12. References

[1] S. Northcutt and J. Novak, "Network Intrusion Detection: An Analyst's Handbook," 2nd Edition, New Riders Publishing, Berkeley, 2000.

[2] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," *NIST Special Publication*, February 2007, pp. 800-94

[3] A.-S.Mohammad and Z. Mohammad, "Efficacy of Hidden Markov Models over Neural Networks in Anomaly Intrusion Detection," 30*th Annual International Computer Software and Applications Conference*, Chicago, 2006, pp. 325-332.

[4] K. Ilgun, R. A. Kemmerer and P. A. Porras, "State Transition Analysis: A Rule-based Intrusion Detection Approach," *IEEE Transactions on Software Engineering*, Vol. 21, No. 3, March 1995, pp. 181-199. doi:10.1109/32.372146

[5] M. Crosbie and E. Spafford, "Applying Genetic Programming to Intrusion Detection," *GECCO* '96 *Proceedings of the First Annual Conference on Genetic Programming* 1996..

[6] F. Jemili, M. Zaghdoud and M. B. Ahmed, "A Framework for an Adaptive Intrusion Detection System using Bayesian Network," *IEEE Intelligence and Security Informatics*, May 2007, pp. 66-70. doi:10.1109/ISI.2007.379535

[7] A. El-Semary, J. Edmonds, J. Gonzalez and M. Papa, "A Framework for Hybrid Fuzzy Logic Intrusion Detection Systems," 14*th IEEE International Conference on Fuzzy Systems*, May 2005, pp. 325-330. doi:10.1109/FUZZY.2005.1452414

[8] R. Bace and P. Mell, "Intrusion Detection Systems," 2001. http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf

[9] S. Forrest, *et al.*, "A Sense of Self for UNIX Processes," *Proceeding of* 1996 *IEEE Symposium on Research in Security and Privacy*, 1996, pp. 120-128.

[10] J. O. Kephart, *et al.*, "Blueprint for a Computer Immune System," *Proceedings* 1997 *Virus Bulletin International Conference*, San Francisco, 1-3 October 1997.

[11] A. Abraham, *et al.* "Fuzzy Online Risk Assessment for Distributed Intrusion Prediction and Prevention Systems," 10*th International Conference on Computer Modeling and Simulation*, *UKSim/EUROSim*, Cambridge, 2008, pp. 216-223.

[12] F. Y. Leu, J. C. Lin, M. C. Li, C. T. Yang and P. C. Shih, "Integrating Grid with Intrusion Detection," *Proceedings of* 19*th International Conference on Advanced Information Networking and Applications*, 2005, pp. 304-309.

[13] Jose Nazario, "Defense and Detection Strategies Against Internet Worms," Artech House, London, 2004

[14] A. Abraham *et al.* "DIPS: A Framework for Distributed Intrusion Prediction and Prevention Systems Using Hidden Markov Model and Online Fuzzy Risk Assessment," *Proceedings of* 3*rd International Symposium on Information Assurance and Security*, Manchester, 29-31 August 2007, pp. 183-188.

[15] I. Mukhopadhyay, *et al.*, "Implementation of Kalman Filter in Intrusion Detection System," *Proceeding of International Symposium on Communications and Information Technologies*, Vientiane, 21-23 October 2008.

[16] RFC 791, "Internet Protocol," http://www.faqs.org/rfcs/rfc791.html

[17] "Assigned Internet Protocol Numbers," 17 May 2010. http://www.iana.org/assignments/protocol-numbers/protocol-numbers. xml,

[18] Version of the Internetwork General Protocol, 27 June 2007. http://www.isi.edu/in-notes/iana/assignments/version-numbers

[19] RFC 793, "Transmission Control Protocol," http://www.faqs.org/rfcs/rfc793.html

[20] RFC 768, "User Datagram Protocol," http://www.faqs.org/rfcs/rf c768.html

[21] E. Guillen, D. Padilla and Y. Colorado, "Weakness and Strength Analysis over Network-Based Intrusion Detection and Prevention System," *IEEE Latin-American Conference on Communications,* 2009.

[22] K. Byung-Joo and K. Il-Kon, "Kernel Based Intrusion Detection System," *Proceedings of* 4*th Annual ACIS International Conference on Computer and Information Science*, Jeju Island, 14-16 July 2005, pp. 13-18. doi:10.1109/ICIS.2005.78

[23] Danilo Bruschi, Lorenzo Martignoni and Martia Monga, "Code Normalization for Self-Mutating Malware," *IEEE Security* & *Privacy*, Vol. 5, No. 2, 2007. pp 46-54.