Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 16 Issue 4 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Towards Configured Intrusion Detection Systems

By Gagan Deep Sharma & Vivek Kumar
GGSIP University

Abstract- This paper studies the challenges in the current intrusion detection system and comparatively analyzes the active and passive response systems. The paper studies the existing IDS and their usefulness in detecting and preventing attacks in any type of network and control traffic with the performance of the system to be improved. The study also evaluates the emerging avenues in Intrusion Detection System and explores the possible future avenues in intrusion detection scheme. It is observed that the detection-based systems have started to gain popularity in the IT security domain. The paper highlights the need to implement an appropriately configured IDS since an optimally configured IDS deters hackers, thus, reducing the need for investigation by security experts for security violations.

Keywords: *Intrusion detection system, response systems, detection-based systems, configured IDS, security violations.*

GJCST-E Classification : *C.2.1 C.2.2*

Strictly as per the compliance and regulations of:

Towards Configured Intrusion Detection Systems

Gagan Deep Sharma [α] & Vivek Kumar [σ]

Abstract- This paper studies the challenges in the current intrusion detection system and comparatively analyzes the active and passive response systems. The paper studies the existing IDS and their usefulness in detecting and preventing attacks in any type of network and control traffic with the performance of the system to be improved. The study also evaluates the emerging avenues in Intrusion Detection System and explores the possible future avenues in intrusion detection scheme. It is observed that the detection-based systems have started to gain popularity in the IT security domain. The paper highlights the need to implement an appropriately configured IDS since an optimally configured IDS deters hackers, thus, reducing the need for investigation by security experts for security violations.

Keywords: Intrusion detection system, response systems, detection-based systems, configured IDS, security violations.

I. Introduction

Data systems and computer networks are central in modern social club. The more data stored and processed, the more significant it is to secure computer systems. Widespread use and proliferation of computer network has increased the attacks on new age information systems. These attacks are attempts to take illegal/unauthorized access to information available with an intention of misusing the same. These attacks result in major financial loss to organizations in the form of mistrust of customers, loosing goodwill. Any set of processes that attempt to compromise the integrity, confidentiality, or availability of a computer resource, is known as intrusion *(Zamboni, 2001)*. Generally an intruder is defined as a system, program or person who tries to and may become successful to stop into an information system or perform an action legally not allowed *(Graham, 2000)*.

The act of detecting actions that try to compromise the integrity, confidentiality, or availability of a computer resource can be referred as intrusion detection *(Zamboni, 2001)*. Intrusion Detection (ID) refers to all processes used in discovering unauthorized uses of network or computer devices through specifically designed software with a sole purpose of detecting unusual or abnormal activity. *Denning (1987)* proposes intrusion detection as an approach to counter, the information processing system and networking attacks and misuses *(Denning, 1987 and Botha & Solms, 2004)*.

Intrusion detection is carried out by an intrusion detection scheme. There are many commercial intrusion detection systems available and most of these commercial implementations are relatively ineffective and insufficient, which gives rise to the need for research on more dynamic intrusion detection schemes. An intrusion detection system is a device or software application that monitors network and/or system actions for malicious actions or policy violations and produces reports *(Scarfone and Mell, 2007)*. IDS is also understood as an instrument that complements a spacious scope of users used to experience some tier of protection *(Vigna et al, 2002)*.

For an IDS to be efficient, it must run continuously adapt to behavioral alterations and large sums of data, be configurable, do not apply too much memory resources of the machine and after system failures, be reusable without new learning *(Zamboni, 1998)*.

Traditional methods for intrusion detection are based on extensive knowledge of attack signatures that are provided by human experts. The signature database has to be manually revised for each new type of intrusion that is discovered. A significant limitation of signature-based methods is that they cannot detect novel attacks. In addition, once a new attack is discovered and its signature developed, often there is a substantial latency in its deployment. These limitations have led to an increasing interest in intrusion detection techniques based upon data mining, which generally fall into one of two categories: misuse detection and anomaly detection.

To prevent attacks or reduce their severity, many solutions exist, but no one can be considered satisfactory and all over. The intrusion detection schemes are one of the most efficient solution. Their purpose is to recognize intrusions or intrusion attempts by users or abnormal behavior by the identification of an onslaught from the stream network data. Different methods and approaches are available in the design of intrusion detection systems.

There are a variety of tools providing a certain level of comfort with acceptable risks used in the defence and surveillance of computer networks. Defence-in-Depth is a term encompassing

Author α: University School of Management Studies, GGSIP University, New Delhi. e-mail: angrishgagan@gmail.com
Author σ: Bureau Veritas Consumer Products Services India, Noida, Uttar Pradesh, India. e-mail: bhardwaj.vivekkumar@gmail.com

comprehensive analyst training, hardware deployed in strategic positions and a strong security policy necessary for achieving this objective. There are tools available to reach this goal. The aggregation of data comes from routers, the host itself, firewalls, virus scanners and IDS, the tool strictly designed to catch known attacks (SANS Institute, 2001).

Since the introduction of IDS, Cyber-attacks have been a real threat. With their wide variety and specialty, they can have catastrophic consequences. To prevent attacks or reduce their severity, many solutions exist, but no one can be considered satisfactory and complete. The intrusion detection systems are among the most effective solution. Their role is to recognize intrusions or intrusion attempts by users or abnormal behavior by the recognition of an attack from the stream network data.

Anderson (1972) delineates the fact the United States Air Force [USAF] "became increasingly aware of computer security problems. This problem feels virtually in every aspect of USAF operations and governance".

USAF faces the daunting tasks of providing shared use of their computer systems, which contained various levels of classifications in a need to know environment with a user base holding various levels of security clearance. Thirty years ago, this created a grave problem that is still with us today. The problem remains: "How to safely secure separate classification domains on the same network without compromising security?" *(Anderson, 1972)*.

Denning (1984) and Neumann (1986) undertake the R&D project with the first model of a real-time IDS. This prototype was named the Intrusion Detection Expert System (IDES). This IDES was initially a rule-based expert system trained to detect known malicious activity.

Some of the most common terms in context of IDS are as follows:

(a) Host-Based
(b) Network-Based
(c) Anomaly Detection Model
(d) Misuse Detection Model

These models are used as terms in Intrusion Detection user and research community. People from different areas have researched and developed few systems to deal with these kind of issues *(SANS Institute, 2001)*

a) Why should Intrusion Detection Systems be used?

Intrusion detection allows organizations to protect their systems from the threats that come with increasing network connectivity and reliance on information systems. Passed on the grade and nature of modern network security threats, the question to security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to employ.

IDSs have gained acceptance as a necessary addition to every organization's security infrastructure. Despite the documented contributions intrusion detection technologies make to system security, in many organizations one must still justify the acquisition of IDSs *(Bace and Mell, 2001)* There are several compelling reasons to acquire and use IDSs:

(a) To prevent problem behaviors by increasing the perceived danger of discovery and punishment for those who would assault or otherwise misuse the scheme.
(b) To detect attacks and other security violations that are not prevented by other protection criteria.
(c) To identify and handle with the preambles to attacks (commonly viewed as network probes and other "doorknob rattling" activities).
(d) To document the existing threat to an establishment.
(e) To act as quality control for security design and administration, especially of large and complex enterprises.
(f) To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors.

b) Major types of IDSs

IDSs have various types, characterized by different monitoring and analysis approaches. Each approach has inherent advantages and disadvantages. Furthermore, all approaches can be described in terms of a generic process model for IDSs. *(Bace & Mell, 2001)*

Many IDSs *(Bace & Mell, 2001)* can be described in terms of three fundamental functional components:

- Information Sources – the different sources of event information used to determine whether an intrusion has taken place. These roots can be traced from different stages of the system, with network, host, and application monitoring. On this basis, the following types of IDS have been observed –

o *Network-based IDSs:* The majority of commercial intrusion detection systems is network based. These IDSs detect attacks by capturing and analyzing network packets. Listening on a network segment or switch, one network-based IDS can monitor the network traffic affecting multiple hosts that are connected to the network segment, thereby protecting those hosts. Network-based IDSs often consist of a set of single-purpose sensors or hosts placed at various points in a network. These units monitor network traffic, performing local analysis of that traffic and reporting attacks to a central management console. As the sensors are limited to running the IDS, they can be more easily secured against attack. Many of these detectors are designed to function in "stealth" mode, in

parliamentary procedure to attain it more unmanageable for an assailant to influence their presence and placement.

- *Host-based IDSs:* Host-based IDSs operate on information collected from within an individual computer system (Application-based IDSs are actually a subset of host-based IDSs). This vantage point allows host based IDSs to analyze activities with great reliability and precision, determining exactly which processes and users are involved in a particular attack on the operating system. Furthermore, unlike network based IDSs, host-based IDSs can "see" the outcome of an attempted attack, as they can directly access and monitor the data files and system processes usually targeted by attacks. Host-based IDSs normally utilize information sources of two types, operating system audit trails, and system logs. Operating system audit trails are usually generated at the innermost (kernel) level of the operating system, and are therefore more detailed and better protected than system logs. However, system logs are much less obtuse and a lot smaller than audit trails, and are furthermore far easier to grasp. Some server-based IDSs are designed to sustain a centralized IDS management and accounting infrastructure that can tolerate a single management console to pass over many hosts. Others generate messages in formats that are compatible with network management systems.
- *Application-based IDSs:* Application-based IDSs are a special subset of host-based IDSs that analyze the events transpiring within a software application. The most common information sources used by application-based IDSs are the application's transaction log files. The ability to interface with the application directly, with significant domain- or application-specific knowledge included in the analysis engine, allows application-based IDSs to detect suspicious behavior due to authorized users exceeding their mandate. This is because such problems are more likely to appear in the interaction between the user, the data, and the application.

- Analysis – the part of intrusion detection systems that actually organizes and makes sense of the events derived from the information sources, deciding when those events indicate that intrusions are occurring or have already taken place. The most common analysis approaches are misuse detection and anomaly detection. The following forms of IDS are observed on this basis –

- *Misuse Detection:* Misuse detectors analyze system activity, looking for events or sets of events that match a predefined pattern of events that describe a known attack. As the patterns corresponding to known attacks are called signatures, misuse detection is sometimes called "signature-based detection." The most common form of misuse detection used in commercial products specifies each pattern of events corresponding to an attack as a separate signature. Nevertheless, in that respect are more sophisticated approaches to doing misuse detection (called "state-based" analysis techniques) that can leverage a single signature to find groups of approaches.
- *Anomaly Detection:* Anomaly detectors identify abnormal, unusual behavior (anomalies) on a host or network. They operate on the assumption that attacks are different from "normal" (legitimate) activity and can therefore be detected by systems that distinguish these conflicts. Anomaly detectors construct profiles representing normal behavior of users, hosts, or network connections. These profiles are built from historical information accumulated over a period of normal functioning. The detectors then collect event data and use a variety of measures to determine when monitored activity deviates from the norm.

- Response – the set of actions that the system takes once it detects intrusions. These are typically grouped into active and passive measures, with active measures involving some automated intervention on the part of the system, and passive measures involving reporting IDS findings to humans, who are then expected to take action based on those reports. The forms of IDS under this section are as under –

- *Active IDS:* Active IDS responses are automated actions taken when certain types of intrusions are detected. In that respect are three categories of active responses.

  - *Collect additional information over time:* In the IDS case, this might involve increasing the level of sensitivity of information sources (for instance, turning up the number of events logged by an operating system audit trail, or increasing the sensitivity of a network monitor to capture all packets, not just those targeting a particular port or target system.) Collecting additional information is helpful for several reasons. The additional information collected can help resolve the detection of the attack. (Assisting the system in diagnosing whether an attack did or did not take place.) This option also allows the organization to gather information that can be used to support investigation and apprehension of the attacker, and to support criminal and civil legal remedies.
  - *Technological Change and the Environment:* Another active response is to stop an attack in advance and then block subsequent access by the assailant. Typically, IDSs do not possess the power

to stop a specific person's access, but instead block Internet Protocol (IP) addresses from which the attacker seems to be doing. It is very difficult to block a determined and knowledgeable attacker, but IDSs can often deter expert attackers or stop novice hackers by (a) injecting TCP reset packets into the attacker's connection to the victim system, thereby terminating the connection ; (b) reconfiguring routers and firewalls to block packets from the attacker's apparent location (IP address or site); (c) reconfiguring routers and firewalls to block the network ports, protocols, or services being used by an attacker; (d) in extreme situations, re-configuring routers and firewalls to sever all connections that use certain network interfaces.

- *Take Action Against the Intruder:* Some who follow intrusion detection discussions, particularly in information warfare circles, consider that the first option in active response is to call for action against the trespasser. The most aggressive form of this response involves launching attacks against or attempting to actively gain information about the attacker's host or site. However tempting it might be, this response is ill advised. Due to legal ambiguities about civil liability, this option can represent a bigger peril that the attack it is designated to stop. The first reason for approaching this option with a large deal of carefulness is that it may be illegal. Furthermore, as many attackers use false network addresses when attacking systems, it carries with it a high risk of causing damage to innocent Internet sites and users. Finally, strike back can escalate the attack, provoking an attacker who originally thought just to browse a site to contain more aggressive activity.

o *Passive IDS:* Passive IDS responses provide information to system users, relying on humans to take subsequent action based on that information. Many commercial IDSs rely solely on passive responses.

- *Alerts and Notifications:* Alerts and notifications are generated by IDSs to inform users when attacks are discovered. Most commercial IDSs allow users a large deal of latitude in finding out how and when alarms are generated and to whom they are exhibited. The most usual kind of alarm is an onscreen alert or popup window. This is displayed on the IDS console or on other systems as specified by the user during the configuration of the IDS. The information provided in the alarm message varies widely, ranging from a notification that an intrusion has taken place to extremely detailed messages outlining the IP addresses of the source and target of the attack, the specific attack tool used to gain access, and the outcome of the attack. Another set of options that are of utility to large or distributed organizations are those involving remote notification of alarms or alerts. These allow organizations to configure the IDS so that it sends alerts to cellular phones and pagers carried by incident response teams or system security personnel.

- *SNMP Traps and Plug-ins:* Some commercial IDSs are designed to generate alarms and alerts, reporting them to a network management system. These uses SNMP traps and messages to post alarms and alerts to central network management consoles, where they can be serviced by network operations personnel. Several benefits are associated with this reporting scheme, including the ability to adapt the entire network infrastructure to respond to a detected attack, the ability to shift the processing load associated with an active response to a system other than the one being targeted by the attack, and the ability to use common communications channels. *(Bace and Mell, 2001)*

c) IDS Framework/ Architecture

Various intrusion detection system (IDS) frameworks/architectures have evolved over a period of time. These broadly include the following.

- The STAT Framework – The Web STAT intrusion detection system has been developed using the STAT framework *(Vigna et al., 2002)*. The framework provides the implementation of a domain-independent analysis engine that can be extended in a well-defined way to perform intrusion detection analysis in specific application domains. The STAT framework centres around an intrusion modeling technique that characterizes attacks in terms of transitions between the security states of a system. This approach is supported by the STATL attack modeling language. The STATL language provides constructs to represent an attack as a composition of states and transitions. States are used to characterize different snapshots of a system during the evolution of an attack. Obviously, it is not feasible to represent the complete state of a system (e.g., volatile memory, file system); therefore, a STATL scenario uses variables to record just those parts of the system state that are needed to define an attack signature (e.g., the value of a counter or the source of an HTTP request). A transition has an associated action that is a specification of the event that can cause the scenario to move to a new state. For example, an action can be the opening of a TCP connection or the execution of a CGI script. The space of possible relevant actions is constrained by a transition assertion, which is a filter condition on the events that can possibly match the action. For example, an assertion can require that a TCP connection be opened with a specific destination port or that a CGI application be invoked with specific parameters. It is possible for several

occurrences of the same attack to be active at the same time. A STATL attack scenario, therefore, has an operational semantics in terms of a set of instances of the same scenario specification. The scenario specification represents the scenario's definition and global environment, and a scenario instance represents a particular attack that is currently in progress.

The STAT Core module is the run-time for the STATL language. The Core implements the concepts of state, transition, instance, timer, etc. In addition, the STAT Core is responsible for obtaining events from the target environment, and matching this event stream against the actions and assertions corresponding to transitions in the active attack scenarios. The STATL language and the Core runtime are domain independent. They do not support any domain-specific features, which may be necessary to perform intrusion detection analysis in particular domains or environments. For example, network events such as an IP packet or the opening of a TCP connection cannot be represented in STATL natively. Therefore, the STAT framework provides a number of mechanisms to extend the STATL language and the runtime to match the characteristics of a specific target domain.

In summary, a STAT-based sensor is created by developing a language extension that describes the particular domain of the application, an event provider that retrieves information from the environment and produces STAT events, and attack scenarios that describe attacks in terms of state transition models of STAT events. In addition, it is possible to create response libraries that are specific to a certain domain. The response functions in the library can be dynamically associated with the states modeled in the attack scenarios.

- Distributed Intrusion Detection System (DIDS) – DIDS is the second major IDS system having evolved in recent times (Snapp et. al., 2003). The DIDS architecture combines distributed monitoring and data reduction with centralized data analysis. This approach is unique among current IDS's. The components of DIDS are the DIDS director, a single host monitor per host. and a single LAN monitor for each broadcast LAN segment in the monitored network. DIDS can potentially handle hosts without monitors since the LAN monitor can report on the network activities of such hosts. The host and LAN monitors are primarily responsible for the collection of evidence of unauthorized or suspicious activity, while the DIDS director is primarily responsible for its evaluation. Reports are sent independently and asynchronously from the host and LAN monitors to the DIDS director through a communications infrastructure.

High level communication protocols between the components are based on the ISO Common Management Information Protocol (CMIP) recommendations, allowing for future inclusion of CMIP management tools as they become useful. The architecture also provides for bidirectional communication between the DIDS director and any monitor in the configuration. This communication consists primarily of notable events and anomaly reports from the monitors. The director can also make requests for more detailed information from the distributed monitors via a "GET" directive, and issue commands to have the distributed monitors modify their monitoring capabilities via a "SET" directive. A large amount of low level filtering and some analysis is performed by the host monitor to minimize the use of network bandwidth in passing evidence to the director.

The DIDS director consists of three major components that are all located on the same dedicated workstation. Because the components are logically independent processes, they could be distributed as well. The communications manager is responsible for the transfer of data between the director and each of the host and the LAN monitors. It accepts the notable event records from each of the host and LAN monitors and sends them to the expert system. On behalf of the expert system or user interface, it is also able to send requests to the host and LAN monitors for more information regarding a particular subject. The expert system is responsible for evaluating and reporting on the security state of the monitored system. It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole. The expert system is a rule-based system with simple learning capabilities. The director's user interface allows the System Security Officer (SSO) interactive access to the entire system. The SSO is able to watch activities on each host, watch network traffic (by setting "wire-taps"), and request more specific types of information from the monitors.

The host monitor is currently installed on Sun SPARC stations running SunOS 4.0.x with the Sun C2 security package. Through the C2 security package, the operating system produces audit records for virtually every transaction on the system. These transactions include file accesses, system calls, process executions, and logins. The contents of the Sun C2 audit record are: record type, record event, time, real user ID, audit user ID, effective user ID, real group ID, process ID, error code, return value, and label.

All possible transactions fall into one of a finite number of events formed by the cross product of the actions and the domains, and each event may also succeed or fail. Note that no distinction is made between files, directories or devices, and that all of these are treated simply as objects. Not every action is applicable to every object; for example, the terminate action is applicable only to processes. The choice of

these domains and actions is somewhat arbitrary in that one could easily suggest both finer and coarser grained partitions. However, they capture most of the interesting behavior for intrusion detection and correspond reasonably well with what other researchers in this field have found to be of interest. By mapping an infinite number of transactions to a finite number of events, not only can the operating system dependencies be removed, but also restrict the number of permutations that the expert system will have to deal with. The concept of the domain is one of the keys to detecting abuses. Using the domain allows us to make assertions about the nature of a user's behavior in a straightforward and systematic way. Although this leads to loss of some details provided by the raw audit information, that is more than made up for by the increase in portability, speed, simplicity, and generality.

The LAN monitor uses heuristics in an attempt to identify the likelihood that a particular connection represents intrusive behavior. These heuristics consider the capabilities of each of the network services, the level of authentication required for each of the services, the security level for each machine on the network, and signatures of past attacks. The abnormality of a connection is based on the probability of that particular connection occurring and the behavior of the connection itself. Upon request, the LAN monitor is also able to provide a more detailed examination of any connection, including capturing every character crossing the network (i.e., a wire-tap). This capability can be used to support a directed investigation of a particular subject or object. Like the host monitor, the LAN monitor forwards relevant security information to the director through its LAN agent.

DIDS utilizes a rule-based (or production) expert system. The expert system is currently written in Prolog, and much of the form of the rule base comes from Prolog and the logic notation that Prolog implies. The expert system uses rules derived from the hierarchical Intrusion Detection Model (IDM). The IDM describes the data abstractions used in inferring an attack on a network of computers. That is, it describes the transformation from the distributed raw audit data to high level hypotheses about intrusions and about the overall security of the monitored environment. In abstracting and correlating data from the distributed sources, the model builds a virtual machine which consists of all the connected hosts as well as the network itself. This unified view of the distributed system simplifies the recognition of intrusive behavior which spans individual hosts. The model is also applicable to the trivial network of a single computer.

- Intrusion Detection System for Cloud Computing – Cloud computing provides application and storage services on remote servers *(Shelke, Sontakke, Gawande, 2012).* The clients do not have to worry about its maintenance and software or hardware up-gradations. Cloud model works on the „concept of virtualization" of resources, where a hypervisor server in cloud data center hosts a number of clients on one physical machine. Deploying HIDS in hypervisor or host machine would allow the administrator to monitor the hypervisor and virtual machines on that hypervisor. But with the rapid flow of high volume of data as in cloud model, there would be issues of performance like overloading of VM hosting IDS and dropping of data packets. Also if host is compromised by an offending attack the HIDS employed on that host would be neutralized. In such a scenario, a network based IDS would be more suitable for deployment in cloud like infrastructure. NIDS would be placed outside the VM servers on bottle neck of network points such as switch, router or gateway for network traffic monitoring to have a global view of the system. Such NIDS would still be facing the issue of large amount of data through network access rate in cloud environment. To handle a large number of data packets flow in such an environment a multi-threaded IDS approach has been proposed in this paper. The multi-threaded IDS would be able to process large amount of data and could reduce the packet loss. After an efficient processing the proposed IDS would pass the monitored alerts to a third party monitoring service, who would in turn directly inform the cloud user about their system under attack. The third party monitoring service would also provide expert advice to cloud service provider for mis-configurations and intrusion loop holes in the system. The cloud user accesses its data on remote servers at service provider's site over the cloud network. User requests and actions are monitored and logged through a multi-threaded NIDS. The alert logs are readily communicated to cloud user with an expert advice for cloud service provider.

Proposed multi-threaded NIDS model for distributed cloud environment is based on three modules: capture & queuing module, analysis/ processing module and reporting module. The capture module, receives the in-bound and out-bound (ICMP, TCP, IP, UDP) data packets. The captured data packets are sent to the shared queue for analysis. The analysis and process module receives data packets from the shared queue and analyze it against signature base and a pre-defined rule set. Each process in a shared queue can have multiple threads which work in a collaborative fashion to improve the system performance. The main process will receive TCP, IP, UDP and ICMP packets and multiple threads would concurrently process and match those packets against pre-defined set of rules. Through an efficient matching and analysis the bad

packets would be identified and alerts generated. Reporting module would read the alerts from shared queue and prepares alert reports. The third party monitoring and advisory service having experience and resources would immediately generate a report for cloud user's information and sends a comprehensive expert advisory report for cloud service provider. Figure above depicts the flow chart of proposed multi-threaded Cloud IDS.

- An implementation of intrusion detection system using genetic algorithm – *Hoque, Mukit and Bikas (2012)* identify the following problems with the existing systems.

o *Snort:* A free and open source network intrusion detection and prevention system, was created by Martin Roesch in 1998 and now developed by Sourcefire. In 2009, Snort entered InfoWorld's Open Source Hall of Fame as one of the "greatest open source software of all time". Through protocol analysis, content searching, and various pre-processors, Snort detects thousands of worms, vulnerability exploit attempts, port scans, and other suspicious behavior.

o *OSSEC:* An open source host-based intrusion detection system, performs log analysis, integrity checking, rootkit detection, time-based alerting and active response. In addition to its IDS functionality, it is commonly used as a SEM/SIM solution. Because of its powerful log analysis engine, ISPs, universities and data centers are running OSSEC HIDS to monitor and analyze their firewalls, IDSs, web servers and authentication logs.

o *OSSIM:* The goal of Open Source Security Information Management, OSSIM is to provide a comprehensive compilation of tools which, when working together, grant network/security administrators with a detailed view over each and every aspect of networks, hosts, physical access devices, and servers. OSSIM incorporates several other tools, including Nagios and OSSEC HIDS.

o *Bro:* An open-source, Unix-based network intrusion detection system Bro detects intrusions by first parsing network traffic to extract its application-level semantics and then executing event-oriented analyzers that compare the activity with patterns deemed troublesome.

o *Fragroute/Fragrouter:* A network intrusion detection evasion toolkit. Fragrouter helps an attacker launch IP-based attacks while avoiding detection. It is part of the NIDS bench suite of tools by Dug Song.

o *BASE:* The Basic Analysis and Security Engine, BASE is a PHP-based analysis engine to search and process a database of security events generated by various IDSs, firewalls and network monitoring tools. A Genetic Algorithm (GA) is a programming technique that mimics biological evolution as a problem-solving strategy. It is based on Darwinian's principle of evolution and survival of fittest to optimize a population of candidate solutions towards a predefined fitness. When using GA for solving various problems three factors will have vital impact on the effectiveness of the algorithm and also of the applications. These include:

- the fitness function;
- the representation of individuals;
- the GA parameters. The determination of these factors often depends on applications and/or implementation.

The present paper aims to understand the emerging avenues in Intrusion Detection System, as to what all models, architectures are available for detecting intrusions and how to prevent those intrusions to occur in any network traffic. The paper further focuses on challenges in the current intrusion detection system while also comparatively analyzing the Active and Passive Response Systems. Finally, the paper explores the possible future avenues in intrusion detection scheme.

II. Findings and Discussion

IDS is an emerging trend in network security as intrusions are increasing day by day due to internet availability with high level of usage among people across the globe. With improvements in the network is required to protect one`s information lying unsecured over the internet and should not be revealed to unauthorized people or groups. Cloud computing is another emerging trend which has shot up demand of security over free network, i.e. Internet *(Shelke et. al, 2012).*

On the basis of analysis done from available systems in Intrusion Detection proposed by people in different geographical areas, different network or environment requires a different level of security and infrastructure is another concern to implement IDS or related services.

Currently, networked computer systems play an ever more major function in our fellowship and its economic system. They have become the targets of a wide array of malicious threats that invariably turn into real intrusions. This is the reason computer security has become a vital concern for network practitioner. Too often, intrusions cause disaster inside LANs and the time and cost to renovate the damage can grow to extreme proportions. Instead of using passive measures to repair and patch security hole once they have been exploited, it is more efficient to take up a proactive measure to intrusions *(Gomez, Dasgupta, 2002).*

Intrusion Detection Systems (IDS) are primarily focused on identifying probable incidents, monitoring

information about them, tries to stop them, and reporting them to security administrators in real-time environment, and those that exercise audit data with some delay (non-real-time). The latter approach would in turn delay the instance of detection. In addition, organizations apply IDSs for other reasons, such as classifying problems with security policies, documenting existing attacks, and preventing individuals from violating security policies. IDSs have become a basic addition to the security infrastructure of almost every organization *(Hassan, 2013).* A usual Intrusion Detection System is demonstrated in Figure 1 below.

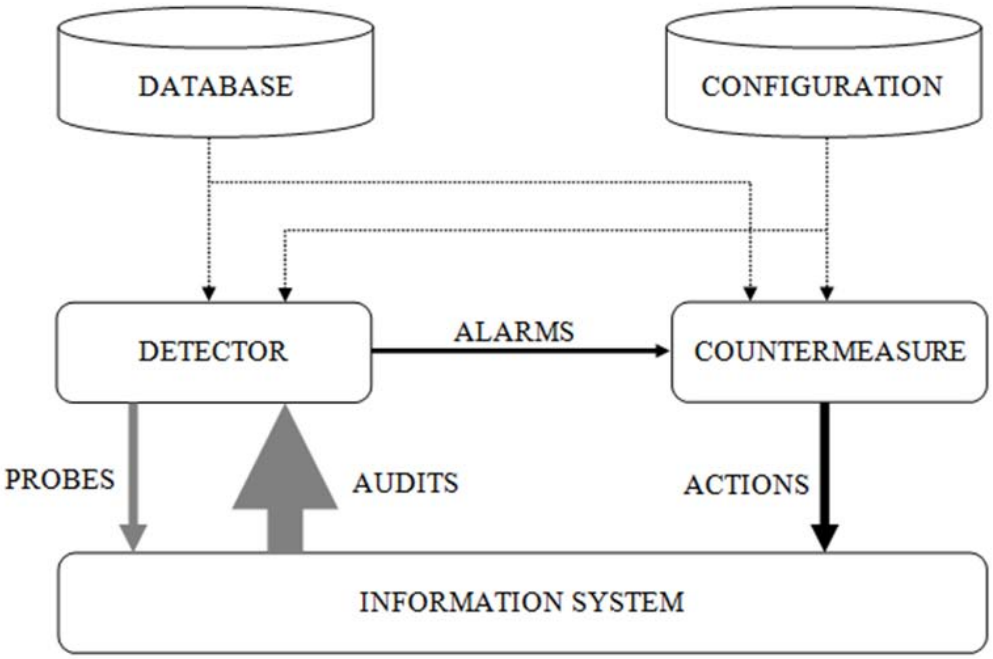

Figure 1

Note: The arrow lines symbolize the amount of information flowing from one component to another

Very Simple Intrusion Detection System

One of the major problems encountered by IDS is large number of false positive alerts that is the alerts that are mistakenly analyzed normal traffic as security violations. An ideal IDS does not produce false or inappropriate alarms. In practice, signature based IDS found to produce more false alarms than expected. This is due to the very general signatures and poor built in verification tool to authenticate the success of the attack. The large amount of false positives in the alert logs generates the course of taking corrective action for the true positives, i.e. delayed, successful attacks, and labor intensive.

The normal and the abnormal intrusive activities in networked information processing systems are hard to forecast as the limits cannot be easily explained. This prediction process may generate false alerts in many anomaly based intrusion detection schemes. However, with the introduction of fuzzy logic, the false alarm rate in determining intrusive activities can be minimized; a set of fuzzy rules (noncrisp fuzzy classifiers) can be employed to identify the normal and abnormal behavior in computer networks, and fuzzy inference logic can be applied over such rules to determine when an intrusion is in progress. The primary problem with this procedure is to make good fuzzy classifiers to detect intrusions *(Tillapart, 2002).*

The intrusion detection strategies concern four primary issues. First is the dataset that is captured from network communications. The second is Genetic Algorithms (GA) which use mutation, recombination, and selection applied to a population of individuals in order to evolve iteratively better and better solutions and a way to generate fuzzy rules to characterize normal and abnormal behavior of network systems. The third is to generate alerts and reports for malicious traffic behavior, and the fourth is the maintenance of the ids for observation of placement of sensors, and qualified trained intrusion analysts so that the latest malicious traffic is being detected.

The following future trends are clearly visible in intrusion detection systems.

- *Genetic Algorithm (GA):* GA is a programming technique that uses biological evolution as a problem solving strategy. It is based on Darwinian's theory of evolution and survival of fittest to make effective a population of candidate result near a predefined fitness. The proposed GA based

intrusion detection system holds two modules where each acts in a dissimilar stage. In the training stage, a set of classification rules are produced from network audit data using the GA in an offline background. In the intrusion detection phase, the generated rules are employed to classify incoming network connections in the real-time environment. Once the rules are generated, the intrusion detection system becomes simple, experienced and efficient one.

GA applies an evolution and natural selection that employs a chromosome-like data structure and evolve the chromosomes by means of selection, recombination and mutation operators. The process generally starts with randomly generated population of chromosomes, which signify all possible solution of a problem that are measured candidate solutions. From each chromosome different positions are set as bits, characters or numbers. These positions are regarded as genes. An evaluation function is employed to find the decency of each chromosome according to the required solution; this function is known as "Fitness Function". During the process of evaluation "Crossover" is applied to have natural reproduction and "Mutation" is applied to mutation of species. For survival and combination the selection of chromosomes is partial towards the fittest chromosomes *(Hassan, 2013).*

- *Fuzzy Logic:* A fuzzy expert system consists of three different types of entities: fuzzy sets, fuzzy variables and fuzzy rules. The membership of a fuzzy variable in a fuzzy set is determined by a function that produces values within the interval [0, 1]. These functions are called membership functions. Fuzzy variables are divided into two groups: antecedent variables, that are assigned with the input data of the fuzzy expert system and consequent variables, that are assigned with the results computed by the system.

The fuzzy rules determine the link between the antecedent and the consequent fuzzy variables, and are often defined using natural language linguistic terms. For instance, a fuzzy rule can be" if the temperature is cold and the wind is strong then wear warm clothes", where temperature and wind are antecedent fuzzy variables, wear is a consequent fuzzy variable and cold, strong and warm clothes are fuzzy sets *(Hassan, 2013).*

The process of a fuzzy system has three steps. These steps are Fuzzification, Rule Evaluation, and Defuzzification. In the fuzzification step, the input crisp values are transformed into degrees of membership in the fuzzy sets. The degree of membership of each crisp value in each fuzzy set is determined by plugging the value into the membership function associated with the fuzzy set. In the rule evaluation step, each fuzzy rule is assigned with a strength value. The strength is determined by the degrees of memberships of the crisp input values in the fuzzy sets of antecedent part of the fuzzy rule. The defuzzification stage transposes the fuzzy outputs into crisp values.

III. Conclusion

The study focused on studying existing IDS and their usefulness in detecting and preventing attacks in any type of network and control traffic with the performance of the system to be improved as well. It is found that intrusion has different meaning and scenarios defining need of attack detection and prevention of attacks.

Deterrence is the key to the value of IDS. The benefit of deploying an IDS depends on how much it prevents hackers from committing intrusions. Although IDSs are classified as detective controls because they detect attacks that were not prevented, they implicitly act as preventive controls by changing the behavior of attackers in the first place, and thus eliminating attacks.

The presence of a network-based IDS can put hackers on notice that their actions may lead to legal action. Host-based systems provide very similar deterrent effect. People who know that their actions may be monitored are less likely to commit misuse.

Optimally configured IDSs always provide non-negative value to their adopters. By using the out-of-box configuration, firms may be taking the easy way out, but they may be hurting themselves. Current widespread complaint against IDSs is that they produce many false alarms: False positives are tremendous time wasters and drive up operational labor costs.

IDS developers should also pay close attention to the configuration issue. They should design IDSs that are easy to configure, especially in light of high false positive rates associated with IDSs. Most vendors do not provide these data. Various groups, including academic institutions, research labs, and commercial organizations, have tested commercial and government sponsored IDS products.

All the IT security concerns are integral part of security programs and therefore, should be carefully designed and deployed. Recently, organizations realized that it is impossible to eliminate all security risks. As a result, detection based systems have started to gain popularity in the IT security domain. Today, IDSs are the most popular detective controls. Although IDS has been the fastest-growing security product in the market for the last few years, the security community is uncertain about their value.

An improperly configured IDS may encourage more hacking, resulting in a higher loss for the firm. An optimally configured IDS deters hackers, thus, reducing the need for investigation by security experts for security violations. To firms that are using default configuration

or that have not adopted an IDS because of doubts about its value, our results provide incentives to implement an appropriately configured IDS.

IV. Limitations and Future Research Directions

As with all models, the model parameters were common knowledge to the firm and users. One region that looks particularly interesting is games with incomplete information, in which either the assembly or the user is unsure about the other's payoffs. This perspective allows incorporation of uncertainty about the nature of the game being played.

It may be more realistic to consider a multiperiod model in which the firm revises its estimates every period based on its observations of the hacker's strategy in previous periods. Such learning has been analyzed in game theory.

Security experts take appropriate actions after receiving alarms from IDSs. This approach, also called passive response, is the current trend in commercial IDSs. Another response option is to let the IDS take an action without human intervention (active response). Current IDSs provide little or no guidance to security management once an attack has been identified.

IDSs are here to stay, with billion dollar firms supporting the development of commercial security products and driving hundreds of millions in annual sales. Nevertheless, they remain hard to configure and operate and often can't be effectively utilized by the very novice security personnel who demand to benefit from them most.

References Références Referencias

1. Anderson, J. P. (1972). Computer Security Technology Planning Study Volume II. *Electronic Systems Division (AFSC)*.
2. Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems. *National Institute of Standards and Technology, CA*.
3. Balasubramaniyan, J. S., Garcia-Fernandez, J. O., Isacoff, D., Spafford, E., & Zamboni, D. (1998). An Architecture for Intrusion Detection Using Autonomous. *Paper, COAST Technical Report, Purdue University*.
4. Botha, M., & Solms, R. v. (2004). Utilizing Neural Networks For Effective Intrusion Detection. *Port Elizabeth Technikon, South Africa*.
5. Denning, D. E. (1987). An Intrusion Detection Model.
6. Gomez , J., & Dasgupta, D. (2002). Evolving Fuzzy Classifiers for Intrusion Detection. *IEEE*.
7. Graham, R. (2000, January 07). *FAQ: Network Intrusion Detection Systems.* Retrieved from www.robertgraham.com.
8. Hassan, M. M. (2013). Current Studies on Intrusion Detection System, Genetic Algorithm And Fuzzy Logic. *International Journal of Distributed and Parallel Systems*.
9. Hoque, M. S., Mukit, M., & Bikas, M. N. (2012). An Implementation of Intrusion Detection System using Genetic Algorithm. *International Journal of Network Security & Its Applications (IJNSA)*.
10. Institute, S. (2001). Understanding Intrusion Detection Systems. *SANS Institute*.
11. Scarfone, K., & Mell, P. (2007). Intrusion Detection and Prevention Systems. *National Institute of Standards and Technology Special Publication*.
12. Shelke, M. K., Sontakke, M., & Gawande, D. D. (2012). Intrusion Detection System for Cloud Computing. *International Journal of Scientific & Technology Research*.
13. Snapp, S. R., Brentano, J., Dias, G. V., Goan, T. L., Heberlein, L. T., Ho, C.-L., et al. (2002). DIDS (Distributed Intrusion Detection System) - Motivation, Architecture, and An Early Prototype. *University of California, Davis*.
14. Tillapart, P., Thumthawatworn, T., & Santiprabhob, P. (2002). Fuzzy Intrusion Detection System. *Thesis, Assumption University,Bangkok*.
15. Vigna, G., Robertson, W., Kher, V., & Kemmerer, R. A. (2002). A Stateful Intrusion Detection System forWorld-Wide Web Servers. *University of California, Santa Barbara*.
16. Zamboni, D. (2001). Using internal sensors for computer intrusion detection. *Center for Education and Research in Information Assurance and Security, Purdue University*.