***INTRUSION DETECTION***

***SYSTEM***

***Submitted in partial fulfillment of the requirements for the award of the degree***

**B.Tech in COMPUTER SCIENCE**

Supervisor Submitted by



**University School of Information Technology**

**GGS Indraprastha University, Delhi – 78**

**ABSTRACT**

Intrusion-detection systems aim at detecting attacks against computer systems and networks or, in general, against information systems. Indeed, it is difficult to provide provably secure information systems and to maintain them in such a secure state during their lifetime and utilization. Sometimes, legacy or operational constraints do not even allow the definition of a fully secure information system. Therefore, intrusion detection systems have the task of monitoring the usage of such systems to detect any apparition of insecure states. They detect attempts and active misuse either by legitimate users of the information systems or by external parties to abuse their privileges or exploit security vulnerabilities.

Intrusion detection based upon computational intelligence is currently attracting considerable interest from the research community. Characteristics of computational intelligence (CI) systems, such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information ﬁt the requirements of building a good intrusion detection model. Here we want to provide an overview of the research progress in applying CI methods, such as fuzzy logic and genetic algorithm to the problem of intrusion detection. The scope of this review will be on core methods of CI, fuzzy systems, evolutionary, Genetic algorithm, and soft computing.

**Introduction**

A correct firewall policy can minimize the exposure of many networks however they are quite useless against attacks launched from within. Hackers are also evolving their attacks and network subversion methods. These techniques include email based Trojan, stealth scanning techniques, malicious code and actual attacks, which bypass firewall policies by tunneling access over allowed protocols such as ICMP, HTTP, DNS, etc. Hackers are also very good at creating and releasing malware for the ever-growing list of application vulnerabilities to compromise the few services that are being let through by a firewall.

IDS arms your business against attacks by continuously monitoring network activity, ensuring all activity is normal. If IDS detects malicious activity it responds immediately by destroying the attacker's access and shutting down the attack. IDS reads network traffic and looks for patterns of attacks or signatures, if a signature is identified, IDS sends an alert to the Management Console and a response is immediately deployed.

**Definitions**

What is intrusion?

An intrusion is somebody attempting to break into or misuse your system. The word "misuse" is broad, and can reflect something severe as stealing confidential data to something minor such as misusing your email system for Spam.

What is an IDS?

An IDS is the real-time monitoring of network/system activity and the analysing of data for potential vulnerabilities and attacks in progress.



Intrusion Detection Systems is a topic that has recently garnered much interest in the computer security community. In the last few years, this interest level has spurred the development of a variety of approaches to providing IDS capabilities that are both reliable and low-impact in terms of management or cost. When presented with different types of IDS one might be tempted to assume that one approach or another was inherently superior. In fact, the mixture of approaches used for IDS offers the security analyst a unique opportunity in terms of the synergies inherent in combined techniques. Intrusion Detection Systems are like a burglar alarm for your computer network. They detect unathorized access attempts. They are the first line of defence for your computer systems.

# Need for IDS

# Who are attacked?

Internet Information Services (IIS) web servers – which host web pages and serve them to users – are highly popular among business organizations, with over 6 million such servers installed worldwide. Unfortunately, IIS web servers are also popular among hackers and malicious fame-seekers – as a prime target for attacks!

As a result, every so often, new exploits emerge which endanger your IIS web server’s integrity and stability. Many administrators have a hard time keeping up with the various security patches released for IIS to cope with each new exploit, making it easy for malicious users to find a vulnerable web server on the Internet. There are multiple issues which can completely endanger your Web server – and possibly your entire corporate network and reputation.

People fell there is nothing on their system that anybody would want. But what they are unaware of is that, there is the issue of legal liability. You are potentially liable for damages caused by a hacker using your machine. You must be able to prove to a court that you took "reasonable" measures to defend yourself from hackers. For example, consider if you put a machine on a fast link (cable modem or DSL) and left administrator/root accounts open with no password. Then if a hacker breaks into that machine, then uses that machine to break into a bank, you may be held liable because you did not take the most obvious measures in securing the machine.

# How are they attacked?

# An intruder normally hacks into your system only after he has carefully accessed you and your security and he attacks you in a systematic way to cause maximum damage. The normal steps towards intrusion are:

# Outside reconnaissance: The intruder will find out as much as possible without actually giving himself away. They will do this by finding public information or appearing as a normal user. In this stage, you really can't detect them. The intruder will do a 'whois' lookup to find as much information as possible about your network as registered along with your Domain Name (such as foobar.com. The intruder might walk through your DNS tables (using 'nslookup', 'dig', or other utilities to do domain transfers) to find the names of your machines. The intruder will browse other public information, such as your public web sites and anonymous FTP sites. The intruder might search news articles and press releases about your company.

# Inside reconnaissance: The intruder uses more invasive techniques to scan for information, but still doesn't do anything harmful. They might walk through all your web pages and look for CGI scripts (CGI scripts are often easily hacked). They might do a 'ping' sweep in order to see which machines are alive. They might do a UDP/TCP scan/strobe on target machines in order to see what services are available. They'll run utilities like 'rcpinfo', 'showmount', 'snmpwalk', etc. in order to see what's available. At this point, the intruder has done 'normal' activity on the network and has not done anything that can be classified as an intrusion. At this point, a NIDS will be able to tell you that "somebody is checking door handles", but nobody has actually tried to open a door yet.

# Exploit: The intruder crosses the line and starts exploiting possible holes in the target machines. The intruder may attempt to compromise a CGI script by sending shell commands in input fields. The intruder might attempt to exploit well-known buffer-overrun holes by sending large amounts of data. The intruder may start checking for login accounts with easily guessable (or empty) passwords. The hacker may go through several stages of exploits. For example, if the hacker was able to access a user account, they will now attempt further exploits in order to get root/admin access.

# Foot hold: At this stage, the hacker has successfully gained a foot hold in your network by hacking into a machine. The intruder's main goal is to hide evidence of the attacks (doctoring the audit trail and log files) and make sure they can get back in again. They may install 'toolkits' that give them access, replace existing services with their own Trojan horses that have backdoor passwords, or create their own user accounts. System Integrity Verifiers (SIVs) can often detect an intruder at this point by noting the changed system files. The hacker will then use the system as a stepping stone to other systems, since most networks have fewer defenses from inside attacks.

# Profit: The intruder takes advantage of their status to steal confidential data, misuse system resources (i.e. stage attacks at other sites from your site), or deface web pages.

**Types of IDS**

There are two primary types of IDS:

#### Network based IDS

A Network Intrusion Detection system (NIDS) transparently monitors network traffic, looking for patterns indicative of an attack on a computer or network device. By examining the network traffic, a network based intrusion detection system can detect suspicious activity such as a port scan or Denial of Service (DOS) attacks.

A NID monitors the network traffic it has access to, by comparing the data in the TCP/IP packet to a database of attack signatures. In a network environment, it can see packets to and from the system(s) that it monitors. In a switched environment, it can see packets coming to and from the system(s) that it monitors, providing it can see all data traffic on the ports that connect to the systems. Once a NIDS detects an attack, the following actions may be taken:

bd14753_ Send email notification

bd14753_ Send an SNMP trap to a network management system

bd14753_ Send a page (to a pager)

bd14753_ Block a TCP connection

bd14753_ Kill a TCP connection

bd14753_ Run a user defined script

In general terms a NID will be deployed on a DMZ. This assumes that you have a firewall in place and that you have a DMZ configured. When deployed behind the firewalls, the NID will detect attacks from protocols and sources allowed through the firewall and from internal users. By taking an action, such as sending an SNMP trap or a page, it can alert network staff that an attack is in progress and enable them to make decisions based on the nature of the attack. It is recommended that the IDS is used for detection and alerting only and not for proactive defence i.e. killing/blocking TCP connections as this can often cause more problems.



**Host based IDS**

In most cases, a Host Intrusion Detection System (HIDS) component is made up of two parts: a centralised manager and a server agent. The manager is used to administer and store policies, download policies to agents and store information received by agents. The agent is installed onto each server and registered with the manager. Agents use policies to detect and respond to specific events and attacks. An example of a policy would be an agent that sends an SNMP trap when three concurrent logins as root have failed on a UNIX server. System logs and processes are also monitored to see if any actions that violate the policy have occurred. If a policy has been violated, the agent will take a predefined action such as sending an email or sending a SNMP trap to a network management system. Host based intrusion detection system may further be divided into

System integrity verifiers (SIV): monitors system files to find when a intruder changes them (thereby leaving behind a backdoor). The most famous of such systems is "Tripwire". A SIV may watch other components as well, such as the Windows registry and chron configuration, in order to find well known signatures. It may also detect when a normal user somehow acquires root/administrator level privileges. Many existing products in this area should be considered more "tools" than complete "systems": i.e. something like "Tripwire" detects changes in critical system components, but doesn't generate real-time alerts upon an intrusion.

**Log file monitors (LFM):** monitor log files generated by network services. In a similar manner to NIDS, these systems look for patterns in the log files that suggest an intruder is attacking. A typical example would be a parser for HTTP server log files that looking for intruders who try well-known security holes, such as the "phf" attack. Example: swatch



IDS vs. Firewalls

A common misunderstanding is that firewalls recognize attacks and block them. This is not true.

Firewalls are simply a device that shuts off everything, then turns back on only a few well-chosen items. In a perfect world, systems would already be "locked down" and secure, and firewalls would be unneeded. The reason we have firewalls is precisely because security holes are left open accidentally. Thus, when installing a firewall, the first thing it does is stops ALL communication. The firewall administrator then carefully adds "rules" that allow specific types of traffic to go through the firewall. For example, a typical corporate firewall allowing access to the Internet would stop all UDP and ICMP datagram traffic, stops incoming TCP connections, but allows outgoing TCP connections. This stops all incoming connections from Internet hackers, but still allows internal users to connect in the outgoing direction.

A firewall is simply a fence around you network, with a couple of well chosen gates. A fence has no capability of detecting somebody trying to break in (such as digging a hole underneath it), nor does a fence know if somebody coming through the gate is allowed in. It simply restricts access to the designated points.

In summary, a firewall is not the dynamic defensive system that users imagine it to be. In contrast, an IDS is much more of that dynamic system. An IDS does recognize attacks against the network that firewalls are unable to see.

For example, in April of 1999, many sites were hacked via a bug in ColdFusion. These sites all had firewalls that restricted access only to the web server at port 80. However, it was the web server that was hacked. Thus, the firewall provided no defense. On the other hand, an intrusion detection system would have discovered the attack, because it matched the signature configured in the system.

Another problem with firewalls is that they are only at the boundary to your network. Roughly 80% of all financial losses due to hacking come from inside the network. A firewall a the perimeter of the network sees nothing going on inside; it only sees that traffic which passes between the internal network and the Internet.

Some reasons for adding IDS to your firewall are:

bd14753_ Double-checks misconfigured firewalls.

bd14753_ Catches attacks that firewalls legitimate allow through (such as attacks against web servers).

bd14753_ Catches attempts that fail.

bd14753_ Catches insider hacking.

**Use case description of ids**

In our system IDS is installed on the server side, which serves local hosts and users over internet as shown in figure. There are four actors in the system namely monitor, user, network and system administrator. User sends request to the server over the internet or LAN and IDS will analyze the packets received by the server. This IDS detects both internal and external intrusions. If it detects any intrusion then it alerts system administrator.

System Administrator

User App1

IDS

Server

User App2

Internet

Router

Hostn

Host2

Host1

User Appn

**System Description**

Network Intrusion Detection system (NIDS) is a system which monitors network intrusion. Intrusion may be detected by techniques like anomaly detection, signature pattern matching etc. Anomaly detection is a method in which normal network behavior is captured and any abnormality in the network is detected such as a sudden increase in network traffic rate (number of IP packets per second). Signature pattern matching is a method in which network data is compared with the known attack techniques that are saved in a database. For example an IDS that watches web servers might be programmed to look for string “phf” as an indicator for a CGI program attack.

Intrusion is detected and system administrator is alerted about the kind of intrusion when any one of the following events takes place:

1. If a foreign entity has been detected in a log entry.
2. If user tries to access information which is beyond his/her access.
3. Baseline for critical system resources is measured such as cpu utilization, file entries, disk activity, user logins etc. Then the system can trigger when there is a deviation from this baseline.

**Use Cases**

**Actors:**

1) User

User sends request to server and server responds by providing the requested service.

2) Network

Network carries the IP packets from source to destination.

3) IDS

IDS takes the packets from the network, analyses the packets.

4) System Administrator

System Administrator is alerted by the IDS of any suspicious activity or whenever intrusion is detected.

**Use Case Description**

1. IP Packets

Network gives the IP Packets to IDS which does further processing of these packets.

1. Anomaly Detection

If IDS Detects any abnormality in the network traffic, then it triggers the alert system

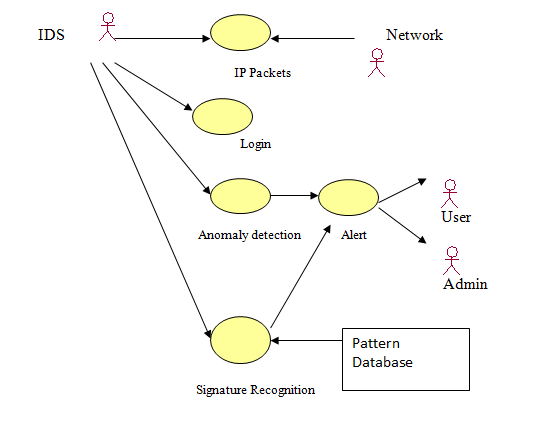
1. Signature recognition

IDS examines the traffic looking for well-known patterns of attack, which are saved in pattern database and triggers the alert system, if a match is found.

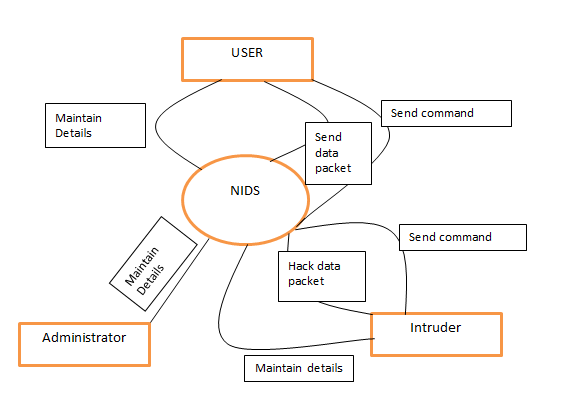
1. Alert System

Whenever triggered by anomaly detection or signature recognition, it alerts the system administrator.

**Use Case Diagram**

****

**Data flow diagram**



**SOFTWARE AND HARDWARE REQUIREMENTS**

* **External Interface Requirements:**
* User Interfaces
* Help and Tooltips are available for easy understanding.
* Graphical interface is available for ease and convenience of the user.
* Most functions require mouse click thus simplifying operations.
* Tools strip menu is available for faster access of menus.
* Shortcut keys are available for experienced users.
* Hardware Interfaces
* Mouse is required for use of application
* Keyboard is required for use of application
* Monitor is required for use of application
* Network interface card is required for packet capture
* 1.5 MB of hard disk space.
* 1 GB RAM (Random Access Memory)
* Software Interfaces

This software requires following software interfaces:

* WinPcap V4.0.1
* Communications Protocols

Net Vigilant should capture packets on wired as well as wireless networks.

* **Software Product Feature**:

The following features are included in Net Vigilant Network Monitor:

1. Network Monitor for wired networks [Ethernet]

This feature will provide the facility to capture network packets. The details of the packet will be listed in a table. These packets can be stored in XML serialized formats. Packets can be retrieved later for viewing and analysis.

1. Network monitor for wireless networks [802.11]

This feature will provide the facility to capture network packets. The details of the packet will be listed in a table. These packets can be stored in xml serialized formats. Packets can be retrieved later for viewing and analysis.

1. Network Utilities [Ping, TCP Statistics, UDP Statistics]  
   The above mentioned utilities will be provided for network traffic analysis.
2. Packet Store/Retrieve

Packets can be stored in an XML file and later retrieved for viewing or analysis.

1. Packet Filtering

The packets can be filtered by protocol type TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ARP (Address Resolution Protocol), ICMP (Internet Control Message Protocol) and IGMP (Internet Group Management Protocol).

1. Windows based Graphical User Interface (GUI)

Windows based GUI for better user experience will be provided.

* **Software System Attributes**:
* Reliability

This software has been tested and found to be reliable.

* Availability

Since this system has been tested for defects and fixed, the downtime is low and therefore is available.

* Security

Security features are also provided by .NET.

* Maintainability

Ease of maintenance is one of the advantages of .NET.

* Portability

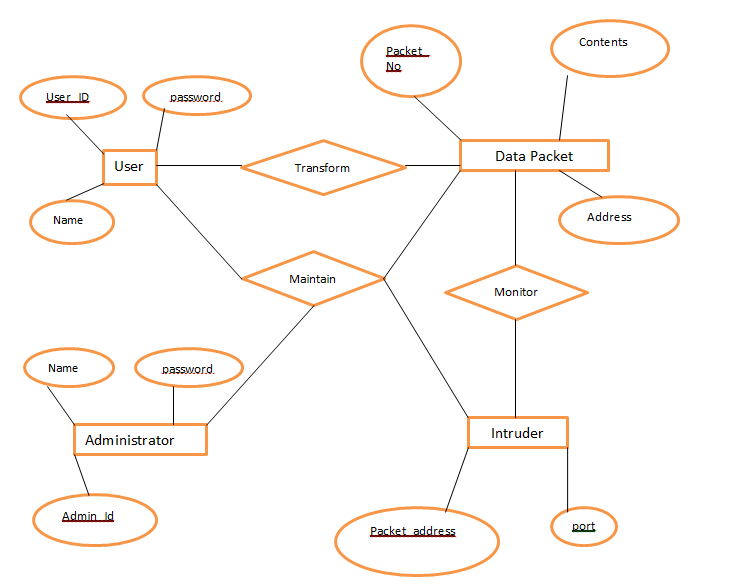
This application is supported on following Operating Systems.

Windows(98, XP, 2000, Vista, 7) and Linux(Ubuntu Fedora etc).

* Performance

Performance of this application is good on a small network. It has yet to be tested on a larger network.

**e-r diagram for intrusion detection system**



**System design**

* **CLIENT-SERVER MODEL**

Coordination model in a distributed system.

It defines:

• Which process may begin the interaction.

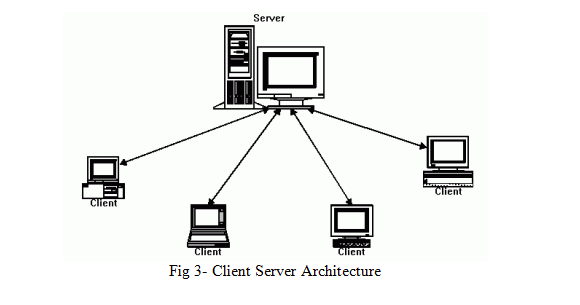
• Which process may answer.

• How error conditions may be managed.

• Although an Internet system provides a basic communication service, the protocol software cannot initiate control with, or accept contact from, a remote computer.

• Of course, two application involved in a communication cannot both wait for a message to arrive. One application must actively initiate interaction while the other application passively waits.

• Most network applications use a form of communication known as the client –server paradigm. A server application waits passively for contact, while a client application initiates communication actively.



**Process classification**

• Client process, the process that requires a service

•Server process, the process that provides the required service

•The client requires a service, the server provides the service and makes available the results to the client.

* **Client functions**

In generally, client software:

* is an arbitrary application program that becomes a client temporarily when remote access is needed, but also performs other computation locally.
* is invoked locally by a user, and executes only for one session
* runs locally on a user personal computer
* actively initiates contact with a server can access multiple services as needed, but actively contacts one remote server at a time.
* does not require special hardware or a sophisticated operating system.
* **Server functions**
* Is a special purpose, privileged program dedicated to providing one service, but can handle multiple remote clients at the same time.
* Run on a shared computer(i.e. not a user’s personal computer).
* Wait passively for contact from arbitrary remote clients
* Accepts contact from arbitrary clients, but offers a single service
* Requires powerfull hardware and a sophisticated operating system

A server must guarantee:

**authentication**: client identity verification

**authorization**: verification of the possibility for a client to access to a particular service

**data security**: guarantee that specific data cannot be read and/or modified

* **Characteristics of a client-server architecture**

• Client and server machines need different amount of hardware and software resources.

• Client and server machines may belong to different vendors.

• orizzontal scalability (increase of the client machines) and vertical scalability (migration to a more powerful server or to a multiserver solution)

• A client or server application interacts directly with a transport layer protocol to establish communication and to send or receive information.

•The transport protocol then uses lower layer protocols to send or receive individual messages. Thua, a computer needs a complete stack of protocols to run either a client or a server.

• A single server-class computer can offer multiple services at the same time; a separate server program is needed for each service.