

Analysis of Intrusion Detection Systems and Effective Intrusion Detection Mechanism

Kalyani Kundeti¹, Dr. M.V. Vijaya Saradhi²

¹(PG Scholar, Department of IT, Aurora's Engineering College, Bhongir, Andhra Pradesh, India)

²(Professor & HOD, Department of IT, Aurora's Engineering College, Bhongir, Andhra Pradesh, India)

ABSTRACT

Intrusion detection is the process of identifying activity that is malicious or unauthorized. The Intrusion Detection System (IDS) is designed to monitor for known attack signatures and sniff out suspicious behaviour. Today's Security infrastructure are becoming extremely complex, it includes firewalls, identification and authentication systems, access control product, Virtual private networks, encryption products, virus scanners, and more. Failure of one of the above component of our Security infrastructure puts the system in risk which they are supposed to protect. Even if our perimeter systems are fully up to date, new attacks that signature files don't recognize will still get through. Even though companies use three separate layers of antivirus protection from three separate vendors, none can identify Intrusions because of its unusual design and persistence. In this paper we present the anatomy of Intrusion Detection Systems and we are proposing a mechanism aimed at Intrusion detection and taking action in the context of corporate business model. It is capable of keeping the track on the system activities. This method can understand the system information and identifies the suspicious system activities

Keywords - Intrusion Detection, security, attack, Intruder, Red team report

I. INTRODUCTION

Computer Systems have become more comprehensive and a higher value asset of organizations, intrusion detection systems has been incorporated as elements of operating systems, although not typically applications. Intrusion detection involves determining that some entity, an intruder, has attempted to gain, or worse, has gained unauthorized access to the system. The objectives of IDS are Confidentiality, Integrity, Availability, and Accountability [1].

Intruders are classified into two groups. External intruders do not have any authorized access to the system they attack. Internal intruders have at least some authorized access to the system. Internal intruders are further subdivided into the following three categories. Masquerades are external intruders who have succeeded in gaining access to the system and are acting as an authorized entity. Legitimate intruders have access to both the system and the data but misuse this access (misfeasors). Clandestine intruders have or have obtained supervisory (root) control of the system and as such can either operate below the level of auditing or can use the privileges to avoid being audited by stopping, modifying, or erasing the audit records.

Security is an important issue for all the networks of companies and institutions at the present time and all the intrusions are trying in ways that successful access to the data of these companies and Web services and despite the development of multiple ways to ensure that the infiltration of intrusion to the infrastructure of the network via the Internet, through the use of firewalls, encryption, etc. Fig-1 shows the number of machines that were attacked over the last few months [2].

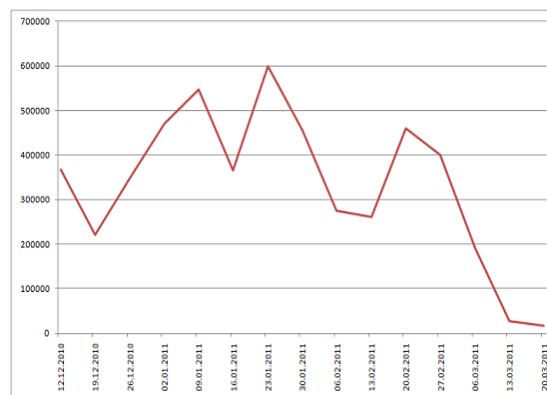


Fig-1 Graphical representation of number of machines that were attacked over the last few months

The main task of intrusion detection systems is defence of a computer system by detecting an attack [3]. Detecting attacks depends on the number and type of appropriate actions. Intrusion prevention requires a well-selected combination of “baiting and trapping” aimed at both investigations of threats. Diverting the intruder’s attention from protected resources is another task. Both the real system and a possible trap system are constantly monitored. Data generated by intrusion detection systems is carefully examined (this is the main task of each IDS) for detection of possible attacks (intrusions).

II. IDS ACTIVITIES

- Monitor and analyse user and system activities
- Auditing of system and configuration vulnerabilities
- Asses integrity of critical system and data files
- Recognition of pattern reflecting known attacks
- Statistical analysis for abnormal activities
- Data trail, tracing activities from point of entry up to the point of exit
- Installation of decoy servers (honey pots)
- Installation of vendor patches (some IDS).

III. WHAT IDS CAN NOT DO

- Compensate for weak authentication and identification mechanisms
- Investigate attacks without human intervention
- Guess the content of your organization security policy
- Compensate for weakness in networking protocols, for example: IP Spoofing
- Compensate for integrity or confidentiality of information
- Analyze all traffic on a very high speed network
- Deal adequately with attack at the packet level
- Deal adequately with modern network hardware

IV. WHY DO I NEED AN IDS, I HAVE A FIREWALL?

- IDS are a dedicated assistant used to monitor the rest of the security infrastructure
- Today’s security infrastructure are becoming extremely complex, it includes firewalls, identification and authentication systems, access control product, virtual private networks, encryption products, virus scanners, and more. All of these tools

performs functions essential to system security. Given their role they are also prime target and being managed by humans, as such they are prone to errors

- Failure of one of the above component of your security infrastructure jeopardized the system they are supposed to protect
- Not all traffic may go through a firewall i:e modem on a user computer
- Not all threats originates from outside. As networks uses more and more encryption, attackers will aim at the location where it is often stored unencrypted (Internal network)
- Firewall does not protect appropriately against application level weaknesses and attacks
- Firewall are subject to attacks themselves
- Protect against misconfiguration or fault in other security mechanisms

V. TYPES OF INTRUSION DETECTION SYSTEMS

Intrusion detection systems (IDS) can be classified into different ways. The major classifications are Active and passive IDS, Network Intrusion detection systems (NIDS) and host Intrusion detection systems (HIDS)

An active Intrusion Detection Systems (IDS) is also known as Intrusion Detection and Prevention System (IDPS). Intrusion Detection and Prevention System (IDPS) is configured to automatically block suspected attacks without any intervention required by an operator. Intrusion Detection and Prevention System (IDPS) has the advantage of providing real-time corrective action in response to an attack [7].

A passive IDS is a system that’s configured to only monitor and analyse network traffic activity and alert an operator to potential vulnerabilities and attacks. A passive IDS is not capable of performing any protective or corrective functions on its own [10].

1. NETWORK INTRUSION DETECTION SYSTEMS (NIDS) AND HOST INTRUSION DETECTION SYSTEMS (HIDS)

Network Intrusion Detection Systems (NIDS) usually consists of a network appliance (or sensor) with a Network Interface Card (NIC) operating in promiscuous mode and a separate management interface. The IDS is placed along a network segment or boundary and monitors all traffic on that segment.

A Host Intrusion Detection Systems (HIDS) and software applications (agents) installed on workstations which are to be monitored. The agents monitor the operating system and write data to log files and/or trigger alarms. A host Intrusion detection systems (HIDS) can only monitor the individual workstations on which the agents are installed and it cannot monitor the entire network. Host based IDS systems are used to monitor any intrusion attempts on critical servers.

The drawbacks of Host Intrusion Detection Systems (HIDS) are

- Difficult to analyse the intrusion attempts on multiple computers.
- Host Intrusion Detection Systems (HIDS) can be very difficult to maintain in large networks with different operating systems and configurations
- Host Intrusion Detection Systems (HIDS) can be disabled by attackers after the system is compromised.

2. Knowledge-based (Signature-based) IDS and behaviour-based (Anomaly-based) IDS

A knowledge-based (Signature-based) Intrusion Detection Systems (IDS) references a database of previous attack signatures and known system vulnerabilities. The meaning of word signature, when we talk about Intrusion Detection Systems (IDS) is recorded evidence of an intrusion or attack. Each intrusion leaves a footprint behind (e.g., nature of data packets, failed attempt to run an application, failed logins, file and folder access etc.). These footprints are called signatures and can be used to identify and prevent the same attacks in the future. Based on these signatures Knowledge-based (Signature-based) IDS identify intrusion attempts.

The disadvantages of Signature-based Intrusion Detection Systems (IDS) are signature database must be continually updated and maintained and Signature-based Intrusion Detection Systems (IDS) may fail to identify unique attacks.

A Behaviour-based (Anomaly-based) Intrusion Detection Systems (IDS) references a baseline or learned pattern of normal system activity to identify active intrusion attempts. Deviations from this baseline or pattern cause an alarm to be triggered.

Higher false alarms are often related with Behaviour-based Intrusion Detection Systems (IDS). Following table shows strengths and weaknesses of Host based and IDS based IDSs.

Network based IDS	Host based IDS
<ul style="list-style-type: none"> • Broad in Scope • Examine Packet headers and entire packet • Near real time response • Host independent • Bandwidth dependant • No overload • Slow down the networks that have IDs clients installed • Detects network attacks, as payload is analysed • Not suitable for encrypted and switches network • Does not perform normally detection of complex attacks • High false positive rate • Lower cost of ownership • Better for detecting attacks from outside and detect attacks that host based IDS would miss 	<ul style="list-style-type: none"> • Narrow in scope, monitor specific activities • Does not see packet headers • Responds after a suspicious activity • Host dependant • Bandwidth independent • Overload • Slow down the hosts that have IDS clients installed • Detects local attacks before they hit the network • Well suited for encrypted and switches environment • Powerful tool for analysing a possible attack because of relevant information in database • Low false positive rate • Require no additional hardware • Better for detecting attacks from inside and detect attacks that network based IDS would miss

Table-1 Evaluation of Host based and Network based IDS

VI. INTRUSION DETECTION TECHNIQUES

There are two basic approaches to intrusion detection. The first approach, anomaly detection [4], attempts to define and characterize correct static form of data and/or acceptable dynamic behaviour. The second approach, called misuse detection, involves characterizing known ways to penetrate a system in the form of a pattern. Rules are defined to monitor system activity essentially looking for the pattern. Intrusion detection systems have been built to explore both approaches: anomaly detection and misuse detection [5]. In some cases, they are combined in a complementary way in a single intrusion detector.

	Advantage	Disadvantage
Misuse Detection	Accurately and generate much fewer false alarm	Cannot detect novel or unknown attacks
Anomaly Detection	Is able to detect unknown attacks based on audit records	High false-alarm and limited by training data

Table-2 Misuse Detection vs. Anomaly Detection

VII. CHARACTERISTICS OF IDS

After analysing the approaches taken by IDS at the operating system and network levels, some generic characteristics of intrusion detection became apparent.

It is possible for the IDS to evaluate all relations immediately after each *event*, the results of actions taken by users, processes, or devices that may be related to a potential intrusion. However, this may place an intolerable processing burden on the IDS. Therefore, events are typically collected in audit records over a period of time. Audit records entries can be reduced by combining some events into a single entry for analysis. For example, a single, failed log-in attempt is most likely insignificant, but many failed log-in attempts over a relatively short period of time may indicate a possible intrusion. The period of time between audit record analyses may be determined using real time or logical time where the relations are evaluated after a certain number of events have occurred. Audit records only deal with notions defined by the OS. Many aspects of the application are not visible to the OS and thus are not in the audit records. Fig-1 shows the Components if IDS [6].

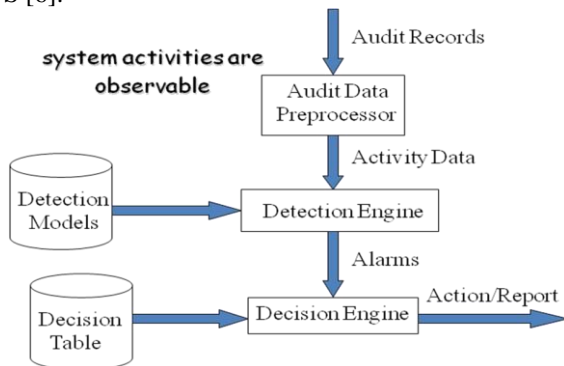


Fig-1 Components of Intrusion Detection System

VIII. REAL TIME INTRUSION DETECTION MECHANISM

The EIDM (Effective Intrusion Detection Mechanism) proposed here is independent of any particular system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose intrusion-detection expert system, The model is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of *system usage* [9].

The following examples illustrate:

- *Attempted Break-in:* Someone attempting to break into a system might generate an abnormally high rate of password failures with respect to a single account or the system as a whole.
- *Masking or Successful Break-in:* Someone logging into a system through an unauthorized account and password might have a different login time, location, or connection type from that of the account's legitimate user. In addition, the penetration's behaviour may differ considerably from that of the legitimate, user-, in particular, he might spend most of his time browsing through directories and executing system status commands, whereas the legitimate user might concentrate on editing or compiling and linking programs. Many break-ins have been discovered by security officers or other users on the system who have noticed the alleged user behaving strangely [11].
- *Penetration by Internal Users:* A user attempting to penetrate the security mechanisms in the operating system might execute different programs or trigger more protection violations from attempts to access unauthorized files or programs. If his attempt succeeds, he will have access to commands and files not normally permitted to him.
- *Leakage by Legitimate User:* A user trying to leak sensitive documents might log into the system at unusual times or route data to remote printers not normally used.
- *Trojan Horse:* The behaviour of a Trojan horse planted in or substituted for a program may differ from the legitimate program in terms of its CPU time or I/O activity.

IX. OVERVIEW OF THE MOEL

Today, damaging intrusions can occur in a matter of seconds. Intrusion detection has received increasing attention in recent years. One reason for this is the explosive growth of the internet and the large number of networked systems that exists in all types of organizations. The increase in the number of networked machines has led to an increase in unauthorized activity, not only from external attackers, but also from internal sources such as unsatisfied employees and people abusing their privileges for personal gain [8].

An intrusion mainly enters into a system by doing modifications on OS programming files of a system. Every operating system has its own set of critical files, whose access is generally protected by access control mechanisms, native to the operating system [5]. The importance of such files also simultaneously invites their inspection, unauthorized modification and tampering. So, the need for preserving the authenticity of these critical files along with tracking any unauthorized access to them demands paramount importance. This addresses the need of a good file-system intrusion detection system which is capable of monitoring and tracking any accidental, benign, malicious, intentional changes made to the files that reside in the file-system [6].

Hence to stop the intrusions into a system it is absolutely necessary to detect them first. The intrusion in a system's file system can be detected by presence of a new unfamiliar file in the system. An intrusion enters by doing modification of files in OS programming files folder like changes in the SYS32 folder files (for example .dll files). Hence each and every modification of a system should be monitored effectively to detect the intrusions.

The basic idea is to monitor the standard operations on a target system: logins, command and program execution's, file and device accesses, etc., looking only for deviations in usage. The model does not contain any special features for dealing with complex actions that exploit a known or suspected security flaw in the target system; indeed, it has no knowledge of the target system's security mechanisms or its deficiencies. Hence the motivation for using this Intrusion Detection technology is to collect forensic information to locate intruders [12].

The model has five main components:

1. **Initiators;** Initiates the activity on a target system- normally users.
2. **Resources;** managed by the system-files, commands, devices, etc. files, programs, messages, records, terminals, printers, and user- or program-created structures
3. **Activities;** Generated by the target system in response to actions performed or attempted by Initiators
4. **Reports;** Generated when initiator initiates the checking process.
5. **Time Stamp;** time period between configuration of the safe system and current system.

When the initiator starts his work in an organization, runs the checking process of the system and the report which is generated will be saved as safe system configuration report. In this model the Java code has been developed to scan the entire system resources.

Static `getCurrentSystemConfiguration()` SafeStore
Public -Used to read the current system configuration.

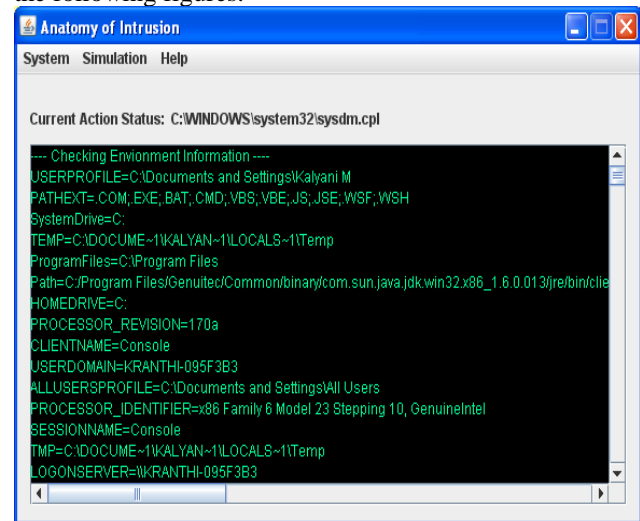
Static `getSystemSafeConfiguration()` SafeStore
Public - will get the system safe configuration

The system is scanned and the report is saved in the system for future verification.

Scanning means, we consider

- Which soft wares are installed?
- Present state of the system
- Storage space of the system

And this report is saved in the system as shown in the following figures.

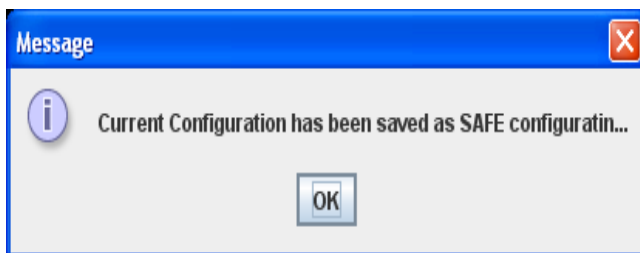
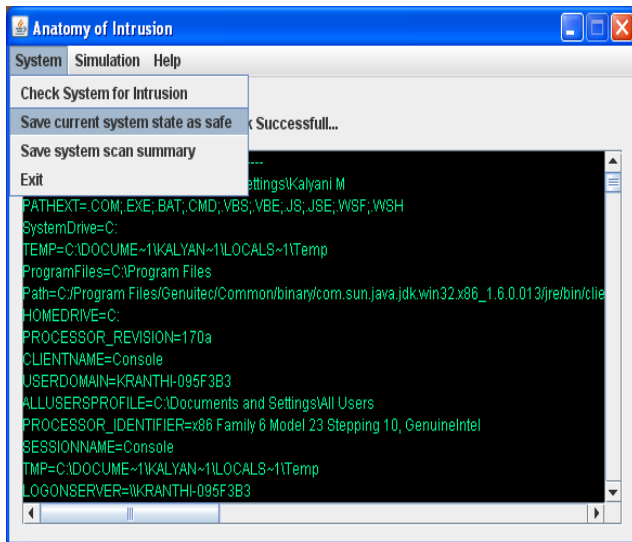
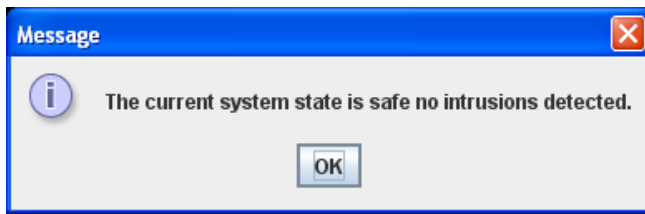


```

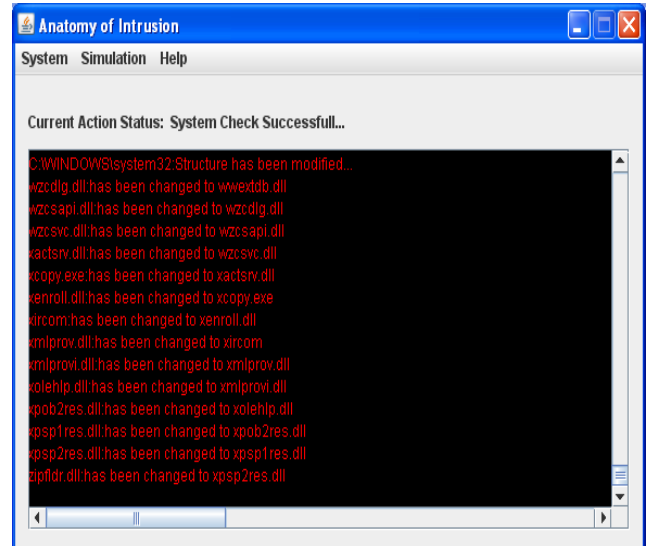
Anatomy of Intrusion
System Simulation Help

Current Action Status: C:\WINDOWS\system32\sysdm.cpl

--- Checking Environment Information ---
USERPROFILE=C:\Documents and Settings\Kalyani M
PATHEXT= .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
SystemDrive=C:
TEMP=C:\DOCUME~1\KALYAN~1\LOCALS~1\Temp
ProgramFiles=C:\Program Files
Path=C:\Program Files\Genuitec\Common\binary\com.sun.java.jdk.win32.x86_1.6.0.013\jre\bin\clle
HOMEDRIVE=C:
PROCESSOR_REVISION=170a
CLIENTNAME=Console
USERDOMAIN=K-RANTHI-095F3B3
ALLUSERSPROFILE=C:\Documents and Settings\All Users
PROCESSOR_IDENTIFIER=x86 Family 6 Model 23 Stepping 10, GenuineIntel
SESSIONNAME=Console
TMP=C:\DOCUME~1\KALYAN~1\LOCALS~1\Temp
LOGONSERVER=\\K-RANTHI-095F3B3
  
```



changes have been done to SYS32 folder. We have added a new .dll file to the SYS32 folder and run the scanning process. Reports generated can be seen in the following figures.



When the initiator finished his/her shift or finished the work with the system again the checking process will be done and the report which is generated after scanning will be saved as current system configuration report. The safe system report and current system report will be compared and analysed to easily locate the intruders.

The comparison is basically on the following

- Modifications in the storage space
- Any software installed or not
- Whether any person used or not
- Any changes in the registry

In the following figures we can see the screenshots of the reports which are generated when some

The current system report gives the detailed view of all installations and modifications which have been done during the time stamp. If an intruder wants to attack on a targeted system, then modification of files in OS programming files folder is necessary. Hence each and every modification of a system should be monitored effectively to detect the intrusions. The mechanism proposed here gives a detailed report of each and every modification done during the time zone. The report contains date, time, modified file name and actions performed on that particular resource. For example if .dll file has been modified in SYS32 folder during the time zone the report contains each and every information about that modification.

The method can be used by employees of the organization in different shifts. The security management should make it mandatory to run the checking process and saving the reports for each login in to the system. The generated reports can be analysed by security management of the organization to locate the intruders. By analysing these Red Team reports management can easily detect abusing activities and internal intruders who misuse their privileges.

X. CONCLUSION

We believe the EIDM model provides a basis for developing powerful real-time intrusion detection capable of detecting a wide range of intrusions related to attempted break-ins, masquerading (successful break-ins), system penetrations, Trojan horses, leakage and other abuses by legitimate users, and certain covert channels.

But there are several open questions like Soundness, Completeness, System Design, and Feedback of this approach. Although we believe that the approach can detect most intrusions, it may be possible for a person to escape. For example, because it is not practical to monitor individual page faults, a program that leaks data covertly by controlling page faults would not be detected-at least by its page-fault activity. The work in intrusion detection techniques and methodologies which has been a major focus of information security-related research in the past two decades is certain to continue. The area of intrusion detection is continuing to evolve. While a number of methodologies and tools have been designed to assist in the identification of intruders, no definable standard has been developed which could serve as the basis for a deployable intrusion detection tool. However, as the processing capabilities of computer systems improve and the innovative approaches to intrusion detection continue to be developed, the creation of an effective intrusion detection standard is inevitable.

REFERENCES

- [1] John McHugh, Alan Christie, and Julia Allen ,Software Engineering Institute, CERT Coordination Center, Defending Yourself: The Role of Intrusion Detection Systems
- [2] SANS Institute, Adrian Brindley, Denial of Service attacks and the emergence of "Intrusion Prevention Systems, November 1, 2002
- [3] Shari Lawrence Pfleeger, *RAND Corporation*, Anatomy of Intrusion,
- [4] James Cannady Jay Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies", Georgia Tech Research Institute Georgia 30332-0800
- [5] "A Comparative Analysis of Current Intrusion Detection Technologies", James Cannady Jay Harrell Georgia Tech Research Institute Georgia Tech Research Institute Georgia Institute of Technology Georgia Institute of Technology Atlanta, Georgia 30332-0800 Atlanta, Georgia 30332-0800
- [6] Tyrone Grandison and Evimaria Terzi, "Intrusion Detection Technology" IBM Almaden Research Center 650 Harry Road, San Jose, CA 95120 ftyroneg.eterzig@us.ibm.com September 7, 2007
- [7] Robert S. Sielken, Anita K. Jones, "Application Intrusion Detection Systems: The Next Step", University of Virginia, September 1999
- [8] Neumann, P.G. (1985). "Audit Trail Analysis and Usage Collection and Processing. Technical Report Project 5910", SRI International.
- [9] Storage-based Intrusion Detection: Watching storage activity for suspicious behavior Adam G. Pennington, John D. Strunk, John Linwood Griffin, Craig A.N. Soules, Garth R. Goodson, Gregory R. Ganger
- [10] R. Lippmann et al., "Evaluating Intrusion Detection Systems: The 1998 DAPA Offline Intrusion Detection Evaluation," *Discex 2000*, Vol. 2, IEEE Computer Soc. Press, Los Alamitos, Calif., 2000, pp. 12–26
- [11] Nicholas J. Puketza, Kui Zhang, Mandy Chung, Biswanath Mukherjee*, Ronald A. Olsson, A Methodology for Testing Intrusion Detection Systems
- [12] "Network Security – A Layered Approach", Kanchan Bala1, Barjeena Lucky2, Surinder Pal Garg3 *IH. I. M. T. Greater Noida, 2D. N. College, Hisar, 3G. P. Hisar*
- [13] *Guide to Intrusion Detection and Prevention Systems*, Karen Scarfone, Peter Mell, February 2007.

Authors:

1. **KALYANI KUNDETI**, PG Scholar, Department of IT, Aurora's Engineering College, Bhongir, Andhra Pradesh, India.



2. **Dr.M.V.Vijaya Saradhi**

received his Ph.D degree from Faculty of Engineering, Osmania University (OU), Hyderabad, Andhra Pradesh, India. He is Currently Working as Professor in the Department of Information Technology (IT) at Aurora's Engineering College, Bhongir, Andhra Pradesh, India. His main research interests are Software Metrics, Distributed Systems, Object-Oriented Modeling, Mobile Environment, Data Mining, Design Patterns, Object-Oriented Design Measurements and Empirical Software Engineering. He is a life member of various Professional bodies like MIETE, MCSI, MIE, MISTE. Contact him at