ABSTRACT
Internet provides huge information and value to the users but at the same time access to the internet is prone to increasing number of attacks. Tracing the source of the attacking packet is very difficult because of stateless and destination based routing infrastructure of Internet. If the attacks are detected successfully, then preventive measures for attacks can be taken. Host based and network based intrusion prevention systems are available in the market. Host based Intrusion Prevention Systems are designed to protect information systems from unauthorized access, damage or disruption. To counteract the rapidly evolving threats presented by the latest generation of worms, software and network exploits knowledge of intrusion detection and prevention is very much important. This paper provides the in depth study of methodologies and models used for intrusion-detection. Details about prevention technologies are also stated. Attacks on intrusion prevention systems and counter measure is also described. We propose our approach for intrusion detection and prevention in the networks.

KEYWORDS Network, Security, Attacks, Intrusion, Prevention

I. INTRODUCTION
Unauthorised intrusions into a computer system or network are one of the most serious threats to computer security. All the attacks on hardware and software relate to network security because system entry can be achieved by means of a network. Anderson identified three classes of intruders

- Masquerader:-- an individual who is not authorised to use the computer and who penetrates a system's access controls to exploit a legitimate user's account.
- Misfeasor:-- A legitimate user who accesses data, programs or resources for which such access is not authorised or whoever is authorised for such access but misuses his or her privileges.
- Clandestine user:-- An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

Intruder attacks range from the simple to serious levels. There are many people who wish to exploit internet which may cause unintentional intrusion. These kinds of intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. At the serious level people who are attempting to read privileged data, perform unauthorised modifications to data or disrupt the system. Various techniques are used for intrusions. The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Generally this requires the intruder to acquire information that should have been protected. Intruders can get access to a system by exploiting attacks such as buffer overflows on a program that runs. The paper is organised as below: In section II we describe the different intrusion detection approaches and models used for it. In section III different Information sources for intrusion detection are discussed. Section IV describes the Intrusion prevention system, attacks on IPS and countermeasures. In Section we propose our approach for intrusion detection and prevention. Section IV concludes the paper.

II. INTRUSION-DETECTION APPROACHES AND MODELS
Intrusion detection is the process of monitoring the events occurring in a computer system or network and analysing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. Intrusions are caused by attackers accessing the systems from the Internet, authorised users of the systems who attempt to gain additional privileges for which they are not authorised and authorised users who misuse the privileges given them. [2]. Intrusion detection and this has been focus of research is recent years. This interest is motivated by a number of considerations including the following.

- if an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised. Even if the detection is not sufficiently timely to pre-empt the intruder, the sooner that the intrusion is detected, the less the amount of damage and more quickly that recovery can be achieved.
- An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusion.
- Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Intrusion detection is based on the assumption that the behaviour of the intruder differs from that of a legitimate user in ways that can be quantified. The typical behaviour of an intruder differs from the typical behaviour of an authorised user, there is overlap in these behaviours, thus a loose interpretation of intruder behaviour, which will catch more intruders, will also lead to a number of false positives. On the other hand, an attempt to limit false positives by a tight interpretation of intruder behaviour will lead to an increase in false negatives. In Anderson's study, it was postulated that one could distinguish between a masquerader and a legitimate
user. Patterns of legitimate user behaviour can be established by observing past history and significant deviation from such patterns can be detected. Anderson suggests that task of detecting a misfeasor is more difficult, in that the distinction between abnormal and normal behaviour may be small. Anderson concluded that such violations would be undetectable. Following approaches are used for intrusion detection

- **Statistical anomaly detection**: Involves the collection of data relating to the behaviour of legitimate users over a period of time. Then statistical tests are applied to observed behaviour to determine with a high level of confidence whether that behaviour is not legitimate behaviour.
  - **Threshold detection**: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.
  - **Profile based**: A profile of the activity of each user is developed and used to detect changes in the behaviour of individual accounts.
- **Rule based detection**: Involves an attempt to define a set of rules that can be used to decide that a given behaviour is that of an intruder.
  - **Anomaly detection**: Rules are developed to detect deviation from previous usage patterns.
  - **Penetration identification**: An expert system approach that searches for suspicious behaviour.

The approaches stated above are discussed in detail below

**A) Statistical Anomaly Detection**

Statistical approaches attempt to define normal or expected behaviour, whereas rule base approaches attempt to define proper behaviour. Statistical anomaly detection is effective against masqueraders who are unlikely to mimic the behaviour patterns of the accounts they appropriate. On the other hand, such techniques may be unable to deal with illegitimate users.

A fundamental tool for intrusion detection is the audit record. Some records of ongoing activity by users must be maintained as input to an intrusion detection system.

Two techniques are used for this

**1. Native audit records:**

Usually all multi user operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional software is required. The disadvantage is that the native audit records may contain unwanted information and we may require additional processing of the information.

**2. Detection specific audit records:**

A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. Advantage of this approach is that it could be made vendor independent and ported to a variety of systems.

Each audit record contains following important fields.

**Subject-** A subject is typically a terminal user but might also be process acting on behalf of users or groups of users. All activity arises through commands issued by subjects. Subjects may be grouped into different access classes and these classes may overlap.

**Action:** Operation performed by the subject on or with an object like read, execute etc.

**Object:** Recipient of action. Examples include files, programs, messages, records, terminals, printers.

When a subject is the recipient of an action, such as electronic mail then that subject is considered an object. Objects may be grouped by type. Object granularity may vary by object type and by object environment. Database actions may be audited for the database as a whole or at the record level.

**Exception:** Denotes which exception condition is raised on return.

**Resource:** Usage: A list of quantitative elements in which each element gives the amount used of some resource.

**Time-Stamp:** Unique time and date stamp identifying when the actions took place.

Statistical Anomaly Detection falls into two categories: threshold detection and profile based systems. Threshold detection involves counting the number of occurrences of a specific event type over an interval of time. If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed. Threshold analysis is a crude and ineffective for even moderately sophisticated attacks. Both the threshold and time interval determination is required. Because of the variability across users, such thresholds are likely to generate either a lot of false positives or a lot of false negatives. However simple threshold detectors may be useful in conjunction with more sophisticated techniques.

Profile based anomaly detection focuses on characterising the past behaviour of individual users or related groups of users and then detecting significant deviations. A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert. Audit records serve to define typical behaviour. The intrusion detection model analyses incoming audit records to determine deviation from average behaviour. Examples of metrics that are useful for profile based intrusion detection are the following

**1. Counter:**

A non negative integer that may be incremented but not decremented until it is reset by management action. Typically a count of certain event types is kept over a particular period of time.

**2. Gauge:**

A non negative integer that may be incremented or decremented. Typically a gauge is used to measure
the current value of some entity. Interval timer: the length of time between two related events.

3. Resource Utilization:
Quantity of resources consumed during a specified period. Using these general metrics the statistical test like mean and standard deviation of a parameter can be conducted. This gives a reflection of the average behaviour and its variability. A multivariate model is based on correlations between two or more variables. Intruder behaviour may be characterized with greater confidence by considering such correlations. A markov process model is used to establish transition probabilities among various states. A time series model focuses on time intervals, looking for sequences of events that happen too rapidly or too slowly. An operational model is based on judgement of what is considered abnormal rather than an automated analysis of past audit records. Intrusion is suspected for an observation outside limits.

B) Rule based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious. Rule based anomaly detection is similar in terms of its approach and strengths to statistical anomaly detection. With the rule based approach, historical audit records are analyzed to identify usage patterns. Rules may represent past behaviour patterns of users, programs, privileges, time slots, terminals and so on. Current behaviour is then observed and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behaviour.

III. INFORMATION SOURCES FOR INTRUSION DETECTION

For attack detection and analysis information gathering plays very important part. Different sources can be used for Intrusion Detection Systems. These information sources can be categorized as network based and host based. They are described in detail below.

A) Network-based Information sources:

1. SNMP Information

The Simple Network Management Protocol (SNMP) Management Information Base (MIB) is a repository of information used for network management purposes. It contains configuration information (routing tables, addresses, names) and counters to measure the traffic at various network interfaces and at different layers of the network. This section describes experiments performed within the SECURENET project [53] to use SNMP V1 common MIB for Ethernet and TCP/IP. Other projects also aim at using SNMPv2 and SNMPv3 for security and intrusion detection [32].

The SECURENET project explored whether the counters maintained in this MIB are usable as input information for a behaviour-based intrusion-detection system. The starting point was to examine the counters at the interface level because this was the only place where one can differentiate between information sent over the wire and information transmitted inside the operating system via the loopback interface. The prototype collected increments on the number of bytes and packets transmitted and received at each interface every five minutes. The outcome of a very simple average/standard deviation analysis of this data was not satisfactory, as the standard deviation was larger than the average for almost all sets collected during daytime activity.

MIB counters at higher levels of the network do not contain much more information. On the IP, TCP and UDP layers, the counters exhibited similar behaviour but, owing to the larger number of counters at these layers, we did not compute all possible correlations. The ICMP counters show more consistency with respect to their statistical modelling, but ICMP attacks [4] have not been tried to validate this approach. This study shows that SNMP MIBs are a potentially interesting candidate as an audit source for intrusion detection systems. The demise of SNMPv2 owing to a lack of consensus on the security features has certainly lowered the interest of the intrusion-detection community in it. However, with the deployment of SNMPv3, new projects are exploiting its features for intrusion-detection tools [32].

2. Network packets

As the popularity of network sniffers for gathering information has grown in the attacker’s community, they are nowadays also regarded as an efficient means for gathering information about the events that occur on the network architecture. This is consistent with the trend of moving from a centralized to a distributed computing model, and the pace of change has even increased with the widespread use of the Internet. Most accesses to sensitive computers today take place over a network, and therefore capturing the packets before they enter the server is probably the most efficient way to monitor this server.

It is also consistent with the occurrence of denial-of-service attacks. As companies put valuable information on the Internet, and even depend on it as a source of revenue, the prospect of simply shutting down a web site creates an effective threat to the organization running it. Most of these denial-of-service attacks originate from the network and must be detected at the network level, as a host-based intrusion-detection system does not have the capability to acquire this kind of audit information. There is an inherent duality in network sniffers, which is also apparent in the firewall world with its differences between application-level gateways and filtering routers [3]. If the analysis is carried out at a low level by performing pattern matching, signature analysis, or some other kind of analysis of the raw content of the TCP or IP packet, then the intrusion-detection system can perform its analysis quickly. This is a stateless approach that does not take session information into account because the latter could span several network packets. If the intrusion-detection system acts as an application gateway and analyzes each packet with respect to the application or protocol being followed, then the analysis is more
thorough, but also much more costly. This is a stateful analysis. Note that this analysis of the higher levels of the protocol also depends on the particular machine being protected, as implementations of the protocols are not identical from one network stack to another.

The detection of network-specific attacks

There are a number of network attacks, particularly denial-of-service attacks that cannot be detected in a timely fashion by searching for audit information on the host, but only by analyzing network traffic. The impact of auditing on the host performance. Information is entirely collected on a separate machine, with no knowledge of the rest of the network. Therefore, installation of such tools is facilitated because they do not impact the entire environment in terms of configuration and performance. The heterogeneous audit trail formats. The current de facto standardization towards TCP/IP facilitates the acquisition, formatting, and cross-platform analysis of the audit information. Certain tools analyze the payload of the packet, which allows attacks against hosts to be detected by signature analysis. However, an effective analysis requires knowledge of the type of machine or application for which the packet is intended. But it also has a number of drawbacks:

- It is more difficult to identify the culprit when an intrusion has been discovered. There is no reliable link between the information contained in the packets and the identity of the user who actually submitted the commands on the host.
- With switched networks (switched-Ethernet, switched Token-Ring, ATM), selecting an appropriate location for the sniffer is not straightforward. Some tools are located on switches, others at gateways between the protected system and the outside world. The former gives better audit information but also incurs a higher cost. Note, however, that switched networks are also much less vulnerable to sniffer attacks [6, 44] and actually are a recommended solution to improve the security of a network.
- Encryption prevents the analysis of the payload of the packets, and therefore hides a considerable amount of important information from these tools. Also, even without encryption, it is possible to obfuscate the contents of the packet to evade detection [44] if the signatures are not sufficiently comprehensive.
- Systematic scanning, for example at the firewall, is difficult because it might create bottlenecks. This will only worsen as the bandwidth to access the Internet is increased at sensitive sites (e.g. banks, electronic-commerce web sites).
- Finally, these tools are inherently vulnerable to denial-of-service attacks if they rely on a commercial operating system to acquire network information. Just as the network stacks of these commercial operating systems are vulnerable to attacks, so is the intrusion-detection system.

Network packets currently are the source of information used by several recent commercial products [7, 28, 59], and several projects in the research community also pursue this track [41, 45, 46, 54]. A recent evaluation of these products by Ptacek and Newsham [44] shows that the sniffer approach, or at least the current implementations, have flaws that make it possible for a skilled attacker to evade detection. In particular, they show that IP fragmentation is not handled well, and that the use of wildcards and control sequences in protocols such as http makes it possible to evade detection by signature. Also in this area is research being conducted? After IDES and NIDES, SRI is now developing a prototype called Emerald [42] to deal with the analysis of network traffic. Other network sniffers such as Bro [41] or Network Flight Recorder [45] have been developed as network data-acquisition tools and therefore do not support intrusion-detection.

B) Host-based information sources

Host audit sources are the only way to gather information on the activities of the users of a given machine. On the other hand, they are also vulnerable to alterations in the case of a successful attack. This creates an important real-time constraint on host-based intrusion-detection systems, which have to process the audit trail and generate alarms before an attacker taking over the machine can subvert either the audit trail or the intrusion-detection system itself.

1. Accounting

Accounting is one of the oldest sources of information on system behavior. It provides information on the consumption of shared resources, such as processor time, memory, disk or network usage, and applications launched, by the users of the system. Accounting is found almost everywhere, in network equipment, in mainframes as well as in UNIX workstations. This omnipresence has led some designers of intrusion-detection prototypes to try to use it as an audit source. In the UNIX environment, accounting is a universal source of information. The format of the accounting record is the same on all UNIXes, the information is compressed to gain disk space, and the overhead introduced by the recording process is very small. It is well integrated in modern operating systems, and easy to set up and exploit.

However, accounting information also has a number of drawbacks, which make it untrustworthy for security purposes. In particular, the information identifying the command launched as well as the time stamps are too imprecise to allow efficient detection of attacks.

2. Syslog

Syslog is an audit service provided to applications by the operating system (UNIX and others). This service receives a text string from the application, prefixes it with a time stamp and the name of the system on which the application runs, and then archives it, either locally or remotely.

Syslog is known to have security vulnerabilities, as syslog daemons on several UNIX operating systems have been the subject of CERT documents [5] showing that buffer overflows in the syslog daemon can be exploited to execute arbitrary code. Syslog is very easy to use, and that has incited many application developers to use it as their audit trail. A number of applications and network services use it, such as login, sendmail, nfs, http, including security-related tools such as sudo, klaxon, or TCP wrappers. Therefore, a few intrusion-detection tools have been developed that use information provided by the syslog daemon, for example Swatch [22]. Although syslog is a lightweight audit source that does not generate a large amount of audit data per machine, a large network can generate a large number of messages, very few of which are security-relevant. Swatch [3] reduces the burden of the system administrator by correlating messages (reports from several machines that an nfs server is down would be aggregated into one) and highlighting security-related ones.

3. C2 security audit

The security audit records all potentially security-significant events on the system. As the US government has required that all computer systems it purchases certified at the C2 level of the TCSEC [56], all operating-system vendors competing in this area have had to include an "accountability" feature. This translates into security audit trails such as SUN's BSM and Shield packages, or AIX audit. All these security audit trails have the same basic principle: They record the crossing of instructions executed by the processor in the user space and instructions executed in the Trusted Computing Base (TCB) space [3]. This security model postulates that the TCB is trusted, that actions in the user space cannot harm the security of the system, and that security-related actions that can impact the system only take place when users request services from the TCB.

In the UNIX environment, the TCB is basically the kernel. Therefore, the audit system records the execution of system calls by all user-launched processes. Compared with a full system-call trace, the audit trail provides limited abstraction: context switches, memory allocation, internal semaphores, and consecutive file reads do not appear in the trail. On the other hand, there is always a straightforward mapping of audit events to system calls.

The UNIX security audit record contains a vast amount of information about the events. It includes detailed user and group identification (from the login identity to the one under which the system call is executed), the parameters of the system call execution (i.e. names including path, command line arguments, etc.), the return code from the execution, and the error code.

The main advantages of the security audit are a strong identification of the user, its login identity, its real (current) identity, its effective (set-user-id bit) identity, its real and effective (set-group-id bit) group identities; a repartition of audit events into classes to facilitate the configuration of the audit system; a fine-grained parametrization of the information gathered according to user, class, audit event, and failure or success of the system call, and a shutdown of the machine if the audit system encounters an error status (usually a running out of disk space).

The main drawbacks of the security audit are a heavy use of system resources when detailed monitoring is requested. Processor performance could potentially be reduced by as much as 20%, and requirements for local disk space storage and archiving are high; a possible denial-of-service attack by the audit file system; difficulty to set up the audit service owing to the number of parameters involved. Standard configuration delivered by vendors minimize the performance hit by recording only classes of rare events (administrative actions, logins and logouts).

The auditing requirements of an intrusion-detection tool demand more detailed information, such as file accesses, processes executed; difficulty to exploit the information obtained owing to its size and complexity. This is compounded by the heterogeneity of audit-system interfaces and audit record formats in the various operating systems and the parametrization of the audit system involving subjects (users) and actions (system calls or events), and only very rarely objects (on which the action is performed). Important objects should be monitored by an intrusion-detection tool, and this is done primarily by scanning the entire trail. The C2 security audit is the primary source of audit information for the majority of host-based intrusion detection prototypes and tools because it currently is the only reliable mechanism for gathering detailed information on the actions taken on an information system. Work has been conducted by several groups [21, 39, 43, 55] to define which information should be included in the security audit trail as well as a common format for audit trail records, but this is an ongoing research effort.

IV. INTRUSION DETECTION AND PREVENTION SYSTEMS

Figure 1. Intrusion Detection and Prevention Systems

Intrusion Prevention Systems (IPSs) are software or hardware products that automate the monitoring and analysis process of intrusion-detection [2]. 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. Some of the measures that are used for Intrusion Detection and type of intrusion detected by them is given in Table no.1 [1]

**Table 1: Intrusion detection Model**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Model</th>
<th>Intrusion Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login frequency</td>
<td>Mean and</td>
<td>Intruders trying to log in during off hours</td>
</tr>
<tr>
<td>day and time</td>
<td>standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deviations</td>
<td></td>
</tr>
<tr>
<td>Time since last login</td>
<td>Operational</td>
<td>Dead account</td>
</tr>
<tr>
<td>Password failures</td>
<td>Operational</td>
<td>Password guessing and cracking</td>
</tr>
<tr>
<td>Program execution</td>
<td>Mean and</td>
<td>Intrusions which have successfully gained access to privileged programs.</td>
</tr>
<tr>
<td>frequency</td>
<td>standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deviation</td>
<td></td>
</tr>
<tr>
<td>Execution denials</td>
<td>Operational</td>
<td>May detect penetration attempts.</td>
</tr>
<tr>
<td>Records read, written</td>
<td>Mean and</td>
<td>Sensitive data access by interference and aggregation</td>
</tr>
<tr>
<td></td>
<td>standard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deviation</td>
<td></td>
</tr>
</tbody>
</table>

Almost all IDSs will output a small summary line about each detected attack. This summary line typically contains the information about time/date, sensor IP address, vendor specific attack name, standard attack name (if one exists), source and destination IP address, source and destination port numbers, and network protocol used by attack. Many IDSs will also provide a generic description of each type of attack. This description is important as it enables the operator to correctly gauge the impact of the attack. This description usually contains the following text description of attack, attack severity level, type of loss experienced as a result of the attack, the type of vulnerability the attack exploits, list of software types and version numbers that are vulnerable to the attack, patch information so that computers can be made invulnerable to the attack, and references to public advisories about the attack or the vulnerability it exploits.

Intrusion detection systems perform the following functions:

- Monitoring and analysis of system events and user behaviours
- Testing the security states of system configurations
- Baselining the security state of a system, then tracking any changes to that baseline
- Recognizing patterns of system events that correspond to known attacks
- Recognising patterns of activity that statistically vary from normal activity
- Managing operating system audit and logging mechanisms and the data they generate
- Alerting appropriate staff by appropriate means when attacks are detected
- Measuring enforcement of security policies encoded in the analysis engine
- Providing default information security policies

**Limitations of Intrusion Detection Systems**

Intrusion detection systems cannot perform some of the functions. IDS can not compensate for weak or missing security mechanisms in the protection infrastructure. Such mechanisms include firewalls, identification and authentication, link encryption, access control mechanisms, and virus detection and eradication. Instantaneous detection, reporting, and response to an attack, when there is a heavy network or processing load is not possible. Detecting new attacks or variants of existing attacks can not be done in IDS. Automatically investigating attacks without human intervention is also difficult. IDS can not deal effectively with switched networks.

Attacks on Intrusion Prevention System and protection. Intrusion detection technology is a necessary addition to every large organisation’s computer network security infrastructure. However, given the deficiencies of today’s intrusion detection products, and the limited security skill level of many system administrators, an effective IDS deployment requires careful planning, preparation, prototyping, testing, and specialised training.

We recommend consideration of a combination of network-based IDSs and host based IDSs to protect an enterprise-wide network. We furthermore recommend a staged deployment, starting with network-based IDSs as they are usually the simplest to install and maintain. Next, protect critical servers with host-based IDSs. Utilize vulnerability analysis products on a regular schedule to test IDSs and other security mechanisms for proper function and configuration.

When an intrusion prevention system is deployed, it becomes the natural primary target of hostile attacks, with the aim of disabling the detection feature and allowing an attacker to operate without being detected.

Disabling the intrusion prevention system can happen in the following ways:
- Denial-of-service attacks:
  - Denial-of-service attacks are a powerful and relatively easy way of temporarily disabling the intrusion prevention system. The attack can take place against the detector, by forcing it to process more information than it can handle for example by flooding a network link. This usually has the effect of delaying detection of the attack or, in the worst case, of confusing the detector enough so that it misses some critical element of the attack. A second possibility is to saturate the reaction capability of the process handling the intrusion prevention system.
  - When the process is presented with too many alarms,
he can easily miss the important one indicating penetration, even if it is present on the screen. Several techniques have been developed to evade detection of an attack by intrusion prevention systems. Network-based tools, the most popular tools today, particularly suffer from these attacks involving hand-crafted network packets:

1. Intrusion prevention systems have difficulties reassembling IP packets if packets are fragmented. Therefore, splitting an attack artificially into multiple packets creates a mismatch between the data in the packet and the signature, thus hiding the attack.

2. Attack via the TTL (Time To Live). By altering the TTL of IP packets, it is possible to make the intrusion-detection system see packets that will not arrive at the target of the attack. By inserting fake data into the communication stream, an attacker can interleave the attack with incorrect information, thus hiding the attack from the intrusion detection system while the target correctly reconstructs this attack data and reacts to it.[6]

Intrusion prevention is defined as the complete process of identifying and responding to malicious activities targeted at computing and network resources [8].

V. CONCLUSIONS

Intrusion detection currently attracts considerable interest from both the research community and commercial companies. Research prototypes continue to appear, and commercial products based on early research are now available. In this paper, I have given an overview of the current state of the art of intrusion detection, based on a proposed taxonomy illustrated with examples of past and current projects. The taxonomy clearly highlights the properties of these intrusion-detection systems, covering both past and current developments adequately. Information sources for these tools are either a C2 audit trail, syslog, or network packets. Whereas system sources were widely used in the early stages of research, the current focus of research prototypes as well as products is on protecting the infrastructure rather than the end-user station, and this paradigm has led to the use of network sniffers that analyse packets. As shown, quite a number of research issues concerning the efficiency of both network and host audit sources, the formatting and existence of a common audit trail format, and even the contents of the audit trail itself, still await an answer. There are also a number of unsolved issues concerning the analysis of the audit trail. Signature analysis is clearly in the commercial domain now, but has been shown to be insufficient for detecting all attacks. Therefore, work is still in progress to experiment with new approaches to both knowledge-based and behavior based intrusion detection. The detection of abuse-of-privilege attacks (primarily insider attacks) is also the subject of ongoing work. This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

REFERENCES


