Network Defense: Approaches, Methods and Techniques

Rup Kumar Deka\textsuperscript{a}, Kausthav Pratim Kalita\textsuperscript{a}, D. K. Bhattacharya\textsuperscript{a}, Jugal K. Kalita\textsuperscript{b}

\textsuperscript{a}Department of Computer Science and Engineering, Tezpur University, Napaam, Assam, India
\textsuperscript{b}Department of Computer Science, College of Engineering and Applied Science, University of Colorado, Boulder, CO, United States

Abstract

To defend a network from intrusion is a generic problem of all time. It is important to develop a defense mechanism to secure the network from anomalous activities. This paper presents a comprehensive survey of methods and systems introduced by researchers in the past two decades to protect network resources from intrusion. A detailed pros and cons analysis of these methods and systems is also reported in this paper. Further, this paper also provides a list of issues and research challenges in this evolving field of research. We believe that, this knowledge will help to create a defense system.

Keywords: DoS, Intrusion, Defense, Response, Tolerance.

1. Introduction

Computerization and internetization of the world is happening at an astonishing speed. In spite of growth at breakneck pace, service providers are doing their best to provide the highest quality of service. At every step, an aspect that stands out is security, which is indeed a very serious topic of concern. An intrusion or attack may be fast or slow. When an attack uses large size packets or extremely high volume traffic within a very short time, say a fraction of a minute, to disrupt service, it can be termed a fast attack. On the other hand, some attacks take minutes or hours to complete the process, and are referred to as slow attacks.

Frequently, network or system activities are carried out with malicious intentions or other network policy violations take place. This type of attempt or activity can be termed intrusion and its creator is known as an intruder. The goal of intrusion detection is to make the whole network secure by thwarting attempts to compromise confidentiality, integrity or availability of resources.

1.1. Motivation

There are several published surveys on approaches to intrusion detection and/or prevention such as Patel et al. (2010), Bhuyan et al. (2014), Hoque et al. (2013), Kumar (2007), Richhariya and Srivastava (2013), Patel et al. (2013). These authors usually provide details of a few approaches although some cover a larger number of defense systems. Bhuyan et al. (2014) present a comprehensive survey of DDoS attacks, detection methods and tools used in wired networks. Hoque et al. (2013) provide a taxonomy of attack tools and also present a comprehensive and structured survey of existing tools and systems that can support both attackers and network defenders. An exhaustive survey of intrusion defense systems is presented by Patel et al. (2013), where the authors discuss approaches against intrusion by creating a layered taxonomy in addition to discussing cloud-based intrusion defense systems. Neither of the surveys by Patel et al. (2010) and Richhariya and Srivastava (2013) include issues of defense, challenges and solutions. In this paper we present a structured and comprehensive survey of defensive approaches, in terms of general overview, modules of a defense architecture, infrastructure and a taxonomy. We also attempt to present challenges in developing effective defensive approaches.

This paper provides a structured and comprehensive survey of approaches to counter intrusions. The major contributions of this survey are the following.

- Our presentation is more streamlined. First, we describe a defense system, in particular whether it detects or prevents intrusions considering the modules it contains. Then we focus on various detection techniques. Infrastructure needs, location and control of defense systems are also discussed.
- Most existing surveys do not fully cover the large number of issues, related to intrusion defense systems, but we do.
- We present a taxonomy to ensure that we cover a large area within the intrusion defense process.
- We also identify challenges encountered by approaches to prevent intrusions.

1.2. Prior Surveys

Richhariya and Srivastava (2013) address issues of information security and describe the security needs of an organization to protect its critical information from attacks. A well trained staff of analysts is required to continuously monitor...
Table 1: Comparison with Existing Surveys

<table>
<thead>
<tr>
<th>References</th>
<th>IDS</th>
<th>IPS</th>
<th>Different Approaches</th>
<th>Defense challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bai and Kobayashi (2003)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Murali (2005)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Kabiri and Ghorbani (2005)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Kumar (2007)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Patel et al. (2010)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sandhu et al. (2011)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Rathore (2012)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Patel et al. (2013)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Richhariya and Srivastava (2013)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Bhuyan et al. (2014)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>This Paper</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

the system. In such an environment, a huge amount of effort is required to construct new security strategies. Patel et al. (2010) review current trends in intrusion detection together with a study of implemented technologies. Kabiri and Ghorbani (2005) identify main categories of intrusion detection and prevention systems. They also provide a comparison of various approaches. Rathore (2012) also provides a survey of different approaches to intrusion detection. Sandhu et al. (2011) reviews methods for building Intrusion Detection and Prevention Systems (IDPS) and uses a cost-effective intrusion detection and prevention method based on the concept of intelligent mobile agents to design an effective Agent based Intrusion Prevention System (AIPS). AIPS works well in a distributed environment due to the use of software agents.

Murali (2005) surveys recent IDPSs and alarm management techniques by providing a comprehensive taxonomy and investigating possible solutions to detect and prevent intrusions in cloud computing systems. Considering the desired characteristics of IDPSs and cloud computing systems, a list of requirements is identified and four concepts of autonomic computing, viz., self-management, ontology, risk management, and fuzzy theory are leveraged to satisfy these requirements.

A survey of technologies for defense against intrusion is given in Patel et al. (2013). This paper discusses aspects of intrusion defense systems and data collection techniques. Data mining-based and data fusion-based IDSs are discussed to emphasize the need for large-scale data collection. Current defense technologies face powerful challenges and these are also described here, along with some suggested methods to overcome them.

Bai and Kobayashi (2003) describe detailed designs of both signature and anomaly-based NIDS (Network based Intrusion Detection System). Requirements of such systems are thoroughly discussed. Kumar (2007) presents a nomenclature of IDSs that he uses for his survey. This paper also identifies strengths as well as the limitations of several IDSs.

Our survey differs from these previous surveys in the following ways.

- In all the papers mentioned in this section, there is little information regarding where to deploy IDSs and other details of issues in deployment of IDSs.
- Most papers, which are mentioned in this section, do not provide any discussion of challenges faced when an intrusion defense system is deployed.
- We describe modules of an intrusion defense model in this paper. A thorough understanding of these modules is necessary to develop successful defense systems. Such discussions are not usually found in other survey papers.

1.3. Organization

The rest of the paper is organized as follows. Concepts related to approaches and intrusion defense systems are discussed in Section 2. This section also presents a classification of intrusion detection, prevention, response and tolerance systems. A selection of such systems is presented in Section 3. Section 4 is dedicated to issues and challenges in building a defense system. Finally, we present conclusions in Section 5.

2. Intrusion Defense Solutions

We can visualize three different types of attack which are mentioned below.

In Scanning attack, an attacker tries to gather information such as network topology, types of network traffic allowed by the firewalls, versions of operating system and kernel hosted on a network and identities and versions of server software running. The whole process may be initiated and accomplished by sending a stealth SYN packet. It is stealthy because the attacker just tries to create a half open TCP connection. When the server receives a SYN packet, server responds with a SYN/ACK and goes to a listening state, allowing the attacker to gather the required information. If no SYN/ACK is received by the attacker, it is assumed that the server is in a closed state.

In a penetration attack, an attacker tries to access a system and its resources without authorization. The attacker seeks to acquire the privileges of root to execute code easily and exploit system resources. After compromising the system, the attacker
can use the machine as a launchpad for different types of attacks.

A Denial of Service (DoS) attack tries to exhaust the resources of a network or a system. An attack can be carried out by a few malformed packets that exploit vulnerabilities in the host or by a vast number of legitimate packets that exhaust the victim’s network bandwidth or resources (Bhattacharya et al., 2013). As a precursor, a distributed DoS attacker may access many machines to lunch a coordinated distributed DoS attack. A DoS attack causes frequent congestion, hindering legitimate communication.

With the rapid emergence of external and internal threats to networks and resources, we must think about security all the time. As a result, researchers and practitioners have looked at a variety of approaches such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Intrusion Response System (IRS) and Intrusion Tolerance System (ITS). IPS and IDS are important components of a layered security infrastructure.

Four main steps (Bhattacharya et al., 2013) taken by an attacker prior to executing an intrusion into a network or system are as follows.

(a) Prepare: In this first step, the attacker attempts to collect network configuration information using port scanners to identify vulnerabilities in the network (Bhuyan et al., 2011). Port scanning gathers information such as computer IP addresses, operating systems, open ports with identities and version of listening software.

(b) Exploit: Once vulnerabilities are identified, in the second step, the attacker attempts to exploit these vulnerabilities. The attacker may execute multiple attempts during this step.

(c) Leave Behind: If the lunching of an attack is successful, the attacker installs additional software to create continued access to the network. This process, termed as leave behind, includes installation of network sniffer or additional back-door network services.

(d) Clean Up: At last, the attacker tries to clean up any evidence left due to the actions in the previous steps. This step may include restarting daemons crashed during the second step, clearing logs and other information, and installing modified system software designed to override the presence of other software from normal system commands.

2.1. Based on Approach Used

Based on the approach used to counter intrusions the four main intrusion defense systems such as: intrusion detection systems, intrusion prevention systems, intrusion response systems and intrusion tolerance systems are found to work as follows.

2.1.1. Intrusion Detection System

An intrusion detection system (IDS) (Ertoz et al., 2004) monitors a network or system for malicious activities or policy violations. Some systems or approaches may try to stop an intrusion attempt, but this is neither required nor expected of a monitoring system. If an IDS detects any threat, it alerts the system or network administrator. The objective of an IDS is to detect and inform active defenders about intrusions. An IDS also uses techniques that can detect abnormalities both at the network and host levels. Figure 1 shows a generic view of an IDS. The components are: a managing system, a monitoring component and a detection component.

- The managing system oversees traffic flow in the network. It provides traffic information to the monitoring component for analysis.
- The monitoring component monitors traffic and analyzes the behavior of the network.
- The detection component detects any suspicious behavior with respect to the normal working nature of the network. If any abnormal behavior is detected, it is communicated to the reaction component.
- The reaction component reacts to the situation. After detection of abnormality, it raises an alarm so that the intrusion can be handled appropriately.

2.1.2. Intrusion Prevention System

An IPS is considered an upgraded version of an intrusion detection system (Desai, 2009). They both monitor network traffic and/or system activities for malicious activity, but the main difference is that intrusion prevention systems are able to actively prevent intrusions that are detected. An IPS executes
steps such as sending an alarm, dropping malicious packets, resetting the connection and/or blocking traffic from the offending IP addresses. Figure 2 presents a generic view of an intrusion prevention system. The managing system, monitoring component and detection component are similar to those in an IDS, but in the reaction component prevention procedures are applied by the prevention engine.

- In the **reaction component**, the prevention engine applies procedures according to the pattern of behavior of the suspicious traffic by working closely with the **Managing system**.
- The **managing system** manages the traffic flow and applies the procedures provided by prevention engine.
- The **monitoring system** and the **detection component** work similarly to those in an IDS.

### 2.1.3. Intrusion Response System

An intrusion response system (IRS) (Stakhanova et al., 1991) continuously monitors system health based on IDS alerts, so that malicious or unauthorized activities can be handled effectively by applying appropriate actions to prevent problems from worsening the situation and to return the system to a healthy mode. A notification system generates alerts when an attack is detected. An alert can contain information like attack description, time of attack, source IP and user accounts used to attack. An IRS automatically executes a preconfigured set of response actions based on the type of attack. An automated approach requires no human intervention, unlike an IDS where there is a delay between intrusion detection and response. Figure 3 shows the generic structure of an intrusion response system. It is comprised of a reaction component, a detection component, a monitoring component and a managing system. In particular,

- The **reaction component** has a response system, and
- It responds to the intrusion using a predefined approach in an automated manner.

![Figure 3: Intrusion Response System: A generic View](image)

### 2.1.4. Intrusion Tolerance System

An Intrusion tolerance system (Deswarte et al., 1991) takes a fault-tolerant design approach to defend information systems against malicious attacks. In lieu of the general aim of preventing all intrusions, intrusion tolerance uses mechanisms that prevent intrusions from leading to system security failure. As a matter of fact, intrusion tolerance is not a new concept. Classical fault tolerance techniques are useful for tolerating intrusion and error detection and recovery. Error hiding techniques can also be applied to provide data integrity or service availability despite intrusions. However, such fault-tolerance techniques are usually considered harmful for data confidentiality due to the redundancy that they imply. Figure 4 provides a generic view of an intrusion tolerance system. The managing system, the monitoring component and the detection component are similar to those in an IDS, but the reaction component uses tolerance techniques.

- In the **reaction component**, intrusion tolerance techniques try to prevent intrusions from causing system failure.
- Classical techniques may be useful and efficient.
- A tolerance approach differs from the conventional way of preventing the attacks.

![Figure 4: Intrusion Tolerance System: A Generic View](image)

### 2.2. Modules of A Defense System

In this section, we discuss the components of a generic defense system.

#### 2.2.1. Monitoring

Network monitoring collects data on the state of the network (Conorich, 2004). Traffic analysis requires inspection of services being used on a network or system and comparing them against activities that are expected. This allows one to identify suspicious services within a network. To perform basic network monitoring, one needs to collect traffic characteristics at various points within the network. Although it is necessary to look carefully at network borders, if there are internal hosts providing unauthorized services for other internal hosts, one will miss this traffic if one only looks at the borders. There are four different types of TCP activity that should be considered.

- Are three-way handshakes being completely executed or not?
- Are three-way handshakes being initiated but never successfully completed?
2.2.2. Detection

A detection module provides reports (Mukherjee et al., 1994) to a management section. Some detection modules may try to stop an attack but this is neither required nor expected. The intrusion detection module is primarily focused on identifying possible incidents, logging information about them and reporting intrusion attempts. A detection module can be used for various purposes such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. A detection module acquires and analyzes information from various areas within a computer or a network to identify possible security breaches, vulnerabilities, which include both intrusions and misuse. This module may use scanning, which is a technology developed to assess security vulnerabilities of a computer system or network.

2.2.3. Reaction

Typically, a defense system reacts using a two-step process. The first set of procedures constituting the passive component, involves inspection of the system’s configuration files to detect inadvisable settings, inspection of the password files to detect inadvisable passwords, and inspection of other system areas to detect policy violations. The second set of procedures constitute the active component. Here mechanisms are set in place to react to known methods of attack and to generate system responses. IDSs can respond to suspicious events in several ways, which include displaying an alert, logging the event or even paging an administrator. Alarm management can be categorized into two (Klüft and Staaf, 2012; Pietraszek and Tanner, 2005).

- Alert/alarm quality improvement: This approach tries to improve alert quality by using information such as vulnerability reports or alert contexts. One can prioritize alerts with respect to the vulnerabilities of the victims.

- Alarm correlation: This approach creates a more ambitious goal. It tries to reconstruct higher-level incidents from lower-level alerts. Sometimes, a defense system may generate more alarms than normal within a short period. If a set of alerts are triggered, and knowing this without any additional background knowledge, one cannot determine whether these are coordinated/distributed attacks or independent attacks that happen to be interleaved. If it is a single multistage attack, alarms would have to be generated in a single incident. In the case of multiple attacks, the alerts should be divided into multiple incidents, namely, one incident per attack. Grouping alerts that constitute a single attack into a single meta-alert is aggregation. The task of clustering alerts into incidents is called correlation (Julisch, 2003).

To prevent an attack before damaging the network system, it is need to adopt preventive measure like, creating a database of detected signatures of abnormalities to filter out threatening packets, analyze pattern of network behavior, reconfigure other security controls etc. Trace-backing (Xiang and Zhou, 2004) the source of attack is a good way to prevent the attack in future. But, it also important to look for low collateral damage while trace backing a huge botnet attack. So, reaction procedure in terms of prevention is suitable aspect. Passive systems can attempt to terminate the connection before an attack can succeed, for example, by ending an existing TCP session.

Researcher have demonstrated that no system can assure to detect and prevent from any kind of anomalous activities in a exposed and live network in a generic way. Thus, the solution to react, lead in two different track. One option is that, respond the attack or intrusion in an efficient way to provide the usual service to general user and stop the service to the non-legitimate user. Second option is to provide fault tolerance approach towards attack.

2.3. Based on Nature of Control

In this section, we discuss types of defense systems based on the control structure used to counter attack traffic. There are three basic ways used to control detection and prevention processes, viz., centralized, hierarchical and distributed (Patel et al., 2013).

2.3.1. Centralized

In this type of defense, each detection element produces alerts locally. The generated alerts are sent to a central server that plays the role of a correlation handler and analyzes them. Using centralized control, an accurate detection decision can be made based on all available alert information. The main drawback of this approach is that the central unit is crucially vulnerable; any failure in the central server leads to the collapse of the whole process of correlation. In addition, the central unit should be able to handle the high volume of data which it may receive from the local detection elements in a short amount of time.

2.3.2. Hierarchical

The whole system is divided into several small groups based on features such as geography, administrative control, and software platforms. The IDPSs at the lowest level work as detection elements while an IDPS at a higher level is furnished with both a detection element and a correlation handler, and it correlates alerts from both its own level and lower levels. The correlated alerts are then passed to a higher level for further analysis. This approach is more scalable than the centralized approach, but still suffers from the vulnerability of the central unit. Besides, the higher level nodes have a higher level abstraction of the input, which may limit their detection coverage.
Table 2: Comparison of Control Mechanisms for Defense Against Intrusion

<table>
<thead>
<tr>
<th>Control Mechanism</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Central           | - Each IDPS acts as a detection element.  
                   - Every detection element produces alerts locally.  
                   - Accurate detection decisions can be made based on all available alert information. | - The central unit is crucially vulnerable.  
                   - Any failure in the central server leads to deactivation of the whole process of correlation.  
                   - The central unit handles high volume of data received from local detection elements. |
| Distributed       | - Need not have complete information of network topology.  
                   - Possible to have a more scalable design since there is no central entity.  
                   - Local alarm correlation is simpler in this structure. | - Information about all alerts is not available during decision making about detection.  
                   - Alert information may be too narrow to detect large scale attacks. |
| Hierarchical      | - This approach is more scalable than the centralized approach.  
                   - Higher level nodes have higher level abstractions of the input. | - Suffers from the vulnerability of a central unit.  
                   - At each level, the detection coverage may be limited. |

2.3.3. Distributed

In this approach, there is no centralized coordinator to process the information, and is comprised of fully autonomous systems with distributed management control. All participating IDPSs have their own components communicating with each other. The advantages of the fully distributed IDPS (Leitner et al., 2007) are that although the network entities do not have complete information about the network topology, it is possible to have a scalable design since there is no central entity responsible for doing all the correlation work, and local alarm correlation is simpler in this structure. Meanwhile, the fully distributed approach has its own drawbacks (Zhou et al., 2010): (a) Information on all alerts is not available during decision making, so the accuracy may be reduced; (b) The alert information usually has a single feature (like an IP address), which may be too narrow to detect large scale attacks.

Figure 5: Central Management Structure

Figure 6: Hierarchical Management Structure

Figure 7: Distributed Management Structure

Looking at the different approaches to control we observe the following:

- Each of the three ways of control has advantages and disadvantages. We summarize them in Table 2.
2.4. Based on Defense Infrastructure

In this section, we discuss various defense systems based on the infrastructure used. These are two basic types, viz., host-based and network-based.

2.4.1. Host-based

In this architecture, data is analyzed by individual computers that serve as hosts. The network architecture used is agent-based, which means that a software agent resides on each of the hosts in the system. Thus, a host-based Intrusion Detection and Prevention System (HIDPS) processes data that originates on the computers themselves, such as event and kernel logs. An HIDPS (Yeung and Ding, 2003) can also monitor which program accesses which resources and may flag anomalous usage. An HIDPS also monitors the state of the system and makes sure that everything makes sense, which is necessary for the use of anomaly filters.

2.4.2. Network-based

A network-based detection system examines data exchanged among computers in the network. A Network-based Intrusion Detection and Prevention System (NIDPS) (Vigna et al., 2004) captures network traffic from the wire as it travels to a host. This can be analyzed for a particular signature or for unusual or abnormal behaviors. Several sensors are used to sniff packets on the network; these are computer systems designed to monitor network traffic. If any suspicious or anomalous behavior occurs, it triggers an alarm and passes the message to the central computer system or an administrator, and generates an automatic response.

2.5. Defense Location

An intrusion defense system can be deployed in three possible locations: victim-end, intermediate and source-end. Each has its advantages and disadvantages.

2.5.1. Victim-end defense mechanism

Victim-end detection (Douligeris and Mitrokotsa, 2004) approaches are conventionally employed in the routers of the victim side network. The detection software stores information about known intrusion signatures or profiles of normal behavior. This information is updated by the processing elements as new knowledge becomes available. The processing element in a detection engine frequently stores intermediate results in what is called configuration data. Detecting attacks at the victim end is relatively easy, but requires higher resource consumption. An important drawback is that these approaches detect the attack only after it reaches the victim and thus legitimate clients have already been affected.

2.5.2. Intermediate network defense mechanism

The intermediate network defense scheme (Wang et al., 2001) balances detection accuracy and attack bandwidth consumption, which are the main issues in source-end and victim-end detection approaches. The main difficulty with this approach is deployability. To achieve full detection, all routers on the Internet will have to use this detection scheme, because non-availability of this scheme on only a few routers may cause detection failure. Thus, full practical implementation of this scheme is extremely difficult.

2.5.3. Source-end defense mechanism

This type is somewhat similar to victim-end detection. It is the best option if we want to reliably detect or stop intrusion (Mirkovic and Reiher, 2005). It prevents congestion not only on the victim side, but also in the whole intermediate network. The main difficulty is that during attacks, sources are widely
Table 4: Comparison of Defense Solutions Based on Locations

<table>
<thead>
<tr>
<th>Defense Location</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim-end</td>
<td>- Stored known intrusion signatures are applied for better performance.&lt;br/&gt;- To detect attack at this end is easy because a high amount of resources is available.</td>
<td>- Till the time an attack is detected and stopped, legitimate clients are affected.</td>
</tr>
<tr>
<td>Intermediate</td>
<td>- At this location, we can balance the resources used for detection and bandwidth consumed by an attack&lt;br/&gt;- Suffering of legitimate clients is lower than in the case of victim-end.</td>
<td>- The main problem is deployability.&lt;br/&gt;- All routers in the network must employ the detection scheme.&lt;br/&gt;- Failure at a few routers can cause damage to whole detection process.</td>
</tr>
<tr>
<td>Source-end</td>
<td>- It is the best option that avoids congestion and collateral damage to the whole network.&lt;br/&gt;- Legitimate clients suffer less.</td>
<td>- The resources needed to deploy are lower than at any other network locations.&lt;br/&gt;- It is difficult to distinguish between normal and abnormal traffic because they behave almost similarly.</td>
</tr>
</tbody>
</table>

2.6. Based on Technique Used

Many techniques have been developed to prevent intrusion. We cover a variety of them below under various categories and subcategories.

2.6.1. Misuse Detection

In misuse detection (Kumar and Spafford, 1994), one defines abnormal system behavior first, and then define any other behavior as normal. This is in contrast to anomaly detection, which uses the reverse approach, defining normal system behavior first and defining any other behavior as abnormal. In other words, anything one doesn’t know as bad is normal in misuse detection. Using attack signatures in IDSs is an example of this approach. Such misuse detection IDSs attempt to detect only known attacks based on predefined attack characteristics. The accuracy of such IDSs solely depends on how well the knowledge of known attacks has been preprocessed and fed to the IDS’s detection engine. Well-crafted expert knowledge of known attacks can enable misuse detection IDSs to perform accurately with a low number of false positives.

A. Signature-based approach: A signature detection or misuse detection scheme (Kruegel and Toth, 2003) stores sequences of patterns and signatures of attacks in a database. When an attacker tries to attack, the IDS matches the signature of the particular attack type with predefined signatures that are already stored in the database. If there is successful match, the system generates alarm. In this approach, the semantic characteristics of an attack are analyzed and details are used to form attack signatures. Attack signatures are constructed in such a way that they can be searched using information in audit data logs produced by computer systems. A database of attack signatures is built using well-described known attacks and the detection engine of the IDS compares string log data or audit data against the database to detect an attack. Each time a new attack is discovered, the attack signature database must be quickly updated to obtain up-to-date performance. Many different signature matching algorithms are used in signature based attack detection systems.

B. Rule-based approach: (Porras and Kemmerer, 1992) Rule based systems are built using a number of if-then rules. Rules are developed by analyzing attacks or misuses by experts. These are later used by inference modules of IDSs to compare against monitored data to detect any misuse.

C. State Transition Approach: (Ilgun et al., 1995) In this approach, attacks or misuses are represented as a series of activities and a single activity or a combination of activities can cause transition from one state of a monitored sensor to another state, eventually reaching the security alert state of a monitored system.

2.6.2. Anomaly Detection Technique

Theses techniques first establish the normal behavior of a subject, e.g., a user or a system. Any action that significantly deviates from normal behavior is considered intrusive. If we can establish normal activity profile for a system, we can flag all
The mean and standard deviation model gives higher weights to vals depends on observed data, as it may vary from user to user.

Order to set limits. Second, determining the confidence interval, above or below the moment, is said to be anomalous. Variations are known as moments. An event that falls outside a set interval. A change of state of the system occurs when an event happens and the behavior is detected as anomalous if the probability of occurrence of that event is low. Transitions between certain commands may lead to anomaly detection, especially where a sequence of commands is necessary to perform a task (Ye, 2000).

A.2. Operational: The count of certain events that occur over a period of time may determine the alarm to be raised. For example, alarm may be raised if an event occurs fewer than m or more than n times. This can be visualized in Win2k lock, where a user is blocked after n unsuccessful login attempts. Here the lower limit is ‘0’ and the upper limit is n. The size of executable files that can be download is restricted in some organizations to about 4 MB. The difficulty in this sub-model is determining appropriate values for m and n.

A.3 Statistical moments: In statistics, mean and standard deviation are known as moments. An event that falls outside a set interval, above or below the moment, is said to be anomalous. The system is updated considering data over time by making changes to the statistical database. There are two major advantages over an operational model (Jyothsna et al., 2011). First, prior knowledge is not required to determine normal activity in order to set limits. Second, determining the confidence intervals depends on observed data, as it may vary from user to user. A threshold model lacks this flexibility. A major variation on the mean and standard deviation model gives higher weights to recent activities.

A.4 Time series: A time series model needs interval timers (Vinnikka et al., 2009) together with event counters or resource measures. The system observes and stores values of order and inter-arrival times. If the probability of occurrence of a new observation is too low, it is considered an anomaly.

B. Data mining or Machine learning: In this approach, historical usage data obtained from a monitored system is normally categorized as acceptable or unacceptable and labeled accordingly. By using a learning algorithm, the system is trained to learn either what is acceptable or what is unacceptable considering usage in a network or system. If any deviation occurs, the system triggers an intrusion alert. Many machine learning (Lee and Stolfo, 1998; Tsai et al., 2009) techniques can be used for data classification. We present the general concepts and a brief description of some popular supervised learning techniques below.

B.1. Decision Trees: Decision Trees represent a well-known and efficient classification algorithm (Peddabachigari et al., 2004). A decision tree consists of non-terminal nodes (a root and internal nodes) and terminal nodes (leaves). The root node is the first attribute with test conditions to move an input data record down the tree toward a leaf node depending on characteristics of the data record. Testing and splitting are repeated at every internal node. First, a decision tree is trained with known labeled data before it can classify new or untrained data. The training process builds the decision tree by identifying attributes and values that would be used to test the input data at each internal node. After training, the tree can classify new data by starting from a root node and traversing down internal nodes based test conditions until it arrives at a leaf (terminal) node consisting of an answer class.

B.2. Artificial Neural Networks: In this method, a dataset of input vectors and corresponding target vectors are used in training the network to associate input with output (Mukkamala and Sung, 2003; Hagan et al., 1996). A neural network can be created from audit data that specify a string of events as abnormal or normal. Ghosh and Schwartzbard (1999) used Artificial Neural Networks (ANNs) to detect both known and novel attacks on a computer system using supervised learning. They used Sun Microsystems’s Solaris Basic Security Module (BSM) generated host audit data to train a multi-layered back propagation feed forward neural network to learn normal system behavior. In order to verify complete session activities, which

<table>
<thead>
<tr>
<th>Table 5: Comparison of Different Misuse Detection Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signature-Based Approach</strong></td>
</tr>
<tr>
<td>- During intrusion, it matches the pattern of attack with stored patterns.</td>
</tr>
<tr>
<td>- When a new attack is discovered, database must be quickly updated.</td>
</tr>
</tbody>
</table>

System states that vary significantly from an established profile. Thus, anomaly based techniques try to detect the compliment of bad behavior. Two types of errors in detection may occur: (1) False positives: Anomalous activities that are not intrusive but are flagged intrusive. (2) False Negatives: Anomalous activities that are intrusive but are flagged non-intrusive. The main advantage of anomaly detection is that it can detect unknown attacks.

A. Statistical: Statistical methods have long been used for anomaly detection (Stibor et al., 2005). In this approach, normal user behavior is first defined based on what is acceptable within system usage policies. Using statistical modeling techniques, user behavior is monitored and if there is a large deviation from predefined normal behavior thresholds, such anomalous activity is considered attack.

A.1. Markov process: Intrusion detection in this model is performed by investigating the system at fixed intervals and keeping track of probability for each state at a given time interval. A change of state of the system occurs when an event happens and the behavior is detected as anomalous if the probability of occurrence of that event is low. Transitions between certain commands may lead to anomaly detection, especially where a sequence of commands is necessary to perform a task (Ye, 2000).

A.2. Operational: The count of certain events that occur over a period of time may determine the alarm to be raised. For example, alarm may be raised if an event occurs fewer than m or more than n times. This can be visualized in Win2k lock, where a user is blocked after n unsuccessful login attempts. Here the lower limit is ‘0’ and the upper limit is n. The size of executable files that can be download is restricted in some organizations to about 4 MB. The difficulty in this sub-model is determining appropriate values for m and n.

A.3 Statistical moments: In statistics, mean and standard deviation are known as moments. An event that falls outside a set interval, above or below the moment, is said to be anomalous. The system is updated considering data over time by making changes to the statistical database. There are two major advantages over an operational model (Jyothsna et al., 2011). First, prior knowledge is not required to determine normal activity in order to set limits. Second, determining the confidence intervals depends on observed data, as it may vary from user to user. A threshold model lacks this flexibility. A major variation on the mean and standard deviation model gives higher weights to
Table 6: Summary of Different Anomaly-based Defense Techniques

<table>
<thead>
<tr>
<th>Anomaly Detection Technique</th>
<th>Description</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical</td>
<td>- Any activity is considered attack if there is any deviation from predefined normal behavior</td>
<td>- Markov process</td>
<td>- Keeps track of the different states the system is in at a specific time interval. The occurrence of a state is low if it corresponds to abnormality.</td>
</tr>
<tr>
<td></td>
<td>threshold.</td>
<td>- Operational</td>
<td>- An alarm detecting abnormality is raised if count of events that occur is lower than a threshold value or greater than a threshold value.</td>
</tr>
<tr>
<td></td>
<td>- A normal user behavior is defined using statistical modeling techniques by computing mean and standard deviation</td>
<td>- Statistical Moments</td>
<td>- An event that falls outside the set interval, above or below the moment can be treated as anomalous. Standard deviation and other summaries are known as moments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Time Series</td>
<td>- If the probability of occurrence of a new observation is too low, it is considered an anomaly.</td>
</tr>
<tr>
<td>Data Mining or Machine Learning</td>
<td>- Data mining or machine learning algorithms are good at finding similarities or patterns in data. Knowledge gained by training on data is used to classify new data.</td>
<td>- Decision Trees</td>
<td>- A decision tree is trained with known data before it can classify new or untrained data. After training, it can classify new data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Artificial Neural Networks</td>
<td>- A dataset of input vectors and corresponding target vectors are used in training the network to associate input with output.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bayesian Networks</td>
<td>- A neural network is created from audit data to determine a string of events as abnormal or normal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Fuzzy Logic</td>
<td>- In this graphical technique, nodes in the graph represent random variables. The Bayesian network learns causal or dependency relations among attributes in the training data set before classifying new data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Outlier Detection</td>
<td>- Such a system learns the characteristics of network traffic by applying fuzzy logic. Signatures are developed by analyzing network protocols.</td>
</tr>
<tr>
<td>Cognition-based or Expert System</td>
<td>- Detection of intrusion is influenced by a set of predefined parameterized rules that classify training data.</td>
<td>- Finite State Machines</td>
<td>- If a data point is very different from the rest of the data, it is an outlier. Abnormal traffic that is very different from acceptable traffic based on metrics, is identified as outlier.</td>
</tr>
</tbody>
</table>
consist of multiple BSM events, they applied the leaky bucket algorithm (Lee and Un, 1993) to capture recent activities from neural network generated outputs.

B.3. Bayesian Networks: A Bayesian Network (Burroughs et al., 2002) involves a probabilistic model representing random variables and conditional independence using a directed acyclic graph. Nodes in the graph represent random variables whereas edges represent conditional dependencies. Thus nodes that are not connected represent variables that are conditionally independent of each other. The Bayesian Network learns causal relations among attributes and class labels from the training dataset before it can classify unknown data.

B.4. Fuzzy logic: Such a system learns the characteristics and behavior of network traffic by applying fuzzy logic (Klir and Yuan, 1995). Signatures are developed by learning or analyzing network protocols. Abouzakkar and Manson (2003) proposed a neuro-fuzzy technique for detecting distributed network attacks such as denial of service (DoS). The proposed system learned characteristics of network traffic by applying fuzzy logic. Chavan et al. (2004) proposed a neuro-fuzzy adaptive IDS for IP networks where a database of pattern signatures was built to complement the SNORT signature database (Snort, 1999). These signatures were developed by analyzing network protocols and adaptive learning based on combination of artificial neural networks and fuzzy inference techniques.

B.5. Outlier Detection: If a data point is very different from the rest of the data, it is an outlier. According to statistical distribution of data points based on a given mean and a standard deviation, data points that do not fall under a specific range are considered outlier. In network intrusion detection, anomalous network traffic data or abnormal network behavior is different from acceptable traffic data or normal network behavior based on appropriate metrics and thus may be identified as outlier. There are various methodologies used to detect outliers (Gogoi et al., 2011).

C. Cognition-based Expert System: Cognition-Based (also called knowledge-based or expert systems) (Jyothsna et al., 2011) detection techniques classify the audit data with the help of a set of predefined rules. These rules may be created from training data or may be built by expert or a combination of the two.

C.1. Finite state machines: A Finite State Machine (FSM) (Hershey et al., 1995) or finite automation is a model of behavior captured in states, transitions and actions. A state denotes a situation. Any changes in the input may cause transitions. An action is a description of an activity to be performed. There are several action types: entry action, exit action and transition action.

2.6.3. Applications and protocols

An application-protocol-based intrusion detection system (APIDS) (Boukerche and Notare, 2000; Dreger et al., 2006) monitors dynamic behavior and states of the protocol and typically consists of a system or agent that sits between a process, or a group of servers, monitoring and analyzing the application protocol between any two connected devices. A typical place for an APIDS is between a web server and the database management system, monitoring the SQL protocol specific to the middleware/business logic as it interacts with the database. A protocol-based intrusion detection system (PIDS) is typically installed on a web server, and is used in monitoring and analysis of the protocol in use by the computing system. A PIDS monitors the dynamic behavior and state of the protocol and typically consists of a system or agent that sits at the front end of a server, monitoring and analyzing the communication between a connected device and the system it is protecting. A typical use for a PIDS is at the front end of a web server monitoring the HTTP (or HTTPS) stream. Because it understands the HTTP data stream relative to the web server/system it is trying to protect, it can offer greater protection than less in-depth techniques such as filtering by IP address or port number alone.

3. Defense Systems

In this section we discuss several defense systems and analyze their pros and cons.

(a) STAT: State Transition Analysis Technique (STAT) (Porras, 1992) is a suite of tools for misuse detection. It uses the state transition mechanism to identify intrusive activities in computer systems. This suite includes the STATL (Eckmann et al., 2002) language to define attack scenarios with the help of domain independent attack attributes in a high level language at an abstract level. These definitions must be included by security system developers to meet the needs of a specific environment. The basic concept behind detecting an attack or misuse is that before penetration by an attacker, a computer system is in an initial secure state. A series of activities by an attacker causes a system to transition to various intermediate states before reaching a final state where the system has been successfully penetrated by an intruder.

(b) USTAT: UNIX State Transition Analysis Tool (USTAT) (Ilgun, 1992) is the first STAT based tool to analyze audit logs generated by UNIX-based systems for misuse identification. It can act in real time.

(c) ARMD: ARMD (Adaptable Real-time Misuse Detection System) is an abstraction based misuse detection system, designed by Lin et al. (1998). In a UNIX environment, this system uses its own high level language called MuSigs for misuse signature abstraction from audit log. Using MuSigs, misuses are described in easily understandable abstract forms or signatures. ARMD uses a monitoring algorithm that searches a given event history looking for matching signatures and if any matching signatures are found, they are reported. Using ARMD as a part of the monitoring system, real time misuses could be detected. Other parts of the system are monitored off-line.

(d) NIDES: The continuation of research work on the IDES system led to the Next Generation Intrusion Detection System (NIDES) (Anderson, 1995). NIDES is built using client-server architecture. Log data from various hosts on a network are gathered at a specific host. On that specific host, rule based anomaly detection is performed. The P-BEST expert system (Lindqvist and Porras, 1999) is used for misuse detection.
(e) NSTAT: USTAT was originally capable of analyzing audit log of a single UNIX host. Its extension to analyze audit logs of multiple UNIX systems changed it into NSTAT (Kemmerer, 1997). It gathers audit records of a set of distributed hosts. All these records are processed in a central system for misuse detection.

(f) EMERALD: The EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) system (Porras and Neumann, 1997) also uses the P-BEST expert system. EMERALD is an environment for misuse and attack detection in a large scale network, employing both anomaly and misuse detection techniques.

(g) NetSTAT: NetSTAT (Vigna and Kemmerer, 1998), a real-time network misuse detection system, is also based upon the STAT framework. NetSTAT represents the network topology as a hypergraph model and uses STAT definitions of network based attacks to map specific misuses to specific network configurations. It preprocesses network traffic log given as input and filters network packets for relevant network events and generates abstract events.

(h) Bro: Bro is an open source UNIX based network intrusion detection system (Paxon, 1999). It uses a signature based approach and detects known attacks and events using predefined attack signatures. Bro tries to detect uncommon activities such as failed connection attempts.

(i) Snort: Snort is a popular signature based lightweight network IDS (Snort, 1999). It uses a database consisting of user defined attack signature rules and uses the Boyer-Moore pattern matching algorithm (Boyer and Moore, 1977; Knuth et al., 1977) against the database for each network traffic packet. Snort has three modes. The first one is the packet sniffing mode, which enables it to monitor and display traffic packets. The second one is the network traffic logger mode where Snort writes network traffic log into a database. The last one is the IDS mode in which it has intrusion detection and prevention capabilities, both in real time.

(j) MADAM ID: Lee et al. (1999) apply various data mining techniques such as association rules and classification techniques to develop an automated misuse detection model using data from audit logs and system calls. They develop an intrusion detection framework called MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection).

(k) Ghosh and Schwartzbaxd (1999): The authors use Artificial Neural Networks (ANNs). Both known and unknown attacks against a computer system are detected using supervised learning. It uses audit data to train a multi-layered back propagation feed forward neural network to learn normal system behavior.

(l) Chavan et al. (2004): They propose a Neuro-Fuzzy adaptive IDS for IP networks where a database composed of patterns of signatures is built to complement the SNORT signature database. These signatures are developed by analyzing network protocols and using adaptive learning based on combination of artificial neural networks and fuzzy inference techniques.
<table>
<thead>
<tr>
<th>Year</th>
<th>System</th>
<th>References</th>
<th>Approach</th>
<th>Technique</th>
<th>Infrastructure</th>
<th>Real time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>ARMD</td>
<td>Lin et al. (1998)</td>
<td>Detection</td>
<td>Abstraction based misuse detection</td>
<td>Host-based</td>
<td>Yes</td>
</tr>
<tr>
<td>1999</td>
<td>Snort</td>
<td>Snort (1999)</td>
<td>Detection</td>
<td>Signature based</td>
<td>Network-Based</td>
<td>Yes</td>
</tr>
<tr>
<td>1999</td>
<td>MADAM ID</td>
<td>Lee et al. (1999)</td>
<td>Detection</td>
<td>Data mining</td>
<td>Host-based</td>
<td>No</td>
</tr>
<tr>
<td>1999</td>
<td>Ghosh and Schwartzbaxd</td>
<td>Ghosh and Schwartzbaxd (1999)</td>
<td>Detection</td>
<td>Artificial Neural Networks based</td>
<td>Host-based</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>PHAD</td>
<td>Mahoney and Chan (2001)</td>
<td>Detection</td>
<td>Data mining techniques</td>
<td>Network-based</td>
<td>No</td>
</tr>
<tr>
<td>2003</td>
<td>WHIPS</td>
<td>Battistoni et al. (2004)</td>
<td>Prevention</td>
<td>Monitoring critical Windows system calls</td>
<td>Host-based</td>
<td>Yes</td>
</tr>
<tr>
<td>2005</td>
<td>FLIPS</td>
<td>Locasto et al. (2005)</td>
<td>Prevention</td>
<td>Signature based, anomaly detection</td>
<td>Host-based</td>
<td>Yes</td>
</tr>
<tr>
<td>2009</td>
<td>Luo et al.</td>
<td>Luo et al. (2009)</td>
<td>Detection</td>
<td>Game-theory based, Multi-stage attack defense</td>
<td>Host-based</td>
<td>Yes</td>
</tr>
<tr>
<td>2010</td>
<td>Huang et al.</td>
<td>Huang et al. (2010)</td>
<td>Detection</td>
<td>Distributed multi agent based</td>
<td>Host-based</td>
<td>Yes</td>
</tr>
<tr>
<td>2011</td>
<td>Computer worm defense system</td>
<td>Aziz (2011)</td>
<td>Detection, Prevention</td>
<td>Sensor based</td>
<td>Network-based</td>
<td>Yes</td>
</tr>
</tbody>
</table>
(m) NFIDS: Mohajerani et al. (2000) use fuzzy logic and neural networks to build the Neuro-Fuzzy Intrusion Detection System (NFIDS). They use neural networks to learn fuzzy rules for each type of attack listed by the system administrator offline. After learning the fuzzy rules, the neural network performs a fuzzy inference process to detect intrusions.

(n) Abouzakhar and Manson (2003): They also propose a neuro-fuzzy technique for detecting distributed network attacks such as denial of service (DoS) attacks. The proposed system learns the characteristics of network traffic by applying fuzzy logic.

(o) PHAD: The Packet Header Anomaly Detector (PHAD) system detects anomalous field values using a data mining technique from the information contained in the data link, and network and transport layers protocol headers (Mahoney and Chan, 2001). The algorithm is trained with normal network traffic data so that it can learn the normal range of allowable values for each field. The algorithm calculates the number of previously unseen values and frequencies of each value for each field and assigns an estimated probability to a given field value being abnormal. An abnormality field score is calculated using the time since the last abnormality was observed in that field. Finally a packet score is calculated by summing up all the abnormality field scores in the packet.

(p) Qiao et al. (2002): The authors use a Hidden Markov Model (HMM) built using system calls to detect intrusions. To determine various state transitions that a special UNIX based process goes through from the start to the end, they collect all the system calls specific to that process and train an HMM on system calls associated with such processes.

(q) MINDS: The Minnesota Intrusion Detection System (MINDS), described in (Ertoz et al., 2004), successfully applies outlier detection techniques on network traffic data to detect attacks on a computer network. MINDS uses a suite of data mining algorithms in its various detection modules. Each module detects a specific type of computer attacks and intrusive activities in a networked environment.

(r) Kruegel and Toth (2003): They apply the decision tree approach to match attack signatures instead of traditional signature matching techniques such as in Snort and achieved improved detection speed.

(s) WHIPS: Battistoni et al. (2004) propose WHIPS, a host-based IPS for Windows operating systems. The system solely monitors critical Windows system calls in the kernel mode. The proposed system identifies privileged processes and identifies harmful processes by examining access tokens.

(t) Yao et al. (2005): The authors develop a hybrid intrusion detection system using both fuzzy logic (Klir and Yuan, 1995) and Support Vector Machines (SVM) (Mukkamala and Sung, 2003). They apply SVMs on network traffic data multiple times by changing values of parameters and obtain sets of support vectors during the training phase of the SVMs. Then they apply fuzzy logic to develop fuzzy rules to make decision from various sets of support vectors obtained from SVM training.

(u) FLIPS: Locasto et al. (2005) introduce a hybrid adaptive intrusion prevention system called FLIPS (Feedback Learning IPS). Host-based FLIPS uses both signature matching and anomaly based classification. Its goal is to detect and prevent code injection attacks. It uses an intermediate emulator to detect injected malicious attack code and does not generate attack signatures.

(v) Weinsberg et al. (2006): They propose a high performance string matching algorithm and use it to develop. It works as a network-based IPS. The hardware-based algorithm has the ability to match multiple patterns at a time, making it faster.

(w) NFR: Network Flight Recorder (NFR) (Ranum, 2008) is a commercially used powerful network intrusion detection and analysis tool. It uses signatures of known attacks to raise alarm if any attack is detected. NFR uses a scripting language called n-code for generating signatures and for network packet analysis.

(x) Zhang et al. (2004): They describe a network-based IPS which is distributed in action. The rule-based IPS uses a network management module with a number intrusion detection modules. Multiple IDSs are placed in various locations on the network. The IPS was created using application-specific integrated circuits (ASICs) and therefore, it had fast processing ability.

(y) Luo et al. (2009): This system presents a game theory method to monitor and analyze the risk and impact of multi-stage attacks in intrusion detection system. They developed an algorithm to defend attack, termed as multi-stage attacker defender (MAD) and to provide help to administrator in defending. The believes of the attacker and the administrator are updated based on the analysis of the life-cycle for the multi-stage attacks to reduce the horizon effect.

(z) Huang et al. (2010): It is a distributed defense system which detects intrusion and perform as supplement of firewall for more effective protection means. It used a Multi-Agent-Based Distributed Intrusion Detection System. It assures good detection accuracy and detection speed and enhance the system security.

(aa) Computer worm defense system: This system consists of multiple containment systems bind together by a management section (Aziz, 2011). Each containment system can be deployed on a separate communication network and contains a worm sensor and a blocking system. Worm sensor generates computer worm identifiers from every containment system and it can provide distributed blocking system.

4. Defense Issues and Challenges

Flaws in the architecture of a network system allows intrusions to take place. When intrusions happen, regular benign traffic to the victim’s network suffers and the victim loses access to other networks. The victim can try to prevent these attacks or intrusions by installing firewalls or intrusion detection systems. However, developments on the Internet take place rapidly and the quality and intensity of attacks also keep on improving and escalating, respectively. As a result, security at the victim-end often begins to deteriorate. This causes legitimate connections
to suffer during attacks. Inter-dependencies on the Internet are very high and as a result, the probability of getting attacked by an attacker depends on the rest of the global Internet.

- Effective prevention mechanisms should be able to perform well in real time. Good intrusion detection and prevention systems must perform well by two metrics: accuracy and timely performance. Performance in time has received relatively lower attention. If not addressed properly, lack of efficient performance can be the roadblock to large-scale adoption of real-time intrusion prevention solutions. Specifically, the overhead associated with monitoring (e.g., data collection), analysis (e.g., signature-matching), and response, in terms of their impact on foreground tasks is not well understood.

- A successful prevention mechanism must be dynamic. Dynamism is a strategy to provide system components with security awareness and adaptability to address runtime policy changes. In addition, if a system can be constructed to be protocol independent, it is even more desirable.

- There are no common characteristics among traffic streams comprising various attacks that can facilitate early detection and filtering. Additionally, if the attack is distributed, participating machines usually observe no higher outgoing load than usual.

- When faced with an intrusion, if all resources of a system must be engaged to rectify the situation, legitimate users suffer unnecessarily. So, it is necessary not only to detect and restrain attacks, but also provide good service to legitimate traffic between the network where the solution mechanism is deployed and the victim. The solution should have low false positives and have a low deployment cost. This requires that source-end response must be selective. Source-end response must also be flexible to compensate for poor detection and to offer deployment incentive.

- It is ideally necessary to have a distributed and coordinated response system. It is also crucial that the response be mounted at many points on the Internet to cover the network. Since the Internet is administered in a distributed manner, wide deployment of any defense system (or even various systems that cooperate) cannot be enforced or guaranteed. This discourages many researchers and practitioner from even designing distributed solutions.

- Internet resources are limited. Most intelligence and resources needed to service clients but fight intrusion are located at the end hosts. However, the high bandwidth pathways needed for large throughput reside in the intermediate network. As a result, resources present in unwitting parts of the network are widely exploited by attackers to launch successful attacks.

- Many vendors make bold claims that their solution handles the intrusion problem completely. However, there is currently no standardized approach for testing intrusion defense systems that would enable true comparison and characterization.

- The defense should have a fast response time and should be able to respond quickly to any change in the traffic pattern. It is necessary to execute the defense functions only at a time of attack and at the other times, it should act as a normal system and not expend much system resources.

5. Conclusion

Many defense techniques are available in the research community and many techniques have been proposed to provide better defense against intrusion. In this paper, we have covered a general overview of different approaches and also discussed current defense issues and challenges. We started this paper with a brief description of defense structures, control mechanisms and infrastructure layouts to mount a good defense. We also have provided discussions of some popular IDSs and IPSs.

References


Locasto, M., Wang, K., Keromytis, A., & Stolfo, S. FLIPS:


