An Overview of Intrusion Detection Systems

Sriram Sundar Rajan
Masters in Software Engineering
Malardalen University
ssr09002@student.mdh.se

Vijaya Krishna Cherukuri
Masters in Software Engineering
Malardalen University
vci09002@student.mdh.se

Abstract

Network security is one of the most important non-functional requirements in a system. Over the years, many software solutions have been developed to enhance network security and this paper provides an insight into one such solution which has become prominent in the last decade: Intrusion Detection System (IDS). In this paper, we have provided an overview of different types of intrusion Detection Systems, the advantages and disadvantages of the same. The need for IDS in a system environment and the generic blocks in IDS is also mentioned. Finally, the details of examples of intrusion detection systems proposed by other authors have been elaborated. The examples are as follows: (1) Misuse intrusion detection system that uses state transition analysis approach, (2) Anomaly based system that uses payload modelling and (3) Hybrid model that combines the best practices of Misuse and Anomaly based intrusion systems.

Keywords: Anomaly IDS, Payload Modelling, Misuse IDS, Manhattan Distance, Mahalanobis Distance, Penetration, False Positive, False Negative, State Transition Analysis

1. Introduction

An intrusion is defined, by Heady et al. [1], as any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource [2].

Intrusion could be in many forms such as:

- Unauthorized personnel trying to gain access to the resources in a system or network.
- Malicious programs that spoil the system resources, degrades the system performance and manipulates the system data.
- Authorized personnel trying to gain additional privileges or access to confidential information, thus compromising the system's security policy.

Internet has changed the face of communication and computation. It is a highway to a lot of useful information. As internet has been illustrating an exponential growth, concurrently, so have computer attacks and cyber crimes by exploiting vulnerabilities in the system. It is reported in [3] that, during the period of 1991-1996, information theft rose by 250% and 99% of all major companies reported one incident of major security breach and 10 billion dollars were lost in the US due to telecom and computer related frauds.

Even if any computer system is very secure, the abuse of the privileges by authorized users is still an issue of threat. Moreover, a transaction logging mechanism; generally referred to as audit collection; is the need of the hour. It helps to understand and assess the damage caused by an intrusion. The automation of this audit collection is referred to as Intrusion detection system.

Intrusion detection system (IDS) monitors all the system activities in order to detect the Intrusion. It searches for security violation incidents, recognizes unauthorized accesses, identifies information leakages and intervention of malicious programs.

IDS play a role as a reactive agent rather than a proactive agent [3] in the security landscape of the system, whose primary job is to inform the system administrator in the event of an intrusion.

A deeper understanding of computer attacks is required to identify intrusion and security threats. A typical computer attack can be generalized into a five step approach [5].

Reconnaissance: The attacker collects high level information of the system.

Scanning: Using the information collected in the previous step, the attacker identifies potential vulnerabilities in the system and collects detailed information about the network such as network topology, ports used and firewall rules.

Gaining Access: There are two ways to gain access to the system depending upon the legitimacy of the user. An authorized user exploits the loop holes in the operating system or other applications running in the system. An illegitimate user makes use of the network to attach the system. DoS (Denial of Service) is one such example in which the web server is bombarded with multiple requests simultaneously that it eventually crashes.

Maintaining Access: The attacker has access to the system and tries to extract information from the system and retain control.
Covering Tracks: In order to exercise continuous control over the system, the attacker modifies system logs and other relevant information to ensure that there is no trace of contravention in the security system.

Section 2 discusses about the taxonomy of IDS. The building blocks of IDS are explained in Section 3. Section 4 presents an overview of examples of Misuse IDS, Anomaly IDS and a hybrid model combining both Misuse and Anomaly IDS. Section 5 provides a conclusion of this paper.

2. Taxonomy

Figure 1. Taxonomy of IDS

Intrusion detection system can be broadly classified based on two parameters as shown in Figure 1:

(a) Analysis method used to identify intrusion, which is classified into Misuse IDS and Anomaly IDS.

(b) Source of data that is used in the analysis method, which is classified into Host based IDS and Network based IDS

2.1. Misuse IDS

Misuse based IDS is a very prominent system and is widely used in industries. Most of the organizations that develop anti-virus solutions base their design methodology on Misuse IDS. The system is constructed based on the signature of all-known attacks. Rules and signatures define abnormal and unsafe behavior. It analyzes the traffic flow over a network and matches against known signatures. Once a known signature is encountered the IDS triggers an alarm. With the advancement in latest technologies, the number of signatures also increase. This demands for constant upgrade and modification of new attack signatures from the vendors and paying more to vendors for their support [6].

The advantages of this model are easy creation of attack signature databases, faster and easier implementation of IDS and minimal usage of the system resources.

The main weakness of the traditional and established rule based techniques is that rule based detection is highly dependent on the audit results. This one-to-one correspondence between rules and audit records makes the system inflexible. For example, given a particular penetration scenario, the audit results may vary in the sequences of events. This results in variations in the detection outcome. This may lead to large number of false positives (Section 3) and in some cases, false negatives (Section 3) too.

Another limitation includes the inability to predict a mishap and take preemptive action. The rule-based technique only helps in prevention of an intrusion when the details and patterns of it are available.

Rules are framed when a set of administrators are interviewed, different observed penetrations are recorded, rules are set to those penetration scenarios based on the expected outcomes from the analysis of audit records. Therefore, updating of rules is expensive in terms of time and money.

2.2. Anomaly IDS

Anomaly IDS is built by studying the behavior of the system over a period of time in order to construct activity profiles that represent normal use of the system. The anomaly IDS computes the similarity of the traffic in the system with the profiles to detect intrusions.

The biggest advantage of this model is that new attacks can be identified by the system as it will be a deviation from normal behavior.

The drawbacks of this model are summarized based on [3]

(a) There is no defined process or model available to select the threshold value against which the profile is compared.

(b) They are computationally expensive because the profiles have to be constantly updated and compared against.

(c) User behaviors generally vary with time and hence the model must provide a provision to revise and update it.

2.3. Host Based IDS

When the source of data for IDS comes from a single host (System), then it is classified as Host based IDS. They are generally used to monitor user activity and useful to track
intrusions caused when an authorized user tries to access confidential information.

2.4. Network Based IDS

The source of data for these type of IDS is obtained by listening to all nodes in a network. Attacks from illegitimate user can be identified using a network based IDS.

Commercial IDSs are always a combination of the two types mentioned above. The possible kinds of IDS are host based misuse IDS, network based misuse IDS, host based anomaly IDS and network based anomaly IDS. However, with greater interest and research in this field, new models are being developed such as Network Security monitor, IDS using autonomous agents and many others [3].

3. Key Functionalities and Metrics

The Key functionalities of an IDS can be divided into 4 parts [5] as shown in Figure 2:

![Figure 2. Key functionalities of an IDS](image)

(1) **Data Collection**: This module captures the real time data in the system and passes it as an input to the IDS. The kind of data that is written to a file for analysis depends on the type of IDS which is explained in sections 2.1 and 2.2. In case of network based IDS, data packets in transit are collected and in host based IDS, details such as disk usage, system process, call stack etc are logged.

(2) **Feature Selection**: Large amount of data is available in the network and a subset of it is usually analyzed for intrusion. For example, the Internet Protocol (IP) address of the source and target system, protocol type, header length and size could be studied for possible intrusion.

(3) **Analysis**: This module defines the method that is used to analyze data. One method is use of rule based IDS where the incoming traffic is checked against pre-defined signature or pattern. Another method is use of anomaly based IDS where the system behavior is studied and mathematical models are employed.

(4) **Action**: This module defines how the system must react to possible attacks in the system. It can either inform the system administrator with all the required data through email/alarm icons or it can play an active part in the system by dropping packets so that it does not enter the system or closing ports.

Two important measures, that help to evaluate the effectiveness of an intrusion detection system, are false positive errors and false negative errors [4]. A false positive error is one that is classified as intrusion by the IDS when it is a normal expected user behavior. A false negative error is one when the IDS fails to raise an alarm in the event of an intrusion since it classifies the user behavior as legitimate.

4. Types of IDS

This section provides an overview of architecture of different types of intrusion detection system and the advantages and disadvantages of the same.

The below section discusses a new type of method called Penetration state transition analysis and this is used in the development of Intrusion detection system. The rest of the discussion is about the results of [7]. In order to overcome the disadvantages mentioned in section 2.1, this new method is implemented. A *state* is a snapshot of the system at a particular moment that contains information about all the system resources such as temporary and permanent values at the memory locations. The series of events of attack or penetration are recorded as states in the state transition diagram, previously based on the known signatures. Consequently, the states obtained from the recordings of audit trial are compared with the states of former penetrations. Unlike regular rule based IDS where the audit records are compared with pattern matching sequences of known penetrations, this state transition approach will also investigate on the effects of the penetration on the state of the system. This system is referred henceforth in this paper as State Transaction Analysis Tool (STAT).

4.1. State Transaction Analysis Approach (STAT)

STAT is based on some basic prerequisite ideas. Any personnel intending to intrude into a system, for accessing security related information or with a motivation of disrupting the system’s security policy, should have access to at least one possible medium concerned to the system. It can represent a simple network cable, wireless internet connection or some bare minimum access privileges. Intrusion is performed with a specific motto such as gaining access to confidential information, trying to gain additional privileges, or harm others by mimicking their user privileges.
The intrusion is defined in the form of states in this analysis approach. Initial state is the immediate state before the penetration process is started by the attacker. Compromised State is the state of the system at which the attacker has succeeded in the intrusion task. This marks as the completion of the penetration process. There can be one or more transactions called state transactions in between the initial state and compromise state. The actions that lead to some of these states are known as signature actions. The actions without whose existence the penetration is unsuccessful are called signature actions.

The approach delivers an abstract analysis by observing the minimum number of signature actions in the various states starting from initial state to compromise state. A diagrammatic scheme is developed using these key signature actions. These schemes are similar to the state machine graphs. The representation initially contains two layers using which the penetration states and the state diagram evolve. The top layer is the command or system call representation and the lower layer is the English description. Since both the layers are ambiguous in terms of self explanation and clear understanding, the transition state diagram allows the analyst to simplify the problem.

In their paper “Penetration State Transition Analysis - A Rule-Based Intrusion Detection Approach”, Porras and Kremmerer describes the transition using a simple example of attacker trying to gain access to root privileges by using the mail utility provided by UNIX.

The attacker ensures that the mail utility fails to reset the setuid of the file. It also appends a message and changes the owner. In that way, the user can mimic the make by creating a setuid shell script and can execute it as a root with root privileges.

Table 1. Layers of the penetration scenario. Ref: [7]

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. user cp /bin/csh /usr/spool/mail/root</td>
</tr>
<tr>
<td>2. user chmod 4755 /usr/spool/mail/root</td>
</tr>
<tr>
<td>3. user touch x</td>
</tr>
<tr>
<td>4. user mail root &lt; x</td>
</tr>
<tr>
<td>5. user /usr/spool/mail/root</td>
</tr>
</tbody>
</table>

In Table 1, the user first creates a csh script and names it after the mail’s root file. The attacker only does that if there is no unread mail by root. Therefore he/she has to wait till all the mails are read. In the next step he changes the privileges of the copied file by activating the setuid bit of the file. In the consecutive steps the attacker creates an empty file and sends a message to root using the mail utility. As a result of mail failing to reset the setuid bit of the root, the attacker can execute root’s mail file without even having the actual privileges.

As said earlier, some of the prerequisite assumptions are that the attacker can access mail file, able to execute commands such as cp, chmod, touch etc. Here it is clear that the user should have some access to some preliminary system resources to initiate any attack through penetration. And the motive here is to execute a script with root privilege without actually possessing the actual root privilege.

For modeling the above scenario first we need to identify the initial and compromise states. Apparently, at first glance, the compromise seems to be done when the user is trying to gain root privilege but once a close observation is made, it is revealed that the system allows to reset or activate the setuid bit of the mail file. This is the actual initial security violation. Here the initial state is that the /usr/spool/mail/root does not exist and attacker doesn’t have root privilege. By the enabling of setuid bit of the mail file, the compromise state is accounted even though the user is not having the root privileges. Figure 3 graphically represents this idea.

![Figure 3. Initial and Compromise states. Ref:[7]](image)

The dotted line represents the sequence of actions that the system undergoes during the penetration process. It contains many or a few transition states. Further we need to identify different tasks or actions that take place from the initial state, to target compromise state. It is difficult to determine the next state as it can be any action such as testing the functionality or learning the command. Therefore, it is always easier to analyze by tracing back from the compromised state. We need to identify minimum number of key actions to perform this penetration. Actions can be in respect to the state changes that the system has undergone. It should not be necessarily the commands in the sequence. For example, Step 4 in Table 1 not only mails to root but also starts the mail program with contents of the file named ‘X’ as the input. Hence, it is better to start the analysis from Step 4 as it was identified as the compromise state earlier.

Likewise, explore all the sequences in both the initial and compromise states and start connecting the sequence actions. For example, the compromise state contains actions such as change of owner attribute of the file /usr/spool/mail/root and enabling of setuid bit. This can be correlated to one particular consequence of actions from initial state i.e., modification of owner attribute of the file...
Based on the above explanation, the state diagram can be modified (Figure 4)

```
S1
|
|--|--|--|
| S0d | S0c |
|
```

```
Exists(object) = false
Owner(object) = attacker
Setuid(object) = enabled
```

```
Object = usr/spool/mail/root
```

**Figure 4. Refining Intermediate steps. Ref: [7]**

A back tracing of all such analysis states leads to identification of minimum key signature actions (Figure 5).

```
S1
|
|--|--|--|
| S0d | S0c | S0a |
|
```

```
Exists(object) = false
Owner(object) = attacker
Setuid(object) = enabled
```

```
Object = usr/spool/mail/root
```

**Figure 5. Final distinct states. Ref: [7]**

Based on the final outcome of key or signature states in the state transaction analysis, analysts can frame the rules that can handle the possible penetration scenarios. One important point to note is that every penetration leads to a compromise state, but not every compromise is a result of variations in only system attributes but also fall in scope outside the state transaction analysis of penetration scenario.

With the derivation of such a model, the question arises with its application in real time computer systems. One way of utilizing this approach to real time systems by converting the state transition approach to a detection mechanism is to find out a systematic development of a recording mechanism of the states of the system. This accounts for the recording of changes in the system attributes for every penetration action performed. Modern audit logging mechanisms facilitates this requirement. The audit reports are fed as an input to an analysis tool which formulates the changes by observing the changes in system attributes for all the actions. The output will be the state transaction diagram, which will help the user of the tool to identify the key signatures. Those key signatures will be helpful for the formulation of rules. Since the list of attribute changes to be recorded for a system is exhaustive, all the attributes cannot be recorded. This may not give all the possible penetrations and it ends up as fewer transaction states. This may also result in inappropriate or redundant evolution of signature actions. This is one limitation of the usage of state transaction analysis for rule-based approach of detecting intrusion in a system. Moreover, this approach was used and tested successful in the Reliable Software Group [7], where the prototype implementation is referred as Unix State Transition Analysis Tool (USTAT).

4.2. Payload Modelling

In this section, an overview of a network based anomaly detector; proposed by N.Srinivasan and V.Vaidehi in [8]; is provided.

The basic idea is to let the system learn about its behaviour so that it can distinguish between an intrusion and normal data flow. As intrusion is a relative concept, what is classified as intrusion in one system may not be the case in another. Therefore, the system learns about the traffic that flows and uses this knowledge to identify intrusions in the system. The system reads the payload of every network packet and a byte frequency distribution is computed based on two parameters namely payload length and port number.

The rationale behind choosing the two parameters is:
(a) The kind of data received by each port is different. For instance, port 21 used for FTP (File Transfer Protocol) and port 80 used for HTTP (Hypertext Transfer Protocol) handle different types of data.
(b) Payloads of different length are handled by each port. Hence it is very difficult to maintain a single model for various lengths that are processed by a port.

A model based on port and payload length facilitates an accurate representation of the payload.

The system is divided into four phases namely Training phase, Incremental phase, Clustering phase and Detection phase which are briefly explained below:

**Training Phase:** N-gram analysis introduced by M.Damashek [9] is used to compute the byte distribution model. The following values are computed.
(a) In this model, N is taken to be 1 i.e. Relative frequency of each character in the payload is calculated.
(b) Both mean and variance are computed for each character
(c) Total Number of packets processed
(d) Port Number
(e) Payload length

The parameters mentioned below, together constitute the centroid model represented as Mj (where I represent length and j represents port) and are stored as a record in the database table as shown in Figure 6.
Incremental Phase: It is a part of both training and detection phase and is used to ensure the completeness and correctness of the centroid model. Systems evolve with time and the models need to be updated to ensure accuracy. When a new packet is sampled, the model $M_{ij}$ is updated based on 3 parameters as shown if Figure 7:

(a) Byte frequency of the character in the centroid model
(b) Total number of packets processed
(c) Byte frequency of the character in the processed packet

To eliminate redundancy, neighbouring models were compared based on Manhattan distance, which measures the similarity between two centroid models. If the measured Manhattan distance was lesser than the defined threshold value for the system, the models were merged. This process was done recursively so that all adjacent models within the acceptable Manhattan distance were merged as shown in figure 8.

Detection Phase: The system analyzed every single data packet for possible intrusion. The byte distribution of the payload of the incoming packet was compared against the centroid model using Mahalanobis distance; which is a standard metric to compare two statistical distribution and an useful way to compare an unknown set against a new one[8]. When the Mahalanobis distance is greater than the defined threshold value, the packet is classified as an intrusion and an alert is generated as shown in Figure 9.

The advantage of this system is that it significantly reduces the number of false alarms in the system using Mahalanobis distance to evaluate the similarity of the incoming packet. Novel attacks can be detected early as the payload is completely analyzed. The system constantly updates itself in an incremental manner based on its behavior and hence the centroid model always reflects the state of the system.

The disadvantage of this system is that encrypted data cannot be read and therefore cannot be detected for anomaly. Secondly, during the training phase, it is essential that the traffic introduced in the system is completely intrusion free. This is because the system assumes that all the payload data analyzed during the training phase is normal and builds the centroid model based on it.
4.3. Hybrid Model

There is no perfect answer to the question as to which of above mentioned methods are useful. Both the detection mechanisms have their own capabilities. Porras and Kemmerer in their paper [7] cite a good example. In an academia, teachers and students tend to use different types of programs, switch to different tools and technologies, have varied fields of investigation etc. Accordingly, patterns also keep varying. In this scenario, it is the rule-based technique that supports strongly than anomaly based technique as the probability of number of false positives is more. At the same time, anomaly based techniques will be helpful in identifying the intruders with pretended identities and authorizations. Also in many software organizations, where variety of projects are dealt but all are dependent on the network traffic internally and externally, anomaly based techniques come in hand to detect the intrusions using behavior patterns. A blend of both the techniques will not only result in better signatures but also will catalyze the performance of dependent intrusion preventive systems.

Lubomir Nistor’s publication [6] deals with such a possibility of using the rule-based IDS solutions as if they were anomaly-based. The primary idea is to modify the signatures. Whole network security is identified with the network traffic by classifying the allowed, suspicious and not-allowed traffic. The basic problem is that the system owner doesn’t know the type of traffic they require. So during the analysis phase of the system, a thorough understanding of the types of network packets will be helpful for designing the efficient system. Also, awareness of all the fields derived in a network packet (may be derived from a tool like Ethereal) is an additional advantage. Information about Source and Destination IP addresses, the type of network protocol, data format, data series, and other information such as TTL, flags etc is essential for studying the network traffic.

In the ALLOWED category, only the traffic that is meeting the user/system requirements and also acceptable communication between the peers is allowed to enter the network premises. In the SUSPICIOUS category, alarm is raised for the traffic that does not offer any useful services in the user’s network even though it is allowed by port and firewall. NOT-ALLOWED Category includes the remaining traffic that doesn’t fall in ALLOWED or SUSPICIOUS category.

First solution is to admit only the ALLOWED category of traffic and raise alarm for the rest. The main advantage of doing this is that it catches all the traffic packets which are non-standard such as slow port-scan. At the same time, such a monitoring will lead to huge amount of logs and frequent false-alarms. There will not be any priority based filtering of messages and this can be implemented well in areas which demand high security and don’t require any non-standard traffic.

Another solution is to alert on any traffic whose service is not used. It is not useful measure to monitor the traffic that is not used by any of the resources involved in the network. One good example suggested by Nistor in [6] is the http protocol using POST method in which none of the web-servers in the network are listening or using the method. In such a scenario, the packet(s) have to be suspected. Unlike the previous solution, this reduces the number of log entries but such a system design requires detailed implementation of rules and frequent update of rules database.

Apart from these solutions, trend-analyzers, packet-analyzers and statistical analyzers serve as alternate ways of utilizing the blend of the features of Rule-based and Anomaly-based intrusion detection techniques.

5. Conclusion

Intrusion detection mechanisms play a crucial role in the security landscape of an organization. State transition analysis approach and payload modelling clearly illustrate that there are inherent drawbacks with every model of IDS and there is no single version of it that can be used as a standard solution against all possible attacks. Secondly, the hybrid model shows that a combination of misuse and anomaly based intrusion systems can help to develop a potent and robust system that can identify attacks at the outset.
References


