ANOMALY BASED HOST INTRUSION DETECTION SYSTEM USING ANALYSIS OF SYSTEM CALLS

1Shijoe Jose, 2D.Malathi, 3Bharath Reddy, 4Dorathi Jayaseeli
1,2,3,4Department of Computer Science and Engineering, SRM University, Kattankulathur, Chennai, India
1shijoe8@gmail.com, 2malathi.d@ktr.srmuniv.ac.in, 3bharathreddy.s@ktr.srmuniv.ac.in, 4dorathijayaseeli.jd@ktr.srmuniv.ac.in

Abstract: Security is always a major concern of any organization. An intrusion detection system (IDS) is necessary to detect the malicious activities over a network or single system. An ID is the most useful approach to safe guard single system and the computer networks from various malicious activities. There are two categories of intrusion detection system is available host based intrusion detection system (HIDS) and network based intrusion detection system (NIDS). NIDS is mainly used to detect the malicious activates in the network. HIDS is used for detecting attacks in the single system. For an intruder, it is difficult to compromise a system being protected with Host Based Intrusion Detection System. HIDS continuously monitors the system audit and event logs to safe guard the execution of programs. Design a host based intrusion detection system with less false alarm rate is major challenge. The proposed scheme of anomaly based host intrusion detection method is to detect the malicious activities based on the analysis of system calls with less false alarm rate. An increase in detection rate in the existing anomaly host-based intrusion detection systems results in increased in false alarm rate. This leads to the development of a new method of host-based intrusion detection system with high detection rate and less false alarm rate. The gathered system calls sequences are analysed to determine the activities are normal or malicious. It is very difficult to identify the malicious activity by analyzing huge log file. Here system is designed, which uses Harmony search based K-Means clustering approach for detecting the intrusion in system calls.

Keywords: Intrusion Detection Systems, System call traces, Anomaly-based Detection, Harmony search, K-means Clustering.

1. Introduction

Many intrusion detection systems have been introduced but it is still difficult to provide a complete solution to determine malicious activity in a system. Intrusion Detection System (IDS) is one of the system security tools works along with firewalls, anti-viruses and other security features methods used to provide the total system safety against malicious activities. An IDS provides a better solution for system monitoring, malicious activity detection and efficient reporting in an organization. Intrusion Detection systems define rules to distinguish malicious activity from the normal activity, if there is any violation happens then an alarm will be raised.

A. Types of Intrusion Detection systems

There are two types of intrusion detection systems are available. Host based intrusion detection system (HIDS) which uses information gathered from a single host such as audit trail, log files, system call sequences etc. Network based intrusion detection system (NIDS) uses the gathered data by analyzing the computer network traffic.

HIDS often monitors and analyses the single host events, generally the host based models uses rule based pattern matching approaches. For each user a profile is created and the HIDS continuously monitors and compares the current audit record and the present user profiles. If there is deviation above a particular threshold value then the current activity is considered as a malicious activity.

The increased use of the computers and computer related resources, resulted in increased number of devices and users connected to the networks. A network based intrusion detection system (NIDS) monitors and analyzes the network traffic to protect a system from network-based malicious activities. NIDS works at selected system on a network and which analyzes network traffic, packet by packet to detect intrusion. If NIDS is installed in the network then it decreases the workload of intrusion detection on every individual system.

B. Intrusion Detection Techniques

To detect known attacks is comparatively less difficult with the help of known attack signatures. But unknown
attacks are difficult to distinguish from normal patterns with the given footprints. Consequently, there are two methods that have been used to detect malicious activities. Anomaly Detection system works based on the anomaly (deviation of normal pattern) present in the malicious activity. There are static and dynamic anomaly detection methods. A static anomaly detector assumes that the certain system environments will not change. The hardware parts need not to be tested as intruder cannot make changes to it and it considers only the software changes. For example, OS kernel and basic boot strap information will be static and need not to be modified. If intruder tries to change these static parts that activity will be considered as a malicious activity. Hence integrity checking is the main approach used in static anomaly detection. Dynamic detection uses the network traffic information and audit records. The audit files of operating systems records the events happen in the system. The cumulative information of resource utilization is analysed in this approach. If the cumulative information reflects a deviation from the nominal resource consumption then it is considered as anomalous resource consumption.

Misuse detection mainly concerned the system vulnerability. The known vulnerabilities can be removed by a system security administrator. The intrusion scenario indicates that there is a security flaw present in the environment which can be used by the intruder. This type of IDS can repeatedly compare present activities to known intrusion scenarios to ensure that no intruder can make use of known vulnerabilities. The difference in modeling the system performance makes the misuse systems different. The rule based misuse detection systems define normal events and malicious activities. Analyzing huge number of rule set can be computationally expensive. The rule set is used to detect the events that lead to intrusion scenario. Misuse detection systems can use the system calls for monitoring live events and can use audit records for monitoring recorded events.

The major limitation of misuse detection system is that its detection capability is limited to the known set of attacks. Unknown attacks and variants of common attacks are difficult to detect. Every day new security flaws in software are discovered and these are not identified with current intrusion scenario. The main merit of anomaly detection method is the capacity to detect unknown attacks and deviations of usage of various programs independent to the user privileges. The disadvantage of the anomaly detection method is that some known attacks may not be detected, if they come from the authorized profile of the user. Once detected, it is then difficult to classify the nature of the attack for the forensic reports. Another disadvantage of anomaly detection methods is that if the intruder knows that the activities are being profiled then the intruder can modify the user activities gradually. Therefore the user profile which is stored in the IDS will be modified to accommodate the malicious activity of the intruder. Hence it produces a high false positive alarm rate for a small trained detection method and produces a large false negative rate for a widely trained anomaly detection method.

2. Related Works

The Intrusion Detection System (IDS) is a system which provides timely detection of malicious activities of computer system or a network, so that the administrator can take proper remedial action against intrusion. The HIDS is a main part of the IDS. One of the main advantages of HIDS is that it can detect attacks on a single system. The main disadvantage of HIDS is that it uses the resources of the host system for its working. An IDS designer can use various models like anomaly based system, which checks deviation from a normal behavior but this approach has high false alarm rate. The other method is the signature based system which uses the available signatures to detect known attacks but these systems are unable to detect new attacks.

HIDS uses various metric from the host system such as system call information, system call names, log file based analysis etc. The information regarding all the activities will be in the log files. The main disadvantage in using log files is that, firstly log files contain interpreted data. Secondly the creation of log files is a difficult process. On the other side system calls represents raw data. System calls provide information regarding interaction between the kernel and the user program [2].

The basic HIDS using system call analysis was suggested by Forrest [5]; it uses collection of repeated system calls of fixed length n. For each new calculated trace (sequence of system calls) of process, are grouped based on the same n length and are matched towards the stored information and the comparison metric is calculated and rely on this result intrusion is detected. One of the other approaches that use the data flow rather than the control flow is suggested by Oualid Koucham [8]. The introduction of Artificial Neural Network (ANN) as part of the decision mechanism helps in improving the performance of intrusion detection. ANN uses nonlinear approach for information processing. ANN uses heavily parallel distributed processors for efficiency. The usage of weight considerations for connection helps in determining the final result. It is like the brain acquires the knowledge through the learning process and the inter neuron connection weight that is the weights are used to store the knowledge rate [10].

Forrest et al [1] is suggested a mechanism for determine malicious activity at sequence of system calls identified by the privilege user process. In that identified send mail process traces of UNIX for gathering system calls normal behaviour sequences. Forrest proposes a method for creating the normal activities by collecting small sequence of system calls in an active process. Sequence
may be 6, 8, 10 and 12. Once normal activity information is created then it searches the new traces information for malicious activity and also searching for sequence system calls that are not available in the normal database. After abnormal and normal system calls gathered these can be verified for intrusion detection. The approaches suggested by Warrender et al [3] uses extension of this method in embedding of sequence time-delay (stide), that remembered all the sequences of known, fixed lengths during learning. An anomaly error count was identified and the number of changes in a temporally local region. A threshold value was defined for the scoring of anomaly above the threshold a sequence is defined anomalous, indicating a possible intrusion. A.Hafmer, Martin and Lee continued the Forrest work Y.Li [4] and others are suggested a method by using Hidden markov model. These methods trace the system call sequences gathered and analysed the abnormal behavior and normal behavior by HMM. From the past research work on the system call approaches the normal behavior is collected from the normal sequences. This will helps to identify the new system calls sequences of the current process that may be abnormal or normal.

3. Proposed System

The proposal of this paper is an anomaly based intrusion detection system which works by analyzing set of system call sequences, the use of system call name of user processes and their return values to determine whether the system has been compromised or not. The main aim is to decrease false alarm rate and increase detection rate. The propose system is an anomaly based detection model, which construct a normal behavior database and checks for deviation in this stored database. The normal behaviour and abnormal behaviour sequence of system calls are present in data set. The Linux command, named S-trace is used to trace system call sequence of system calls. A normal behavioral database is constructed by analyzing the sequence of system calls traced in this way. This normal behavioral database is used to detect abnormal behavioral. Normal behavioral system call sequence data sets are used to train the IDS.

The anomaly activities that floats up in system calls can be find out by following steps

- The normal sequences of system calls are collected.
- Gathering abnormal activities sequence of system calls.
- As preprocessing step, consider the system call sequence as small sized chunks of data.
- Analyze the system call sequence with normal sequence of system calls by using Modified Harmony Search based k-means clustering.
- Using Decision system find out whether any mismatch sequence is present or not, if present then a positive alarm will be raised or it will be considered as a normal sequence.

A. Normal Trace and Abnormal Trace

System call sequences of ‘sendmail’ process.

All the normal system calls are gathered in the normal trace step. In abnormal trace step, abnormal system call sequences are gathered. UNM data set is used for collecting normal and abnormal traces. The stide, xlock, ps and login processes sequences of system calls are collected from UNM data set. The repeated execution of these processes generates system call sequences which are recorded in separate files. Each trace system call sequence contains ten to thousand system calls. These traces are collected while there are no malicious activities. The examples of abnormal processes are iprep, buffer overflow, sun sendmailcp etc. The traces with abnormal behaviour in UNM data set contain many system traces like intrusions which exploit known problems in Unix systems. For example,
sendmail has a script called `sunsendsmtpcp` which is used to copy an email message into the file. If it is used executed upon a file like `/rhosts`, a local user can acquire root access. Another example is Syslog attack. It uses the interfaces like syslog which makes buffer overflow in send mail. Intrusion traces contains three sunsendsmtpcp attacks, five error conditions of forwarding loops, 2 traces of syslog-local attacks, two traces of the syslog-remote attacks, and two traces of decode attacks. Each trace contains two attributes: process ID and a system call value. The process ID is used to identify the specific system call. An abnormal process will not have the sequences of normal system calls. The current sequence of system calls can be compared with the sequence of normal system calls stored and deviations can be detected [5][6].

**B. Data set preprocessing**

After collecting the system call sequences of from the active process, the next step is preprocessing of data. The gathered information about system call is basic raw collection data. The techniques used for preprocessing have to be applied on raw data to make the data set into processing dataset. A unique number will be assigned to each and every system call name. For instance 8 for open, 9 for close, 74 for mmap etc. The unique numbering will make it is easy to access the system call, reduces data complexity and convenient format for processing. With proper sliding window mechanism, long system call sequence numbers can be processed. The normal behavioural data base uses the window size of 3.

For example the normal behaviour database can be created from the following system call sequence: Open, read, mmap, mmap, open, read, close

For the given sequence, the system calls will be put in position 1, position 2 and position 3 as shown in below table. The window size decides the pairs generated.

<table>
<thead>
<tr>
<th>Current</th>
<th>Position 0</th>
<th>Position 1</th>
<th>Position 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>read</td>
<td>mmap</td>
<td>mmap</td>
</tr>
<tr>
<td>Read</td>
<td>mmap, close</td>
<td>mmap</td>
<td></td>
</tr>
<tr>
<td>Mmap</td>
<td>mmap</td>
<td>mmap, open</td>
<td></td>
</tr>
<tr>
<td>Mmap</td>
<td>open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mmap</td>
<td>mmap</td>
<td>open, read</td>
<td>close</td>
</tr>
<tr>
<td>Open</td>
<td>read</td>
<td>close</td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>close</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By analyzing the data set it is found that certain system calls are executed frequently. These system call executions may be followed by different system calls. For example, the read is followed by different system calls and executed two times. Therefore every system calls are recorded first and then, expanded the database for different sequences.

The expanded format is given in the following table.

**Table 2. Sequences of System Calls Expanded**

<table>
<thead>
<tr>
<th>Current</th>
<th>Position 0</th>
<th>Position 1</th>
<th>Position 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>read</td>
<td>mmap</td>
<td>mmap</td>
</tr>
<tr>
<td>Read</td>
<td>mmap, close</td>
<td>mmap</td>
<td>Open</td>
</tr>
<tr>
<td>Mmap</td>
<td>mmap, open</td>
<td>open, read</td>
<td>close, read</td>
</tr>
</tbody>
</table>

Using the sliding window, many system call sequences are produced and stored in database. After data base is preprocessed from raw information, normal behavior rule can be easily formed from this data set.

**C. Harmony Search based K-means Clustering**

**Algorithm:** Hybridizing IC-means as one step of Harmony Search Algorithm

**Require:** \( D = \{d_1, d_2, d_3, \ldots, d_n\} \) // set of n data points.

- \( k \) // Number of Clusters
- \( \text{HMCR} \) // Harmony Memory Considering Rare
- \( \text{PARmin} \) // Minimum Pitch Adjusting Rate
- \( \text{PARmax} \) // Maximum Pitch Adjusting Rate

**Ensure:** A set of solutions.

**Steps:**

1. Repeat
2. Improvise a new solution
3. Calculate cluster centroids for the new solution
4. Use K-means to reassign each data point to the cluster with the nearest centroid
5. If the result of K-means has better fitness than those in HM then
6. Replace it with a worse solution in harmony memory
7. Endif
8. Until the termination criterion is met.

The clustering constructs a rule set that determine the clusters with less amount of error. The harmony memory is updated by matching the average distance value of solutions in the harmony memory and the generated solution. This avoids the worst case solution.
The main advantage of modified $K$-means is that its ability to consider all possible solutions by treating all the divergent points in the dataset as initial centroids of clusters. In other words, modified $K$-means constructs clusters with all the cases. Hence, Harmony Search based $K$-means will distribute the dataset instances to clusters with best accuracy. The misclassified training examples are considered as errors [7][8]. The set of rules are built to define abnormal sequence and normal system calls. Once the normal database is constructed, the new sequence of system calls can be compare with normal database and result can be produced. The possible results are normal or abnormal. All the mismatch sequences are not abnormal behaviour. It may be a legal sequence of error code. Clustering rules used to identify the abnormal system calls and raise the alarm.

The UNM dataset is used to gather system calls information. The gathered data contains sequences of system calls along with their process ids. Popular performance metrics are accuracy (Acc), detection rate (DR), and false alarm rate (FAR).

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{DR} = \frac{TP}{TP + FN}$$
$$\text{FAR} = \frac{FP}{TP + FN}$$

The if-then rule can also be used to take decision. For these if-then rules, use various parameters.
- If mis_match is m and confid_avg is m and rule_count_difference is m and score_mismatch is l then the decision is normal.
- If mis_match is m and confid_avg is m and rule_count_difference is m and score_mismatch is l then the decision is normal.
- If mis_match is l and confid_avg is l and rule_count_difference is s and score_mismatch is l then the decision is normal.
- If mis_match is l and confid_avg is l and rule_count_difference is l and score_mismatch is l then the decision= abnormal

The given rules are used to detect abnormal behaviour system calls and to raise the alarm.

### 4. Results

The proposed system performance can be measured using False Alarm Rate (FAR) and Detection Rate (DR). For experiment UNM dataset has been used for system calls. The comparison results of Fuzzy c means, K-means and Harmony Search based K-means Clustering are shown below.

<table>
<thead>
<tr>
<th>Method</th>
<th>False Alarm Rate (%)</th>
<th>Detection Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy</td>
<td>2.6</td>
<td>82</td>
</tr>
<tr>
<td>K-means</td>
<td>2.4</td>
<td>81</td>
</tr>
<tr>
<td>HS K-means</td>
<td>2.3</td>
<td>84</td>
</tr>
</tbody>
</table>

![Figure 3. Graph of accuracy - False Alarm Rate.](image3)

![Figure 4. Graph of accuracy - Detection Rate.](image4)

### 5. Conclusion

The numbers of security breaches are increased with the increased number of users interacting with organizations network. The proposed system is to improve the security of the single system. In this work, a modified Harmony search based $K$-means algorithm is used for HIDS which improve the clustering accuracy. The false alarm rate has been reduced and produce high detection rate. This offline intrusion detection system helps security analyst to detect the activity of intruder and...
record the pattern mismatch sequences. The clustering method used here gives better clustering hence the results are improved.

The IDS accuracy can be increased further by combining the proposed method with other techniques. The proposed system can be trained with the other datasets and real data network to make system more suitable for real environment.

References


Journal of Computer Science and Telecommunications
vol. 2, 2011.


