Anomaly Based HIDS using System Call Names and Return Value (AHIDS-SCN&RV)

Geejamol Gladus

Abstract—The paper mainly proposes about the use of using system call names as well as the return value as the metric for Anomaly based Host Based Intrusion Detection System (HIDS). Here we make use of the concept of corpus, which is used in Natural Language Processing, where the semantic tool such as the dictionary is used. The data dictionary constructed containing every possible combinations of system call names of particular phrase length for the selected privilege processes, and for the return value of these processes we construct a database. Five features are extracted, four from the data dictionary and one from the database. These features are then given as input to the decision engine. The decision engine used is the Extreme Learning Machine. The new approach helps in detecting the Mimicry Attack with high detection accuracy and low false alarm rate. The data set used for the evaluation and testing of the HIDS is KDD99.

Index Terms—Anomaly, BP, Data Dictionary, ELM, HIDS

I. INTRODUCTION

One of the most popular and researched attack present in today’s world is commonly known as the “MIMICRY ATTACK”[6]. They are attacks which try to mimic the activities of the host process so as to evade the preventative mechanisms and to gain entry into the target system. Once they have gain access into the system they try to obfuscate their payload during the system call execution by either inserting No-Ops or by patiently waiting for the right opportunity [6]. Examples of such types of attacks are the Flame Virus, the Zotob Worn, the Stuxnet Virus, the 2011 Play station Virus etc. Most of which remained undetected for more than five years while others caused the loss of profit to business.

An IDS (Intrusion Detection System) is a system which monitors and analyses a network or a computer system to check whether they have been compromised or not by the intruders [10][1]. If so they will send alert messages to the administrator or concern user. There are different types of IDS such as the HIDS (Host based IDS), which select a metric from the host for detection, the NIDS (Network based IDS), which checks whether the packet arriving is from the correct person or not [1]. With these IDS models we can use the any one of the two detection model such as the Anomaly Based detection model, which creates a normal behavior and checks for deviation in this behavior, but they tend to produce high false alarm rate. The second detection model is the Signature based detection model, which checks for known signature patterns. They are unable to detect new attacks[1].

This paper mainly focuses on the Anomaly Based HIDS, as most of the NIDS system has difficulties in detecting the internal network attack. Hence most of the IDS available today have both. The paper proposes the use of System call name of privilege processes as well as their return values to check whether the system has been compromised or not.

We can construct the data dictionary of different phrase length for the system call name as proposed in[2] and database of the return values of system call executed for the corresponding process. Features are extracted and the corresponding values are fed in to the Decision Engine (DE). Here, for the DE we use the Single Hidden layer Feed Forward Neural Network and for the learning algorithm we use the ELM to improve the detection rate and to reduce the false alarm rate. The data set used is the KDD99. BSM (Basic Security Model) audit data from Solaris 2.5 version [9].

II. RELATED WORKS

The IDS is a system which includes the timely and proper detection of computer system or a network, so that the administrator or the user can take proper action against intrusion. The HIDS is a part of the IDS just like the NIDS is. One of the main advantages of HIDS is they can help in detecting attacks on a single system or group of system or from within the network. In spite of the advantage they have disadvantages like ,uses the resources of the host system for its working. With any IDS we can use any one of the detection model such as the Anomaly based ,which create a normal behavior pattern and checks for deviation in them. They suffer from high false alarm rate. The second one is the Signature based system which creates a data base of known attacks and they are unable to detect new attacks.

In order for the HIDS to work we have to choose a metric from the host system. Some of the examples such metrics are the system call Information, system call names, log file based analysis [4] etc. The log files records all the activities taking place inside the system. The main disadvantage in using log files is that, firstly they represent interpreted data. Secondly the production of log files is a
We detect the missing also can only be used well as the corresponding return value. These are placed in the database. Whenever a new trace appears they are passed through the same process and they are compared against the data base so as to obtain the number of match. The system call names alone are taken and dictionaries of different phrase length is found out similar to the approach specified in [2]. The reason to take the dictionary phrase length 5 is that more than that, increases the computational complexity. Here phrase length of 1 is excluded. These constitute to the Corpus which we use for feature extraction. When a new trace comes they system call name alone are passed through the same method. Their number of matching count is taken for each dictionary. From this method we obtain four feature and the fifth feature is obtained from the data base. These are then given as the input to the Decision Engine.

**ELM Algorithm:**
For the given training set, activation function and the number of hidden neuron k:
1. Assign random input weights and biases.
2. Calculate the hidden layer output matrix.
3. Calculate the output weight.

The ANN is characterized by the following:

1. **The Architecture:** i.e. the connection between neurons. In this paper we are using the single hidden layer feed forward neural network.
2. **Training:** i.e. determines the weights on the connection. In this paper we use the Extreme Learning Machine as opposed to the traditional Back Propagation (BP) algorithm. The main advantage compared to the gradient descent method is that in ELM we don’t require parameter tuning where as in BP all parameters are tuned iteratively, in ELM they reach a minimum training error also considers the weights where as in the BP method, they don’t consider the weights. ELM requires only one time training. The disadvantage is that ELM require high processing also can only be used with single hidden feed forward neural network.
3. **Activation Function:** i.e. this gives the responses of the neuron.

**III. SYSTEM ARCHITECTURE**

This section describes about the proposed system. The section is mainly divided into two parts the first one describes about the feature extraction and the second one about the decision engine.

**A. Corpus Creation & Feature Extraction.**

From the Xml file we extract the system call names, its id as well as the corresponding return value. These are placed in the database. Whenever a new trace appears they are passed through the same process and they are compared against the data base so as to obtain the number of match. The system call names alone are taken and dictionaries of different phrase length is found out similar to the approach specified in [2]. The reason to take the dictionary phrase length 5 is that more than that, increases the computational complexity. Here phrase length of 1 is excluded. These constitute to the Corpus which we use for feature extraction. When a new trace comes they system call name alone are passed through the same method. Their number of matching count is taken for each dictionary. From this method we obtain four feature and the fifth feature is obtained from the data base. These are then given as the input to the Decision Engine.

**IV. CONCLUSION**

The proposed system can be evaluated using the samples taken from the KDD99 data set. BSM (Basic Security Model) audit data from Solaris 2.5 version.

In this paper we are proposing the use of system call return value. The return values specify the result of an action, they help in revealing insufficient privileges to operate on specific files which are not opened e.g. the presence of No –Ops[11]. We construct a data base using the return values. The data dictionaries from the system call name are created and a feature extracted from this. It is then given as input to the decision engine.
The Decision engine used here is the ELM, which helps in reaching the result faster. Also, they require only one time training and the learning speed is high. All these come with a cost of high processing time. Apart from this, it is still one of the best learning approaches.

The performance of the HIDS can be measured using the following [2]

Detection rate = \( \frac{\text{No of detected attacks}}{\text{No of attacks present}} \) * 100

False Alarm Rate = \( \frac{\text{No of false rate}}{\text{No of traces in validation data}} \) * 100

In the future work, we can use the clustering method for the system call arguments values as well as with it we can use the return values so as to provide more information to the Decision engine so as to increase the accuracy of the IDS. From this, features can be extracted, thereby increasing the portability of the HIDS compared to the method proposed here where all the return values of the process taken has to be specified.

REFERENCES


