A Host Based Intrusion Detection System Using Improved Extreme Learning Machine

Megha Raj
PG Student
Department of Computer Science & Engineering
Jawaharlal College of Engineering and Technology, Palakkad, Kerala

Shijoe Jose
Assistant Professor
Department of Computer Science & Engineering
Jawaharlal College of Engineering and Technology, Palakkad, Kerala

Ambikadevi Amma T
Professor
Department of Computer Science and Engineering
Jawaharlal College of Engineering and Technology, Palakkad, Kerala

Abstract

HIDS is very challenging due to high false alarm rate. Host based systems are based on building some reference models from execution traces to characterize the system behavior. These models are then used to classify the normal as well as abnormal behavior. Most of the popular techniques are based on using Extreme Learning Machine (ELM).First analyze the discontiguous patterns of system calls and extract the important feature using ELM. This method provides powerful solution to IDS problems. For dynamic nature interpret the semantic structure between system calls and programming languages. However the use of ELM requires long training time due to the large size of typical system call traces which makes ELM computationally infeasible. So in order to overcome this problem this paper introduces a new host based intrusion detection system using Improved Extreme Learning Machine (I-ELM), in an attempt to reduce the training overhead problem while increasing the detection rate. The key concept is to apply N-gram extraction algorithm. This method mainly focuses on mining the frequent common patterns (N-grams) in the system call traces instead of considering each trace. Thus the length of training sequence is reduced significantly compare to traditional ELM while keeping the accuracy rate.

Keywords: Intrusion detection system (IDS), Host-based IDS (HIDS); Extreme Learning Machine (ELM), Improved ELM, N-gram extraction algorithm

I. INTRODUCTION

Nowadays the use of networks especially internets has become an integral part of our daily life. Many private as well as government organizations are now storing their valuable data over the network. Due to the widespread use of internets numerous intrusions are also increasing. So Intrusion Detection Systems (IDS) has been used as an additional safeguard mechanism to protect our valuable data. Intrusion Detection Systems (IDS) is mainly focused on detecting, tracking and identifying the intruders. The main goal of IDS is to set up alarms when an intruder or attacker tries to access the valuable and sensitive data in the security perimeter. An IDS first gathers information from various areas either within a computer or network and then analyzes the data or the information in order to identify the possible security breaches which consist of both anomaly and misuse detection. The main functions of IDS are:

1) Monitoring and analyzing both user and system activities.
2) Analyzing system configurations and vulnerabilities.
3) Accessing system and file integrity.
4) Ability to recognize patterns typical of attacks.
5) Analysis of abnormal activity patterns.
6) Tracking user policy violations.

The performance of IDS is measured by comparing the attack detection rate and false alarm rate. This means whenever detection rate is low then the false alarm rate is also low, this reduces the burden of system administrator. But due to the low detection rate the chances of attack is increased consequently. Conversely when the detection rate is high protection will better, but there may be numerous false alarm accompanied by an increase in the involvement of system administrator. As a result there is a decrease in the actual effectiveness of the system. The efficiency of typical IDS is evaluated mainly using three parameters:

1) Accuracy:

Accuracy deals with the proper detection of attacks and also the absence of false alarms. Inaccuracy occurs when IDS flags normal action as an abnormal or intrusion.

2) Performance:
Performance of IDS deals with the rate at which audit events are processed. Poor performance of IDS refers to poor detection rate ie real-time detection is not possible.
3) Completeness:
Completeness is the property of an ID to detect all types of intrusions and attacks accurately. Incompleteness refers to a situation where IDS fails to detect an attack.

- There are two additional properties that also determine the efficiency:

1) Fault tolerance:
An IDS should be designed with a goal in mind that it should be resistant to attack especially denial-of-service attack. Most of the IDS run above commercially available operating system or hardware, which is vulnerable to attacks. So it is important to build fault tolerant IDS.
2) Timeliness:
IDS must detect the attack or intrusion as quickly as possible to prevent the attacker from subverting the audit source or IDS itself. IDS has to perform and propagate its analysis within a short timeline to enable the security of sensitive data and also to react to the situation before much damage has been done.

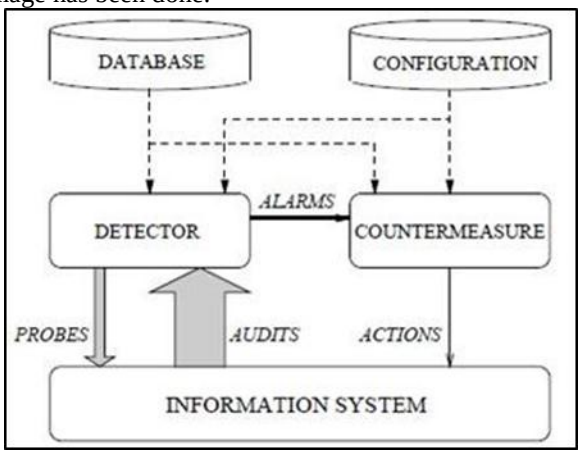

Fig. 1: Very Simple Intrusion-Detection System

Host based intrusion detection systems (HIDS) mainly focuses on protecting single computer and prevent the execution of malicious code in that system. Host based systems looks for signs of intrusions in the local host such as logins, improper file access, unapproved privilege escalation or alterations on system activities. HIDS are usually implemented by selecting the features present in the host and using this as an input to the decision engine for training to detect intrusions. Host based systems can be extremely powerful tool for analyzing the possible attack. For example: HIDS detect exactly what attacker did, which malicious command he ran, what system calls and files are opened and executed.

HIDS provide much more detailed information about intrusion than NIDS. The possibility of false alarm is less in HIDS as compared to NIDS. This is because commands executed in a specific host are much more focused than the types of traffic flow across a network. Thus the complexity of host based engines can be reduced to an extent. Host based systems are mainly used in such environments where broad intrusion detection is not needed or where the bandwidth is not available for sensor-to-analysis communications. Most of the HIDS are self-contained and it prevents the attacker from disabling the IDS. HIDS are less risky to configure with an active response for example: terminating a service or logging off an offending user. And the HIDS is more difficult to spoof into restricting access from legitimate sources.

The remainder of this paper is organized as follows. Section II reviews the related work. In Section III, the methodology of intrusion detection using ELM and I-ELM, followed by result and the conclusion of the system.

II. RELATED WORK

S.S.Murtaza, W.Khreich, A.H-Lhadj, M.Couture proposed that system calls can be represented as kernel modules. Analyzes the interactions between the states and then by comparing the probabilities of occurrences of states of system calls in normal and anomalous traces the anomalies or intrusions are detected. The technique proposed here allows for a visual understanding of system behaviour, and hence it provides more information in making accurate decision. The major drawback of this scheme is it cannot able to detect the intrusions accurately also the training time is quite long.

L. Ying, Z. Yan, and O. Yang-jia, proposed a new method that is based on the Host system only. paper a new Host- based intrusion detection system is proposed which is a combination of both log file analysis technology and BP neural network technology. Log file analysis is an approach of misuse detection, where as BP neural network is an approach of anomaly detection. By combining of these two kinds of detection technologies, the efficiency and accuracy of the host based intrusion detection system can be improved effectively. But this method also failed to reduce the training overhead problem.

A. Sultana, A. Hamou-Lhadj, & M. Couture proposed a method based on an improved Hidden Markov Model for anomaly detection using frequent common patterns. Here the paper focuses on reducing the training overhead problem, which is a serious issue in most of the semantic based intrusion detection system. The basic technique is to build an improved Hidden Markov Models (HMM). The models are build based on extracting the largest n-grams (patterns) in the traces instead of taking each trace event on its own. So that the learning time of the model is improved, which in turn result in the reduction of training time.

III. METHODOLOGY

This method mainly focuses on using semantic features in contiguous and discontiguous system calls in an attempt to detect the intrusions.

A. Contiguous and Discontiguous Semantic Analysis:

High false alarm rate, large trace size, high processing times are the key issues in host-based intrusion detection systems over two decades. So in an attempt to reduce these false alarms and processing time, a new method is proposed that is based on semantic feature. The main concept is to apply the semantic structure to both contiguous and discontiguous system calls in order to reflect abnormities or intrinsic activities hidden in the system calls and programming languages which can understand the anomaly behavior much better. By analyzing the discontiguous system calls patterns, the semantic feature can be derived and it is then given as an input to the decision engine. Here ELM is used as the decision engine because it offers better performance in sustained deployments. By counting the occurrences of semantic phrases, the ELM are able to apply the semantic feature more simply and rapidly. In this approach, contiguous and discontiguous system calls are first analyzed and then system calls are considered as "letters" , with continuous string of letters forms a "word". After the word dictionary formation the words are then combined to form "phrases" with every possible combination of the word up to a specified phrase length.

The following steps are required to apply the semantic feature to the host based intrusion detection system:

1) First the training data must be processed and then dictionary is extracted from the training data containing every contiguous system calls traces present in the training dataset. Each entry in the dictionary forms a phrase of length "1".
2) The words are again combined to further dictionaries of "phrases". The phrases represent every possible combination of the word up to a specified length.
3) After phrase dictionary construction next step is to process the dictionaries once again to extract the occurrences count of different length phrases. The training data is re-examined in this step, the incoming system calls are compared against the theoretically possible phrase list. After comparison the number of phrases consisting of discontiguous words in each training sample is obtained.
4) After normalization and standard data treatment routines, the information is used to train the decision engine (ELM).
5) When a new data is arrived, it is first compared with the existing phrase dictionary in order to classify it as either "normal" or "abnormal" data.

B. Algorithm Used:

Function GETWORDS (system call traces)
for all traces in the training data
set the counter to 1
for all the system calls in trace do
word =system call + next counter calls
If word is already present in the word Dictionary then
Increment the word count
Else
Add the new word to word Dictionary
end If
counter =counter+1
end for
end for
Return the word Dictionary
end Function
Function GENPHRASES (word dictionary, length)
Create new phrase dictionary for phrases of given length from the word dictionary.
for all words present in the word dictionary do
while length > current phrase length do
set current Phrase ← current Word
for current Word in the word dictionary do
current Phrase = current Phrase+ next dictionary : word
end for

end while
Add phrase to the phrase Dictionary
Increment word list start position
end for
Return the phrase Dictionary
End Function
Function GETPHRASECOUNT (system call trace)
feature Vector = new array with length=number of dictionaries
for all Phrase Dictionaries do
set i as the phrase length for dictionary
set phrase Count ← 0
for all Phrases in the Dictionary do
If phrase is present in trace then
Increment the phrase Count
end If
set feature Vector[i] ←phrase count
end for
e nd for
Return the feature Vector
End Function
Function EVALUATE SYSTEMCALL TRACE (new trace)
set new Feature ← get Phrase Count (new Trace)
Normalize the new Feature
Then assign feature → trained decision engine
Now deResult ← decision engine output: decision engine output is stored in deResult
If global threshold < deResult then
classification ← anomalous : classify the new trace as intrusion
or else
classification ← normal : classify the new trace as normal data
End If
Return classification
End Function
Disadvantages of Contiguous and Discontiguous Semantic approach

1) Training time is quite long and it result in training overhead.
2) Computationally infeasible due to the large size of typical traces.

In order to overcome the drawbacks of existing system here a new method for host based intrusion detection system has been generated. The main method proposed in this paper is to use N-gram extraction algorithm in an attempt to reduce the training time without affecting the accuracy of the system.

C. N-Gram Extraction Algorithm:

This method uses an N-gram extraction algorithm to overcome all limitations of the existing system and ensures the intrusion detection with reduced training time. The main strategy used in this new model to reduce the size of traditional ELM identifies the frequent common sub-sequences or patterns in a string or phrase, where the length of the patterns varies from "1" to "n". First the algorithm analyzes the training sequences and then extracts the n-grams (frequent common patterns) from the training dataset. Then based on the extracted patterns a new model is constructed; I-ELM (Improved Extreme Learning Machine). The process of constructing the I-ELM is very similar to the construction of ELM. Here the set of observables in I-ELM are "n" grams instead of original system calls. Since common patterns (n-grams) are frequently found in the trace sequences at most "n" number of system calls can be replaced by "1" particular "n"-gram, which is then given as input to the I-ELM. So longer "n" means the training sequence for I-ELM is shorter. Due to the training sequence the cardinality of the observable set is reduced. The above two factors result in minimizing the overall training time in I-ELM over ELM. The number of hidden states remains the same as the number of hidden states in ELM. The state transition probability for I-ELM is randomly assigned and then the training model is adjusted iteratively till it reaches the acceptable threshold.

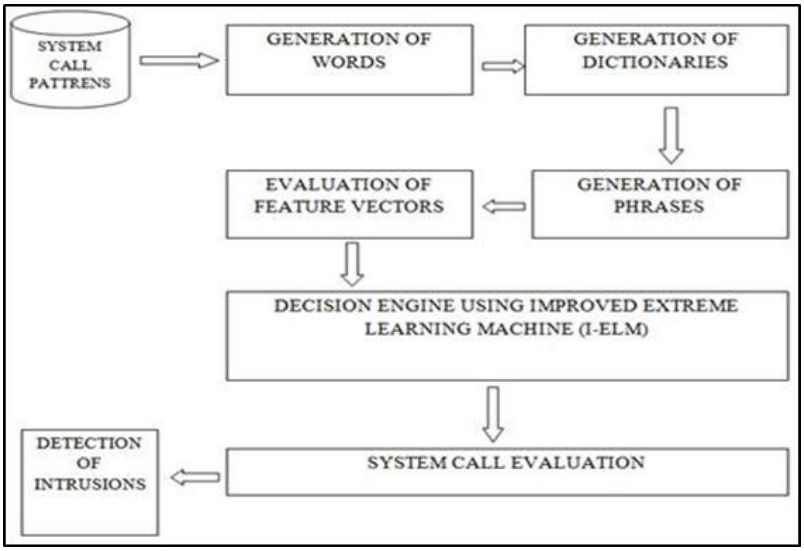

Fig. 2: System Architecture of HIDS with I-ELM

For example, if XYZU, YZUW and XWYZU are the input sequences, where X, Y, Z, W, U are the valid 1-grams. Remaining steps consist of combining, two n-grams of length n in order to make an n-gram of length n+1. A sub-sequence or pattern pn+1 forms an n-gram, if frequency of pn+1 is greater than alpha multiplied by the minimum frequency of qn and rn. Here, pn+1 is generated from qn and rn (two valid n-grams of length n). Therefore, a model with a smaller alpha takes most of the n-grams as valid n-grams, even with very low frequency, and becomes very flexible. Due to very low value of alpha high false negative rate will be raised. Similarly if the value alpha is significantly large then model take only very few n-grams with high frequency. A very high value of alpha may lead to generate high false positive rate. If we take alpha =0.6 in previous example, and combine the two valid 1-grams X and Y, we get XY that is present in the sequence.

However, the frequency of XY is 1 in our input data which is less than alpha(= 0.6) minimum frequency of X and Y (= 2). Therefore, XY does not qualify as a valid 2-gram in this model. Whereas, YZ is a composition of 2 valid 1-grams Y and Z, and the frequency of YZ is 3 which is greater than alpha (= 0.6) minimum frequency of Y and Z (= 3). Thus, YZ is a valid 2-gram in the model. Similarly, DB is also another valid 2-gram in the model. Though, the 2-grams XY, UW, XW, WY are present in the input, they do not qualify as valid 2-grams because of the low frequency. In the next step, YZ and ZU are combined to make YZU. The 3-gram YZU is valid since the frequency 3 is higher than alpha (= 0.6) minimum frequency of YZ and ZU (= 3). Since there is no more than one 3-gram to compose a 4-gram, we stop at this point. That makes the highest n-grams to be 3-grams. In data processing step, all valid n-grams are extracted from pre-processed trace sequences by setting alpha= 0.6. Then mark each n-gram with a unique identification number for future use. Then, replace the n-grams in the trace sequences with their corresponding unique identification numbers (n-gram id). Before replacing the n-grams,], sort the n-grams according to their lengths, where longer n-grams were replaced before the shorter ones. If there was a tie in their lengths, the one with higher frequency got the priority.

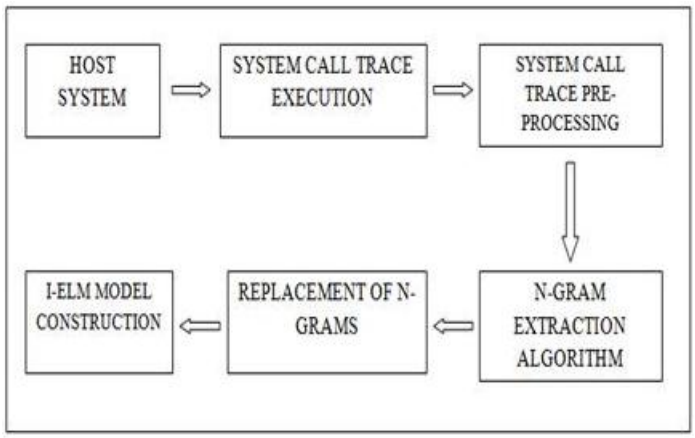

Fig. 3: N-Gram Extraction Method in I-ELM

IV. RESULT

Results are generated using ten different areas of traces in training and testing dataset.

Table.-1:
Result obtained

Training data	Testing data	Intrusion present/not present	Final result
n1	t1	no	no
n2	t2	no	no
n3	t3	no	no
n4	t4	yes	Intrusion detected
n5	t5	yes	Intrusion detected
n6	t6	yes	Intrusion detected
n7	t7	no	Intrusion detected(no training data)
n8	t8	yes	Intrusion detected
n9	t9	yes	Intrusion detected
n10	t1	no	no

In this table N1, N2 ....., N10 are training dataset and T1, T2....., T10 are testing data set. Row seven shows intrusion as it won't find matching training Data and alarms it as intrusion which is a false alarm. It shows that hundred percent detection and no false alarm with proper training.

V. CONCLUSION

The objective of this study was to improve core performance of intrusion detection and at the same time try to reduce heavy burden of false alarms present in traditional approaches. Application of semantic rules and use of multiple decision engines has helped to solve the objective. The semantic theory used defines a scalable set of rules governing the combination of terminating units. Decision Extreme Learning Machine is the method used to differentiate between legitimate and malicious activities against base line of normal behaviour. The ELM methodology has been verified as applicable to the IDS problem, with potential synergies uncovered due to the rapidity of decision engine training possible using this scheme. These techniques, however, require long training time of the models, which makes them computationally infeasible, the main reason being the large size of typical traces.

An improved extreme learning machine has been proposed using the concept of frequent common patterns. These replacements considerably reduce the size of the observable sequences (i.e. trace) and the number of unique observables, hence contribute to important reduction of training time. The objective is to improve core performance of intrusion detection and at the same time try to reduce heavy burden of false alarms present in traditional approaches. Detection of these malicious activities results in system level lock for the host and provides protection against threat. Public dataset were used for evaluation of the new algorithm in this project to allow comparison with existing approach.

If proper training is provided to IDS hundred percent results can be achieved in respect of detection rate as well as false alarm rate. To achieve this regular and rapid training I-ELM algorithm is used. The more rapid training and smaller on-going footprint of an I-ELM reduces the long term burden imposed by the IDS, without unduly affecting decision granularity.

REFERENCES

[1] S.S.Murtaza, W.Khreich, A.H-Lhadj, M.Couture, “A Host-based Anomaly Detection Approach by Representing System Calls as States of Kernel Modules”, (SBA) Research Lab, Department of Electrical and Computer Engineering,2013.
[2] A. Sultana, A. Hamou-Lhadj, & M. Couture, “An improved Hidden Markov Model for anomaly detection using frequent common patterns”, in 'ICC' , IEEE, , pp. 1113-1117June 2012.
[3] F. Bin Hamid Ali and Y.Y. Len, “Development of Host Based Intrusion Detection System for Log Files,” Proc. IEEE. Symp. Business, Eng. and Industrial Applications (ISBEIA), pp. 281-285, Sept. 2011.
[4] A. Liu, X. Jiang, J. Jin, F. Mao, and J. Chen, “Enhancing System- Called-Based Intrusion Detection with Protocol Context,” Proc. Fifth Int’l Conf. Emerging Security Information, Systems and Technologies (SECURWARE ’11), pp. 103-108, Aug. 2011.
[5] L. Ying, Z. Yan, and O. Yang-jia, “The Design and Implementation of Host-Based Intrusion Detection System,” Proc. Third Internatinal.Symp. Intelligent Information Technology and Security Informatics (IITSI), pp. 595-598, Apr. 2010.
[6] C. Feng, J. Peng, H. Qiao, and J.W. Rozenblit, “Alert Fusion for a Computer Host Based Intrusion Detection System,” Proc. 14th Ann. IEEE Int’l Conf. Workshops on the Eng. Computer-Based Systems (ECBS ’07), pp. 433-440, Mar. 2007.
[7] D. Yeung and Y. Ding, “Host-Based Intrusion Detection Using Dynamic and Static Behavioral Models,” Pattern Recognition, vol. 36, no. 1, pp. 229-243, 2003.
[8] X.D. Hoang, J. Hu, and P. Bertok, “A Multi-Layer Model for Anomaly Intrusion Detection Using Program Sequences of System Calls,” Proc. 11th IEEE Int’l. Conf.Networks, pp. 531-536, 2003.
[9] S.A.Hofmeyr, S. Forrest, and A. SoMayaji, “Intrusion Detection Using Sequences of System Calls,” J. Computer Security, vol. 6, no. 3, p. 151, 1998.
[10] W. Lee, S.Stolfo, and K.Mok, “A Data Mining Framework for Building Intrusion Detection Models,” Proc. IEEE Symp. Security and Privacy, pp. 120-132, 1999.