Review of A Semantic Approach to Host-based Intrusion Detection Systems using Contiguous and Dis-contiguous System Call Patterns

Mr. Kulkarni Sagar S.[#1], Prof. Kahate Sandip A.[*2]

[#]Student & Department of ComputerDepartment & Savitribai Phule Pune University
[#]Assistent Prof. & Department of ComputerDepartment & Savitribai Phule Pune University,
Otur, Pune, Maharastra, India

Abstract- *Use of security tools are increased over recent years as a result of increased number of malicious events. To detect possible anomalous events security administrator makes use of intrusion detection system. Earlier intrusion detection systems have higher FPR and lower detection rate. This motivates many researchers for designing different models for detection. Designing host based intrusion detection is difficult task as there are various number of operating environment and difficulty in selecting features to be monitored for intrusion detection. This paper describes one of such host based intrusion detection system that has taken different approach for detecting anomalous events.*

Keywords- *Anomaly Detection; Intrusion Detection; Semantic Theory.*

I. Introduction

There is increased number of malicious events in recent years. These malicious events are motivated from different perspective, but their impact on business is huge. Therefore, organization often uses different security tools to secure their data. The problem with most of the security tools is that they are very good at detecting known intrusions, but incapable of detecting zero-day intrusions. To detect anomalous events system administrator preferably makes use of intrusion detection system. This is because IDS is specially designed for detecting intrusions and they can detect zero-day intrusions. There are two types of IDS, network based and host based [1]. Each IDS can be divided as misuse based and anomaly based, this categorization is based on the approaches used for detecting intrusions. The misuse based systems uses rules for detecting intrusions, thus they are not capable of detecting zero-day intrusions. The anomaly based detection system uses normal behaviour of monitoring entity and any large deviation to normal profile is flagged as intrusion. Anomaly based IDS suffers from high FPR and they are dependent on chosen threshold [1][2][4].

This paper makes review of system proposed by Creech et al [13]. This system uses semantic concept for finding out hidden activities in system call patterns and makes it possible to detect intrusions. The system is designed to work with incomplete training data and for increasing detection rate with lower FPR.

The rest of paper is organized as follows: Section 2 covers literature review. Section 3 presents theories which support this system. Section 4 contains system design, Section 5 gives algorithms, Section 6 experimental result and Section 7 contains concluding remarks.

II. Literature

Initially use of system call patterns for intrusion detection was proposed by Forrest et al [4], in which Fixed-length patterns were used for anomaly detection. Forrest et al [5] provided extension to their earlier model, in which hamming distance was used as process similarity measure. Later, Warrender et al [6] uses fixed-length patterns and local mismatches for detecting intrusions. The t-STIDE model provided by Warrender et al [6] in which pattern rarity was extracted for anomaly detection but it did not perform good.This is because of their definition of rarity. Use of variable length patterns for constructing normal profile of a program was proposed by Wespi et a [7]. Wespi et al model uses Teiresias pattern matching algorithm for extracting variable length patterns. In experimentation it was found that Variable length patterns database is very compact than fixed length patterns. The use of rarity index for intrusion detection was extended by Vardi et al [8], but no experimentation was done over provided over publicly available dataset. Liao et al [9] provided text categorization technique for detecting anomalous sequence, their model views system call traces as documents and then categorizes by clustering those documents. The limitation with their model is that, it does not work in real time. Xuan et al [10] used multilayer model for minimizing FPR. Their model uses HMM for determining cause of mismatch event, such that mismatches because of anomaly can be separated. Zhang et al [11] proposed model uses variable length patterns for normal profile generation and detection was done by taking average hamming distance between extracted patterns. However, the method used for getting average hamming distance was problematic. Use of kernel events for intrusion detection was done by Syed et al [12], in which system call are converted into their respective kernel events. The anomaly detection was carried out by determining the probability of portion in normal trace.

This system uses semantic information of system call patterns for anomaly detection. The advantage of this system is that it is robust to incomplete training data. Further, once training is completed then anomaly detection is easy task.

III. Preliminaries

A. *Semantic Theory*

Semantic theory in computer system often takes form of grammatical rules of a language. These grammatical rules specify how valid programming statements can be formed. To understand why semantic theory can be useful for system call analysis consider, whenever program is executed it will generate system call patterns. As programs are written in higher level languages which follows some grammar, the system calls generated after execution of these programs also follows similar grammatical rules. Thus if the grammatical rules are devised then it is possible to test for valid and invalid system call patterns. To use semantic concept over system call patterns, the each system call is converted into unique letter, then sequence of system calls are considered as 'words' and sequence of words is termed as 'phrases' [13].

B. *Definitions*

1. Let T={architecture specific system calls}
2. Let $N=\{3x Є\ T:\ y=xi,xj,....\}$ or $N=\{$*all possible system call sequences*$\}$
3. Let Z is a known normal trace and A represents a known anomalous trace.

C. *Syntactic Development*

Using notion of CFG, T can be called as terminating units and N represents non-terminating units. Then system call trace can be represented as,

$$3x\ ЄT:\ Z\rightarrow xZ'$$

The meaning of above rule is that any normal trace G can be subdivided into any number of subsequence's and it is still possible to get G only if these subsequence's are combined in specific order.

Using the above rule new set B is constructed which consist of phrases made by combining different sequences of system calls. This set is given as

$$B=\{3x_i,x_j,x_k,\ldots x_n\ Є\ T\ |\ (x_i,x_j,..)\ Є[Z_0,Z_1,Z_2,\ldots Z_n]\}$$

The $Z_0,Z_1,Z_2,\ldots Z_n$ are normal training traces and there is no guarantee that every sequence from *B* set is seen in normal training traces. This is because of construction of phrases is done by combining every possible word with each other [13].

D. *Semantic Detection Hypothesis*

To detect anomalous sequence it must be deviated from normal profile. Thus, if anomalous sequence occurred then the number of phrases seen in abnormal trace is lower than that of normal trace and detection possible [13], It can be written in mathematical rule

$$|\{3x\ ЄB:\ Z_n\rightarrow xZ'_n\}|>>|\{3x\ ЄB:\ \mathrm{A}\rightarrow xA'_n\}|$$

IV. Implementation

Architecture of system is given in figure 1. This system functions in two steps firstly it needs training and secondly it is used for detecting anomalous events. Further training is divided into three steps given as pre-processing, word dictionary generation, phrase dictionary generation, updating semantic information. The detection phrase includes finding invalid phrase count of valid phrases in system call trace. The threshold over semantic count used for detecting anomaly.

A. *Training*

- *Pre-processing*

  In this step system calls are mapped to unique letters, this is done with the help of mapping file consisting of system calls with their id. The newly processed traces are now available for detection.

- *Word Generation*

  In this step, a sliding window of different length is scrolled over system call traces and each system call pattern is extracted. The length of sliding window can ranges from 2 to *n*, where n is length of system call trace. This is done as given below,

  Assume T as trace given as, T = $\{s_1,s_2 \ldots s_n\}$ where $s_1,s_2,\ldots s_n$ are system calls in a trace T, w is sliding window where, l <= |w|<= n,. Initially set |w| = l and slide window across all thetraces extracting all patterns of length |w|. If w<=mthen increases |w| = |w| + 1 and do same procedure.

- *Phrase Extraction*

  This step consists of building all possible phrases using word dictionary. The phrases extracted are consisting of different lengths. The phrase extraction process is a lengthy process and often takes several hours; therefore it must be carried out in offline environment.

- *Updating Semantic Information*

  To get semantic information about system call patterns, one extra pass over training file is required in which occurrence count of each phrase in training file is determined which acts as semantic information. According to semantic theory if phrase is valid phrase then its occurrence count in training file must be high, as opposed to phrases that are invalid.

B. Detection

The detection phase is very simple as compared to training phase. The test system call patterns are initially pre-processed like in training phase. Then occurrence count of phrases seen system call traces is extracted. If during detection the count is decreased below threshold level then such trace is flagged as anomalous.

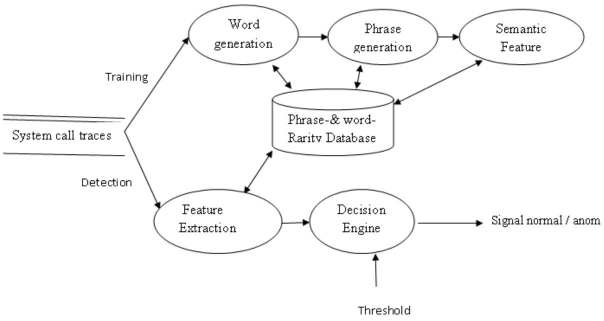

Fig. 1 System Architecture

Decision engines such as ANN, HMM, ELM, SVM are trained using these semantic feature so that usefulness of semantic concept can be tested.

V. ALGORITHM

A. Algorithm For Fixed-Length Pattern Extraction

```
procedure GETFIXEDWORDS(syscalltraces)
for all traces do
    counter = 1
    fixedWordDic←empty
      for system call in traces do
        word = systemcall + nextcountercalls
        if word is not in fixedWordDic then
          add word to fixedWordDic
        end if
      end for
    return fixedWordDic
end procedure
```

B. Algorithm Phrase Generation

```
procedure GETPHRASES(syscalltraces, initWordDic)
    phraselength ←1
    for all fixed length patterns do
    while phraselength<MAX do
      extract and combine all possible Fixed-Length Patterns to form phrase of length phraselen and store in phraseTable
    end while
    end For
    end procedure
```

C. Algorithm For Detection

```
procedure updateSemanicInfo (trainingdata)
for all phrases in phraseTable do
p← phrase
cnt← get count of training traces containing p
update phraseTable Info set count←cnt
end For
end procedure
```

D. Algorithm For Detection

```
procedure decisionEngine(testtrace)
extractedFeatures←getPhraseCount(testtrace)
normalize extractedFeatures
extractedFeatures→DE
 if result> threshold then
     signal←anomalous
 else
     signal ← normal
  end If
return signal
end procedure
```

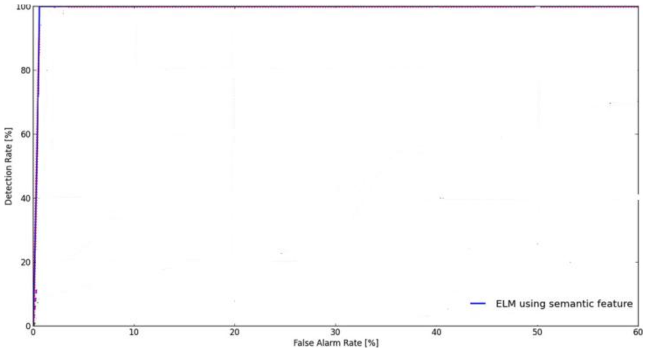

FIGURE 2. Evaluation Results on DARPA 98 dataset

VI. EXPERIMENTAL RESULTS

The use of semantic concept for system call based intrusion detection was evaluated using UNM, DARPA and ADFA-LD. The ADFA-LD dataset was created by Creech et al [12] for evaluating this system. The ADFA-LD dataset was created as there was lack of standard intrusion detection dataset that can represent current system environment and attacks. For experimenting 500 normal traces were used for training purpose, validation traces consist of 4500 normal traces and 37 attack traces were used for testing purpose. The result shows improved results after using semantic features for intrusion detection.The evaluation results for DARPA dataset are shown in figure 2.

VII. Conclusion

The semantic theory is very useful for anomaly detection. But extracting semantic information from system call patterns is time consuming process even if reduced set is used while constructing phrases. The DR of this system rises up to 88% with FPR of 12% on UNM dataset [13]. This may be because of training over combined data of different processes. The main limitation of this system is its training time requirements. The training time can be reduced by limiting longer words extracted, because longer sequences are not representative of normal profile.

REFERENCES

[1] John McHugh, Alan Christie, and Julia Allen, “The Role of Intrusion Detection Systems, IEEE SOFTWARE, SEP 2000.

[2] Mehdi Bahrami and Mohammad Bahrami, “An overview to Software Architecture in Intrusion Detection System”, Soft Computing and Software Engineering (JSCSE), 2011.

[3] Herve Debar, “An Introduction to Intrusion-Detection Systems”, IBM Research, 2011

[4] S. Forrest,S. A. Hofmeyr and A. SoMayaji, ”A sense of self for Unix Processes”, IEEE Symposium, May 1996..

[5] S. Forrest,S.A. Hofmeyr and A. SoMayaji, ”Intrusion Detection Using Sequences of System Calls”, IEEE Symposium, May 1996.

[6] C. Warrender, S. Forrest, and B. Pearlmutter, ”Detecting intrusions using system calls: alternative data models”, Proceedings of the 1999 IEEE Symposium,1999.

[7] John Andreas Wespi and Herv Debar, “An intrusion detection system based on the teiresias pattern discovery algorithm”, Proceedings of EICAR, 1998.

[8] Wen-Hu Ju and Yehuda Vardi, “Profiling UNIX Users And Processes Based On Rarity of Occurrences Statistics with Applications to Computer Intrusion Detection”, Fourth Aerospace Computer Security Applications Conference, October 1988.

[9] John Y. Liao, V. R. Vemuri,”Use of K-Nearest neighbor classifier for intrusion detection”, Computer Security, 2002.

[10] Xuan Dau Hoang, Jiankun Hu, Peter Bertok, "A Multi-layer Model for Anomaly Intrusion Detection Using Program Sequences of System Calls", The 11th IEEE International Conference on Networks ICON2003, Oct. 2003.

[11] Ye Du, Ruhui Zhang, and YouyanGuo, "A Useful Anomaly Intrusion Detection Method Using Variable-length Patterns and Average Hamming Distance”, Journal of Computers, Aug 2010.

[12] Syed Shariyar Murtaza, Wael Khreich, Abdelwahab Hamou-Lhadj, Mario Couture, "A Host-based Anomaly Detection Approach by Representing System Calls as States of Kernel Modules", IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), 2013.

[13] G. Creech and J. Hu. ,”A Semantic Approach to Host-based Intrusion Detection Systems Using Contiguous and Dis-contiguous System Call Patterns”, IEEE Transactions on Computers, 2014.

[14] UNM intrusion detection dataset available at http://www.cs.unm.edu/~immsec/systemcalls.htm.