VNU Journal of Science, Natural Sciences and Technology 24 (2008) 71-81

A program anomaly intrusion detection scheme based on fuzzy inference

Dau Xuan Hoang[1,*], Minh Ngoc Nguyen[2]

[1]*Department of Computer Science, Faculty of Information Technology,*
The Posts and Telecommunications Institute of Technology (PTIT),
122 Hoang Quoc Viet, Cau Giay, Hanoi, Vietnam
[2]*Vietnam Posts and Telecommunications (VNPT), 10th Floor, Ocean Park Building,*
No.1 Dao Duy Anh, Dong Da, Hanoi, Vietnam

Received 31 October 2007

Abstract. A major problem of existing anomaly intrusion detection approaches is that they tend to produce excessive false alarms. One reason for this is that the normal and abnormal behaviour of a monitored object can overlap or be very close to each other, which makes it difficult to define a clear boundary between the two. In this paper, we present a fuzzy-based scheme for program anomaly intrusion detection using system calls. Instead of using crisp conditions, or fixed thresholds, fuzzy sets are used to represent the parameter space of the program sequences of system calls. In addition, fuzzy rules are used to combine multiple parameters of each sequence, using fuzzy reasoning, in order to determine the sequence status. Experimental results showed that the proposed fuzzy-based detection scheme reduced false positive alarms by 48%, compared to the normal database scheme.

Keywords: anomaly intrusion detection, fuzzy logic, hidden Markov model, program-based anomaly intrusion detection.

1. Introduction

One of the most difficult tasks in anomaly intrusion detection is to determine the boundaries between the normal and abnormal behavior of a monitored object. A well-defined boundary helps an anomaly detection system correctly label the current behavior as normal or abnormal. Unfortunately, the border between the normal and abnormal behavior may not always be precisely defined since the normal and abnormal behavior can overlap or be very close to each other [1-3]. This leads to an increase in false alarms and a decrease in the detection rate. This paper proposes a fuzzy-based solution to reduce false alarms for program anomaly detection using system calls.

The boundaries between the normal and abnormal behavior of a monitored object can be divided into two types: hard boundaries and soft boundaries. A hard boundary is usually represented in the form of crisp conditions or

[*] Corresponding author. E-mail: dauhoang@vnn.vn

fixed thresholds. For example, in the normal database detection scheme [4], a short sequence of system calls is labeled as normal if it is seen in the training set. Otherwise, it is classified as abnormal. In the second test of the hidden Markov model-based (HMM) two-layer detection scheme [5], a probability threshold $\hat{P}$ is used to determine the status of short sequences. If a sequence's probability P, generated by the HMM model, is equal or greater than the probability threshold ($P \geq \hat{P}$), it is considered normal. Otherwise, it is considered abnormal ($P < \hat{P}$).

In contrast, a soft boundary is usually represented by fuzzy sets and rules, instead of crisp conditions, or fixed thresholds. In [1], five fuzzy sets are used to represent the space of each input network data source. In addition, a set of fuzzy rules is defined to combine a set of inputs in order to produce an output which is the status of current network activity. In [6], a set of fuzzy association rules is generated to represent the normal behavior of network traffic.

Anomaly detection approaches, based on soft boundaries in general, or fuzzy sets and rules in particular, can produce better detection results than those based on hard boundaries, because of the following reasons:

- Since normalcy and abnormalcy are not truly crisp concepts, it is difficult to define a hard boundary that can create a sharp distinction between the normal and abnormal. Therefore, it is natural to use fuzzy sets to define a "soft" border between them [1,3]. In fuzzy logic terms, the normal is represented by the degree of normalcy. Similarly, the abnormal is represented by the degree of abnormalcy.
- Anomaly detection systems, based on fuzzy inference, can combine inputs from multiple sources, which leads to better detection performance [1].

Although the application of fuzzy inference in anomaly intrusion detection is still in an early stage, promising results have been reported by several fuzzy-based anomaly detection approaches. Cho [7] reported a high detection rate and a significant reduction in the false positive rate, when using fuzzy inference to combine inputs from three separate HMM models deployed in a user anomaly detection system. Luo et al [8] presented a real-time anomaly intrusion detection system, in which a set of fuzzy frequent episode rules was mined from the training data to represent the abnormality. The proposed approach [8] reportedly had lower false positive rates than those, based on non-fuzzy frequent episode rules. Good detection results were also reported in [1,3,6].

In this paper, we propose a fuzzy-based detection scheme which is based on the HMM-based two-layer detection scheme proposed in our previous work [5], and the normal database detection scheme [4]. The proposed detection scheme aims at reducing false alarms and increasing the detection rate. We employ fuzzy inference to evaluate each short sequence of system calls, by combining the sequence's multiple parameters. Each short sequence is represented by three parameters: the sequence probability generated by the HMM model, the sequence distance and the sequence frequency produced by the normal database [4]. Instead of using crisp conditions or fixed thresholds, a group of fuzzy sets is defined to represent each parameter's space. A set of fuzzy rules is created to combine these input sequence parameters, in order to produce an output of the sequence status. Experimental results showed that our fuzzy-based detection scheme reduced false alarms significantly, compared to the two-layer detection scheme [5] and the normal database detection scheme [4].

The rest of this paper is organized as follows: Section II gives a brief introduction to fuzzy logic, fuzzy sets and fuzzy rules. Section III describes the proposed fuzzy-based scheme for program anomaly intrusion detection using system calls. Section IV presents some experimental results and discussion. Section V is our conclusion.

2. Fuzzy logic

Fuzzy logic is an extension of Boolean logic, which specifically deals with the concept of partial truth. The mathematical principles of fuzzy sets and fuzzy logic were first presented in 1965 by professor L.A. Zadeh [4]. Since then, fuzzy logic has rapidly become one of the most successful technologies in the development of control systems. The application of fuzzy logic is ranging from simple, small and embedded micro-controllers to large data acquisition and control systems [9].

While a truth value in classical logic can always be expressed in binary terms (0 or 1, True or False, Yes or No), a truth value in fuzzy logic is represented by the degree of truth. The degree of truth can be any value in the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth.

2.1. Fuzzy sets

Mathematically, a fuzzy set *A* is defined as follows:

$$A = \{(x, \mu_A(x)) \mid x \in U\}$$

where $\mu_A(x)$ is the membership function of the fuzzy set *A*, and *U* is the *Universe of Discourse*. A Universe of Discourse, or Universe in short, is the range of all possible values for an input to a fuzzy system.

2.2. Fuzzy rules

Rules in fuzzy logic are used to combine and interpret inputs in order to produce an output. Fuzzy rules are usually expressed in the IF/THEN form as:

IF <variable> IS <fuzzy set> THEN <output>

A rule is said to fire if its truth value is greater than 0. It is also noted that there is no "ELSE" clause in a fuzzy rule. All available rules in a fuzzy control system are evaluated because an input can belong to more than one fuzzy set.

Like classical logic, fuzzy logic also supports AND, OR and NOT operators, which can be used to create more complex fuzzy rules. Let *x* and *y* be two fuzzy variables, and $\mu(x)$ and $\mu(y)$ be the degrees of membership of *x* and *y*, respectively, AND, OR and NOT operators are defined as:

$$x \text{ AND } y = \min(\mu(x), \mu(y))$$
$$x \text{ OR } y = \max(\mu(x), \mu(y))$$
$$\text{NOT } x = (1 - \mu(x))$$

For more on fuzzy logic, interested readers are referred to [9],[10].

3. The proposed fuzzy-based program anomaly detection scheme

3.1. The proposed fuzzy-based detection scheme

Fig. 1 shows the proposed fuzzy-based detection scheme which is developed in two stages: (a) training stage and (b) testing stage. In the training stage, the detection model is constructed from the training data which is normal traces of system calls of a program. In the testing stage, the constructed detection model is used to evaluate test traces of system calls in order to find possible intrusions. The two stages of the proposed scheme can be described as follows:

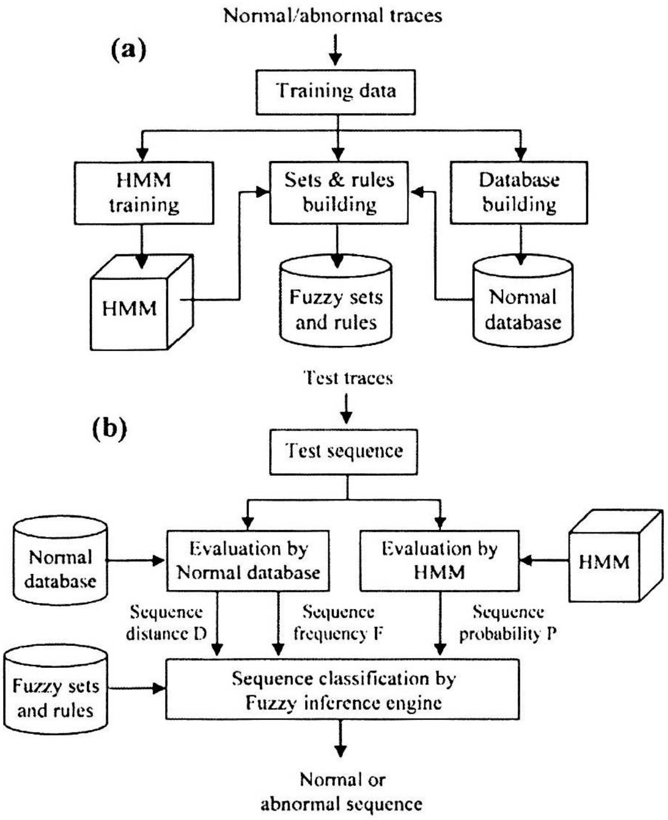

Fig. 1. The proposed fuzzy-based detection scheme: (a) Training stage and (b) Testing stage.

- Training stage: a normal database, an HMM model and fuzzy sets are built from training data.
  - Normal database: The database is an ordered list of all unique short sequences of system calls found in training data. The database is created from normal traces of system calls using the method given in [4]. Each short sequence in the normal database has k system calls. In addition, the occurrence frequency of each short sequence in training data is also recorded in the normal database.
  - HMM model: The HMM model is trained using normal traces of system calls, based on the HMM incremental training scheme, given in [11].
  - The fuzzy sets are created, as discussed in Section 0.0.
- Testing stage: First, short sequences are formed from test traces of system calls using the sliding window method [4]. The sequence length is k system calls. Then, each short sequence is evaluated in two steps as follows:
  - Evaluation of the short sequence by the normal database and by the HMM model: In this step, the normal database and the HMM model are used to compute the input parameters for the fuzzy inference engine.
  - Classification of the test sequence by the fuzzy inference engine: In this step, the fuzzy inference engine applies the fuzzy sets and rules to interpret the input parameters in order to produce the output which is the status of the short sequence: normal or abnormal.

3.2. Fuzzy inference for sequence classification

As discussed in Section III.A, the fuzzy inference engine is used to evaluate each short sequence to find anomalies by combining multiple sequence parameters. Fig. 2 shows the fuzzy inference engine for the classification of short sequences of system calls. The engine accepts the sequence's parameters as the input, and then applies the fuzzy sets and rules to produce the sequence's status as the output. The sequence parameters include the sequence probability P generated by the HMM model, and the sequence distance D and frequency F produced by the normal database.

Creation of fuzzy sets and rules

As shown in Fig. 2, fuzzy sets and rules are used by the fuzzy inference engine to interpret the input and generate the output.

Creation of fuzzy sets

We empirically created fuzzy sets to represent the space of each sequence parameter as follows:

- Four fuzzy sets, namely *Very Low*, *Low*, *High* and *Very High*, are created for the sequence probability *P*, to represent very low, low, high and very high sequence probabilities, respectively.
- Four fuzzy sets, namely *Zero*, *Small*, *Medium* and *Large*, are created to represent zero (for matched sequences), small, medium and large sequence distances, respectively.
- Three fuzzy sets, namely *Low*, *Medium* and *High*, are created to represent low, medium and high sequence frequencies, respectively.
- Two fuzzy sets, namely *Normal* and *Abnormal*, are created to represent the space of the output sequence anomaly score parameter. The anomaly score fuzzy sets are used in the defuzzification process to convert the output fuzzy set to the actual anomaly score of the sequence.

Creation of fuzzy rules

Since the input sequence parameters of the fuzzy rules, which include probability *P*, distance *D* and frequency *F*, are generated by the HMM model and the normal database, our fuzzy rules inherit the assumptions used by the normal database and the HMM-based detection schemes. These assumptions are as follows:

- A sequence, which is produced with a likely probability by the HMM model, is considered to be normal.
- A sequence, which is produced with an unlikely probability by the HMM model, is considered to be abnormal.
- A mismatched sequence is more suspicious than a matched sequence. The larger the distance between a test sequence and normal sequences is the more likely the test sequence is abnormal.
- A matched sequence with a low occurrence frequency is more suspicious than a sequence with high occurrence frequency.

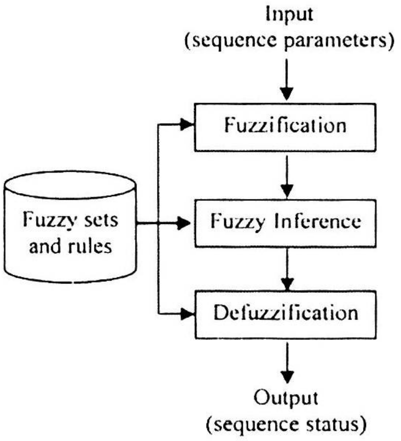

Fig. 2. The fuzzy inference engine for the classification of short sequences of system calls.

Based on the above assumptions, we manually devised a set of 17 fuzzy rules for the sequence classification. An example of such a rule reads "IF probability IS *Low* AND distance IS *Zero* AND frequency IS *Low* THEN the test sequence IS *abnormal*". We do not present all rules in this paper due to space limitation.

Sequence classification using fuzzy reasoning

The fuzzy reasoning process, as shown in Fig. 2, evaluates each sequence of system calls in three phases: fuzzification, fuzzy inference and defuzzification. Fuzzification is the process of transforming crisp input values into linguistic values which usually are fuzzy sets. There are two tasks performed in the fuzzification process: input values are converted into linguistic values which are represented by fuzzy sets, and membership functions are applied to compute the degree of truth for each matched fuzzy set.

Defuzzification is the process of transforming the fuzzy value into a crisp value. In our fuzzy inference engine, the output anomaly score fuzzy set is defuzzified to produce the sequence's anomaly score. There are many defuzzification techniques available, such as centroid method, max-membership method and weighted average method. We used the max-membership method to compute the crisp output from the output fuzzy set.

In the fuzzy inference phase, all rules in the fuzzy rule-base are applied to input parameters in order to produce an output. For each rule, first, each premise is evaluated, and then all premises connected by an *AND* are combined by taking the smallest value of their degree of membership as the combination value of rule's truth value. The final output fuzzy set of the fuzzy rule-base is the *OR* combination of results of all individual rules that fire. It is noted that the truth value of a rule that fires is non-zero. The output fuzzy set is defuzzified to produce a crisp output value.

4. Experimental results and discussion

4.1. Data set

We used *sendmail* traces of system calls collected in a synthetic environment, as given in [12]. The format of system call traces and the data collection procedures were discussed in [4]. The data sets include:

- Normal traces are those collected during the program's normal activity. Normal traces of the *sendmail* program include 2 traces with the total of 1,595,612 system calls.
- Abnormal traces are those that come from a program's abnormal runs generated by known intrusions. In the case of *sendmail* abnormal traces, they consist of 1 trace of *sm5x* intrusion, 1 trace of *sm565a* intrusion, 2 traces of *syslog-local*, and 2 traces of *syslog-remote* intrusion.

4.2. Experimental design

In order to measure the detection rate and the false alarm rate of our fuzzy-based detection model, our experiments were designed as follows:

- *Measurement of the false positive rate*: In this test, we use the proposed fuzzy-based detection scheme to classify normal traces of system calls, which were not used in the construction of the normal database, the HMM model and the fuzzy sets. Since the normal traces do not contain any intrusions, any reported alarms are considered false positives. This experiment was set up as follows:
  - Select first 1,000,000 system calls of *sendmail* normal traces as the full training set.
  - Form 4 training sets which account for 30%, 50%, 80% and 100% of the size of the full training set.
  - Construct normal databases and HMM models from these training sets. The chosen values for the sequence length are k = 5, 11 and 15 system calls.
  - For each training set and on each selected sequence length, construct membership functions to fuzzy sets of three sequence parameters, as discussed in Section III.B.
  - Select three test traces, each trace of 50,000 system calls from the *sendmail* normal traces, which are not used in the training process, to test for false positive alarms of our scheme, the normal database scheme [4] and the two-layer scheme [5]. Reported abnormal short sequences are counted for each test trace.
- *Measurement of anomaly signals and the detection rate*: In this test, we use the proposed fuzzy-based scheme to classify abnormal traces of system calls to find possible intrusions. Since the abnormal traces have been collected from the

program's abnormal runs, generated by known intrusions, reported alarms in this case can be considered true alarms or detected intrusions. This experiment was designed as follows:

- Construct a normal database and an HMM model for *sendmail* program from normal traces of 1,000,000 system calls. We choose the sequence length k = 11 to construct the normal database from normal traces, and to form short sequences from abnormal traces for testing.
- Construct membership functions to fuzzy sets of the three sequence parameters, as discussed in Section 0.
- Use the proposed fuzzy-based detection scheme to evaluate abnormal traces to find abnormal sequences.
- Use temporally local regions to group individual abnormal sequences to measure the anomaly signals. The selected region length is r = 20.

4.3. Experimental results

False positive rate

Table 1 shows the overall false positive rate for three test traces with a total of 150,000 system calls (each trace consisting of 50,000 system calls), as reported by the normal database scheme [4], by the two-layer detection scheme [5] and by the fuzzy-based detection scheme, on different training sets with the sequence length k = 5, 11 and 15. The total number of short sequences in the test traces is dependent on the sequence length and is also given in Table 1.

Table 1. Overall false positive rate of the normal database scheme, the two-layer detection scheme and the fuzzy-based detection scheme with the short sequence length, k = 5, 11 and 15

Training data sets (% of full data set)	Normal database scheme [1] (%)	Two-layer scheme [5] (%)	Fuzzy-based scheme (%)
Sequence length, k = *5*; 3 test traces with the total of 149,988 sequences			
30%	0.131	0.112	0.067
50%	0.099	0.079	0.057
80%	0.094	0.069	0.049
100%	0.094	0.069	0.049
Sequence length, k = *11*; 3 test traces with the total of 149,970 sequences			
30%	0.194	0.170	0.099
50%	0.155	0.115	0.081
80%	0.150	0.107	0.077
100%	0.147	0.107	0.077
Sequence length, k = *15*; 3 test traces with the total of 149,958 sequences			
30%	0.225	0.164	0.107
50%	0.176	0.121	0.091
80%	0.174	0.116	0.085
100%	0.171	0.116	0.085

It can be seen from Table 1 that the false positive rate of the fuzzy-based detection scheme is much lower than that of the normal database scheme [4]. For example, the fuzzy-based detection scheme produced 48.23%, 48.89% and 50.96% fewer false positive alarms than the normal database scheme, for the training set of 80% of full set, with the sequence length k = 5, k = 11 and k = 15, respectively.

It is also noted that there is a significant reduction in the false positive rate of the fuzzy-based detection scheme, compared to that of the two-layer detection scheme [5]. For example, the fuzzy-based detection scheme produced 29.81%, 28.13% and 26.44% fewer false positive alarms than the two-layer detection scheme for the training set of 80% of full set, with sequence length k = 5, k = 11 and k = 15, respectively (refers to Table 1).

Fig. 3 shows the dependence of the false positive rate on the size of the training sets with the sequence length k = 11. When the size of the training set increases, the false positive rate of the normal database scheme [4] and the two-layer scheme [5] decreases considerably, especially from the training set of 30% of the full set to the set of 50% of the full set. Since the fuzzy-based scheme has already achieved a low false positive rate at the set of 30% of the full set, there is only a small reduction in the false positive rate when the size of the training set increases.

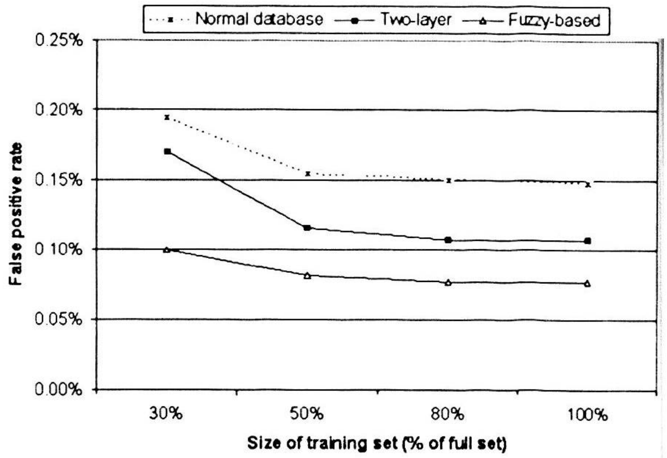

Fig. 3. The relationship between the size of training sets and the false positive rate with k = 11.

Anomaly signals and the detection rate

Table 2 shows a summary of the detection results of the two-layer scheme and the fuzzy-based scheme for some abnormal traces which were generated by some known intrusions. The detection performance results of the normal database scheme are taken from Table 3 of [4]. Similar to the anomaly signal measurement method described in [5], we measure anomaly signals based on temporally local regions. The anomaly score A of a region is computed as the ratio of the number of detected abnormal short sequences in the region to the length of the region r. The average of anomaly scores is computed over abnormal regions that have the anomaly score A greater than the region score threshold $\hat{A}$ ($A \geq \hat{A}$), where $\hat{A}$ = 40.0%.

Table 2. Detection results produced by the normal database scheme, by the two-layer scheme and by the fuzzy-based scheme for some abnormal traces

Name of test abnormal traces	% detected abnormal sequences by [1]	% of detected abnormal regions		Average of scores of abnormal regions	
		Two layer (%)	Fuzzy-based (%)	Two layer (%)	Fuzzy-based (%)
sm565a	0.60	38.46	76.92	68.00	88.00
sm5x	2.70	31.58	67.11	60.42	72.55
syslog-local No.1	5.10	12.00	60.00	73.33	84.67
syslog-local No.2	1.70	16.67	60.26	71.54	86.49
syslog-remote No.1	4.00	28.26	67.39	72.31	86.53
syslog-remote No.2	5.30	24.68	61.04	74.74	83.40

It can be seen from Table 2 that the fuzzy-based scheme produced significantly better detection results than the two-layer scheme [5], in terms of the number of detected abnormal regions and the generated anomaly signal level. For the "*sm5x*" intrusion trace, the rates of detected abnormal regions are 31.58% and 67.11% by the two-layer scheme and fuzzy-based scheme, respectively. Also for this test trace, the fuzzy-based scheme generated the average anomaly score of 72.55%, compared to the average anomaly score of 60.42% produced by the two-layer scheme.

Fig. 4 and Fig. 5 show the anomaly signals produced by the two-layer scheme [5] and the fuzzy-based scheme for *syslog-local* No. 1 and *syslog-remote* No. 1 abnormal traces, respectively, with the sequence length $k = 11$. It is noted that anomaly signals are measured based on temporally local regions for both schemes. It can be seen from these figures that the proposed fuzzy-based scheme generated much stronger and clearer anomaly signals than the two-layer scheme [5].

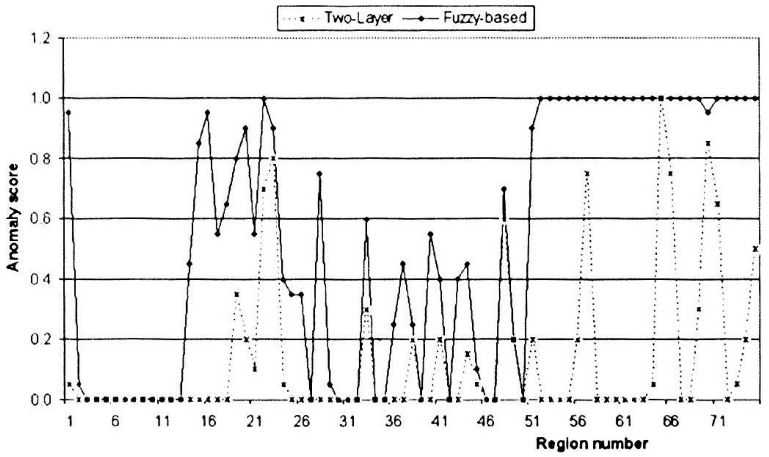

Fig. 4. Anomaly signal generated for *syslog-local* abnormal trace No. 1 by the two-layer and fuzzy-based schemes.

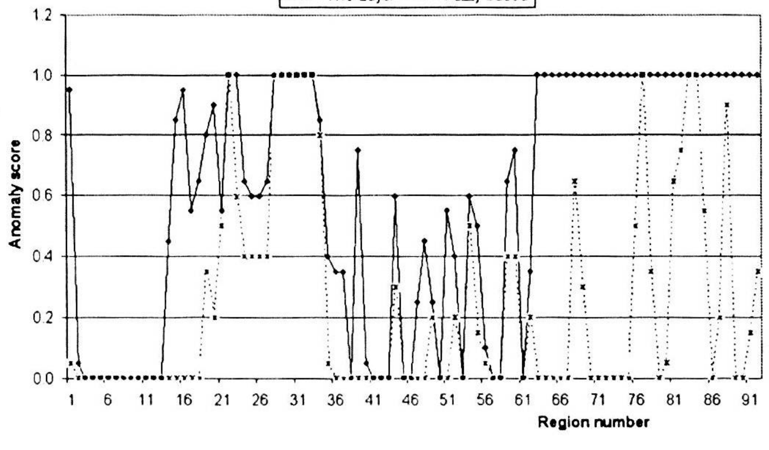

Fig. 5. Anomaly signal generated for *syslog-remote* abnormal trace No. 1 by the two-layer and fuzzy-based schemes.

4.4. Discussion

The proposed fuzzy-based detection scheme generated much fewer false positive alarms than the normal database scheme [4], as shown in Table 1. For example, the false positive rate of the normal database scheme is 0.174%, as opposed to 0.085% of the proposed scheme, or a reduction of 50.96%, when using the training set of 50% of the full set, with $k = 15$.

It is also noted that the proposed detection scheme achieved a much lower false positive rate on small-size training sets than the normal database scheme [4]. On the training set of 30% of the full set, the false positive rate of proposed detection model is lower than that of the normal database scheme on the full training set. This means that the proposed detection model requires significant less training data to achieve a better level of false positive rates than the normal database scheme [4].

According to experimental results presented in Table 2, our detection scheme correctly detected all intrusions embedded in all abnormal traces tested. In contrast, the normal database scheme [4] missed the *sm565a* intrusion, with only 0.6% of abnormal sequences detected. This scheme [4] possibly also missed the *syslog-local* intrusion, embedded in *syslog-local* trace No. 2, with just 1.7% of abnormal sequences detected.

The fuzzy inference engine plays an important role in the reduction of false positive alarms and the increase of the detection rate. The fuzzy inference engine that incorporates multiple sequence information, generated by the normal database and by the HMM model, accurately classifies the sequence. This reduces the false alarms and increases the detection rate.

5. Conclusion

In this paper, we presented a fuzzy-based scheme for program anomaly intrusion detection using system calls. The proposed fuzzy-based detection scheme is based on the two-layer detection scheme [5] and the normal database detection scheme [4]. Instead of using crisp conditions, or fixed thresholds, fuzzy sets are created to represent the space of each sequence parameters. A set of fuzzy rules is created to combine multiple sequence parameters in order to determine the sequence status, through a fuzzy reasoning process. Experimental results showed that the proposed detection scheme reduced false positive alarms by about 48% and 28%, compared to the normal database scheme [4] and the two-layer scheme [5], respectively. The proposed detection scheme also generated much stronger anomaly signals, compared to the normal database scheme [4] and the two-layer scheme [5].

References

[1] J.E. Dickerson, J. Juslin, O. Koukousoula, J.A. Dickerson, "Fuzzy Intrusion Detection," in the *Proceedings of North American Fuzzy Information Processing Society*, Vancouver, Canada, July 25, (2001) 1506.

[2] J. Gòmez, D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection," in the *Third Annual IEEE Workshop on Information Assurance*, New Orleans, Louisiana, USA, June 17-19, 2002.

[3] J. Gòmez, F. Gonzàlez, D. Dasgupta, "An Immuno-Fuzzy Approach to Anomaly Detection," in the *IEEE International Conference on Fuzzy Systems*, Vol.2, May 25-28 (2003) 1219.

[4] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, "A sense of self for Unix processes," in the *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy*, 1996.

[5] X.D. Hoang, J. Hu, P. Bertok, "A multi-layer model for anomaly intrusion detection using program sequences of system calls," in *IEEE International*

Conference on Network – IEEE ICON2003, Sydney, Australia, September (2003) 531.

[6] G. Florez, S.M. Bridges, R.B. Vaughn, "An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection," in the *2002 Annual Meeting of the North American on Fuzzy Information Processing Society*, June 27-29 (2002) 457.

[7] S. Cho, "Incorporating Soft Computing Techniques into a Probabilistic Intrusion Detection System," in *IEEE transactions on systems, man, and cybernetics*, Vol.32, No.2, May 2002.

[8] J. Luo, S. M. Bridges, R.B. Vaughn, "Fuzzy Frequent Episodes for Real-time Intrusion-Detection," in *IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, December 2-5 (2001).

[9] L.A. Zadeh, "Fuzzy sets," in the *Information and Control Journal*, Vol.8 (1965) 338.

[10] E. Cox, "Fuzzy fundamentals," in *IEEE Spectrum*, Vol.29, No.10 October (1992) 58.

[11] X.D. Hoang, J. Hu, "An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls," in *IEEE International Conference on Network-IEEE ICON2004*, Vol.2, Singapore, November (2004) 470.

[12] University of New Mexico's Computer Immune Systems Project web page: http://www.cs.unm.edu/~immsec/systemcalls.htm.