# Rule-based Integration of Multiple Measure-models for Effective Intrusion Detection*

Sang-Jun Han
Dept. of Computer Science
Yonsei University, Korea
sjhan@cs.yonsei.ac.kr

Sung-Bae Cho
Dept. of Computer Science
Yonsei University, Korea
sbcho@cs.yonsei.ac.kr

**Abstract** – *As the reliance on computers gets higher, security of critical computers becomes more important thing. An IDS detects unauthorized usage and misuses by a local user as well as modification of important data by analyzing system calls, system logs, activation time, and network packets Conventional IDSs based on anomaly detection employ several artificial intelligence techniques to model normal behavior. However, they have the shortcoming that there are undetectable intrusions according to types for each measure and modeling method because each intrusion type makes anomalies at individual measure. In this paper, we propose a multiple-measure intrusion detection method to remedy this drawback of conventional anomaly detector. We measure normal behavior by system calls, resource usage and file access events and build up profiles for normal behavior with hidden Markov model, statistical method and rule-base method, which are integrated with a rule-based approach. Experimental results with real data clearly demonstrate the effectiveness of the proposed method that has significantly low false-positive error rate against various types of intrusion.*

**Keywords:** intrusion detection systems, anomaly detection, computer security

## 1 Introduction

Due to worldwide proliferation and rapid progress in networking, faster and more diversified services have become in reality. As the reliance on computers gets higher, security of critical computers becomes more important thing. An IDS detects attacks exploiting illegal uses or misuses and modification of important data by analyzing system calls, system logs, activation time, and network packets of each operating system [18]. There are two general approaches to detect intrusions: misuse detection and anomaly detection.

Misuse detection has the advantage that known attacks can be detected reliably with low false-positive error and economically. The shortcoming is that it cannot detect unknown attacks. Anomaly detection is better than misuse detection in terms of detecting novel attacks and its low false-negative rate. However, it suffers from high false-positive error rate because unseen normal behaviors are considered as attacks. One type of audit record is inadequate for monitoring the whole behavior and modeling method can models all perspectives of malcious behavior. Thus, the detectable types of intrusion are limited according to the measures and modeling methods used.

In this paper, to overcome drawbacks of the conventional anomaly detection techniques, we propose a novel detection technique that combines multiple measures and models. First of all, we develop four appropriate detection methods that use system call events, resource usage of process, file access events as the measure of normal behavior, with the three modeling methods. Next, we combine these detectors with rule-based approach. The proposed detection method is expected better performance because it can model normal behaviors from various perspectives.

The rest of this paper is organized as follows. In Section 2, we give a brief overview of the related works. The overall design and detailed description of the proposed methods are presented in Section 3. Experimental results are shown in Section 4.

## 2 Related Works

Various techniques are used for anomaly-based intrusion detection. Expert system, statistics, artificial neural network, rule learning and hidden Markov model (HMM) are widely used for modeling normal behaviors. The representative studies on anomaly detection are summarized in Table 1.

Statistics is the most widely used technique, which defines normal behavior by collecting data relating to the behavior of legitimate users over a period of time [4]. The representative IDS based on statistics is NIDES (Next-generation Intrusion Detection Expert Systems),

*0-7803-7952-7/03/$17.00 © 2003 IEEE.

Table 1: The representative studies on intrusion detection
ES: Expert System, NN: Neural Network, ST: Statistics, HMM: Hidden Markov Model, RL: Rule Leaning

| Organization | Name | Period | ES | NN | ST | HMM | RL |
|---|---|---|---|---|---|---|---|
| AT&T | ComputerWatch [9] | 1987-1990 | X | | | | |
| UCDavis | NSM [10] | 1989-1995 | | | X | | |
| | GrIDS [17] | 1995- | X | | | | |
| SRI International | IDES [14] | 1983-1992 | | | X | | |
| | NIDES [2] | 1992-1995 | | | X | | |
| | EMERALD [15] | 1996- | | | X | | |
| CS Telecom | Hyperview [8] | 1990-1995 | X | X | | | |
| Univ. of New Mexico | C. Wrannder et. al [19] | 1995 | | | X | X | |
| Yonsei Univ. | Park and Cho [6] | 1999- | | | | X | |
| MIT Lincoln Lab. | R. Lippmann et. al [13] | 1999- | X | | | | |
| Colombia Univ. | MADAM ID [12] | 1998- | | | | | X |

which measures the similarity between a subject's long-term behavior and short term behavior for intrusion detection [2]. The detection rate is high because it can use various types of audit data and detect intrusion based on the previous experimental data. The shortcoming is that it is not sensitive to some behavior and detectable types of intrusion are limited.

Hyperview of CS Telecom is a representative IDS using neural network [8]. It consists of 2 modules: neural network and expert system. The neural network in Hyperview uses temporal sequence of audit data as inputs and 60 types of audit data: CPU usage, memory usage, etc. R. Lippmann et al. have applied neural network to keyword-based detection system [13]. They have used keyword counts from transcripts for telnet session as inputs of neural network. While the artificial neural network has some similarity to statistical techniques, it has advantage of easier representation of nonlinear weight parameters between input and output. The defects of neural networks are that its computational load is very heavy and it is difficult to interpret the relationship between inputs and outputs.

An HMM is useful technique because it is a good for modeling system call sequences. The representative study is the technique proposed by C. Warender of New Mexico University [19]. It uses system call audit trails to measure normal behaviors. While HMM produces better performance in modeling system call events than other methods, it requires very long time for modeling normal behaviors. The solution for this problem might be to improve the performance of computer system or to reduce the training data. The technique that reduces audit trails by filtering audit trails from abstracted information around the change of privilege flows can save the computational costs significantly while maintaining good performance [6].

RIPPER [7], a rule learning tool, has been used for automatic construction of detection models. RIPPER is applied to labeled training data sets and automatically mines the patterns of intrusion in MADAM ID [12]. Though it is good tool for discovering patterns, anomaly detection technique is required for the detection of novel intrusions.

# 3 Proposed Method

Among the intrusion types that frequently occur, buffer overflow, S/W security error, configuration error and denial of service attacks are prevalent on host. According to the hacking trends in May, June, and July 2002 reported by CERTCC the majority of attacks occurred is buffer overflow. Recently massive access to internet raises the issue of denial of service attack. This paper focuses on the two intrusion types to develop sophisticated detection method.

We use common measures for host-based detection system: system call event, file system information and resource usage of process. Though there are several methods for modeling each measure, we select modeling methods appropriate to each measure considering the relationship between the characteristics of intrusion traces and the power of modeling methods. However, because each intrusion type leaves anomalies at individual measure, there are undetectable intrusion types in each measure and modeling method. To overcome this drawback, it is necessary to remedy shortcomings of each method through integrating the results of several methods. We construct the rules to combine multiple measure models and detect intrusions with them. The general architecture of the proposed method can be seen in Fig 1.

## 3.1 HMM with System Call Events

Sun Microsystem's Basic Security Module (BSM) which is auditing facility for Solaris provides an adequate representation of the behavior of the program be-

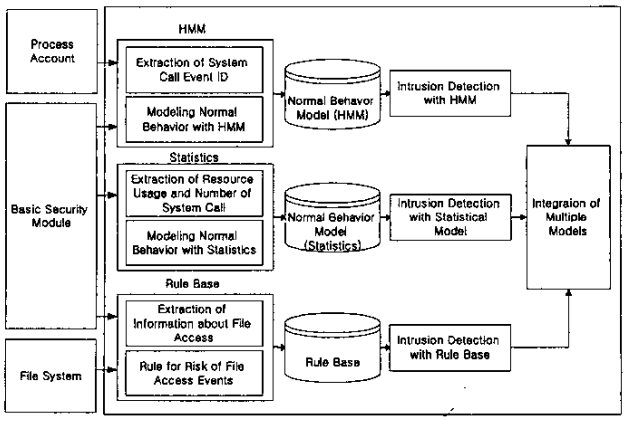


Figure 1: Overall of the proposed method

cause any privileged activities that might be generated by a program are captured by BSM. Usually audit trail from BSM consists of several measures. The victim process of a certain attack generates system call events that are significantly different from events generated at normal situation. Thus, we can detect intrusions effectively by building the model of system call events from normal situation and noting significant deviation from the model. In this paper, for modeling system call event sequences, we use HMM that is widely used for speech recognition because it is very useful for modeling sequence information. HMM can be successfully applied to modeling system call event sequences [5].

Intrusion detection with HMM consists of two phases: normal behavior modeling and anomaly detection. The first phase is determining HMM parameters to maximize the probability $Pr(O \mid \lambda)$. Because no analytic solution is known for it, an iterative method called Baum-Welch reestimation is used [16]. Anomaly detection, the second phase, matches current behavior against the normal behavior model, and calculates the probability with which it is generated out of the model. Forward-backward procedure is used for this purpose. The probability is used to decide whether the current behavior is normal or not with a threshold.

## 3.2 Statistics with Resource Usage and System Call Events

Most of Unix-based operating systems serve Process Account (PACCT) as audit data. It provides the resource usage of processes: CPU time, memory usage, I/O usage, etc. Denial of service attack sharply raises the resource usage of victim process. This type of attack can be detected by noting processes that show unusual resource usage compared with normal behavior using PACCT audit data.

PACCT audit data is useful to detect DoS attacks. Unfortunately, we cannot detect attack type that targets resource not recorded to PACCT. For example, attack that consumes process table leaves no anomalies in PACCT. However, it unusually generates large amount of system call events. In this case, we can detect that by noting process that generates unusual number of system call events.

In this paper, we use statistical technique to model normal resource usage and the number of system call events. This statistical approach is a modified method that was used in NIDES. For each audit record generated by user, we generate a single test statistic value denoted by $T$ that summarizes the degree of abnormality in the user's behavior in the near past. Large values of $T$ indicate abnormal behavior, and values close to zero indicate normal behavior. In the case of PACCT, the $T$ statistic is a summary judgment of the abnormality of measures in PACCT. We denote the individual abnormality of measures by $S$, each of which measures the degree of abnormality of behavior with respect to specific features such as CPU time, memory usage and I/O usage. $T$ statistic has been set equal to the weighted sum of the $S$ statistics as follows:

$$T = a_1 s_1 + a_2 s_2 + \ldots + a_n s_n \qquad (1)$$

where $s_n$ is the $S$ score of each measure and $a_n$ is the weight to each measure.

Each $S$ statistics is derived from a corresponding statistic called $Q$. In fact, each $S$ statistic is a normalizing transformation of the $Q$ statistic so that the degree of abnormality for different types of features can be added on a comparable basis. The value of $Q$ corresponding to the current audit record represents the number of audit records that are arrived in the recent past. In order to transform $Q$ to $S$, we have built a normal historical profile of all previous values of $Q$ and compare the current value of $Q$ with this normal profile to determine if the current value is anomalous.

Small value of $Q$ indicates a recent past that is similar

to historical behavior, and large value of $Q$ indicates a recent past that is not similar to historical behavior. Given $k$ is an index of appropriate audit records, $t_k$ is the time that elapses between the $k$th and most recent audit records, $r$ is the decay rate and $D_k$ is the change that occurs between the $(k+1)$st and $k$th appropriate audit records, $Q$ is defined as follows [11]:

$$Q \sum_{k \geq 1} D_k \times 2^{-rt_k} \quad (2)$$

### 3.3 Rule-base with File Access Events

Generally, attacks tend to access files of which the security level is high. For example, it executes file which has a SETUID privilege to acquire root privilege or attempts to read files owned by root to destroy the computer system or obtain a critical data. Owing to this tendency of abnormal behavior, it is adequately suspicious behavior for ordinary user to access files which have higher security level.

The best-known security policy related with file access is the Bell-LaPadula (BLP) model which has been formulated by Bell and LaPadula [3]. In an access to some information, there are three primary elements: subject, object and access attribute. The subject corresponds to user or program, the object corresponds to files and the access attribute corresponds to the kind of access mode: read, write, execute, etc. In order to determine if a specific access mode is allowed, the BLP model compares the clearance of a subject with the classification of the object and determination is made as to whether the subject is authorized for the specific access mode [1]. The BLP model includes several rules referenced frequently. One of them is no read up rule, which allows access if the current level of the subject dominates the level of the object. It prevents user from accessing information for which they are not allowed to access.

In this paper, we audit file access and evaluate the risk of access event by analyzing the content of that. The security levels of subject and object are divided into 4 categories: root, administrator, ordinary user and guests. The risk of access event is evaluated to one of 21 levels according to the difference of security levels between subject and object. The access event of which the difference between two levels is bigger has higher risk. If the level of object is lower than that of subject the risk is not increased due to the security level difference. If the access event contains file that allows more operations, it has higher risk. For example, when the file mode is 755 the access event is riskier than that of 744.

### 3.4 Integration of Multiple Models

In this paper, we have proposed an intrusion detection technique that integrates detection methods in order to increase detectable attack types and reduce false-positive error rate. We use rule-based approach for integration. The rules are made experimentally considering the relationship between the characteristics of intrusion traces and the capability of modeling methods. The rules used for the integration of multiple detectors are as follows:

```
IF ((HMM/System Call > T1)
              AND (Rule-base/File Access < T4))
  THEN Buffer overflow attack
IF ((Statistics/Resource Usage > T2)
              AND (Statistics/System Call < T3))
  THEN Consumption of Process Table (DoS attack)
IF ((Statistics/System Call > T3)
              AND (Statistics/Resource Usage < T2))
  THEN Consumption of Disk or Memory (DoS attack)
```

(T1: Threshold of HMM/System Call, T2: Threshold of Statistics/Resource Usage, T3: Threshold of Statistics/System Call and T4: Threshold of Rule-base/File Access.)

The first rule integrates HMM with system call event and rule-based method with file access event. This rule prevents HMM from considering unseen normal behaviors as attacks. We can reduce false-positive error rate with this rule because most of buffer overflow attacks attempt to access file that is owned by root or has SETUID privilege.

The second rule integrates statistics with resource usage and statistics with system call event. Statistics with system call event has difficulty to consider normal behavior that uses many system calls. An attack that results in denial of service by excessive amount of system call events shows low resource usage while it generates a huge amount of events. This rule can reduce false-positives by considering as attack the behavior that generates unusual number of system call events although resource usage is normal.

The last rule also integrates statistics with resource usage and statistics with system call event. Statistics with resource usage has difficulty to consider normal behavior that requires much resource. The denial of service attack that consumes resource rapdly shows small number of system call events while the resource usage is very high. This rule can reduce error rate by considering behavior as attack, which shows resource usage although it generates usual number of system call events.

## 4 Experimental Results

We have collected normal behaviors from six graduate students for 2 weeks using Solaris 7 operating system. They have mainly used text editor, compiler and programs of their own writing. Total 13 megabytes (160,448 records) of BSM audit data and 840 kilobytes of PACCT audit data have been collected from 16,470 commands. We also collect audit data that contain labeled attacks for testing in the same operating system. It contains 9 cases of u2r buffer overflow intrusion and 4 cases of denial of services. The attacks used in our experiments are as shown in Table 2.

Table 2: The attacks used in our experiments

| Attack Type | Attack Name |
|---|---|
| Buffer Overflow | kcms_configure vulnerability<br>lpset -r vulnerability<br>xlock vulnerability |
| Denial of Service | Consumption of Disk<br>Consumption of Process Table<br>Consumption of Memory |

We used detection rate, false-positive error rate and discriminability to evaluate the performance of IDS. The discriminability is used for measuring numerically the performance of a detection method and efficiency to compare integration methods with others. It is a measure of the average intensity difference perceived by an observer between samples including the signal and samples not including a signal. It has higher value when detection rate is high and false-positive error rate is low. Generally, it has been noted as $d'$ and defined as follows:

$$d' = z(H) - z(F) \quad (3)$$

where z is the inverse of the normal distribution function, $H$ is detection rate and $F$ is false-positive error rate. ROC (Receiver Operating Characteristics) curve is used to visualize the performance of detection method. A desirable intrusion detection system must show a high detection rate at low false-positive error rate. In ROC curve, top-left curve is more desirable.

At first, we have conducted experiments without the integration of detection methods in order to identify the characteristics of each method and find optimal parameters of each method. The experimental result of single detection methods does not show good performance owing to the characteristics of measure and modeling methods. We have compared the detection methods with the best parameters using ROC curves. The false-positive error rate has been sharply raised after a certain degree of detection rate as shown in Fig. 2.

The experiment with method that integrates detection methods is conducted using the same data set. We have used the parameters of each method that show the best performance in the preliminary experiments: The number of states is 3 and the length of input sequence is 8 in HMM, the weight ratio to resource usage is 2:1:2 (CPU time: memory usage: I/O usage). The thresholds used are -17.6 to HMM, 6.0 to statistics with resource usage, 30 to statistics with system call event and 10 to rule base with file access event. We have compared false-positive error rate and discriminability of each method at 100% detection rate as shown in Table 3. The discriminability is a measure of the average intensity difference perceived by an observer between samples including a signal and samples not including a signal. It has higher value at high detection rate and

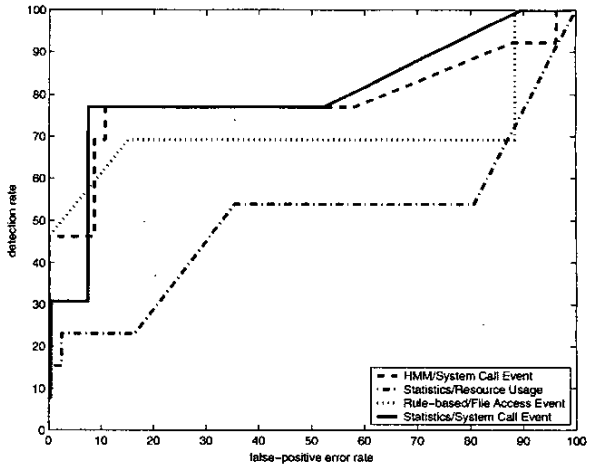


Figure 2: ROC for individual detection methods with best parameters

low false-positive error rate. The result shows the integrated detection method dramatically reduces the false-positive error rate.

# 5 Conclusions

In this paper, we have presented effective modeling methods for three audit data and proposed a novel intrusion detection technique that integrates the detection methods. The proposed method uses multiple measure and modeling methods and integrates the results of individual detection methods with rule-based approach to overcome the drawbacks of the conventional anomaly detection techniques.

We evaluate the performance of each detection method and compare the results with integrated method. The integrated method shows 5.761% false-positive error rate at 100% detection rate whereas the other methods show more than 80% false-positive error rate at the same detection rate. It indicates that we can overcome the drawbacks by integrating several methods.

However, we cannot guarantee that the rules used are optimal because they are made empirically and attack types that are not expressed in the rules cannot be detectable. The machine learning techniques such as decision tree and neural network can be used for the integration to remedy this shortcoming. In the future, it is also needed to develop other measures and modeling methods to increase detectable attack types.

# Acknowledgments

This research was supported by University IT Research Center Project.

Table 3: A comparison of each detection method

| Method | Detection rate | F-P error rate | d' |
|---|---|---|---|
| HMM/System Call Event | 100% | 99.177% | 1.866 |
| Statistics/Resource Usage | 100% | 80.658% | 3.400 |
| Statistics/System Call Event | 100% | 99.177% | 1.866 |
| Rule base/File Access Event | 100% | 88.477% | 3.066 |
| Integration | 100% | 5.761% | 4.665 |

# References


[1] A.G. Amoroso, *Fundamentals of Computer Security Technology*, PTR Prentice Hall, New Jersy, 1994.

[2] D. Anderson, T. F. Lunt, H. Javits, A. Tamaru and A. Valdes, "Detecting unusual program behavior using the statistical components of NIDES," *NIDES Technical Report*, SRI International, May 1995.

[3] D. E. Bell and L. J. LaPadula, "Secure computer systems: Unified exposition and multics interpretation," *Mitre Technical Report ESD-TR-75-306*, Mitre Corporation, March 1976.

[4] E. Biermann, E. Cloete and L. M. Venter, "A comparison of intrusion detection systems," *Computers & Security*, vol 20, no. 8, pp. 676-683, December 2001.

[5] S.-B. Cho, "Incorporating soft computing techniques into a probabilistic intrusion detection system," *IEEE Trans. on Systems, Man and Cybernetics-Part C:Applications and Reviews*, vol. 32, no. 2, pp. 154-160, May 2002.

[6] S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows with hidden Markov model," *Computers & Security*, vol. 22, no. 1, pp. 45-55, 2003.

[7] W.W. Cohen, "Fast effective rule induction," *In Proceedings of the 12th International Conference on Machine Learning*, pp. 115-123, July 1995.

[8] H. Debar, M. Becker and D. Siboni, "A neural network component for an intrusion detection system," *In Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy*1, pp. 240-250, Oakland, CA, May 1992.

[9] C. Dowel and P. Ramstedt, "The computer watch data reduction tool," *In Proceedings of the 13th National Computer Security Conference*, pp. 99-108, Washington DC, USA, October 1990.

[10] T. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J. Wood, and D. Wolber, "A network security monitor," *In Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp. 296-304, Los Alamitos, CA, USA, 1990.

[11] H.S. Javitz and A. Valdes, "The SRI IDES statistical anomaly detector," *NIDES Technical Report*, SRI International, 1991.

[12] W. Lee, S.J. Stolfo and K.W. Mok, "A data mining framework for building intrusion detection models," *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 120-132, 1999.

[13] R. Lippmann and S. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Computer Networks*, vol. 34, no. 4, pp. 594-603, 2000.

[14] T. F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, and P. G. Neuman, "A real-time intrusion-detection expert system (IDES)," *Technical Report Project 6784*, CSL, SRI International, Computer Science Laboratory, SRI International, February 1992.

[15] P. A. Porras and P. G. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," *In Proceedings of the 20th National Information Systems Security Conference*, pp. 353-365, Baltimore, Maryland, USA, October 1997.

[16] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[17] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K Levitt, C. Wee, R. Yip, and D. Zerkle, "GrIDS-A graph based intrusion detection system for large networks," *In Proceedings of the 19th National Information Systems Security Conference*, vol. 1, pp. 361-370, October, 1996.

[18] H.S. Vaccaro and G.E. Liepins, "Detection of anomalous computer session activity," *In Proceedings of IEEE Symposium on Research in Security and Privacy*, pp. 280-289, 1989.

[19] C. Warrender, S. Forrest and B. Pearlmutter, "Detecting intrusion using calls: Alternative data models," *In Proceedings of IEEE Symposium on Security and Privacy*, pp. 133-145, May 1999.