네트워크 패킷에 대한 연관 마이닝 기법을 적용한 네트워크 비정상 행위 탐지[†]

(Network Anomaly Detection using Association Rule Mining in Network Packets)

오 상 현*, 장 중 혁**
(Sang-Hyun Oh, Joong-Hyuk Chang)

요 약 컴퓨터를 통해서 들어오는 다양한 형태의 침입을 효과적으로 탐지하기 위해서 이전에는 오용탐지 기법이 주로 이용되어 왔다. 오용탐지 기법은 이전에 알려지지 않은 침입 방법들을 효과적으로 탐지할 수 있기 때문이다. 하지만, 해당 기법에서는 정상적인 네트워크 접속 형태가 몇 가지 패턴으로 고정되어 있다고 가정한다. 이러한 이유 때문에 새로운 정상적인 네트워크 연결이 비정상행위로 탐지되기도 한다. 본 논문에서는 연관 마이닝 기법을 활용한 침입 탐지 방법을 제안한다. 논문에서 제안되는 방법은 패킷내 마이닝 단계와 패킷간 마이닝 두가지 단계로 구성된다. 제안된 방법의 성능은 대표적인 네트워크 침입 탐지 방법인 JAM과의 비교 실험을 통하여 평가하였다.

핵심주제어 : 비정상행위 탐지, 네트워크 오용, 연관 규칙, 연관규칙 마이닝

Abstract In previous work, anomaly-based intrusion detection techniques have been widely used to effectively detect various intrusions into a computer. This is because the anomaly-based detection techniques can effectively handle previously unknown intrusion methods. However, most of the previous work assumed that the normal network connections are fixed. For this reason, a new network connection may be regarded as an anomalous event. This paper proposes a new anomaly detection method based on an association-mining algorithm. The proposed method is composed of two phases: intra-packet association mining and inter-packet association mining. The performances of the proposed method are comparatively verified with JAM, which is a conventional representative intrusion detection method.

Key Words : Anomaly detection, Network anomaly, Association rule, Association rule mining

1. Introduction

Due to the advance in computer and communication technologies, damages caused by unexpected intrusions and crimes related to computer systems have been increasing rapidly. Intrusion methods have evolved into more sophisticated forms, and many new intrusion methods have been invented as well. As a result, handling the well-known intrusion methods individually is no longer enough to preserve a target domain's security. To

[†] 본 논문은 '항공선진화사업 R&D/36-2007-C-공항'의 일부지원을 받아 수행되었음.

* 유엔비정보기술

** 대구대학교 컴퓨터 · IT공학부, 교신저자

compensate for this situation, the anomaly detection model has been studied.

For anomaly detection, previous works have concentrated on statistical techniques [1, 2, 3]. To represent the characteristics of an activity in an audit data set, various features can be considered, such as CPU usage, system call frequency, the number of file accesses, and so forth. Depending on the type of activity, different features are related. The typical system of statistical analysis is NIDES [2], developed in SRI. In NIDES, the term "measure" is used to denote a feature, and the abnormal rate of each measure is examined independently. NIDES models a user's historical behavior in terms of various features, and generates a long-term profile containing a statistical summary for each feature. To detect an anomaly, the information about the user's online activities is summarized into a short-term profile, and then it is compared to the user's long-term profile. If the difference between the two profiles is large enough, the online activities are considered anomalous behavior. The strong point of statistical analysis is that it can generate a concise profile containing only a statistical summary, which can lessen the burden of computational overhead for real-time intrusion monitoring. However, because statistical analysis represents the diverse behavior in normal activities of a user as a statistical summary, it often fails to accurately model the normal behavioral activities when they deviate widely.

This paper proposes a packet-wise anomaly detection method based on association mining. For this purpose, a network's normal patterns for a long-term profile are generated by mining the network packet data set. However, conventional association mining methods [4, 5] cannot accurately model packet-wise network activities. This is because several packets are contained in a network connection. Therefore, in order to accurately model packet-wise normal patterns, not only intra-packet association mining but also inter-packet association mining should be considered. In mining frequent patterns, although an item is generated repeatedly in a transaction, the number of repetitions is not considered. However, this number should be considered to be significant in anomaly intrusion detection, because an anomalous intrusion (or an attack) may be attempted repeatedly in a short time. Based on this observation, in the process of transforming connection logs to a transactional data set, the number of repetitions for each item is considered significant information. The number of repetitions for each item is not used in mining frequent patterns, but is used to generate a target network audit set's profile, and to decide whether a transaction is an anomaly intrusion. As a result, this method can detect an anomaly more effectively than can previous methods.

The rest of this paper is organized as follows. Section 2 presents various modeling techniques for intrusion detection. Section 3 explains a method of mining frequent patterns among network packets. Section 4 describes an anomaly detection method based on the user's profile. Section 5 comparatively analyzes the proposed anomaly detection method's simulation results in order to illustrate its effectiveness. Finally, conclusions are presented in Section 6.

2. Related Work

Anomaly detection models [1, 2, 3, 6, 7, 8, 9] are classified as either statistical analysis [1, 2, 3], predictive pattern generation [6], or a data mining approach [7, 8, 9]. Statistical analysis maintains a user's historical activities as a statistical profile. For a set of activities, the rate of inconsistency with the profile is regarded as its anomaly one. The typical methods of statistical analysis are IDES [1], NIDES [2], and EMERALD [3], developed in SRI. NIDES, which is the improved version of IDES, utilizes a statistical technique for anomaly detection, as well as a rule-based technique for misuse detection. The EMERALD system, which is similar to NIDES, extends the intrusion detection target from a single host to a network environment. Predictive pattern-generation technique assumes that the sequence of events follows a discernible pattern. This approach uses inductively generated time-based rules that characterize normal users' behavioral patterns. These rules are dynamically modified during the learning phase, and eventually only relevant rules remain in the system. Therefore, an anomaly is detected if the observed sequence of events matches the left-hand side of a rule, but the subsequent events deviate significantly from those events predicted by the rule.

For a network-based anomaly detection system, JAM [7, 8] uses frequent-episode mining [9], which is similar to sequence mining data items. It generates the normal usage patterns of a specific node in a network. These patterns are used to build a base-classifier that determines the network node's abnormality. In order to guarantee correct classification, a sufficient amount of normal and abnormal log data should be gathered during a classifier's learning phase. A set of base-classifiers can be used to build a meta-classifier. Because each base-classifier monitors a different node on the network, an intrusion into the network can be detected by a meta-classifier combining the results of its base-classifiers. Due to the nature of frequent episode mining, however, numeric data, such as the size of a network packet, may be modeled inaccurately. This is because each item should be quantized to one of the predefined ranges, in order to represent it as a categorical data item.

In our previous work [10], we proposed an anomaly detection method based on clustering a data stream. In most conventional clustering methods used on data streams, only a given number of clusters are identified. However, since the number of clusters in a data stream is unknown, the quality of their results can be poor.

3. Mining Frequent Patterns among Network Packets

In anomaly intrusion detection via mining frequent patterns, a very important issue is how to define a transaction for analyzing frequent patterns. Our method divides connection logs generated continuously into several groups depending on their *time stamps*, and those in the same group form a transaction. In this approach, an anomaly-intrusion detection operation can be performed by a transaction. Therefore, the smaller a window is, the more frequently an anomaly-intrusion detection operation can be performed. However, a very small window is inefficient for anomaly intrusion detection, because it is almost impossible to get significantly frequent patterns. The size of window determines the time when an anomaly-intrusion detection operation can be performed, and it affects the detection results' usefulness.

In general, a connection log consists of various features, as shown in Table 1. As a simple and general approach, all connection logs are transformed into a single-target transactional data set, with no considerations as to source hosts, destination hosts, etc. This approach transforms connection logs into a transactional data set as follows:

- Connection logs that are generated in the same time window, i.e., those whose time stamps are in the same time window form a transaction.

Table 1. Features

Feature name	Description	Type
Time stamp	Time stamp of the connection	Continuous
Service	Network service on the destination e.g., http, telnet	Discrete
Source host	System ID of the source host	Discrete
Destination host	System ID of the destination host	Discrete
Flag	Connection status, i.e., normal or error	Discrete

- Among essential features in a connection logs, *service*, *source host*, *destination host*, and *flag* are used to define an item, and *time stamp* is used only to determine which transaction the connection log belongs to.
- For a new connection log, if the values of the four essential features (excluding time stamp) are the same as those of a connection log in a transaction where the new connection log belongs, the new connection log is not considered a new item, but the number of repetitions for the corresponding item is increased by one.
- Even if the values of four essential features in two connection logs are the same, if the logs are generated in different time windows, each connection log is considered as a separate item in each of the connection log's corresponding transactions.

An anomaly intrusion (or an attack) is attempted from multiple source hosts or to multiple destination hosts simultaneously. A source-based transformation approach defines as a transaction only the connection logs that are attempted from the same source host. Therefore, the approach may be unable to detect a general anomaly intrusion that is attempted from multiple source hosts simultaneously. A destination-based approach is also unable to detect the anomalous intrusion. However, to monitor a specified host in a network environment that consists of many hosts, a source-based or destination-based transformation can be applied efficiently. In this paper, network connection logs are transformed into a transactional data set by a simple transformation approach, and an anomalous intrusion operation via mining frequent patterns is performed using the transactional data set.

Let D denote a network log data set. For a transaction $T \in D$, when the number of packets contained in T is m, T is represented by $\{p_1, p_2, \cdots, p_m\}$. Also, when the number of items contained in a packet $p_i \in T$ is denoted by n, p_i is represented by $\{d^i_1, d^i_2, ..., d^i_n\}$.

Definition 1. (Partial element) Let T be a transaction contained in a connection log set D. For a packet $q \in T$, if $p \subseteq q$, then packet p is called the partial element in transaction T, i.e., $p \hat{\in} T$. □

Definition 2. (Partial subset) Let p_s and p_t denote the transaction T's elements. Then a packet-set $P=\{p_1, p_2, ..., p_k\}$ is called the partial subset of the transaction T, where, for all p_i and $p_j \in P$, $p_i \subseteq p_s$ and $p_j \subseteq p_t$ $(p_s \neq p_t)$, i.e., $P \hat{\subseteq} T$. □

Definition 1 describes whether a packet p is contained in a transaction T. In other words, if the packet corresponds to any packet in the transaction or its subset, the packet can become the transaction's partial element. Definition 2 describes the similarity between a packet set and a transaction. In other words, if all packets contained in the packet set are partial subsets, for any packet contained in the transaction, the packet set can become the transaction's partial subset. Therefore, the supports of a packet and a packet set can be calculated using Definition 3.

Definition 3. (Support) Let |*E*| denote the total number of transactions in *E*. The supports of a packet *p* and a packet set *P* are calculated as Equations (1) and (2), respectively.

$$\sup(p) = \frac{|\hat{S}|}{|D|}, \text{ where } \hat{S} = \{p \mid p \mathrel{\hat{\in}} T, T \in D\}$$

.... Equation (1)

$$\sup(P) = \frac{|\hat{R}|}{|D|}, \text{ where } \hat{R} = \{P \mid P \mathrel{\hat{\subseteq}} T, T \in D\}$$

.... Equation (2)

Frequent packet-set mining can be performed as follows:

[Step 1] Using the a priori algorithm [4, 5], intra-packet mining is performed. Each frequent packet is transformed by a unique identifier.

[Step 2] The connection logs are rewritten by the identifiers of frequent packets generated in the first step.

[Step 3] Using the rewritten logs, frequent packet sets can be obtained.

4. Anomaly Detection

To detect an anomaly, a normal activity profile is maintained by two elements: a frequent itemset and its item-occurrence vector. The frequent itemset is a profile for representing relationships among network connections in a transaction, while the item-occurrence vector is a profile for representing the number of same connections in the transaction. When the number of frequent itemsets is *n*, let *P* denote a set of frequent itemset profiles, i.e., $P=\{p_1, p_2, ..., p_n\}$, and each frequent itemset profile be composed of an itemset and an item-occurrence vector. Let V_e denote the item-occurrence vector of itemset *e*, represented by Definition 4.

Definition 4. (Item-occurrence vector) When f_a denotes the item occurrence of an item *a* in an itemset, it means that the number of repetitions for the items *a*, for an itemset $e=(a_1, a_2, \ldots, a_n\}$, in a resulting set of frequent itemsets, its *item-occurrence vector* V_e is defined as follows:

$$V_e=(F_{\alpha_1}, F_{\alpha_2}. \cdots, F_{\alpha_n})$$

When *C*(*e*) denote the number of a frequent itemset *e* in a transaction data set *D*, and D_j is the j^{th} transaction containing *e*; then $F_{\alpha_i}(\alpha_i \in e)$, denoting the average item occurrence of item α_i, is found as $\frac{1}{C(e)} \sum_{j=1, e \subseteq D_j}^{C(e)} f_{\alpha_i}$. □

By comparing a profile set to a new online transaction, any anomalous behavior of the new transaction's activities can be identified. The result of this comparison is expressed by both itemset length and item occurrence differences. The itemset length difference is a measure representing the difference between the length of an itemset in a profile and that of the new transaction. Similarly, the item occurrence difference is a measure representing the Euclidian distance between the item-occurrence vector of a profile in the profile set and the new transaction's item-occurrence vector. These differences are examined as follows. To detect an anomaly in a new transaction *T*, a set of frequent itemsets, which are similar to transaction *T*, are searched for in a profile set. In other words, they are considered to find the two differences for transaction *T*. For a new transaction $T=\{\alpha_1, \alpha_2, ..., \alpha_l\}$, let MFI_T

denote a set of maximally frequent itemsets for transaction T, and let $\pi_e(V_T)$ denote the vector of item occurrences commonly contained in itemset e and transaction T. Then the Euclidean distance between itemset e's and item-occurrence vectors in the transaction T is represented by

$$d(V_{e,}\pi_e(V_T))=\sqrt{\sum_{i=1}^{|e|}(F_{a_1}-\overline{f_{a_i}})^2}.$$

The itemset length and item occurrence differences are calculated as follows:

$$length\_diff(MPI_T,T)=1-|e\cap T|\ (e\in MPI_T)$$

$$occurrence\_diff(MPI_T,T)=\frac{1}{|MPI_T|}\sum_{e\in MPI_T}d(V_e,\pi_e(V_T))$$

Ultimately, the overall abnormality for a new transaction T is represented as follows:

$$\begin{aligned}&abnormality(MPI_T,T)\\&\quad=\beta\cdot length\_diff(MPI_T,T)\\&\qquad+(1-\beta)\cdot occurrence\_diff(MPI_T,T)\end{aligned}$$

In the above equation, the effects of the two differences, *length_diff* and *occurrence_diff*, can be controlled by setting a proper weight β.

In order to decide the rate of abnormal behavior in the new transaction T, a set of different abnormality levels can be defined relative to the normal behavior of historical activities. Our method considers two different abnormality levels (*green*, *red*) in order to determine whether activities of a new object are anomalous or not. The green level is safe and the red is a warning. Let $\gamma(\nu,\lambda,\chi)$ denote the statistics of abnormalities until now. ν, λ, and χ represent the total number of objects occurring within a data set: the linear sum of their abnormalities and the square sum of their abnormalities, respectively. Based on statistic γ, the mean of abnormalities Φ and standard deviation Θ can be calculated as follows:

$$\Phi=\lambda/\nu,\Theta=\sqrt{\chi/\nu-\Phi^2}$$

The new object is in the green or red level, as follows:

- Green level:

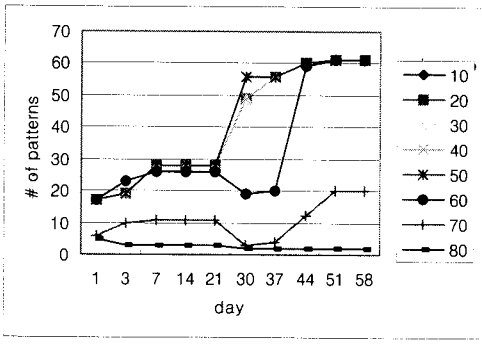
    if $0\le abnormality(\mathrm{MPI}_T,T)\le\Phi+\Theta\times\xi$
- Red level:

    if $\Phi+\Theta\times\xi<abnormality(\mathrm{MPI}_T,T)$.

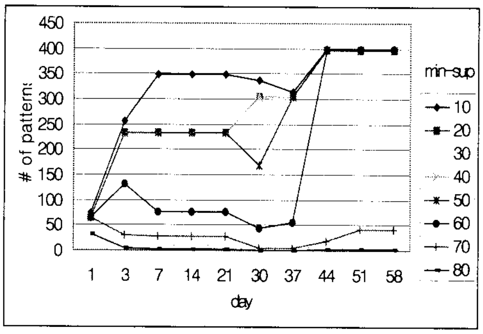
A detection factor ξ is a user-defined parameter which determines how strictly an anomaly of a new object is classified. As factor ξ decreases, a new object is more strictly examined. Given a set of normal objects, its false alarm rate is represented by the ratio of the number of objects within the range of the red level to the total number of normal objects. Similarly, given a set of anomalous objects, its anomaly detection rate is represented by the ratio of the number of objects that are within the range of the red level to the total number of anomalous objects.

5. Experimental Results

The experiments presented in this paper were performed using log data collected in Solaris 2.6 for two months. In order to generate normal user patterns, the connection logs were collected by using *tcpdump* [11]. In all experiments in this section, a predefined time window was set at two seconds. Figure 1 illustrates the number of frequent patterns generated by the proposed method and by JAM. In both of the methods, the 44th day becomes the saturation point of frequent patterns. In this figure, the number of frequent patterns generated by JAM is much larger than by the proposed method, because JAM performs sequential mining in order to obtain normal patterns.



(a) Proposed method



(b) JAM

Figure 1. Number of patterns

Figure 2 illustrates the average abnormalities for normal and abnormal connections, respectively. In this experiment, the minimum support is set to 20%, and the 44th day is selected as the analysis day. As shown in this figure, the proposed method's abnormalities are no different from those of JAM.

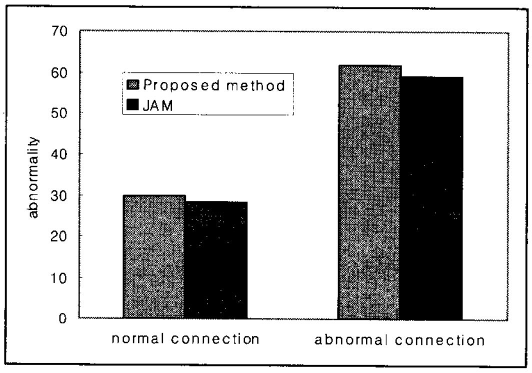


Figure 2. Abnormality

Figure 3 compares the false alarm and detection rates of the proposed method to those of JAM. As shown in this figure, the false alarm rate of the proposed method is similar to that of JAM.

Through the experimental results, we can see that the detection rates of the proposed method and JAM are the same, but the proposed method performs more efficiently than JAM for mining normal patterns.

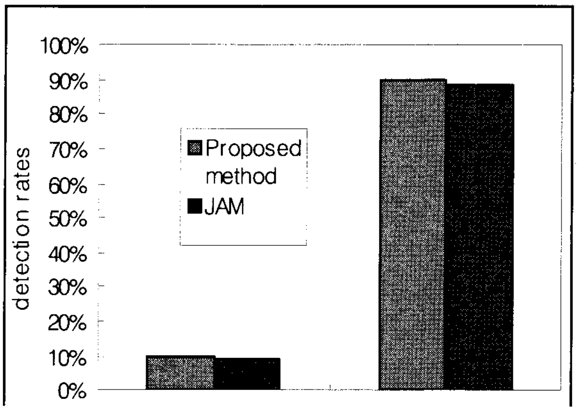


Figure 3. Detection results

6. Conclusions

In previously used statistical methods, network packet information has been treated simply as a rare category, and has been preventing more effective anomaly detection from being realized because the activities of users that are not associated with one another could be managed only as one unit. To resolve these problems, we propose a new anomaly detection method by associating the considerable number of connection logs. The proposed method asserts that to accurately model packet-wise normal patterns, not only intra-packet association mining, but also inter-packet association mining, should be considered. In addition, the number of repetitions for each item is considered for anomaly detection. With the evaluation results using normal users' patterns generated from the proposed scheme, we show that anomalies of a user can be detected more easily and effectively than with JAM.

References

[1] H.S. Javitz and A. Valdes, "The SRI IDES Statistical Anomaly Detector," Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, May 1991.

[2] H.S. Javitz and A. Valdes, "The NIDES Statistical Component Description and Justification," Annual report, SRI International, 333 Ravenwood Avenue, Menlo Park, CA 94025, March 1994.

[3] P.A. Porras and P.G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC, October 1997.

[4] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.

[5] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. of the Int'l Conference on Data Engineering (ICDE), Taipei, Taiwan, March 1995.

[6] H.S. Teng, K. Chen, and S.C. Lu, "Security Audit Trail Analysis Using Inductively Generated Predictive Rules," Proc. of the Sixth Conf. on Artificial Intelligence Applications. pp. 24-29, Piscataway, New Jersey, March 1990.

[7] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. of the 7th USENIX Security Symposium, San Antonio, Texas, January 1998.

[8] S.J. Stolfo, A.L. Prodromidis, S. Tselepis, W. Lee, D. Fan and P.K. Chan, "JAM:Java agents for Meta-Learning over Distributed Databases," Proc. of the workshopon AI Methods in Fraud and Risk Management, 1997.

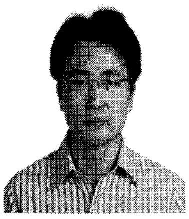
[9] H. Mannila, H. Toivonen, and I. Verkamo, "Discovery of frequent episodes in event sequences," Data Mining and Knowledge Discovery, 1(3), pp.259-289, 1997.

[10] S.-H. Oh, J.-S. Kang, Y.-C. Byun, T. Jeong, and W.-S. Lee, "Anomaly Intrusion Detection Based on Clustering a Data Stream," Proc. of the ISC 2006, pp. 415-426, 2006.

[11] http://www.tcpdump.org/

오 상 현 (Sang-Hyun Oh)

- 1996년 2월 제주대학교 컴퓨터 과학과 (공학사)
- 1998년 2월 연세대학교 컴퓨터 과학과 (공학석사)
- 2004년 8월 연세대학교 컴퓨터 과학과 (공학박사)
- 2007년 9월 ~ 현재 : 유엔비정보기술 기술이사
- 관심분야 : 데이터 마이닝, 침입탐지, 정보보호, 유비쿼터스 데이터 처리



장 중 혁 (Joong-Hyuk Chang)

- 1996년 2월 연세대학교 컴퓨터 과학과 (이학사)
- 1998년 8월 연세대학교 컴퓨터 과학과 (공학석사)
- 2005년 8월 연세대학교 컴퓨터 과학과 (공학박사)
- 2006년 1월 ~ 2008년 7월 : UIUC, Wright State University 박사후 연구원
- 2008년 9월 ~ 현재 : 대구대학교 컴퓨터IT공학부 전임강사
- 관심분야 : 데이터 스트림, 데이터 마이닝, 데이터베이스, 지능형 웹 서비스, USN 환경의 데이터 처리, 바이오인포메틱스

논문접수일 : 2009년 5월 4일
논문수정일 : 2009년 6월 20일
게재확정일 : 2009년 7월 30일