J Comput Virol (2011) 7:133–142
DOI 10.1007/s11416-010-0143-3

ORIGINAL PAPER

An *N*-Gram and STF-IDF model for masquerade detection in a UNIX environment

Dai Geng · Thmohiro Odaka · Jousuke Kuroiwa · Hisakazu Ogura

Received: 9 June 2009 / Accepted: 8 April 2010 / Published online: 13 May 2010

Abstract A masquerader is someone who impersonates another user and operates a computer system with privileged access. Computer security problems caused by masqueraders are serious. Although anomaly detection is considered to be the best way to detect masqueraders, due to the low probability of detection and high error rate, this method is still in the research phase. Thus far, a number of methods, such as the Support Vector Machine (SVM), the Hidden Markov Model (HMM), and the Naïve Bayes (N. Bayes) classifier technique, have been investigated in order to further improve accuracy of detection. In the present paper, a method of integrating Data Mining and Natural Language Processing, namely, the *N*-Gram_Square root Term Frequency-Inverse Document Frequency (*N*-Gram_STF-IDF), is proposed. Using the proposed method, sequences to be detected are segmented via *N*-Gram characteristics, and non-normal users are then detected using a STF-IDF classifier. We perform an experiment using Schonlau and Greenberg data sets and the proposed method and compare the obtained results with results obtained using various other methods.

1 Introduction

Computer security breaches caused by users with unauthorized privileges are serious [1]. In particular, a masquerader, who impersonates another user and operates a computer system with privileged access, is a serious and dangerous intruder [2]. Such a person may be an outsider who has stolen the password of another user or has hacked into the system. The term masquerader can also refer to insiders who hack into a system and abuse their privileges. Once the firewall and the authentication level [3] are compromised, computer security becomes a serious problem. Hence, as a secondary line of defense, namely, online Intrusion Detection Systems (IDS) [4,5], which monitor user behavior, have become necessary.

Intrusion detection systems can be classified as detecting either misuses or anomalies. Misuse detection is based on attack signatures acquired from previous known attacks. Therefore, this type of system cannot detect unknown or new attacks. Anomaly detection is based on the assumption that a significant deviation from past normal behaviors may indicate a possible intrusion. Accordingly, anomaly detection can defend against novel attacks. Since the behavior of a masquerader is supposedly different from that of a real user, anomaly detection has been viewed as a promising method by which to detect masqueraders [6]. However, since anomaly detection suffers from a high false alarm rate, this method is still in the research stage.

In the present study, we focus on host-based IDS, in which masquerade detection is performed by analyzing user behavior when commands are typed in a UNIX environment. Shell commands from a user's account are collected using UNIX's acct auditing mechanism. The profile of each user is constructed from a historically normal command sequence. Subsequently, the normality of an observed command sequence unit can be evaluated by comparing it with the profile of the current user.

Based on differences in working themes, input habits, and the degree of familiarity with the computer system, each user exhibits particular characteristics in issuing a command sequence. These differences among users make masquerade detection possible. Generally speaking, masquerade

D. Geng (✉) · T. Odaka · J. Kuroiwa · H. Ogura
Graduate School of Engineering, University of Fukui, Fukui, Japan
e-mail: geng@u-fukui.ac.jp; koil_geng@hotmail.com

T. Odaka
e-mail: odaka@u-fukui.ac.jp

detection distinguishes intruders by comparing the current behaviors of users with their historic behavior characteristics, so the characteristics that express user behavior must be extracted.

Several methods have been proposed for performing masquerader detection at the command line. One such method is based on behaviors of frequency characteristics, which can distinguish users based on different distributions of command sequence characteristics among users [7]. Another is based on the transition characteristics between behaviors, based upon which transition matrixes between command sequences can be constructed [8,9]. When the command transfer appears to have a higher probability for the current user, the probability that the command sequence belongs to the user is higher. However, the limitations of the above methods include low detection accuracy, high time complexity, and poor results interpretation. Therefore, a masquerade detection method based on the N-Gram frequency characteristics and STF-IDF weight classifier is proposed herein. Both the user's behaviors of frequency characteristics and the transition characteristics between behaviors are taken into account in the proposed method.

Using the proposed method, N-Gram frequencies are recorded during the training phase. During the detection phase, the command sequence to be detected is split into a number of N-Gram sequence combinations in accordance with the N-Gram characteristics, and a STF-IDF weight formula is adopted in order to calculate the scores of these combinations according to the N-Gram characteristics of the current user. The higher the score of a sequence, the higher the likelihood that the sequence belongs to the current user. An experiment based on the Schonlau and Greenberg data set [6,10] reveals that the proposed method is a considerable improvement over previous methods, both in terms of accuracy of detection and the error rate. The proposed method is among the most effective methods available at present. Moreover, since the proposed method has the advantages of easy interpretation and low computational cost, further study would be useful.

The remainder of the present paper is organized as follows. Section 2 reviews related research. Section 3 describes the proposed method. Section 4 describes the experimental configuration and presents the experimental results. Section 5 presents a discussion, and Sect. 6 presents the conclusion.

2 Related research

Masquerade detection based on command sequences has been investigated widely with respect to various aspects and provides a valid approach for securing a Unix system.

Schonlau et al. [6] investigated this issue and attempted to detect the masquerader by determining the behavior characteristics of the "normal user". He experimentally compared six masquerade detection methods, namely, Uniqueness, Bayes one-step Markov, Hybrid multi-step Markov, Compression, IPAM, and Sequence-match. Among these methods, the method is based on the command sequence frequency characteristics, and the remaining five methods are based on the transition information characteristics of command sequences.

Masquerade intrusion detection can also be regarded as a two-class classification problem. Accordingly, a number of conventional classification methods [11] have been proposed for masquerade detection, in which either measures of single events or measures of the transfer relationship from one event to another are taken as feature vectors. The Naïve Bayes classifier technique [12] and the SVM [13] have been proposed for masquerade detection, where frequencies of single commands within a window of fixed length were adopted as feature vectors.

In contrast, other methods use transfer relationships between events to characterize user behavior. A Hidden Markov model (HMM) [14] that postulates a model with a hidden structure producing sequential events has been proposed. In order to further consider correlational relationships between non-adjacent events, the Eigen co-occurrence matrix method (ECM) [15] was proposed. These methods attempt to construct a "normal model", which then attempts to detect abnormal occurrences, based on frequently appearing and relatively stable behaviors.

At the same time, anomaly detection must address the problem of a great quantity of audit data. Accordingly, data mining methods have also been proposed for anomaly detection in sequential data. The Boosting Decision Stumps method [16]was further proposed in order to improve efficiency by filtering and assembling multiple rules based upon which to draw a final conclusion. The main advantage of data mining methods is that extracted patterns or rules can be interpreted by the security administrator.

Statistical methods, which are always computationally efficient, have also been proposed for masquerade detection, and such methods do not require the extended training process required by machine learning methods. In particular, the customized grammars method [17] uses a sequitur algorithm to extract repetitive sequences in the form of a context-free grammar. This method combines high detection accuracy with a low error rate.

3 Method

Figure 1 is a structural diagram of an anonymous detection system running on a UNIX system host, which includes three main components: the audit database module,the learning module and the abnormal detection module. In this system,

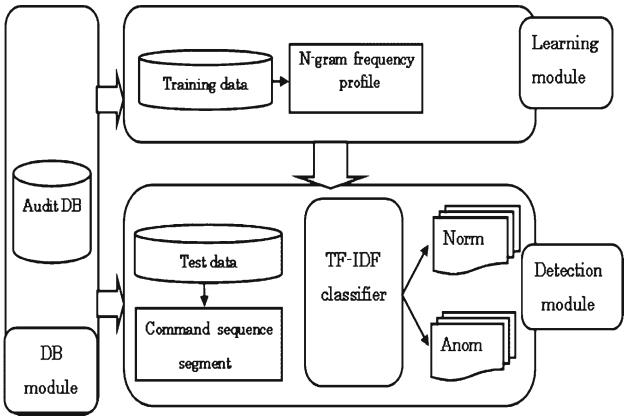

Fig. 1 Structure of an anonymous detection system running on a UNIX system host

the system structure is independent of the source data in the audit database, because the audit data may be a series of records of computer systems, such as a system file log, resources utilization, network packets, calls of system functions, or interface interactive events. The experiment conducted in the present study is based on public command data sets of UNIX command sequences. Using these data sets can effectively compare the detection accuracy of a new method with that of previous methods.

The purpose of the present study is to detect intruders based on the characteristics of the behavior of the normal user observed beforehand. Therefore, the establishment of these characteristics is the first step in the learning module. Here, the N-Gram frequency characteristics are established for individual user. In the detection unit, N-Gram feature extraction will first be conducted from sequences to be detected. In other words, according to previously established N-Gram characteristics, sequences are divided into a series of N-Gram combinations and are then examined for anomalies by a STF-IDF classification formula based on the N-Gram frequency distribution of the characteristics. Each process of the N-Gram_STF-IDF method will be explained in detail in the following.

3.1 N-Gram model for masquerade detection

Under a UNIX environment, different users may use the same command when engaging in the same task, whereas the same user may also enter different commands to complete different tasks over time. Therefore, detecting camouflaged users by using command sequences entered by the user is a very difficult problem. We must extract the same model or a similar model from the user's command sequences as a basis for judgment.

The N-Gram phrase is the most basic and important method in natural language processing [18] and has already

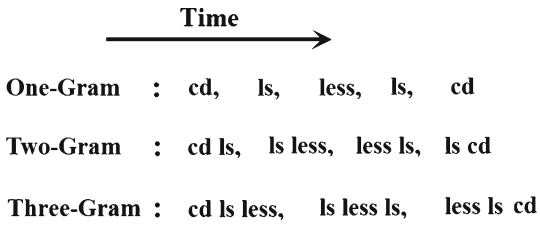

Fig. 2 N-Grams of command sequences

been successfully applied in several fields, such as language decomposition, machine translation, and information retrieval. The key advantage of an N-Gram is that the phrase can itself carry more information than a single element. Thus, the expressing ability of user's behavior by using N-Gram as its characteristics is greater than that of a single word. In addition, since sequences can be divided into a number of N-Gram phrases, any error input by the user affects only a limited part of N-Gram phrases, and the characteristics of the remaining part are maintained, which is conducive to reducing the error in determining sequences of normal users. Therefore, N-Grams have a unique advantage with respect to anomaly detection.

Broadly speaking, an N-Gram refers to N character fragments of a long character sequence. In the present paper, an N-Gram refers to N consecutive commands. Here, we are only concerned with the relationship between adjacent commands, particularly N-Gram fragments that appear several times in a command sequence. This often reveals preferences for certain tasks for users entering commands (Fig. 2).

Generally speaking, a command sequence containing K commands contains $K - N + 1$ N-Gram sequences.

3.1.1 Generating N-Gram frequency profiles for normal users during the learning stage

In the present study, detection of intrusion behaviors using the N-Gram model consists of the establishment of a user's N-Gram frequency characteristics during the learning stage and the extraction of the characteristics of N-Gram sequences to be detected during the pre-detection stage.

The process of establishing N-Gram characteristics is relatively simple, and only the audit data of each user must be scanned separately, and N-Grams having frequencies that are greater than K are extracted as user characteristics. After establishing the user's characteristics, the characteristics are merged and the N-Grams for which the sum of frequencies are higher than M are extracted as the general characteristics of the all the users.

The range of N in the N-Gram characteristics is based on the following observations. Assuming there is a N-Gram characteristics in which the lengthes are N_1 and N_2 respectively ($N_1 \geq N_2$), there exists an arbitrary sequence of commands α in N_1-Gram, and its frequency is f_1, and the user's

frequency is f_1' (the user's frequency is defined as the number of users who use the feature divided by the total number of users). Then, in the N_2-Gram features, there must exists a frequency f_2, and the command sequence of the user whose frequency is f_2' is β (β is a certain prefix of α), where $f_2 \geq f_1$, $f_2' \geq f_1'$. Thus, the N_2-Gram has higher generality than the N_1-Gram. The probability of N_2-Gram features appearing in sequences to be detected is higher, and N_1-Gram features are more specific than N_2-Gram features and can effectively distinguish specific users. However, the probability of N_1-Gram features appearing in the sequences to be detected decreases with increase of the length of N_1-Gram, so the value N of the N-Gram should be limited to within a certain reasonable range in order to provide better user differentiation and greater probability of appearing in sequences to be detected. In the present study, N is set to 3. We will discuss this topic in greater detail later herein.

3.1.2 N-Gram characteristics extraction of command sequences during the detection stage

In UNIX-command-based intrusion detection, a single command is the smallest unit that is able to reflect a user's activities and has a certain meaning. Moreover, a command is also a component of a command sequence. However, a single command is very limited in terms of expressing a user's characteristics. Comparatively, an N-Gram sequence contains more information characteristics. A command sequence is a combination of a number of N-Gram sequences. Hence, according to the characteristics of N-Grams, several combinations can be used to detect possible N-Grams of sequences. Among these, however, there is only one combination that are in accordance with the characteristics of the command sequences of the user. For example, if the original order of inputting command sequences is [cd ls le cd ls le ls le], then the N-Gram combination [cd ls le, cd ls le, ls le] has better agreement with the user's inputting characteristics than the combination [cd ls, le cd, ls le, ls le]. According to the earlier conclusion, in the present study, we segment the sequences to be detected into longer N-Gram combinations in order to increase the accuracy of illegal user detection.

Here, a strategy of maximum matching is adopted. As shown in Fig. 3, for the detection session, we scan the sliding windows one-by-one with a beginning length of $N = 3$ in accordance with commands and then check whether the N-Gram to be considered in each sliding window exists in the all users frequency characteristics. If this N-Gram exists, the window slides back N commands and repeats the initial action. If this N-Gram does not exist, then we change the length of the window to $N - 1$ and continue testing until $N = 1$. At this point, if no command exists in the all users frequency characteristics, then the command is a new command. The window moves backward one command, and

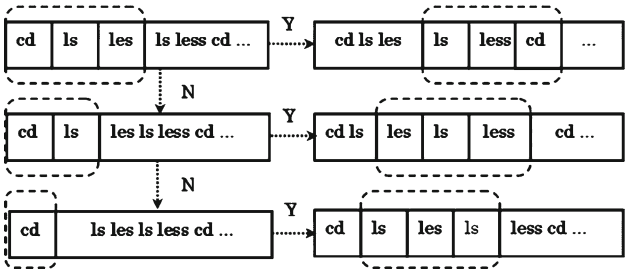

Fig. 3 Process of command sequence segmentation

the above actions are repeated until the end of the sequence to be detected. We then record all generated N-Grams and their frequencies. After this process, the testing sequence should be divided into a series of N-Gram sequence combinations according to the all users characteristics. Subsequently, we need to calculate the similarity between each command sequence to be detected and the current user in order to classify a sequence based on the N-Gram frequency characteristics of the user and the identified N-Gram combinations.

3.2 Masquerade detection using the STF-IDF statistics model

After the sequences to be detected are divided into a series of N-Gram sequence combinations, the weight of each N-Gram sequence must be calculated for the current user in order to evaluate the entire sequence. According to the command input characteristics under the Unix environment and the purpose of abnormal detection, the weight calculation should meet the following conditions.

First, command sequences that appear frequently among users should be assigned greater weight, because these commands represent certain specific behaviors that users engage in frequently. Second, command sequences that appear frequently for a single user but that are rarely used by other users should be assigned a greater weight, because these commands are extremely useful for differentiating users. Finally, basic operation commands that appear frequently for every user should be limited by the weights of these commands, because these commands will often interfere with detection.

Based on the above considerations, the TF-IDF model is introduced in order to calculate the weight of each of the characteristics in the present paper. In this algorithm, which was proposed by Spärck-Jones [19] in 1972, information entropy is properly applied to the information retrieval [20], which is a commonly used weighting technique [21] used in natural language processing (NLP) and information mining (IM). At present, the TF-IDF model is also widely used in searching, text classification, and other related fields. This

Table 1 Notation and terminology

n	Number of all users in the observing data set
m_k	Number of all commands in the observing command sequence k
m'_k	Number of all command sequences in the observing command sequence k
s_i	Number of all users that use command sequence i
f_i	Frequency of command sequence i in the observing data set
f_{ij}	Frequency of command sequence i in the observing user j
f_{ik}	Frequency of command sequence i in the observing command sequence k
c_{kj}	Number of commands in the observing sequence k matched with the user j

model is described in detail below, and the notation and terminology used in the present paper are listed in Table 1.

In masquerade detection studies, the primary reason for introducing the TF-IDF model is that, if the frequency of a command sequence for one user is high and the command sequence appears rarely for other users, then this command sequence has a greater ability to distinguish users and has a high weight value. The basic form of this command sequence is generally as follows:

$$tf_i idf = f_i \log\left(\frac{n}{s_i}\right) \tag{1}$$

where TF primarily refers to the number of appearances of a given N-Gram sequence in detecting current users, and IDF indicates that the less frequently the user uses this sequence, the greater the IDF, which indicates that the sequence has a greater ability to distinguish users. In addition, TF⋆IDF is the degree of importance of the sequence in detecting current users.

In UNIX-based command masquerade detection, the frequencies of some basic operation commands, such as ls, cd, and cat, are often very high, and almost every user operates the computer using these commands. However, for anomaly detection, the frequencies of some basic operation commands are essentially unable to represent the specific input features of the user. Therefore, it is necessary to reduce the influence of command sequences with high frequency to the TF-IDF formula to make sure that characteristics commands have greater weight so as to make the TF-IDF formula have higher identifying accuracy to user's characteristics commands sequences. STF-IDF formula is proposed in the paper, which is used to extract calculate term frequency of TF-IDF and reduces its function, its form is as follows:

$$stf_i idf = \sqrt{f_i} \log\left(\frac{n}{s_i}\right) \tag{2}$$

Moreover, when calculating the weight of a certain sequence for a user based on STF-IDF, the following three factors must also be considered.

(1) The degree of importance of characteristic sequences for current users.
Different sequences for the same user indicate different degrees of importance. The STF-IDF formula is as follows:

$$stf_{ij} idf_i = \sqrt{f_{ij}} \log\left(\frac{n}{s_i} + 1.0\right) \tag{3}$$

(2) The degree of importance of characteristic sequences in the current sequence to be detected.
This is obtained as follows:

$$stf_{ik} idf_i = \sqrt{f_{ik}} \log\left(\frac{n}{s_i} + 1.0\right) \tag{4}$$

(3) The degree of match between sequences to be detected and current users.

The degree of match between sequences to be detected and current users is obtained as the number of commands in the sequences to be detected that appear in the 1-Gram characteristics of the current user. The higher the degree of match, the more likely the sequence is that of a normal user. If the degree of match of a sequence is zero, then the sequence should be designated as an intrusion sequence.

$$n_{kj} = \frac{c_{kj}}{m_k} \tag{5}$$

These three factors influence one another, so the product form is used to reflect the relationship among the three factors. Taking into account the influence of command sequences on the weight, we standardize each component of the additional part and limit the weight range to between 0 and 1 to obtain the following formula:

$$\left[\sum_{i=1}^{m'_k}\left(\frac{stf_{ij} idf_i}{\sqrt{\sum_{i=1}^{m'_k}\left(stf_{ij} idf_i\right)^2}} \frac{stf_{ik} idf_i}{m'_k}\right)\right] n_{kj}^2 \tag{6}$$

This formula is used to calculate the weights of each command unit from the sequences to be detected and to then sum these weights. The obtained result is the score of sequences

to be detected for the current user. In addition, this formula can also be expressed by TF-IDF. We compare the detection accuracies in subsequent experiments.

4 Experiment and results

In this section, the experimental data are based on the Schonlau and Greenberg data set [6,10]. In the Schonlau data set, the influence of N values in the N-Gram features on the detection accuracy is investigated, and the detection results of STF-IDF and TF-IDF are compared. Samples were not been updated during the testing phase of the above two experiments. Therefore, the detection results of N-Gram_STF-IDF and the detection results of other methods are compared under the updated and non-updated conditions. Finally, using the Greenberg data set, the validity of the N-Gram_STF-IDF was verified and the results were compared with the results of other methods.

As usual, receiver operating characteristic (ROC) curves are used to evaluate the intrusion detection capabilities of different methods. Receiver operating characteristic curves are drawn based on the missing alarm rate (MAR) and the false alarm rate (FAR) under different thresholds. Here, MAR is defined as the percentage of intrusion blocks wrongly classifying as normal blocks, and the FAR is defined as the percentage of normal blocks that are incorrectly classified as intrusion blocks. A trade-off relationship exists between the FAR and the MAR, and, in particular, in the ROC curve, the smallest value of the sum of the MAR and the FAR represents the detection performance of the method. Therefore, this sum is defined as the cost, and its value is used to compare detecting effects of different methods. Generally speaking, the smaller the cost, the better the performance.

4.1 Masquerade detection based on the Schonlau data set

Based on UNIX shell commands, Schonlau and others collected a data set that is known as the SEA data set [6]. In the experiment of the present study, this data is used to evaluate the accuracy of masquerade detection in order to compare the obtained results with the results obtained using the other methods. The SEA data set is made up of users' names and associated command sequences (no parameters) for a total of 50 users. For each user, there are 15,000 consecutive commands, which are divided into 150 blocks, each of which is composed of 100 commands. The first 50 blocks are training data, and starting from the fifty-first block, the command blocks besides 50 users randomly replace original command blocks as masquerade command blocks. The objective of the experiment is to correctly detect the masquerade blocks for the normal users. The experimental data sets can be downloaded at http://www.schonlau.net/intrusion.html.

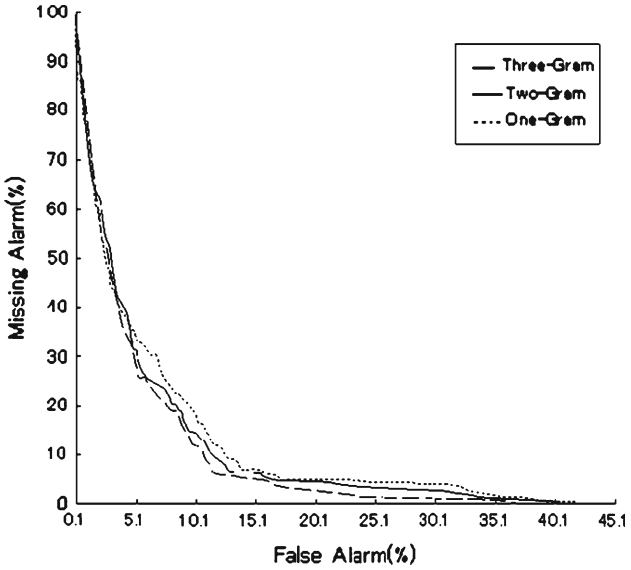

Fig. 4 ROC curves with different N values in N-Gram characteristics

4.1.1 Evaluation of N dereferencing in N-gram characteristics

In order to evaluate the impact of N dereferencing in N-Gram on the detection accuracy, Fig. 4 shows the ROC curves, which show the experimental results for various values of N ($N = 1, 2, 3$). Here, $N = 1$ means that only 1-Grams exist in the characteristics, and $N = 2$ means that 1-Grams and 2-Grams exist in the characteristics, and so on. The figure indicates that the detection accuracy is the highest when $N = 3$, and the detection accuracy for $N = 2$ is higher than that for $N = 1$. For example, when $N = 2$, the accuracy of detecting abnormal blocks increases from 80.2 to 86.3%, and the FAR is 10.0%, which also shows that the characteristics of the 2-Gram and the 1-Gram together can express the user's specific and detailed features better than the characteristics of the 1-Gram alone. Therefore, increasing the length of N-Gram characteristics within a certain range is very helpful for detecting abnormal behaviors.

However, when $N = 3$, the detection accuracy does not increase significantly. The primary reasons for have been described above. Since 3-Gram features have a lower probability of appearing in the command sequences to be detected, as compared to 1-Gram or 2-Gram features. For example, from using 1-Gram ($N = 1$) characteristics to 1, 2-Gram ($N = 2$) characteristics, the value of N increases within a certain range, which can improve the method identifying accuracy to users, so the longer N-Gram characteristics may often stand for user's idiographic behavior characteristic. However, from the experiment and analysis, when the value of N increases to a certain range, the influence to detecting accuracy will be not too great, so it is limited to improve the effect of detecting precision relying on increasing the value of N.

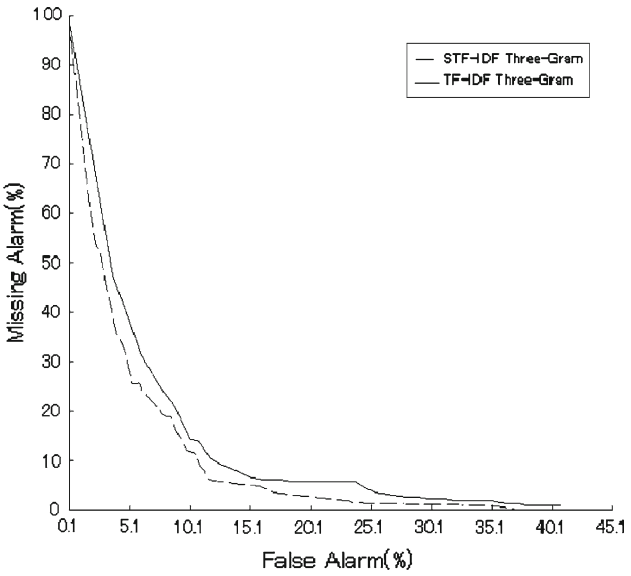

Fig. 5 ROC curves for evaluation of the TF-IDF and STF-IDF classification formula

In the present study, since the addition of "1" to "N" after $N > 3$ requires considerable additional calculation time, with little of no improvement in detection, in our method "N" is set to 3.

4.1.2 *Evaluation of STF-IDF weight calculation and classification formula*

In order to compare the effectiveness of the STF-IDF weight classification formula proposed herein to masquerade detection, the formula is compared with the common TF-IDF weight formula and the ROC curves are drawn, as shown in Fig. 5. As shown in the figure, when the FAR is 7.9%, the detection accuracy of STF-IDF is better than that of TF-IDF. In this case, the intrusion detection rate of STF-IDF reaches 80.9%, primarily because STF-IDF restrains commonly used commands that frequently appear for the user but that cannot be used for user classification. Note that if these commands are deleted, the detection accuracy will decrease because in certain sequences to be detected, this category of commonly used commands accounts for more than half of the total number of commands,so that deleting these commands will lead to too few commands for weight calculation being available for normal users, which results in miscalculation.

4.1.3 *Comparison of intrusion detection performance between N-Gram_STF-IDF and other methods under various conditions*

In a number of previous studies [6,12,13,15–17] based on SEA data sets, Uniqueness, IPAM and other methods have been used for masquerade detection, and the performances of these methods have been investigated. In the present study, the results of these studies are directly compared with the detection accuracy of the N-Gram_STF-IDF method under the updated and non-updated conditions. Here, "updated" refers to the use of data judged to be normal by the detection data to update the sample data, so the updating strategy [6,12] is effective, which can significantly improve performance.

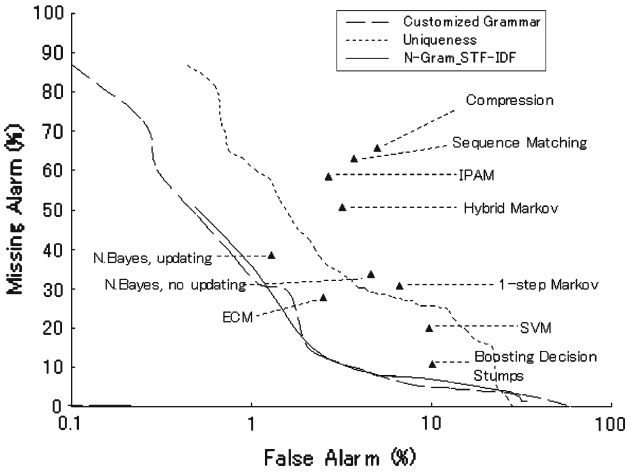

Fig. 6 ROC curves for the N-Gram_STF-IDF and other methods

Figure 6 shows that the performance of the N-Gram_STF-IDF method is better than the performances of the other methods. Furthermore, the masquerade detection accuracy of this method is 91.97% when the FAR is 5.08 and 64.2% when the FAR is 1.0%. Customized grammar[17] has been reported to be the most effective masquerade detection method to date. In addition, for the SEA configuration, these two methods have nearly the same detection efficiency and are superior to previous methods, especially at different trade-off points. Note that different methods have minimum costs at different trade-off points, as shown in Table 2. For the N-Gram_STF-IDF, the minimum cost is 0.128 when the FA rate is 8.03%. Therefore, the N-Gram_STF-IDF method achieves its minimum cost at a particular trade-off value of the FAR and the MAR.

Figure 7 shows the N-Gram_STF-IDF, Customized Grammars, and Uniqueness without updating. Compared to other methods, the N-Gram_STF-IDF method has excellent performance. The masquerade detection accuracy of the N-Gram_STF-IDF method is 93.93% when the FAR is 11.68% and is 98.70% when the FAR is 23.74%. Compared with those results under the updating condition, the detection accuracy has increased, but the false alarm rate has also increased.

4.2 Masquerade intrusion detection based on the Greenberg data set

A data set of command sequences was collected by Greenberg [10]. This data set is composed of 168 users,

Table 2 Results for previous methods based on the SEA experiment

Method	MA	FA	Cost
N-Gram_STF-IDF	4.8	8.0	12.8
Customized grammars	6.0	7.2	13.2
SVM	19.9	9.7	29.6
ECM	27.7	2.5	30.2
N. Bayes (no updating)	33.8	4.6	38.4
N. Bayes (updating)	38.5	1.3	39.8
Uniqueness	60.6	1.4	62.0
IPAM	58.6	2.7	61.3
Hybrid Markov	50.7	3.2	53.9
Sequence matching	63.2	3.7	66.9
Compression	65.8	5.0	70.8
Bayes one-step Markov	30.7	6.7	37.4

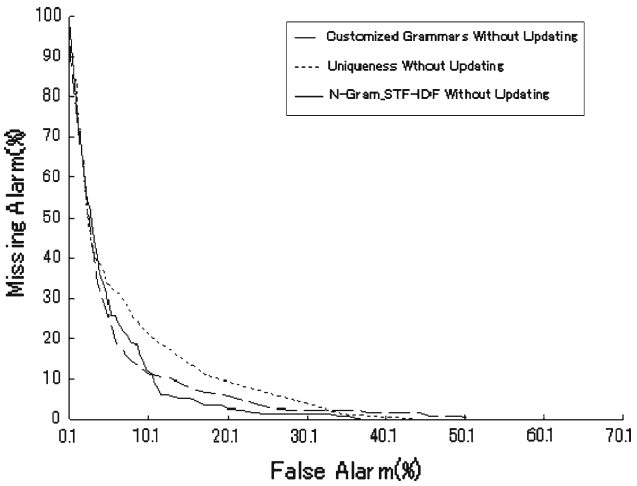

Fig. 7 ROC curves for the *N*-Gram_STF-IDF and other methods

who are divided into four groups consisting of 55 novice users, 36 empirical users, 52 computer-science users, and 25 non-programmers. 50 users with 2,000 commands per user are considered as a test target, and these 2000 commands are divided into 200 blocks, each of which consist of 10 commands (including parameters). The first 100 command blocks are used as the sample data. Starting from the 101st command block, 250 command blocks are randomly selected from the randomly selected 25 users to replace the original data randomly. The Greenberg data set consists of the entire set of command lines. However, in this experiment, all of the parameters except for the commands have been ignored. This type of experiment, based on the Greenberg data sets using truncated command data, is also known as a TCG experiment. Compared with the experiment based on the SEA data set, the TCG experiment is much more similar to the actual situation because, after the implementation of 100 commands, the system's security may have already been destroyed. Therefore,

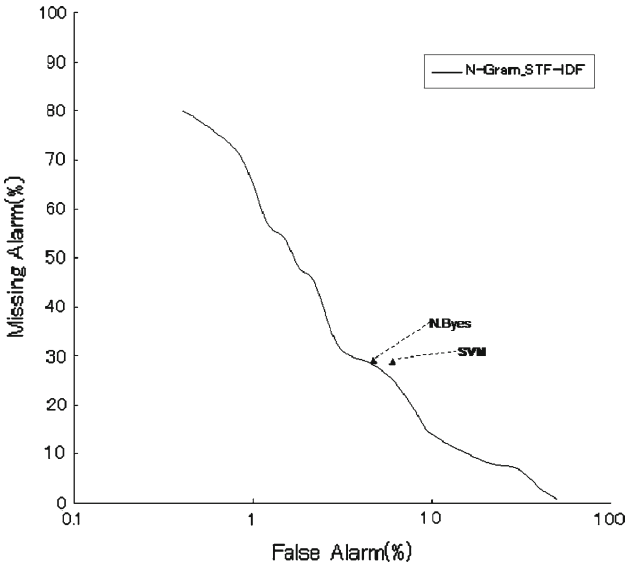

Fig. 8 ROC curves for *N*-Gram_STF-IDF and other methods on Greenberg data set

in the present study, the TCG experiment will be performed in order to evaluate the effectiveness of the new method.

4.2.1 Experimental results

Figure 8 shows the ROC curves for *N*-Gram_STF-IDF and for the SVM and N.Bayes methods based on previous TCG experiments [12,13]. The ROC curves show that the detection accuracy of *N*-Gram_STF-IDF is 74.8% when the FAR is equal to 6.0%, which is higher than the detection accuracies of the other two methods. This confirms that the *N*-Gram_STF-IDF method is valid under more a realistic experimental environment.

5 Discussion

At present, most masquerade detection methods are based on the frequencies during the extraction of the audit data or the transition characteristics, and all of these methods can solve the masquerade detection issue. However, either from the detecting precision and time complexity or from the result explainary, it can't meet the actual need of IDS. Therefore, the *N*-Gram_STF-IDF method mixed considering frequency and transition characteristics were proposed to detect the abnormal behaviors in normal users.

In the present study, 1-Gram, 2-Gram, and 3-Gram command sequences in the audit data were used to generate the characteristics. The results of an experiment confirmed that the *N*-Gram characteristics are effective for identifying some normal command sequence units. For example, for a specific user, a 2-Gram characteristic is extracted as

"java, csh", which relies on the java program included in the user's operation, whereas other users may deal with other subjects related to java programs. Therefore, if the characteristics of the command sequences to be detected include "java, csh", then the user to which the command sequences belong may be identified. In contrast, the 1-Gram characteristic "java" is too common to identify users, so the detection accuracy can be improved by augmenting the N values to a certain range.

The detection accuracies of TF-IDF and STF-IDF proposed herein have also been evaluated. Experiment results reveal that STF-IDF can improve the detection accuracy, primarily because STF-IDF can restrain the weights of command sequences that appear frequently among users but contributes little to the identification of different users. For example, during document display and internet browsing task commonly performed by users, "ls, ls, cd" and "mail, mail" characteristic command sequences will appear frequently. Since these characteristic command sequences will disturb the detection of abnormal behaviors, a reasonable reduction in these characteristic command sequences can further improve the ability to identify different users.

The STF-IDF weight classification formula is very sensitive to the special sequence characteristics that appear for certain users but that are not often used by other users and takes into account not only the frequency factor of characteristic command sequences in the command units to be detected, but also its importance among users. Therefore, the STF-IDF weight classification formula can appropriately compensate these characteristic sequences that are vital for the current user but that do not frequently appear in the sequences to be detected. For example, assuming a certain characteristic sequence is very important to the current user, but appears infrequently in the sequences to be detected, the score of the STF-IDF formula remains very high, which is helpful for improving the detection accuracy.

Compared with the other machine learning methods, the N-Gram_STF-IDF method has higher interpretability. Machine-based methods, such as SVM and HMM, provide only the evaluation values of sequences to be detected, and the reasons for judgments and the processes are not interpretable by security administrators. In contrast, in addition to the evaluation values of the sequences, the N-Gram_STF-IDF method also provides the contribution of each characteristic sequence, which provides security administrators with more information, which can be used to analyze the behaviors of users.

The N-Gram_STF-IDF method has a low calculation cost, and only frequencies need to be calculated. Moreover, 1-Grams, 2-Grams, and 3-Grams must be matched either during the training and testing periods or during the updating period. The calculation cost based on the STF-IDF formula is also very small, so that the detection of the command sequences can be completed in real time. Therefore, N-Gram_STF-IDF can be regarded as a real-time online system.

The present study reveals that the combined use of user's frequencies and transition characteristics can improve the accuracy of masquerade intrusion detection. Therefore, N-Gram ($N \leq 3$)'s frequency characteristics have been constructed, in which not only correlation information among commands, but also frequencies of appearance for each characteristic, were considered. In addition, the STF-IDF weight classification formula considers several important factors in abnormal detection and emphasizes characteristic weights that contribute to the classification of users. Therefore, combine with the above factors, the N-Gram_STF-IDF method provides better intrusion detection accuracy.

Although N-Gram_STF-IDF offers improved detection accuracy of abnormal users on a data set based on UNIX command sequences, even this improved detection accuracy is not satisfactory for application in a practical environment. Although a low MAR is realized, the FAR is still very high. Therefore, in the future, frequency and transition characteristics should be better used to consider user's behavior characteristics, such as non-exact matching, to reduce the FAR under the condition of steady detection accuracy should be investigated. In addition, a further expansion of the use of abnormal data detection, such as System Call Sequence and Network Stream Sequence, will be investigated.

6 Conclusions

The N-Gram_STF-IDF method has been proposed for the detection of camouflaged intrusion, and an evaluation experiment based on a benchmark data set has been conducted. When the FAR is 8.03%, the cost of the N-Gram_STF-IDF method is 0.128. In addition, since the calculation cost of the proposed method is low and implementation is simple, the proposed method may lead to an online real-time intrusion detection system.

References

1. DTI. Information security breaches survey 2006. Technical report, DTI (Department of Trade and Industry, Britain) (2006)
2. Gordon, L.A., Loeb, M.P., Lucyshyn, W., Richardson, R.: CSI/FBI Computer crime and security survey 2006. Computer Security Institute publications (2006)
3. Yampolskiy, R.V.: Human computer interaction based intrusion detection. In: Fourth International Conference on Information Technology, 2007, ITNG'07, pp. 837–842 (2007)
4. Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Department of Computer Engineering, Chalmers University, Tech. Rep. 1:99–15 (2000)

5. Murali, A., Rao, M.: A survey on intrusion detection approaches. In: First International Conference on Information and Communication Technologies, ICICT 2005, pp. 233–240 (2005)
6. Schonlau, M., DuMouchel, W., Ju, W.H., Karr, A.F., Theus, M., Vardi, Y.: Computer intrusion: detecting masquerades. Stat. Sci. **16**, 58–74 (2001)
7. Huang, S.H.S., Wu, H.C.: Analysis of user command behavior and masquerade detection. J. Inf. Assur. Secur. **4**, 265–273 (2009)
8. Liao, Y., Vemuri, V.R., Pasos, A.: Adaptive anomaly detection with evolving connectionist systems. J. Netw. Comput. Appl. **30**(1), 60–80 (2007)
9. Guan, X., Wang, W., Zhang, X.: Fast intrusion detection based on a non-negative matrix factorization model. J. Netw. Comput. Appl. **32**(1), 31–44 (2009)
10. Greenberg, S.: Using unix: collected traces of 168 users. Department of Computer Science, University of Calgary. Technical Report **88**(333), 45 (1988)
11. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. (CSUR) **34**(1), 1–47 (2002)
12. Maxion, R.A., Townsend, T.N.: Masquerade detection augmented with error analysis. IEEE Trans. Reliab. **53**(1), 124–147 (2004)
13. Kim, H.S., Cha, S.D.: Empirical evaluation of SVM-based masquerade detection using UNIX commands. Comput. Secur. **24**(2), 160–168 (2005)
14. Warrender, C., Forrest, S., Pearlmutter, B.: Detecting intrusions using system calls: alternative data models. In: IEEE Symposium on Security and Privacy, pp. 133–145. IEEE Computer Society, USA (1999)
15. Oka, M., Oyama, Y., Abe, H., Kato, K.: Anomaly detection using layered networks based on eigen co-occurrence matrix. Lecture Notes in Computer Science, pp. 223–237 (2004)
16. Jian, Z., Shirai, H., Takahashi, I., Kuroiwa, J., Odaka, T., Ogura, H.: Masquerade detection by boosting decision stumps using UNIX commands. Comput. Secur. **26**(4), 311–318 (2007)
17. Latendresse, M., Navy, U.S.: Masquerade detection via customized grammars. In: Second International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. LNCS, vol. 3548, pp. 141–159. Springer, Berlin (2005)
18. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, pp. 161–175 (1994)
19. Jones, K.S. et al.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **60**, 493–502 (2004)
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval* 1. Inf. Processing Manage. **24**(5), 513–523 (1988)
21. Debole, F., Sebastiani, F.: Supervised term weighting for automated text categorization. In: Proceedings of the 2003 ACM symposium on Applied computing, pp. 784–788. ACM, New York (2003)