

Utilizing Statistical Characteristics of N-grams for Intrusion Detection

Li Zhuowei, Amitabha Das and Sukumar Nandi
School of Computer Engineering,
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
Email: asadas@ntu.edu.sg, zhwei.li@gmail.com, zhwei.li@ntu.edu.sg

Abstract

Information and infrastructure security is a serious issue of global concern. As the last line of defense for security infrastructure, intrusion detection techniques are paid more and more attention. In this paper, one anomaly-based intrusion detection technique (ScanAID: Statistical Characteristics of N-grams for Anomaly-based Intrusion Detection) is proposed to detect intrusive behaviors in a computer system. The statistical properties in sequences of system calls are abstracted to model the normal behaviors of a privileged process, in which the model is characterized by a vector of anomaly values of N-grams. With a reasonable definition of efficiency parameter, the length of an N-gram and the size of the training dataset are optimized to get an efficient and compact model. Then, with the optimal modeling parameters, the flexibility and efficiency of the model are evaluated by the ROC curves. Our experimental results show that the proposed statistical anomaly detection technique is promising and deserves further research (such as applying it to network environments).

1 Introduction

Information security, especially the Internet security, has become a serious issue of global concern. The Internet has brought great benefits to the modern society, but the complexity, accessibility, and openness of the Internet have served to increase the security risk of information system tremendously. According to the survey conducted by CSI/FBI [6], the Internet has become a more frequent attack point for hackers, more and more skilled independent hackers emerge as the main source of attacks, and the total annual loss from computer crime has increased every year. Losses related to computer and network system have increased dramatically year by year as computer systems and networks have suffered from more and more security vulnerabilities and attacks. Unfortunately, there is no panacea

to solve all the security problems in the Internet, and generally, they should be solved individually and hierarchically. As the last line of defense in hierarchical security solutions, intrusion detection techniques first detect the anomalous behaviors of a resource. Then, according to the detection result, other strategies are applied to protect the Internet infrastructure.

Traditionally, there are two methods to detect intrusive behaviors associated with the access and use of a resource in the computer and network system: *misuse detection* and *anomaly detection*. Misuse detection digs out the signatures of attack scenarios manually, and detects the intrusion behaviors based on these signatures. Anomaly detection uses known normal behaviors of a resource to build a model for it automatically, and detects those behaviors violating the model as anomalies.

Since anomaly detection techniques have the potential to detect every possible intrusive behavior which violates the normal model of a resource, they are promising for the protection of computer and network infrastructure. From early 1980s, considerable efforts have gone into studying anomaly detection techniques.

In UNM, Forrest's research group [2] and others [11] [10] [4] [9] used the sequence of system calls of active, privileged processes as the model for anomaly detection. Each privileged process is represented by a trace, which is the ordered list of system calls used by that process from the beginning of its execution to the end. N-grams with the same length (e.g. 6) are abstracted from the traces to construct the model of the self for each privileged process. Their experimental results show that it is a useful technique for modeling privileged processes and detecting anomalies in their execution [1].

Lee et al. [3] applied data mining techniques to solve intrusion detection problems. In their research, the key idea is to use existing data mining techniques to extract consistent and useful profiles of system features which describe normal resource behaviors, and to use the set of relevant system features to mine (inductively learned) classifiers that

can recognize anomalies and known intrusions.

At the same time, there are many researches that focus on using statistical techniques to solve the anomaly detection problems [12] [8] [7]. Ye [12] applied the first-order Markov chain model to detect anomalies in intrusive dataset, in which only the relationship between individual events are considered. The Bayesian networks [8] have been used to solve the anomaly detection problems for any possible networked infrastructure, including Internet.

However, most of the existing anomaly detection techniques suffer from low efficiency due to high false positive rate, which is the fraction of normal behavior that is identified as anomalous. Because of this, there is almost no anomaly detection component in the available market products. In our paper, a statistical anomaly detection technique is proposed to address this shortcoming with an attempt to finding a more flexible and efficient anomaly detection approach.

The paper is organized as follows. Several terms related to anomaly detection techniques are introduced in section 2. In section 3, an anomaly vector for N-grams is defined for building the self model of a resource, and the detection phase is also described. Then, the methodology to get an efficient and compact model in ScanAID is given in section 4. Finally, some experiments are performed which are described in section 5 and corresponding experimental results are analyzed in section 6. Conclusions and further research about ScanAID are discussed in section 7.

2 Related terms and definitions

Central to anomaly detection techniques, there are five main elements: *resource*, *event*, *feature*, *model* and *classifier*. A *resource* is a subject which will be monitored and protected by an anomaly detection system, e.g. user accounts, file systems, network infrastructures, network services. During the access or use of a resource, there is a log associated with the resource, such as audit trail for operating system, network packets for a network infrastructure, and an *event* is an element in the log. From the sequence of events, some specific *features* are abstracted to describe the normal or anomalous behaviors of the resource. Using the features, a *model* can be built to describe the normal behaviors of a specific resource. The model should generalize the normal behaviors in the access sequence of the resource to some extent. After getting a generalized model, a *classifier* should be designed to classify the events being generated by the resource.

After an anomaly detection technique is designed, it should be evaluated by datasets which are collected from live source or generated synthetically. According to the different functions of these datasets, they can be classified as *training*, *testing* and *intrusive* dataset, which are defined as

follows:

Training dataset is a dataset that is used to build the model of a resource for later use (during detection phase). The training dataset should be guaranteed not to include any intrusive behaviors during its collection.

Testing dataset is similar to the training dataset, but its function is to evaluate the efficiency and flexibility of an anomaly detection technique.

Intrusive dataset is a dataset which includes intrusive behaviors during its collection, and it is also used to evaluate the efficiency and flexibility of an anomaly detection technique.

Definition 1. *efficiency*

From the past experiences, there will be some anomalies which can't be detected by an anomaly detection technique, and it causes the false negative outcomes. At the same time, some normal events are detected as anomalies by the anomaly detection technique, and it causes the false positive outcomes. The efficiency of the anomaly detection technique must take into account both the false negative rate and the false positive rate. In our research, the efficiency is used as a measure of the performance of the anomaly detection technique.

For the above reasons, the performance parameter *efficiency* of an anomaly detection technique is defined as:

$$\begin{aligned} \text{efficiency} = & (1 - \text{FalseNegativeRate}) * p(\text{anomaly}) \\ & + (1 - \text{FalsePositiveRate}) * p(\text{normal}) \quad (1) \end{aligned}$$

Where,

FalseNegativeRate is the fraction of processes which are not labeled as anomalies in the intrusive dataset;

FalsePositiveRate is the fraction of processes which are labeled as anomalies in the testing dataset;

$p(\text{anomaly})$ is the probability of anomalous events in the overall dataset (intrusive and testing ones); and

$p(\text{normal})$ is the probability of the normal events in the overall dataset (intrusive and testing ones).

3 Statistical modeling

In Forrest et al's experiments [2], the collections of N-grams, which are event sequences with a specific length, for one privileged process can be used to model the normal behaviors and to detect the anomalous behaviors of the process. In this approach, a profile of normal behaviors is built by enumerating all unique, contiguous sequences of a predefined, fixed length k that occur in the training dataset. For this purpose, a window with length k is slid across each trace, one system call at a time, adding each unique sequence to the normal database.

During detection phase, sequences from the traces in the testing or intrusive dataset are compared to those in the nor-

mal database. Any sequence not found in the database is called a mismatch. Any individual mismatch could indicate anomalous behavior, or it could be a normal sequence that was not included in the normal training data. From their experimental results (Look-ahead pair, tide, stide, t-stide), this technique is quite useful for modeling the processes from system audit trails.

In our ScanAID technique, the statistical dependencies between events (system calls) are considered in its modeling and detecting phases. N-grams, with the absolute probabilities of their occurrence, are used to model the resource to get the model, and to detect anomalies occurring during its execution.

3.1 Training model

During the training phase, w -grams, that is, the N-grams of length w , are abstracted from the training dataset. The absolute probabilities of the occurrences of these w -grams are computed individually. These absolute probabilities comprise an anomaly vector, and it constitutes the model of the resource in ScanAID.

For the sake of convenience, let the w -grams be expressed as: w_1, w_2, \dots, w_k . In the model for ScanAID, the anomaly vector can be expressed as $(p(w_1), p(w_2), \dots, p(w_k))$, and the absolute probability for one specific w -gram w_i is defined as:

$$p(w_i) = \frac{\text{num}(w_i)}{\sum_{j=1}^k \text{num}(w_j)} \quad (2)$$

Where, $\text{num}(w_i)$ is the number of occurrence of the i -th w -gram in the training dataset.

3.2 Detecting Anomalies

After the model of a resource is built, the anomaly detection technique should be able to use it to detect the anomalies existing in the intrusive dataset. In ScanAID, there are two anomaly values which are critical to detect anomalies for a specific resource and to compute its efficiency: *the anomaly value for every event* and *the anomaly value for every trace*. A trace is determined to be anomalous when its anomaly value exceeds a predetermined threshold value t .

3.2.1 Anomaly value for every event (system call)

In ScanAID technique, the original value for every absolute possibility in the anomaly vector is used to calculate the anomaly value for every event in the detection phase. For the N-grams with probability 0, that is, they do not occur in the training dataset, a minimum value (which is the smallest

probability value in the anomaly vector) is assigned to them for calculating the anomaly value.

Based on the absolute probabilities of N -grams for one resource, the anomaly value for every event (system call) in the trace sequence can be computed. Suppose that the events listed below have been generated and checked recently: e_1, e_2, \dots, e_{w-1} . Then, another event, e_w , arrives, producing the sequence: $e_1, e_2, \dots, e_{w-1}, e_w$. The anomaly value of the event e_w is:

$$\text{AnomalyValue}(e_w) = (-1) \times \log_a p(e_1 \dots e_{w-1} e_w) \quad (3)$$

One of the obvious properties of the equation 3 is that the rarest event will hold the largest anomaly value and it is useful to understand the optimizing procedure in ScanAID.

3.2.2 Anomaly value of a trace

The anomaly value of a trace is used to determine whether the trace is anomalous. In ScanAID, it is derived from the anomaly values of individual events in that trace as follows. For every trace, the highest m anomaly values for the events in the trace are remembered, and the sum of them are used as the anomaly value of the trace.

4 Optimizing modeling parameters

Following the design of ScanAID in section 3, its modeling parameters will be optimized and its performance will be evaluated according to the scheme shown in Figure 1. In this scheme, three datasets are needed to optimize the critical modeling parameters of ScanAID and to evaluate its efficiency: *training dataset*, *testing dataset*, and *intrusive dataset*. As the definition in section 2, the training dataset and testing dataset can be generated from a normal dataset.

In our experiments for optimizing the modeling parameters of ScanAID, the starting point for splitting the normal dataset is variable to avert the possible irregularity in the normal dataset and to get an efficient and compact model for ScanAID. Then, the efficient and compact model is used to generate the ROC curve for performance comparisons with other anomaly detection techniques.

4.1 Identifying the critical modeling parameters

From experimental results for stide [2], the length of N-grams strongly affects the efficiency of anomaly-based intrusion detection technique. If a small value for the length of N-grams is chosen, it fails to effectively account for contextual dependencies of individual events. On the other hand, the larger the length of N-grams is, the more storage and computation time it needs. In addition, the effect of infrequent intrusive events becomes blurred as the length

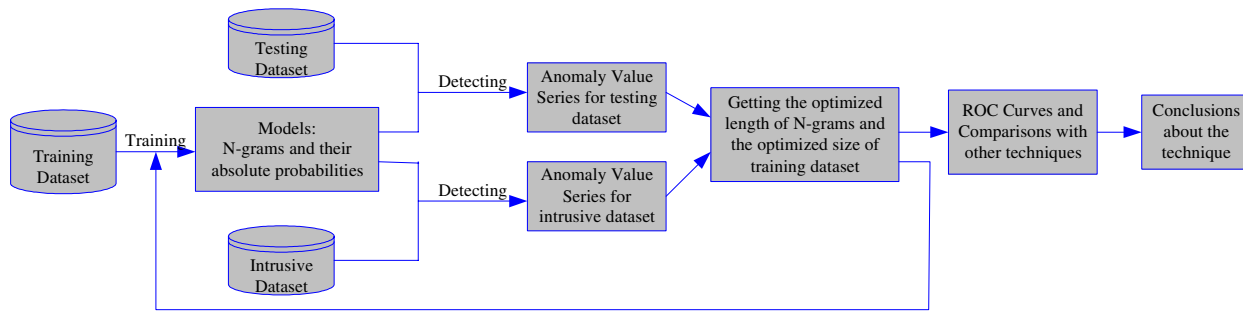


Figure 1. Scheme for Evaluating ScanAID.

of N-grams is increased. Therefore, the length of N-grams should be evaluated and optimized before they are applied to detect anomalies for a resource.

The size of the training dataset is also critical to the efficiency of anomaly-based intrusion detection technique since it can affect the validity of the model created through the training dataset: the inadequacy of training dataset will adversely affect the accuracy of the anomaly detection techniques, and its redundancy will lead to difficulty for collecting training dataset and extra time for training models which is useless for the efficiency of the anomaly detection technique.

Consequently, there are two modeling parameters affecting the efficiency of the ScanAID: *the length of the N-gram*, and *the size of the training dataset*. Both of the modeling parameters should be chosen carefully to get efficient model for anomaly-based intrusion detection.

4.2 Efficiency calculation in ScanAID

The efficiency of an anomaly-based intrusion detection technique is related to the false positive rate found in the testing dataset and the false negative rate found in the intrusive dataset (section 2) in our experiment. For an anomaly detection technique, a threshold t should be predetermined to classify the traces in the testing dataset and intrusive dataset, and the efficiency value of the anomaly detection technique can be calculated based on the threshold t .

Suppose that the number of traces in the testing dataset is n_{Test} , the number of traces in the intrusive dataset is n_{Intru} , the number of normal traces which is classified as anomalous ones is *FalsePositives*, and the number of anomalous traces which is classified as normal ones is *FalseNegatives*, the terms in equation 1 can be expressed as:

$$\begin{aligned} \text{FalseNegativeRate} &= \text{FalseNegatives}/n_{Intru} \\ \text{FalsePositiveRate} &= \text{FalsePositives}/n_{Test} \\ P(\text{anomaly}) &= n_{Intru}/(n_{Test} + n_{Intru}) \end{aligned}$$

$$P(\text{normal}) = n_{Test}/(n_{Test} + n_{Intru})$$

Hence, equation 1 can be simplified as:

$$\text{efficiency} = 1 - \frac{\text{FalsePositives} + \text{FalseNegatives}}{n_{Test} + n_{Intru}} \quad (4)$$

In order to calculate the efficiency parameter of ScanAID, the anomaly value of every trace in the intrusive dataset and the testing dataset is calculated to form two series of anomaly values, and the minimum and the maximum of anomaly values of traces in the testing and intrusive dataset are chosen to define a range ($min - 1, max + 1$) for the threshold t . Since our objective in the experiments is to evaluate the flexibility and efficiency of ScanAID technique, we do care more about the value of efficiency than the value of threshold for this technique. For this reason, every possible threshold in the range are used to calculate the efficiency parameter for ScanAID and the largest one is chosen as the efficiency for ScanAID.

For every possible threshold t_i in the range, the number of false positives and the number of false negatives are enumerated. Then, according to equation 4, the efficiency of ScanAID is calculated for the possible threshold t_i . For the sake of robustness, the average of largest n efficiency values for different threshold values in the range ($min - 1, max + 1$) is regarded as the efficiency value for ScanAID.

4.3 Optimizing the length of N-grams

As redundant training dataset can not degrade the efficiency of an anomaly detection technique, the training dataset is chosen as large as possible. Then, a range of the length of N-grams from 1 to s is used to generate the efficiency data from the testing and intrusive datasets. Using the efficiency data, the optimized value can be picked for the length of N-grams.

4.4 Optimizing the size of training dataset

With the optimized length for N-grams, the training datasets with different sizes (*which range from near zero value to the possible largest value for the size of normal dataset*) are used to train the models for the anomaly detection technique. Then, the intrusive dataset and testing dataset are used to determine the efficiency of detection for different sizes of training dataset. In our experiments, we choose that size as optimal when an increment in the size does not change the efficiency of detection by more than a predetermined margin ε .

5 Experiments over audit trails data

For our experiments, we use the formal datasets used for evaluating stide [2]. The details about these datasets are described in [11] and [2]. In our experiments, only two parts of the dataset are used: 'named' and live 'lpr'-MIT dataset and their related intrusive datasets. The particulars of these datasets are given in table 1 [11].

Table 1. The 'named' and live 'lpr'-MIT datasets.

process	dataset	traces	system calls
Named	Normal	142	9,230,572
	Intrusive	6	1,800
lpr	Normal	2,703	2,926,304
	Intrusive	1,001	169,252

According to the need of the ScanAID technique, their normal dataset is split into training dataset and testing dataset for evaluating the efficiency and flexibility of the technique. However, the starting point to split the normal dataset can influence the efficiency of the anomaly intrusion detection technique if there is irregularity in the normal dataset. To avoid the influence, different starting points are used to get the efficiency values to optimize the two critical parameters. The mean of these efficiency values is taken as the efficiency of the ScanAID technique, which is used to optimize the length of N-grams and the size of the training dataset.

In our experiments for 'named' and 'lpr' process, the three critical control parameters for the ScanAID technique (m, n, s) pick up the following values:

The number of highest anomaly values remembered in a trace, $m=10$;

The number of highest efficiency values remembered for calculating the efficiency of ScanAID, $n=10$;

The maximum length of N-grams, $s=12$.

6 Experimental results

The ScanAID technique is implemented with C under Linux, in a Pentium III 1.0G with 256MB memory. In the following parts, the experimental results will be analyzed for optimizing the parameters for the ScanAID technique and evaluating the efficiency of the ScanAID technique. In this part, only the procedure for optimizing parameters for 'named' process is shown, and the same procedure can be applied to live 'lpr'-MIT process.

6.1 Optimizing the length of N-grams

For the optimization of the length of N-grams, the starting point for training dataset ranges from 92,305th (1%) to 9,138,266th (99%) system call with a step of 830,571 (9%), and the efficiency values for variable lengths of N-grams (1..s) are computed.

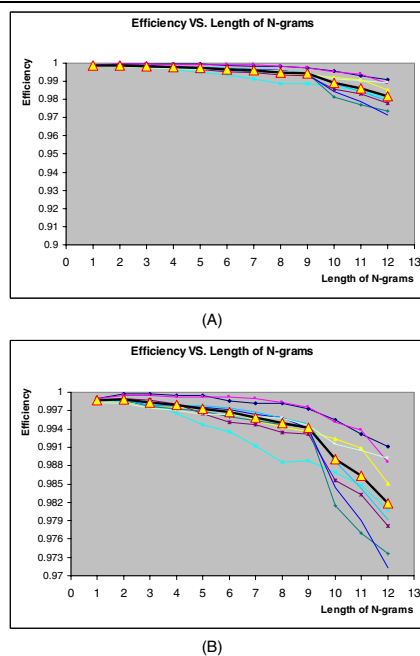


Figure 2. Length of N-gram (1~12) VS. Efficiency of ScanAID. The light lines show standard efficiency curves for different starting points for splitting the normal dataset; the dark line with big triangles shows the mean of them. Chart (B) shows the curves with the efficiency axis magnified.

From the above two charts in Figure 2, the obvious optimum value for the length of N-grams is 2. However, with the increase of the length of N-grams after the optimum

value 2, the efficiency degrades. This is because the range of anomaly values of events is shrunk with the increase of the length of N-grams making the choice of the threshold difficult. This effectively dilutes the effect of rare N-grams making it more difficult to distinguish them.

6.2 Optimizing the size of training dataset

In order to test the relationship between the size of training dataset and efficiency of the ScanAID technique, the size of training dataset changes from 184,611 (2%) system calls to 3,692,228 (40%) system calls with a step size of 369,222 (4%). The starting point for the training dataset varies from 92,305th (1%) to 9,138,266th (99%) system call with a step size of 1,292,280 (14%) to avoid its influence on the experimental results. The optimized value 2 for the length of N-gram from the last experiment is used in this experiment.

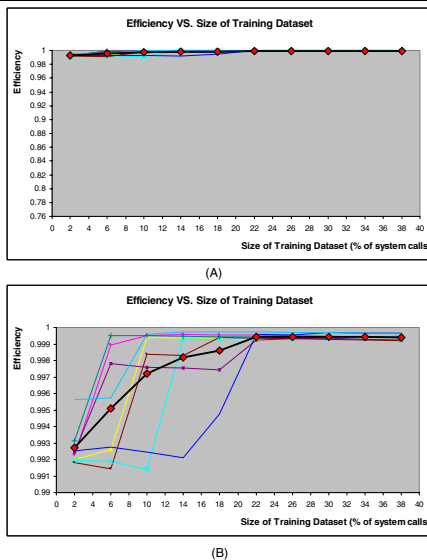


Figure 3. Size of training dataset VS. Efficiency of ScanAID (with optimized length of N-gram, 2). The light lines show standard efficiency curves with different starting points for splitting the normal dataset; the dark line shows the mean of them. Chart (B) shows the curves with the efficiency axis magnified.

As Figure 3 indicates, the optimal size of training dataset is 2,030,725 system calls (22% of the dataset) ($\epsilon = 6e - 4$). Otherwise, the efficiency will be degraded if the size is smaller than it, and the efficiency will not be improved much if the size is bigger than it.

6.3 Comparison of the performances of ScanAID and stide

The ultimate goal of the ScanAID technique is to evaluate whether it is flexible and efficient to detect anomalies in one dataset. Hence the typical ROC (Receiver Operating Characteristic) curve is used to evaluate it. In detection and estimation theory, the ROC curve is used to plot the detection rate as a function of the false positive rate [7], and it can evaluate the performance of a detection technique effectively. In this section, the typical ROC curves of ScanAID and stide are compared.

During the generation of the ROC curve, the optimized parameter values for 'named' generated from the past two experiments are used (the optimal length of N-grams is 2, and the optimal size of training dataset 2,030,725 system calls).

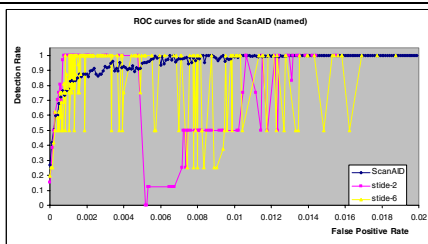
For stide, following [11] [2], the anomaly value for one trace is derived from the number of mismatches occurring in a temporally local region, called a LF (Locality Frame). Then, a LFC (Locality Frame Count) is used as the threshold to determine whether the Locality Frame is anomalous in the trace. The LFC threshold is the primary sensitivity parameter used in the experiments for stide, it ranges from 1 to LF.

In our duplicated experiment, the LF is 20, the same value as in [11]. For the convenience of computing the efficiency of stide, the maximum of mismatches in a trace is remembered as the anomaly value for the trace. After getting the anomaly values in the testing and intrusive dataset, the LFC is dynamic in a range for these anomaly values (similar to the threshold t in section 4.2) to get values for the ROC curve of stide.

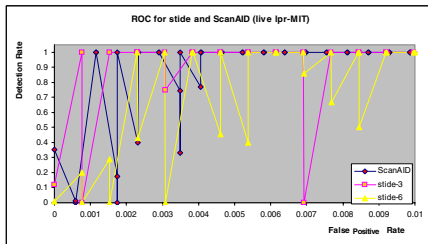
Considering that the optimized length of N-grams for ScanAID is 2, the length of N-grams for stide is also set to 2. Additionally, the magic length 6 of N-grams [5] [2] is also used for the performance comparison between stide and ScanAID to avoid the possibility that the length 2 of N-grams is not enough for modeling the 'named' process.

The same optimizing procedure in section 6.1 and 6.2 was also applied to live 'lpr'-MIT process, and we got that the optimizing length of N-grams is 3 and the optimizing size of dataset is 980 traces. For comparisons, the ROC for live 'lpr'-MIT process is also shown in our paper. The experimental results are described in Figure 4.

Comparing the three ROC curves, stide can achieve good detection rate more quickly than the ScanAID technique with small false positive rate. However, the performance for ScanAID is more stable than stide if the spikes in the ROC curves for stide are considered. Considering both performance and stability, the ScanAID technique has slight edge over stide, at least for the process 'named' and 'lpr' which are used for our experiments.



(A)



(B)

Figure 4. ROC (Receiver Operating Characteristics) curves of stide and ScanAID, in which the curves outlines comparison of ROC curves of stide (with the length of N-gram 2/3 and 6) and ScanAID (with the optimized length of N-grams and the optimized size of training dataset).

During the experiments for comparison, the sizes of the generated models in ScanAID and stide for every starting point in the normal dataset are remembered, and the averages of them are shown in table 2. As indicated by the numbers in the table, comparing with stide, for equal size N-grams (of length 2 for ‘named’ or 3 for live ‘lpr’-MIT), the size of ScanAID has not increased dramatically while it gives a more stable performance for anomaly-based intrusion detection. On the other hand, since stide need N-grams of length 6 for comparable performance, its storage requirements is much larger than that of ScanAID.

7 Conclusions and future research

In this paper we have proposed a new technique called ScanAID for anomaly detection based on statistical analysis of N-grams in the event logs of privileged processes. One of the significances of this technique is that, unlike stide, it considers the event which does not occur in the training dataset as a rare event with a minimum probability during the detection phase.

We have also provided a methodology for optimizing the length of N-grams and the size of the training dataset for obtaining best possible detection efficiency. Performance comparison with another relatively successful anomaly-based intrusion detection method, stide, shows that ScanAID provides equally efficient yet more stable de-

Table 2. Sizes of models for ScanAID and stide.

Process	Models	Integer Values	Prob Values
named	ScanAID	259.78	129.89
	stide-2	259.78	0
	stide-6	3472.92	0
lpr	ScanAID	698.67	232.89
	stide-2	698.67	0
	stide-6	2808.18	0

tection. So far the technique has been applied to host-based intrusion detection. In future, we would like to extend the technique to apply to network based intrusion detection as well.

References

- [1] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff. A sense of self for Unix processes. In *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, pages 120–128. IEEE Computer Society Press, 1996.
- [2] S. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.
- [3] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.
- [4] C. Marceau. Characterizing the behavior of a program using multiple-length n-grams. In *Proceedings of the 2000 workshop on New security paradigms*, pages 101–110. ACM Press, 2000.
- [5] R. Maxion and K. Tan. Benchmarking anomaly-based detection systems. In *Proceedings of the 1st International Conference on Dependable Systems & Networks 2000*, pages 623–630, Los Alamitos, California, 2000.
- [6] R. Power. Computer security issues&trends. 2002 CSI/FBI computer crime and security survey, Computer Security Institute, 600 Harrison St. San Francisco CA 94107, 2002.
- [7] R.S.Puttini, Z. Marrakchi, and L. Mè. A bayesian classification model for real-time intrusion detection. In *the 22nd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, MAXENT*, 2002.

- [8] S. Scott. A bayesian paradigm for designing network intrusion systems. To appear in *Computational Statistics and Data Analysis* (special issue on network intrusion detection), June 20 2002.
- [9] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 144–155, 2000.
- [10] M. Stillerman, C. Marceau, and M. Stillman. Intrusion detection for distributed applications. *Communications of the ACM*, 42(7):62–69, 1999.
- [11] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *IEEE Symposium on Security and Privacy*, pages 133–145, 1999.
- [12] N. Ye. A markov chain model of temporal behavior for anomaly detection. In *Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, 2000*, pages 171–174, 2000.