Research On Markov Chain Model for System Call Anomaly Detection[1]

QIAN Quan, WANG Xu-fa

(Department of Computer Science, University of Science & Technology of China, Hefei 230027, China)

Abstract

Intrusion detection, especially anomaly detection, requires sufficient security background knowledge. It is very significant to identify system anomaly behavior under the special condition of poor domain knowledge. In this paper, Markov chain model used for anomaly detection is deeply discussed from three aspects; those are one-step, multi-step Markov chain and multi-step Markov chain-based sequence prediction. The Sendmail and Wu-ftp privilege program experiments show these methods can detect the anomaly behavior of program, which do not necessarily need any security knowledge.

Keywords: Markov Chain Model, Anomaly Detection

1. Introduction and Related Work

Any computer process, in nature, is a program. Without selection and iteration sentences, the behavior of program is foreseeable. If a process is attacked, i.e. buffer overflow, the normal execution order broken, and we can detect the attack through analyzing the system call sequences. Although system call sequences are somewhat random when concerning selection and iteration sentences, some researches indicate when the amount of system calls reaching certain number, the number of certain length short sequences that generated by sliding window is relatively stable. For instance, if system calls of Sendmail reach 7 million, the short sequences of length 6 are about 1000. Hence, process system call shows certain randomness, but it also has some intrinsic feature which can be used to detect process attacked or abnormal behavior.

About intrusion detection for host system calls, at the very beginning, S. Forrest etc. adopted immune "self" method to detect "nonself". They only depended on short sequence recognition method, which did not consider the frequency and distribution of short sequences[1]. System call database stored series of lookahead pairs with length 0,1,2,…,k. Hofmeyr etc. improved the method that pattern database stored length-fixed continuous short sequences, instead of lookahead pairs and the recognition precision took great improvement[2].

Frequency-based method, initially used in text classification, Helman & Bhangoo put forward frequency-based sequence sort, which mainly analyze the frequency of process normal system calls and rank the frequency by its value. The ranked frequency is used as the evidence for anomaly detection[3]. Wenke Lee employed RIPPER rule learning system in data mining, and learned rules from normal process running patterns, which used as the basis of process monitoring. If there is rule violation, the process is in abnormal[4,5]. Since the incompleteness of short sequence analysis by Stephanie Forrest which need to collect all possible short sequences, Wenke Lee's sliding window sequence analysis method, which not only can detect the appearance number of anomaly sequences, but the distribution discipline of anomaly sequences, overcome the incompleteness of the short sequence method and get a great improvement in detection precision. In [6], it presents a finite state machine(FSM) to construct system calls description language, but the efficiency and practicability are not so good. The [7] provided the online intrusion detection based on system calls.

Markov process model, as a statistic method, has been widely used in sound recognition, information retrieval and some other classification community. Terran Lane used hidden Markov method to model the human-computer interface[8], but Markov model used in network security are not focused sufficiently[9,10].

Anomaly detection, through predefining system "normal" patterns, for example, CPU and Memory utilization ratio, file checksum, etc. compares the system running parameters with the predefined values to get the result of whether system has been attacked[11]. The difficulty of anomaly detection is to determine the system "normal" parameters, which requires sufficient security background knowledge.

[1] This research was supported by NSF Key Research Plan (No. 90104030) and " 21 Century Education Development Plan".

Therefore, it is very significant to identify system anomaly behavior under the condition of poor domain knowledge. In this paper, Markov chain model used for anomaly detection is deeply discussed from three aspects; one-step, multi-step Markov chain and multi-step Markov chain-based sequence prediction. Through system call experiments of Sendmail and Wu-ftp privilege programs, it shows these methods can detect anomaly behavior of system, which do not necessarily need any background security knowledge.

In this paper, section 2 introduces Markov model; Section 3 indicates Markov model for anomaly detection; Section 4 evaluates methods through several experiments; Section 5 gives conclusions.

2. Markov Chain Model

Markov chain is a special case of Markov stochastic process, which state and time parameters are discrete. Stochastic sequence X_u is a Markov Chain, which at any time *t*, can stay at state of $\theta_1,\ldots,\theta_T$, and the state probability of the time of *m+k* is only relevant to the state of the time of *m*, while independent from the state of time before *m*. In practice, each state of Markov chain corresponds to a observable physical event. All state transfer can be described by state transfer matrix. So, state transfer matrix, joining with initial state probability distribution, can depict Markov chain completely.

Hidden Markov Chain, is developed on the basis of Markov chain. Because practical problems are more complicated than what Markov chain depicts, the observed events correspond to states through definite probability distribution, not one by one. HMM can be regarded as a dual stochastic process, one of them is Markov chain which depicting state transfer, the other shows the statistic relation between state and observed value.

3. Markov Chain Model for Anomaly Detection

Among attacking procedures, most of them consist of series of related behavior. Features of related behavior exist obvious difference between normal system calls and attacked ones. Markov chain model, which applied to anomaly detection, makes use of the differences. A state corresponds to a kind of system event and state transition is depicted by state transition probability matrix. An event is a abnormal, only if state transition probability matrix affirms that the transition probability is less than a given value. In this paper, for Sendmail and Wu-ftp programs, events (or states), are system calls. The relativity difference between normal system calls and attacked process will be used to discover the program anomaly behavior.

3.1. Building Markov Chain Model

First of all, to compute single-step transition probability matrix *P* through statistic method. P_{ij} denotes the probability that system stays at state *i* at the time of *t* and at state *j* at the time of *t+1*. The computing function is:

$$P_{ij} = N_{ij} / N_i \qquad (1)$$

N_{ij} is the continuous appearance count of system call *i* and *j*; N_i is the count of system call *i*.

Besides, supposing that the initial state probability vector is $\pi = (\pi_1,\ldots,\pi_n)$, in which:

$$\pi_i = P(\pi_1 = \theta_i) = N_i / N, \quad 1 \le i \le n \qquad (2)$$

Among them, N_i is the count of system call *i*; *N* is the whole count of system calls. Obviously,

$$0 \le \pi_i \le 1 \text{ and } \sum_i \pi_i = 1$$

Moreover, single-step Markov chain model exists two basic hypotheses: (1) State probability distribution of the time of *t+1* is only related to the time of *t*, not the time before *t*. (2) State transition probability from *t* to *t+1*, is independent from *t*. For system call sequences, the two hypotheses are not strictly tenable. We should consider influences of several states ahead of the state. For instance, in sequence $(S_1, S_2, \cdots, S_{t-1}, S_t)$, no only S_{t-1} affects S_t, states before S_{t-1} can also affect it. So we should compute multi-step state transition probability. From the theory of Markov multi-step transition probability, $P^{(t+s)} = P^{(t)} \times P^{(s)}$, we can use single-step transition probability matrix to get multi-step one.

3.2. Using Markov Chain to Analyze Data

3.2.1 Sliding Window Sequences

Firstly, using sliding window to divide long sequence, we can get small sequence sets. Window division, generally, has two methods, which are time window and amount window. Time window uses a definite time interval (i.e. 2 seconds) to divide sequences. Here, we use amount window to divide long system call sequences. For example, using the length *t* sliding window, the sequence $(S_1, S_2, \ldots, S_t, S_{t+1}, \ldots, S_{t+s})$, can be divided into *s+1* sequence sets with length *t* :

$$(S_1, S_2, ..., S_t);\ (S_2, S_3 ..., S_{t+1});\ \cdots\cdots$$
$$(S_{s+1}, S_{s+2}, ..., S_{t+s});$$

3.2.2 Analysis Method

After building Markov chain model of normal system calls, we can use it to analyze data. Analyzing methods are as follows:

(1) Single-Step Markov Chain to Compute Sequence Support Degree

For length *t* sequence $(S_1, S_2, ..., S_t)$, Compute the support degree of normal system patterns. The formula is,

$$P(S_1, S_2, ..., S_t) = \pi_{S_1} \times \sqrt[t-1]{P_{S_1S_2} \times \cdots \times P_{S_{t-1}S_t}} \qquad (3)$$

If the support degree is less than a given threshold value, the sequence is abnormal one, otherwise, a normal one. For all sequences, we can find that the support degree of abnormal system behavior is far bigger than normal one.

(2) Multi-step Markov Method

Because single-step Markov chain's hypotheses do not satisfy in practice, we should adopt multi-step Markov method. For system call sequence $(S_1, S_2, ..., S_t)$ with length *t*, compute appearing probability $P(S_{t+1} \mid S_1, \cdots, S_t)$ of the next system call under the normal system condition. Here, $P(S_{t+1} \mid S_1, \cdots, S_t)$ is the weight average of the transition probability from $S_i\ (1 \le i \le t)$ to S_{t+1}, the computation formula is:

$$P(S_{t+1} \mid S_1, \cdots, S_t) = \frac{\sum_{i=1}^{t} W_i P_{S_iS_{t+1}}}{\sum_{i=1}^{t} W_i} \qquad (4)$$

In which, $P_{S_iS_{t+1}}$ is the probability that the system call is S_i at time *i* and S_{t+1} after *t+1-i* (namely, at time *t+1*). It can be computed by *t+1-i* step state transition probability matrix. W_i is the weight of transition probability from state S_i to S_{t+1}, which represents the influence of S_i on S_{t+1}, at *t+1-i* distance from S_{t+1}. The shorter distance, the bigger influence. So, as *i* becomes bigger, the distance gets shorter and the weight bigger. Here, $W_i = 1^2 (1 \le i \le t)$. If the probability is less than a given threshold value, the sequence is regarded as abnormal, otherwise, normal.

(3) Multi-step Markov Chain based Sequence Prediction

It is using length *t* normal pattern's system call sequence $(S_1, S_2, ..., S_t)$ to predict the next system call S_{t+1}. In concrete, when $\sum_{i=1} W_i P_{S_iS_{t+1}}$ gets the maximal value, the corresponding S_{t+1} makes the state estimate of the time of *t+1* (here W_i is weight, the same as formula (4)). For S_{t+1}, it has,

$$S_{t+1} = \arg\max_{1 \le i \le t} (\sum_i W_i P_{S_iS_{t+1}}) \qquad (5)$$

In practical sequence, when S_{t+1} is consistent with the prediction, then the sequence is normal, otherwise, abnormal. Predicting all sequences, we can find that using normal system patterns to predict the normal system behavior, the rate of abnormal sequences is far less than to predict the attacked system behavior.

3.3 Algorithm Complexity Analysis

The whole algorithm can be divided into 2 parts, Markov chain model building and data analysis. Detailed algorithm can be found in [12]. In model building step, simply scan training data once, the complexity is *O(n)*. Single-step Markov computing sequence support degree, also need to scan training data once, the complexity is also *O(n)*. Multi-step Markov method scan training data once, and for each sequence, compute the weight average probability of the next system call, the complexity is *O(n*t)(here t is sliding window size)*. Multi-step Markov chain based sequence prediction need to scan data once, but at each prediction need to compute the sum of *1* to *t-1* step state transition probability, and the complexity is *O(n*s*t)*. Here *s* is the number of possible appearance system calls, and *t* is the size of sliding window.

4. Experiments and Result Discussion

To evaluate the validity of above models, we use Solaris's Sendmail and Wu-ftp system call records as data source. Solaris operating system, itself, has a security component BSM (Basic Security Module), which can record events related to system security. BSM can record 284 sorts of security audit events, which can be seemed as 284 sorts of possible states of hosts. Each event consists of Event kind, User ID, Group ID, Process ID, etc. In our experiments, we only use Event kind, namely the Process ID.

To gather hosts' security audit data is a complicated task, which should consider almost all possible "normal" behavior of user and system. University of New Mexico and CERT have done lots of works in this area and a great deal of system call data can be available at [13].

In experiments, for Sendmail privilege program, we use two groups of normal system calls, *normal1* and *normal2* as training data, each consists of 1.5

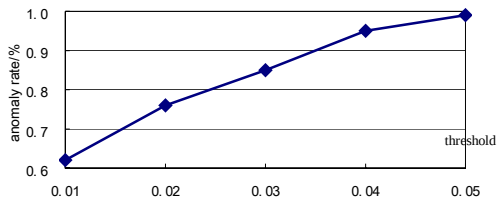
million and 1.7 million system calls. Three groups of abnormal system calls, *intrusion1*, *intrusion2* and *intrusion3*,are regarded as testing data, that each consists 6504, 1812 and 6755 system calls.

4.1. Sequence Support Degree of Normal Pattern

4.1.1 The Relation Between Different Threshold Value and Abnormal Sequence Rate

From experiment, it shows that selecting different threshold value, we can get different abnormal rates. The reason is different threshold value which represents different normal standards. For *intrusion1*, the relation between threshold value and abnormal rate can be depicted in figure 1 *(Here sliding window size is 4).*

Figure 1, Relation between abnormal rate and threshold value



From figure 1, it indicates when the threshold value is bigger; the rate of abnormal sequences is bigger.

4.1.2 The Relation Between Different Window Size and Abnormal Sequences Rate

Given different size of sliding window, we compute the rate of abnormal sequences. The results of 5 groups data are showed in table 1. (In table 1, 2.63 represents that 2.63% sequences are abnormal, and support degree is 0.01.)

Table 1 Analysis result of sequence support degree of SendMail program

Window size （T）	Normal system call （%）	Abnormal system call （%）
2	2.63	61.60
3	2.91	66.36
4	2.71	65.03
5	2.53	62.66
6	2.51	62.33
7	2.50	62.41

In table 1, Normal system call (%) is the average of *normal1* and *normal2*, Abnormal system call (%) is the average of 3 testing data, *intrusion1*, *intrusion2* and *intrusion3*. From table 1, it shows that abnormal rate of attacked sequences is 24 times higher than normal system calls. The support degree of single-step Markov chain model, for normal and abnormal behavior, exists obvious difference, and the result is somewhat independent of sliding window size.

4.2. Multi-step Markov Method to Compute Abnormal Rate

Multi-step Markov method computes the appearance probability of the last system call in a window. If the probability is less than a given threshold, the sequence is abnormal and count the rate of all abnormal sequences at last. As well, threshold determines the “abnormal” standard, it impact the final results directly. Experimental results are depicted in table 2 (threshold value is 0.15).

Table 2 Multi-step Markov method for Sendmail abnormal sequence rate

Window size （T）	Normal System Call （%）	Abnormal system call （%）
2	0.97	24.7
3	0.94	24.0
4	0.91	24.0
5	0.91	24.0
6	0.94	24.2
7	0.95	24.4

From table 2, it concludes that multi-step Markov method is slightly superior to single-step one, the rate difference between them is about 25 times. Also, the window size does not impact result significantly.

4.3. Multi-step Markov Chain based Sequence Prediction

Multi-step Markov chain based sequence prediction is to estimate the next system call based on the normal system call sequences. The difference rates of prediction are expressed in table 3.

Table 3 Multi-step Markov chain based prediction for Sendmail abnormal sequence rate

Window size （T）	Normal System Call （%）	Abnormal System Call （%）
2	35.9	48.7
3	35.9	50.3
4	23.4	44.5
5	39.3	52.7
6	42.5	53.8
7	42.5	55.4

Based on this method, there is about 15%~20% difference between normal system calls and abnormal

ones. Moreover, window size does not greatly impact the result, and size 4 is the maximal difference.

Further, in order to evaluate the model, we test another privilege program Wu-ftp. Also, using normal system call sequences as training data, attacked data as testing data. In which, training data contain 179,916 system calls, testing data contain 1,363. We use the first analysis method to compute normal sequence support degree. The experimental results are in table 4.(The threshold value of support degree is 0.01)

Table 4 Result of sequence support degree of Wu-ftp

Window Size（T）	Normal System Call（%）	Abnormal System Call（%）
2	3.83	66.10
3	3.38	66.74
4	4.20	66.49
5	4.07	66.98
6	4.57	66.37
7	4.24	66.35

From table 4, we can see that, for Wu-ftp, anomaly rate of abnormal system calls is 16 times higher than of normal ones. Obviously, using single-step Markov chain model, there exists an obvious support degree difference between normal and abnormal behavior of Wu-ftp. Furthermore, the variant size of sliding window does not impact the result greatly.

4.4 Analysis of Realtime Detection

The realtime capability affects intrusion detection system the practicability directly. So, we experiment the above three detection methods with different window size to test the real processing time. Software is complied by Microsoft Visual C++ 6.0, and experimental computer is PIII 800 with 256M memory, 30G hard disk. Processing time is the average of 2 groups of normal system calls and 3 groups of abnormal system calls separately. Real processing time of three analysis methods are described in table 5, 6 and 7.

Table 5 Processing time for single-step Markov chain to Sendmail

Window Size（T）	Normal System Calls(s)	Abnormal System Calls(ms)
2	5.69	50
3	5.71	49
4	5.74	49
5	5.69	55
6	5.75	50
7	5.75	49

Table 6 Processing time for multi-step Markov model to Sendmail

Window Size（T）	Normal System Calls(s)	Abnormal System Calls(ms)
2	3.72	74
3	3.99	40
4	4.04	80
5	4.15	36
6	4.31	57
7	4.40	40

Table 7 Processing time for multi-step Markov chain based sequence prediction to Sendmail

Window Size（T）	Normal System Calls(s)	Abnormal System Calls(ms)
2	40.0	121
3	56.2	169
4	79.4	242
5	99.1	304
6	127.1	388
7	148.2	453

From above experimental results, single-step Markov chain that computing sequence support degree, the average time of training step is 5.7 seconds, while testing step needs 50 milliseconds. Multi-step Markov method that computing anomaly sequence rate, training step is 4.1 seconds, and testing step is 55 milliseconds. Moreover, the two methods are almost not relevant to window size. Concerning Multi-step Markov chain based prediction, the minimal training time needs 40 seconds, and minimal testing time is 121 milliseconds. In table 7, the processing time is close relevant to the size of sliding window. The reason is , in multi-step Markov method, each iteration needs to compute the probability of every state. Under the condition of big state space, computing complexity is big, and real time capability is not good.

5. Conclusions

From all above experiments, we can see that single-step Markov chain, used to compute the abnormal sequence support degree by normal system calls, the difference is about 20 times. And multi-step method is about 25 time, while multi-step based prediction is 15%~20%. All three methods can detect the system anomaly behavior.

Secondly, different size of sliding window does not obviously affect experimental results. But different threshold value determines the standards of “normal” and “abnormal”, which influence results greatly.

Thirdly, so as to detection realtime, because of different computation complexity for three methods, Multi-step Markov chain based prediction is the most complicated one. So, single-step Markov chain and

multi-step Markov chain are good to anomaly detection both on precision and computation complexity, which training step needs about 4~6 seconds, and testing step about 50 milliseconds. It concludes, when training step is “offline” and detecting step is “online”, these 2 methods can satisfy the realtime requirement for intrusion detection.

Finally, all above experiments are conducted without any attack background knowledge. Only depending on security audit data (system call sequences), we can find the system abnormal behavior. If we compare system calls with user operations, perhaps Markov method, described above, is also efficient to modeling user behavior and detecting of user anomaly command sequences. Therefore, we can say that Markov model is a good statistic model for anomaly detection.

References

[1] S.Forrest, S.A.Hofmeyr, A.Somayaji, etc. A sense of self for unix processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, CA:IEEE Computer Society Press, pp.120~128, 1996.

[2] S.A.Hofmeyr, S. Forrest, A. Somayaji. Intrusion detection using sequences of system calls. Journal of Computer Security, 6:151--180, 1998.

[3] P.Helman, J.Bhangoo. A statistically based system for prioritizing information exploration under uncertainty. IEEE Transactions on Systems, Man and Cyberneticsm, Part A: Systems and Humans,27(4):449~466,1997.

[4] W.Lee, S.J.Stolfo, P.K.Chan. Learning Patterns from UNIX process execution traces for intrusion detection. In AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, AAAI press:pp50~56,1997.

[5] W.Lee,S.J.Stolfo. Data Mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, 1998.

[6] C.Warrender, S.Forrest, B.Pearlmutter. Detecting intrusions using system calls: Alternative data models. In Proceedings of IEEE Symposium on Security and Privacy, pages 133--145, Oakland, California, May 9-12 1999.

[7] D.Snyder. On-line Intrusion Detection Using Sequences of System Calls. Master's thesis, Department of Computer Science, Florida State University, 2001.

[8] T.Lane. Hidden markov models for human/computer interface modeling. In Proceedings of the IJCAI-99 Workshop on Learning about Users, pp 35-44. 1999.

[9] Nong Ye. A Markov chain model of temporal behavior for anomaly detection[A]. Proceedings of the 2000 IEEE workshop on Information Assurance and Security[C].2000.171-174.

[10] ChangChun Zou. Using hidden Markov model in anomaly intrusion detection[J/OL]. Available at :http://tennis.ecs.umass.edu/~czou/research/HMM.

[11] R. Anderson, A. Khattak. The use of information retrieval techniques for intrusion detection. Web proceedings of the First International Workshop on Recent Advances in Intrusion Detection (RAID'98), Available at: http://www.raid-symposium.org/raid98.

[12] Qian quan, “Research on Intelligent Intrusion Detection Technology”, PhD thesis, Department of Computer Science & Technology, University of Science & Technology of China, China, 2003.

[13] Computer Immune Systems, Data Set and Software. Available at http://www.cs.unm.edu/~immsec/systemcalls.htm