WA1-1 8:00
Proceedings of the 2000 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 6-7 June, 2000

A Markov Chain Model of Temporal Behavior for Anomaly Detection

Nong Ye
Department of Industrial Engineering
Arizona State University
Tempe, Arizona, USA
Email: *nongye@asu.edu*

Abstract

This paper presents an anomaly detection technique to detect intrusions into computer and network systems. In this technique, a Markov chain model is used to represent a temporal profile of normal behavior in a computer and network system. The Markov chain model of the norm profile is learned from historic data of the system's normal behavior. The observed behavior of the system is analyzed to infer the probability that the Markov chain model of the norm profile supports the observed behavior. A low probability of support indicates an anomalous behavior that may result from intrusive activities. The technique was implemented and tested on the audit data of a Sun Solaris system. The testing results showed that the technique clearly distinguished intrusive activities from normal activities in the testing data.

Keywords: Markov chain, intrusion detection, and anomaly detection.

1 Introduction

There are two general methods of detecting intrusions into computer and network systems: anomaly detection and signature recognition [1-9]. For a subject (e.g., user, file, privileged program, host machine, and network) of interest, anomaly detection techniques establish a profile of the subject's normal behavior (norm profile), compare the observed behavior of the subject with its norm profile, and signal intrusions when the subject's observed behavior differs significantly from its norm profile. Signature recognition techniques recognize signatures of known attacks, match the observed behavior with those known signatures, and signal intrusions when there is a match.

Since many intrusions are composed of a series of related computer actions, the temporal profile of action sequence (the temporal behavior profile) is important to detect intrusions. The norm profile of temporal behavior should capture the temporal dependency among computer actions during the normal usage of a computer and network system.

We developed an anomaly detection technique that represents the norm profile of temporal behavior using a Markov chain model, learns the Markov chain model from computer audit data, and detects anomalies based on the Markov chain model of temporal behavior. This technique is presented in this paper. Section 2 describes a Markov chain model. Section 3 defines the intrusion detection problem using the Markov chain model. Section 4 presents and discusses the results of testing the technique.

2 Markov Chain Model

A discrete-time stochastic process specifies how a random variable changes at discrete points in time. Let X_t denote a random variable representing the state of a system at time t, where t = 0, 1, 2, …. A stationary Markov chain is a special type of discrete-time stochastic process with the following assumptions [10]:

- the probability distribution of the state at time t+1 depends on the state at time t, and does not depend on the previous states leading to the state at time t;
- a state transition from time t to time t+1 is independent of time.

Let p_{ij} denote the probability that the system is in a state j at time t+1 given the system is in state t at time t. If the system has a finite number of states, 1, 2, …, s, the stationary Markov chain can be defined by a transition probability matrix [10]:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1s} \\ p_{21} & p_{22} & \cdots & p_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ p_{s1} & p_{s2} & \cdots & p_{ss} \end{bmatrix} \qquad (1)$$

and an initial probability distribution [10]:

$$Q = \begin{bmatrix} q_1 & q_2 & \cdots & q_s \end{bmatrix} \qquad (2)$$

where q_i is the probability that the system is in state i at time 0, and

$$\sum_{j=1}^{j=s} p_{ij} = 1. \qquad (3)$$

The probability that a sequence of states X_1, …, X_T at time 1, …, T occurs in the context of the stationary Markov chain is computed as follows:

$$P(X_1, \cdots, X_T) = q_{X_1} \prod_{t=2}^{T} P_{X_{t-1} X_t} \qquad (4)$$

The transition probability matrix and the initial probability distribution of a stationary Markov chain can be learned from the observations of the system state in the past. Provided with the observations of the system state X_0, X_1, X_2, …, $X_{N\text{-}1}$ at time t = 0, …, N-1, we learn the transition probability matrix and the initial probability distribution as follows [11]:

$$p_{ij} = \frac{N_{ij}}{N_{i.}} \tag{5}$$

$$q_i = \frac{N_i}{N} \tag{6}$$

where

N_{ij} is the number of observation pairs X_t and X_{t+1} with X_t in state i and X_{t+1} in state j;
$N_{i.}$ is the number of observation pairs X_t and X_{t+1} with X_t in state i and X_{t+1} in any one of the states 1, …, s;
N_i is the number of X_t's in state i; and
N is the total number of observations.

3 Problem Definition

Two sources of data have been widely used to capture activities in a computer and network system for intrusion detection: network traffic data and audit trail data (audit data). In this study, we used audit data from a UNIX-based host machine (specifically a Sun SPARC 10 workstation with the Solaris operating system), and focused on intrusions into a host machine that left trails in the audit data.

The Solaris operating system from the Sun Microsystems Inc. has a security extension, called the Basic Security Module (BSM). The BSM extension supports the monitoring of activities on a host by recording security-relevant events. Since there are about 284 different types of BSM audit events on our host machine, we consider 284 event types in this study. An BSM audit record for each event contains a variety of information, including the event type, user ID, group ID, process ID, session ID, the system object accessed, etc. In this study, we extracted and used only the event type that was one of the most critical characteristics of an audit event. Hence, activities on a host machine were captured through a continuous stream of audit events, each of which is characterized by the event type.

Both normal and intrusive activities on a host machine contain sequences of computer actions. Sequences of computer actions induce sequences of audit events. Considering the 284 types of audit events as the 284 possible states of a host machine, the temporal behavior of the host machine can be represented as a discrete-time stochastic process. Discrete points in time for the discrete-time stochastic process are defined not by a fixed time interval but by times when audit events take place.

For an intruder or a normal user, what action to take next is related to the last action as well as preceding actions. This implies a higher order of dependency than the dependency in a Markov chain. Hence, the assumption of a Markov chain as described by formula (1) does not hold for the temporal behavior of the host machine. Although a high-order stochastic process model is appropriate to account for the high-order dependency, a high-order stochastic process model is practically undesirable for its model complexity and computational cost, especially when we deal with a large set of data such as the audit data. Moreover, it is not clear what order of dependency is sufficient to describe the temporal behavior of the host machine.

In this study, we used a Markov chain instead of a high-order stochastic process model to represent the temporal behavior of the host machine. We also made the stationary assumption, that is, assuming that the user's action sequence was not related to the time of using the host machine.

For intrusion detection, we wanted to build a long-term norm profile of temporal behavior, and to compare the temporal behavior in the recent past to the long-term norm profile for detecting a significant difference. Using formulae (5) and (6), we trained and built a stationary Markov chain model (simply referred to as a Markov model) of temporal behavior as the long-term norm profile by learning the transition probability matrix and the initial probability distribution from a stream of audit events that was observed during the normal usage of the host machine.

We defined the temporal behavior in the recent past by opening up an observation window of size N on the continuous steam of audit events to view the last N audit events from the current time t:

$E_{t\text{-}(N\text{-}1)=t\text{-}N+1}$, …, E_t, where E stands for event.

In this study, we let N equal to 100, because we observed that many attack scenarios produced about 100 audit events each scenario in average.

For the audit events $E_{t\text{-}99}$, …, E_t in the window at time t, we examine the type of each audit event and obtain the sequence of states $X_{t\text{-}99}$, …, X_t appearing in the window, where X_i is the state (the type of audit event) that the audit event E_i takes. Using formula (6), we compute the probability that the sequence of states $X_{t\text{-}99}$, …, X_t occurs in the context of the normal usage, that is, the probability that the Markov model of the norm profile supports the sequence of states $X_{t\text{-}99}$, …, X_t.

$$P(X_{t-99}, X_1, \cdots, X_t) = q_{X_{t-99}} \prod_{i=t-98}^{t} p_{X_{i-1}X_i}$$

The higher probability we get, the more likely the sequence of states results from normal activities. A sequence of states from intrusive activities is expected to receive a low probability of support from the Markov model of the norm profile.

It is possible that a sequence of states from a window of the testing data presents an initial state and/or some state transitions which are not encountered in the training and thus have the probabilities of zero in the initial probability distribution or the transition probability matrix of the Markov model. While using formula (4) to infer the probability of support to the sequence of states, the probabilities of zero would dominate the final probability result from formula (4) and make it zero, regardless of the number of non-zero elements in the computation using formula (4). In this study we assigned the small probability value of 0.00001 (or 1E-5 in the scientific expression) to the initial state and state transitions which did not appear in the training data, while using formula (4) to infer the probability of support to a sequence of states. This replacement of zero with a small probability value took place during the inference and after the learning of the Markov model from the training data was completed.

Audit data of normal activities are required for learning a Markov model of the norm profile. In this study, we used a sample of audit data for normal activities from the MIT (Massachusetts Institute of Technology) Lincoln Lab, containing a stream of 3019 audit events. We used the first part of the audit data, consisting of 1613 audit events, for training a Markov model of the norm profile. The second part of the audit data, consisting of 1406 audit events, was used for testing.

The testing data contained the audit data of both normal and intrusive activities. The testing data intrusive activities were generated by simulating 15 intrusion scenarios that we collected over years from various information sources. Some examples of the intrusion scenarios are password guessing, using symbolic links to gain root privileges, attempts to gain an unauthorized access remotely, etc. We simulated these intrusion scenarios in a random order on our host machine to obtain the audit data of these intrusions which included a stream of 1751 audit events.

Hence, we learned and obtained a Markov model of the norm profile using the training data that consisted of 1613 audit events for normal activities. The testing data included the audit data of 1751 audit events from intrusive activities and 1406 audit events from normal activities, which corresponded t0 1652 windows (window no. 1-1652) for intrusive activities and 1307 windows (window no. 1-1307) for normal activities respectively. A sequence of states in each window was evaluated against the Markov model of the norm profile to yield the probability of support.

4 Results and Conclusion

The learning and inference algorithms of the Markov model for intrusion detection were implemented using C++. Figure 1 shows the probabilities that the Markov model of the normal profile supported the testing data from the normal activities. Figure 2 shows the probabilities that the Markov model of the norm profile supported the testing data from the intrusive activities.

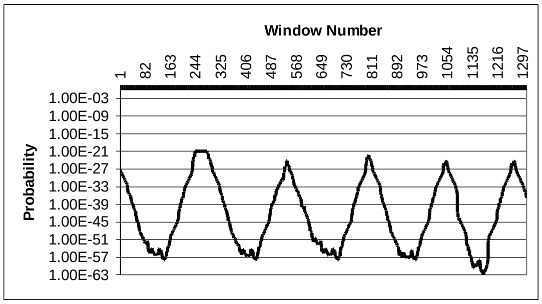

Figure 1: The probabilities of support to the testing data from normal activities

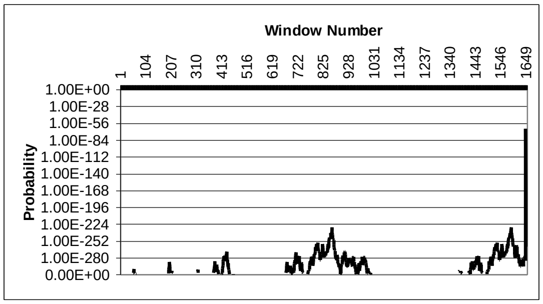

Figure 2: The probabilities of support to the testing data from intrusive activities

As shown in Figures 1 and 2, the probabilities of support to the testing data from the normal activities were much higher than the probabilities of support to the testing data from the intrusive activities. There existed a clear gap between the minimum probability (1.33E-63) for the normal data from the normal activities and the maximum probability (3.88E-65) for the testing data from the intrusive activities. Hence, the probabilities for the normal activities were clearly separate from the probabilities for the intrusive activities. By using any probability value in the gap as the decision threshold to signal intrusions, we were able to clearly distinguish the normal activities from the attack activities with the 0% false alarm rate and the 100% detection rate.

This study has demonstrated the promising performance of the intrusion detection technique based on the Markov model of temporal behavior. The application of the intrusion detection technique using a Markov model of the temporal norm profile is not limited to the temporal behavior of a host machine. The technique is also applicable to the temporal behavior data from other subjects of a larger scale (e.g. a network domain) or a smaller scale (e.g. user, file, and privileged program).

Acknowledgement

The author would like to thank Naiqi Wu for his programming support to this work. This work is sponsored by the Air Force Research Laboratory – Rome (AFRL-Rome) under agreement number F30602-98-2-0005, and the Air Force Office of Scientific Research (AFOSR) under grant number F49620-99-1-0014. The U.S. government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of AFRL-Rome, AFOSR, or the U.S. Government.

References

[1] M. Bishop, S. Cheung, et al. The Threat from the Net. *IEEE Spectrum*, 38(8), 1997.

[2] S. Forrest, S. A. Hofmeyr, and A. Somayaji. Computer immunology. *Communications of the ACM*, 40(10): 88–96, October 1997.

[3] A. K. Ghosh, J. Wanken, and F. Charron. Detecting anomalous and unknown intrusions against programs. In *Proceedings of the 1998 Annual Computer Security Applications Conference* (ACSAC'98), December 1998.

[4] T. F. Lunt. IDES: An Intelligent System for Detecting Intruders. In *Proceedings of the Symposium: Computer Security, Threats and Countermeasures*, Rome, Italy, November 1990.

[5] P. A. Porras and P. G. Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the 20th National Information Systems Security Conference*, pp. 353–365, October 1997.

[6] K. Ilgun. Ustat: A real-time intrusion detection system for UNIX. *Master's thesis*, *Computer Science*, UCSB, July, 1992.

[7] K. Ilgun, A. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection system. *IEEE Transactions on Software Engineering*, 21(3), March 1995.

[8] W. Lee, S. Stolfo, and P. K. Chan. Learning patterns from UNIX process execution traces for intrusion detection. In *Proceedings of AAAI97 Workshop on AI Methods in Fraud and Risk Management*, 1997.

[9] N. Ye, G. Giordano, and J. Feldman. "Detecting information warfare attacks: Current state of the art from a process control viewpoint". *Communications of the ACM*, in press.

[10] W. L. Winston, *Operations Research: Applications and Algorithms.* Belmont, CA: Duxbury Press, 1994.

[11] T. M. Mitchell, *Machine Learning.* Boston, MA: McGraw-Hill, 1997.