Applied Mathematics and Computation 205 (2008) 562–567

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

# Hidden semi-Markov model for anomaly detection

Xiaobin Tan *, Hongsheng Xi

*Department of Automation, University of Science and Technology of China, Hefei 230027, Anhui, PR China*

ARTICLE INFO

*Keywords:*
Intrusion detection
Anomaly detection
Hidden semi-Markov model (HSMM)
Maximum entropy principle (MEP)
Segmental K-means algorithm

ABSTRACT

In this paper, hidden semi-Markov model (HSMM) is introduced into intrusion detection. Hidden Markov model (HMM) has been applied in intrusion detection systems several years, but it has a major weakness: the inherent duration probability density of a state in HMM is exponential, which may be inappropriate for the modeling of audit data of computer systems. We can handle this problem well by developing an HSMM for perfect normal processes of computer systems. Based on this HSMM, an algorithm of anomaly detection is presented in this paper, which computes the distance between the processes monitored by intrusion detection system and the perfect normal processes. In this algorithm, we use the average information entropy (AIE) of fixed-length observed sequence as the anomaly detection metric based on maximum entropy principle (MEP). To improve accuracy, the segmental K-means algorithm is applied as training algorithm for the HSMM. By comparing the accurate rate with the experimental results of previous research, it shows that our method can perform a more accurate detection.

© 2008 Elsevier Inc. All rights reserved.

## 1. Introduction

Intrusion detection [1] attempts to detect possible attacks against software systems in real time before computer systems are compromised. It can be classified into two categories based on its modeling methods: anomaly detection techniques, and misuse detection techniques. Anomaly detection techniques scan audit trails for deviations from the profile of normal or expected behavior that created before. It has been applied to several types of audit trails, e.g., user behavior, privileged process behavior, and network traffic, etc. Misuse detection techniques scan audit trails for specific descriptions or patterns that are indicative of known attacks. These methods include expert systems [2,3], pattern matching based on Petri nets [4], and monitoring for the use of particularly dangerous system calls [5,6], etc.

The most prominent disadvantage of misuse detection approaches is that only pre-trained attacks are able to be detected. Therefore, novel attacks or even variants of common attacks are often ignored. When new security vulnerabilities in software are discovered and exploited every day, these reactive approaches embodied by misuse detection methods are not capable of defeating malicious attacks with unpredictable characters. A major advantage of anomaly detection approaches is their ability to catch unknown attacks, which may include deviations from normal usage of programs, variant forms of known attacks, and novel attacks against software systems.

Hidden Markov model (HMM) has been introduced into intrusion detection field for many years and has achieved many satisfying results [7–13,19], but the major weakness of HMM lies in its high false rejection rate (FRR) and false acceptance rate (FAR). The inherent duration probability density of a state in HMM is exponential, which may be inappropriate for the modeling of audit data of computer systems. We can handle this problem well by developing a hidden semi-Markov model (HSMM) for the normal behavior of computer systems.

* Corresponding author.
*E-mail address:* xbtan@ustc.edu.cn (X. Tan).

0096-3003/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2008.05.028

In this paper, we present a novel anomaly detection approach for intrusion detection based on HSMM. The approach described here applies machine learning techniques to learn the normal behavior of a particular program in order to detect aberrations. By implementing detection at the software process level, multiple, diverse, and overlapping detectors can be embedded within the software infrastructure to provide system-wide coverage.

The remainder of this paper is organized as follows. Section 2 constructs a hidden semi-Markov model for normal behavior of computer system, and proposes an anomaly detection algorithm based on this model. In Section 3, we test the anomaly detection algorithm by using system call sequence collected by University of New Mexico (UNM) and Computer Emergency Response Team (CERT). Finally, we give our conclusion in Section 4.

## 2. Hidden semi-Markov model for computer system

### 2.1. HMM and HSMM

A hidden Markov model is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic process that produces the sequence of observations [14]. HMM is a useful tool to model sequence symbols. The states of HMM represent some unobservable conditions of the system being modeled. In each state, there is a certain probability of producing any of the observable system outputs and a separate probability indicating the likely next states. An HMM can be described using its characteristic parameters. The further information about these parameters can be found in [14].

In previous research, hidden Markov model (HMM) has been applied in intrusion detection systems, but it has a major weakness: the inherent duration probability density of a state in HMM is exponential, which may be inappropriate for the modeling of audit data of computer system. Therefore, we adopt hidden semi-Markov model (HSMM) for intrusion detection, which is introduced in the following.

A semi-Markov HMM (more properly called a hidden semi-Markov model, or HSMM) is similar to HMM except that each state in HSMM can emit a sequence of observations [15]. Because of this difference, the duration probability density of a state in HSMM can be an arbitrary distribution.

An HSMM can be described as

$$\lambda = (N, M, V, A, B, \pi),$$

where

- $N$ is the size of $\Phi = \{0, 1, \ldots, N-1\}$, which is the state space of hidden semi-Markov chain $H_t$, $t = 1, 2, \ldots$;
- $V = \{V_0, V_1, \ldots, V_{M-1}\}$ is visible symbols;
- $M$ is the number of all visible symbols;
- $B = \{b_i(k)\}$, $i \in \Phi$, $1 \leqslant k \leqslant M$, is the distribution of visible symbols $V$;
- $A = [a_{i,j}]_{N \times N}$ is the distribution of state transfer probabilities;
- $\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\}$ is the initial distribution;
- $O_t, t = 1, 2, \ldots, T, O_t \in V$ is visible symbol sequence;
- $T$ is the number of observed visible symbol.

### 2.2. HSMM for anomaly detection

The task of anomaly detection is to analyze and determine whether the audit data of a computer system is produced by normal behavior or anomaly behavior. In a computer or network system monitored by an intrusion detection system, its runtime states can be described by a double-embedded stochastic process, one is an invisible finite semi-Markov chain, and the other is an observable chain related to the previous chain. Both the normal behavior and the anomaly behavior each corresponds to an HSMM. The objective of intrusion detection is to determine which stochastic process produces the monitored data.

Our approach is similar to the test of whether a dice is regular or not by throwing it. For a regular dice, its eccentricity is limited in a certain threshold. A perfect regular dice, whose barcenter overlaps with its geometric center, only exists in unearthly condition. For a perfect dice, the probability of one side facing upwards in the throwing test is equal to the probability of any other side. For a real dice, because its eccentricity is not equal to zero, the probability of one side facing upwards may not be equal to the probability of another side. The probability distribution of any dice is unknown to us; all we know is that this probability distribution is not same for any two dices.

By analyzing the visible symbol sequences of throwing a testing dice, if there is little difference with that of a perfect regular dice (i.e. satisfying some conditions), we consider that the dice under test is a regular dice; otherwise, it is not a regular dice. Based on this principle, we present an approach to distinguish regular dices from irregular dices.

We construct an HSMM for computer systems, whose state space only includes two states: normal state and anomaly state. For convenience, we represent the normal state by 0, and anomaly state by 1. The observed chain is the sample of

system behavior or audit data, and the system's state corresponding to the observed value may belong to the normal state or anomaly state. We construct our hidden semi-Markov model as follows:

- State space: let hidden semi-Markov chain be $H_t$, $t = 1,2,\ldots$, whose state space is $\Phi = \{0,1\}$, where '0' denotes normal state, '1' denotes anomaly state, so $N = 2$;
- the distribution of state transfer probabilities:

$$A = [a_{i,j}]_{N\times N} = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix},$$

  where $a_{i,j} = P\{\text{next\_state} = j | \text{current\_state} = i\}$, $i,j \in \Phi$;
- visible symbol and its distribution: let $V = \{V_0, V_1, \ldots, V_{M-1}\}$ be visible symbol, namely, the set of all visible system behavior, where is the sum of all visible symbol;
- $B = \{b_i(k)\}$, $i \in \Phi$, $1 \leqslant k \leqslant M$ is the distribution of visible symbol; where $b_i(k) = P\{\text{observed\_system\_behavior} = V_k | \text{current\_state} = i\}$
- initial distribution: $\pi = \{\pi_0, \pi_1\}$, $\pi_i = P\{\text{initial\_state} = i\}$, $i \in \Phi$, denotes the initial distribution of normal state and anomaly state;
- visible symbol sequence: $O_t$, $t = 1,2,\ldots,T$, $O_t \in V$, where $T$ is the number of visible symbols.

It is hard to construct an HSMM for attack behaviors because attack methods can differ greatly from one to another and new intrusion methods continue to appear every day. On the other hand, the profile of normal behavior is relatively invariable. Therefore, we only construct HSMM for perfect normal processes of computer system, just like the model developed for testing the perfect regular dice.

Perfect normal processes refer to the processes that their behaviors belong to normal state in an HSMM. Though perfect normal processes do not exist in reality, we use their profile as a reference model. The more the distance of audit data to this model, the more the probability that audit data are generated by anomaly processes.

We define the distribution of state transfer probabilities:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

It means that in a perfect normal process, no matter what current state is, the process will transfer to normal state next time by probability 1. That is to say, the state of perfect normal processes will never leave normal state.

The distribution of visible symbols of normal state is denoted by $\{b_0(k)\}$. That is to say, the appearance probabilities of behaviors under normal state condition are known, but the distribution of behaviors under anomaly state condition is uncertain.

Because the model is constructed for perfect normal processes, we define the initial distribution of normal state and anomaly state as $\pi = \{1,0\}$. It means that under the condition of perfect normal processes, the initial probability of normal state is 1.

### 2.3. Computation of $P\{O|\lambda\}$

After the modeling for perfect normal processes based on HSMM, our anomaly detection algorithm can be divided into two steps:

(1) computing the probabilities of visible symbol sequence under the condition of the HSMM of perfect normal processes, namely, $P\{O|\lambda\}$;
(2) determine whether an anomaly process is detected or not based on the probabilities of visible symbol sequence.

We have constructed the hidden semi-Markov model of perfect normal processes in the last section, our next step is to compute $P\{O|\lambda\}$, where $\lambda$ denotes the model of perfect normal processes.

Consider the forward variable $\alpha_t(i)$ [15] defined as

$$\alpha_t(i) = P\{O_1 O_2 \ldots O_t, H_t = S_i, F_t = 1|\lambda\}. \tag{1}$$

Namely, the probability of a partially observed sequence, $O_1O_2\ldots O_t$, (until time $t$) and the state $S_j$ at time $t$, given model $\lambda$. $F_t$ is a binary indicator variable indicating whether the duration of $H_t$ has finished or not.

We can get:

$$P\{O|\lambda\} = \sum_{i=0}^{N-1} \alpha_T(i). \tag{2}$$

Based on the hidden semi-Markov model we constructed in previous section, we can derive the algorithm of computing as follows:

$$P\{O|\lambda\} = \sum_{i=0}^{N-1} \alpha_T(i) = \alpha_T(0) + \alpha_T(1), \tag{3}$$

$$\alpha_T(i) = \sum_d P(O_{T-d+1}, \ldots, O_T|i,d)P(d|i) \sum_j A(j,i)\alpha_{T-d}(j)[14]. \tag{4}$$

When $i = 0$,

$$\alpha_T(0) = \sum_d P(O_{T-d+1}, \ldots, O_T|0,d)P(d|0) \sum_j A(j,0)\alpha_{T-d}(j) = \sum_d P(O_{T-d+1}, \ldots, O_T|0,d)P(d|0)(\alpha_{T-d}(0) + \alpha_{T-d}(1)).$$

When $i = 1$,

$$\alpha_T(1) = \sum_d P(O_{T-d+1}, \ldots, O_T|1,d)P(d|1) \sum_j A(j,1)\alpha_{T-d}(j) = 0.$$

So,

$$P\{O|\lambda\} = \alpha_T(0) = \sum_d P(O_{T-d+1}, \ldots, O_T|0,d)P(d|0)\alpha_{T-d}(0). \tag{5}$$

We can get $\alpha_T(0)$ by following two assumptions:

1. visible symbol sequence is independent and identically-distributed (i.i.d.);
2. state duration time is uniformly distributed.

Because the visible symbol sequence is emitted by hidden states, a visible symbol is determined by hidden state, not the other visible symbols, then we can get assumption 1; and for the hidden states, either normal state or anomaly state, whose duration time can be arbitrary, for example, an attacker may attack the target system momentarily, so we suppose the distribution of state's duration time is uniform distribution, then we can get assumption 2.

$$P(O_{T-d+1}, \ldots, O_T|0,d) = \prod_{i=1}^{d} b_0(T-d+i) \tag{6}$$

And from assumption 2, we can get:

$$P(d|0) = \frac{1}{k},$$

where $k$ is the parameter of uniform distribution.

Therefore,

$$\begin{aligned} P\{O|\lambda\} = \alpha_T(0) &= \sum_d \left(\prod_{i=1}^{T-1} b_0(T-d+i)P(d|0)\right)\alpha_{T-d}(0) = \frac{1}{k}\cdot\sum_d \left(\prod_{i=1}^{T-1} b_0(T-d+i)\right)\alpha_{T-d}(0) \\ &= \frac{1}{k}\cdot b_0(T)\alpha_{T-1}(0) + \frac{1}{k}\cdot b_0(T)\cdot\sum_d \left(\prod_{i=1}^{T-2} b_0(T-d+i)\right)\alpha_{T-d}(0) = \frac{1}{k}\cdot b_0(T)\alpha_{T-1}(0) + \frac{1}{k}\cdot b_0(T)\cdot\alpha_{T-1}(0) \\ &= \frac{2}{k}\cdot b_0(T)\cdot\alpha_{T-1}(0). \end{aligned} \tag{7}$$

Define the initial value of forward variable as $\alpha_0(0) = 1$, we can determine the observed behavior is normal or not by analyzing $P\{O|\lambda\}$.

### 2.4. Detection algorithm

From maximum entropy principle (MEP) [16], we know that when a computer system is running in normal state, the audit data it generates contains less information than that it generates when running in anomaly state. Namely, the information entropy of anomaly state is larger than that of normal state, so the information entropy can act as the metric in anomaly detection.

But when the length of visible symbol sequence increases, the information entropy of visible symbol sequence will become larger and larger. It only makes sense to compare the value of information entropy among the same-length sequences. In order to use entropy metric on variable-length symbol sequences, we compute the average information entropy (AIE) of visible symbol sequences, and use it as the metric to distinguish between normal behavior and anomaly behavior.

Let $E(N)$ be the average information entropy (AIE) of visible symbol sequences, we can get:

$$E(N) = \frac{-\sum_{i=1}^{N} \ln P_i\{O|\lambda\}}{N}. \tag{8}$$

For convenience of on-line detection, we can use the following iterative algorithm:

$$E(N) = \frac{-\sum_{i=1}^{N} \ln P_i\{O|\lambda\}}{N} = \frac{N-1}{N} \cdot E(N-1) - \frac{\ln P_N\{O|\lambda\}}{N}. \tag{9}$$

Initial value is $E(1) = -\ln P_1\{O|\lambda\}$.

### 2.5. Training algorithm

Because the normal state of a computer system may change over time, so training of hidden semi-Markov model is also an important part in anomaly detection algorithms.

For the hidden semi-Markov model $\lambda = (N, M, V, A, B, \pi)$ we constructed in previous section, both the distribution of state transfer probabilities $A$ and the initial distribution of normal state and anomaly state $\pi$ are fixed values, so only the distribution of visible symbol for normal behavior $B_0 = \{b_0(k)\}$, $1 \leqslant k \leqslant M$ need to be updated.

The training can be implemented by system administrator on normal data sequences. We use the segmental K-means algorithm [17,18] as our training algorithm. The algorithm consists of the following steps:

(1) Randomly choose $N$ normal observed symbols and assign each of the observation symbols to one of these $N$ symbols from which its Euclidean distance is minimum.
(2) Calculate the mean and the covariance for each state:

$$\hat{\mu} = \frac{1}{N} \sum_{t=1}^{N} O_t, \tag{10}$$

$$\widehat{V} = \frac{1}{N} \sum_{t=1}^{N} (O_t - \hat{\mu})^T (O_t - \hat{\mu}). \tag{11}$$

(3) Calculate the symbol probability distributions:

$$\hat{b}_0(O_t) = \frac{1}{(2\pi)^{1/2} |\widehat{V}|^{1/2}} \exp\left[ -\frac{1}{2} (O_t - \hat{\mu}) \widehat{V}^{-1} (O_t - \hat{\mu})^T \right]. \tag{12}$$

## 3. Experimental results

### 3.1. Data set introduction

Short sequences of system calls executed by running programs are a good discriminator between normal and abnormal programs [5,6]. Forrest et al. used system call sequence as the audit data for intrusion detection [6]. The experimental data can be downloaded from http://www.cs.unm.edu/~immsec/systemcalls.htm, which is collected by University of New Mexico (UNM) and Computer Emergency Response Team (CERT). Each system call trace is the list of system calls issued by a single process from the beginning of its execution to the end. Trace lengths vary widely because of differences in program complexity and because some traces are daemon processes and others are not.

### 3.2. Experiment results and analysis

To model HSMM for normal behavior of computer runtime state, we choose system call sequences as our experimental data. We select short system call sequence as visible symbol of HSMM, and use the sequences generated by sendmail, ftpd, lpr, ps and login process.

We divide the experimental data into two classes: training data and test data. Then we build the HSMM using the training data and test anomaly detection algorithm using the test data. Tables 1 and 2 show the experimental results.

After compared with the result of [10,19], which using the same test data set as us, if we chose 8 or 10 as the length of short system calls sequences, our detection rate is much higher.

**Table 1**
The experimental results of UNM data, database denotes the files that record the probability of short system calls sequences

| Length of short system calls sequences | Detection rate (%) | F-P error (%) | Database size |
|---|---|---|---|
| 3 | 92.8 | 7.8 | 893 |
| 5 | 94.5 | 5.8 | 1587 |
| 8 | 100 | 0 | 2159 |
| 10 | 100 | 0 | 2712 |

**Table 2**
The experimental results of CERT data, database denotes the files that record the probability of short system calls sequences

| Length of short system calls sequences | Detection rate (%) | F-P error (%) | Database size |
|---|---|---|---|
| 3 | 93.9 | 6.5 | 721 |
| 5 | 96.8 | 3.3 | 1092 |
| 8 | 100 | 0 | 1536 |
| 10 | 100 | 0 | 1844 |

From experimental results, we know that the length of short system call sequences determines the detection rate and false positive rate. When a longer length of short system call sequences is used, our detection algorithm will exhibit a larger detection rate and smaller false positive rate. But as the length of short system call sequences becomes longer, the database becomes larger, which results in consuming more system resource. From our experiment, we think 8 is an appropriate value of the length of short system call sequences.

## 4. Conclusion

In this paper, hidden semi-Markov model is introduced into intrusion detection systems. We present an algorithm of anomaly detection based on HSMM, which computes the distance between the processes monitored by intrusion detection system and the perfect normal processes. In this algorithm, based on maximum entropy principle (MEP), we introduce the concept of average information entropy (AIE), which is used as detection metric via analyzing variable-length observed symbol sequences. To improve accuracy, the segmental K-means algorithm is applied as training algorithm for the HSMM. Experimental results show that this approach is not only valuable in theory, but also can be effectively applied to monitoring real-time computer systems.

## Acknowledgements

This work is supported by the National 863 High-tech Program of China (No. 2006AA01Z449) and the 42nd National Science Foundation for Post-doctoral Scientists of China (No. 20070420738).

## References

[1] D.E. Denning, An intrusion detection model, IEEE Transactions on Software Engineering 13 (2) (1987) 222–232.
[2] Peng Ning, Yun Cui, Douglas S. Reeves, Analyzing intensive intrusion alert via correlation, in: The 5th International Symposium on Recent Advance in Intrusin Detection, Zurich, Switzerland, 2002.
[3] Frederic Cuppens, Alexandre Miege, Alert correlation in a cooperative intrusion detection framework, in: 2002 IEEE Symposium on Security and Privacy, Oakland, California, 2002.
[4] S. Kumar, E. Spafford, An Application of Pattern Matching in Intrusion Detection, Department of Computer Sciences, Purdue University, CSD-TR-94-013, Coast TR 94-07, 1994.
[5] S. Forrest, S.A. Hofmeyr, A. Somayaji, T.A. Longstaff, A sense of self for Unix processes, in: Proceedings of the 1996 IEEE Symposium on Security and Privacy, Orkland, CA, 1996, pp. 120–128.
[6] S.A. Hofmeyr, S. Forrest, A. Somayaji, Intrusion detection using sequences of system calls, Journal of Computer Security 6 (3) (1998) 151–180.
[7] Sung-Bae Cho, Hyuk-Jang Park, Efficient anomaly detection by modeling privilege flows using hidden Markov model, Computer and Security 22 (1) (2003) 45–55.
[8] C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusion using system calls: alternative data models, in: Proceedings of the 1999 IEEE Symposium on Security and Privacy, IEEE Computer Society, 1999, pp. 133–145.
[9] HaiTao He, XiaoNan Luo, A novel HMM-based approach to anomaly detection, Journal of Information and Computational Science 1 (3) (2004) 91–94.
[10] Weijin Jiang, Yusheng Xu, Yuhui Xu, A novel intrusions detection method based on HMM embedded neural network, Advances in Natural Computation, First International Conference, ICNC 2005, Changsha, China, August 27–29, 2005, in: Proceedings, Part I. Lecture Notes in Computer Science 3610, Springer, 2005, pp. 139–148. ISBN 3-540-28323-4.
[11] Y. Qiao, X.W. Xin, Y. Bin, S. Ge, Anomaly intrusion detection method based on HMM, Electronics Letters 38 (13) (2002) 663–664.
[12] T. Lane, Hidden Markov models for human/computer interface modeling, in: Proceedings of the IJCAI-99 Workshop on Learning About Users, 1999, pp. 35–44.
[13] Kim, In-Young, Study on hidden Markov models for intrusion detection, Master Thesis, School of Computer Science and Engineering, Seoul National University, February 2001.
[14] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 257–286.
[15] K.P. Murphy, Hidden semi-Markov models (HSMMs), June 2002, <http://www.ai.mit.edu/~muphyk>.
[16] E.T. Jaynes, Information theory and statistical mechanics, Physical Review 106 (4) (1957) 620–630.
[17] B.H. Juang, L.R. Rabiner, The segmental K-means algorithm for estimating parameters of hidden Markov models, IEEE Transactions on Acoustics Speech and Signal Processing 38 (9) (1990) 1639–1641.
[18] Rakesh Dugad, U.B. Desai, A tutorial on hidden Markov models, Technical Report No.: SPANN-96.1, May 1996, <http://vision.ai.uiuc.edu/dugad/hmm_tut.html>.
[19] B. Gao, H.Y. Ma, Y.H. Yang, HMMS based on anomaly intrusion detection method, in: Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002.