

A Simple and Efficient Hidden Markov Model Scheme for Host-Based Anomaly Intrusion Detection

Jiankun Hu and Xinghuo Yu, RMIT University
D. Qiu, Sai Global Limited
Hsiao-Hwa Chen, National Cheng Kung University

Abstract

Extensive research activities have been observed on network-based intrusion detection systems (IDSs). However, there are always some attacks that penetrate traffic-profiling-based network IDSs. These attacks often cause very serious damages such as modifying host critical files. A host-based anomaly IDS is an effective complement to the network IDS in addressing this issue. This article proposes a simple data preprocessing approach to speed up a hidden Markov model (HMM) training for system-call-based anomaly intrusion detection. Experiments based on a public database demonstrate that this data preprocessing approach can reduce training time by up to 50 percent with unnoticeable intrusion detection performance degradation, compared to a conventional batch HMM training scheme. More than 58 percent data reduction has been observed compared to our prior incremental HMM training scheme. Although this maximum gain incurs more degradation of false alarm rate performance, the resulting performance is still reasonable.

Intrusion detection has now been widely accepted as an essential component in a decent security system. This is due to the fact that the task of preventing all attacks is impossible. Intrusion detection can detect malicious attacks that have penetrated preventative mechanisms such as firewalls, which can help provide damage assessment, response, deterrence, and prosecution support.

Denning's pioneering work [1] has established the most fundamental principle that the majority of intrusion detection systems (IDSs) have followed. The principle is a hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. It is suggested that profiles are used to represent the behavior of subjects using statistical measures. IDSs can have several different classifications. An IDS can be classified as an HIDS (host-based IDS) and NIDS (network-based IDS) in terms of the target the IDS protects. Also, an IDS can be classified into misuse intrusion detection and anomaly intrusion detection according to whether the features of an intrusion are known or unknown in advance. The misuse IDS retrieves attacks' signatures and establishes a database for the collection. During the detection process, the IDS will retrieve a subject signature and search a match against the established signature database.

An intrusion alert is triggered once a match is found. Such a mechanism is very effective in detecting a priori known attacks, but performs unsatisfactorily in detecting unknown attacks. Anomaly IDSs are promising in detecting unknown attacks. Based on Denning's principle, an anomaly IDS first builds a system's normal behavior profile and then compares operational system behavior against the nominal profile. If a significant deviation is found, an intrusion alert is triggered.

While Denning's intrusion detection model is a host-based IDS, extensive research activities have been shifted to network-based IDSs. There are several factors behind this, summarized as follows:

- Networking factor: With the rapid proliferation of Internet technology, overwhelming computing applications are network based. Many security problems are introduced from this environment such as denial of service (DoS) attacks and other security loopholes related to networking protocols.
- Real-time and computing resource restraints: Ideally, intrusion can be detected as soon as it happens in order to minimize the potential damage. However, audit data collection and processing for detecting intrusion involve large amounts of computing resources. Therefore, a dedicated hardware and software IDS component is required to perform the task efficiently.

Normally, a network-based IDS deduces intrusion from analyzing network packets. It is very effective in detecting DoS attempts originating outside the network. The majority

The work was supported in part by the ARC (Australia research Council) Discovery Grant DP0985838 and Taiwan National Science Council Grant NSC97-2219-E-006-004.

of network IDSs are based on packet traffic analyzing plus inspecting some suspicious network activity such as port scan [2–5]. However, there are some detrimental network attacks that do not generate significant network traffic. Very recently the Zotob worm disabled thousands of computer systems, bringing business to a halt. The Australian car manufacturer Holden lost about \$6 million in this Zotob attack. Other victims included the *Financial Times*, CNN, Canadian Bank of Commerce, Daimler Chrysler, and General Electric [6]. Attacks such as the recent Zotob worm exploits the Microsoft Windows Plug and Play Buffer Overflow Vulnerability on TCP port 445 and installs an FTP server on the victim's machine to download its malicious code that can repeatedly shut down and reboot the machine. Once it gets on a corporate network, it can pass from machine to machine. It can also bypass a network-traffic-based network IDS as it does not generate a large amount of traffic. It has been observed that many network attacks, even some network-traffic-based attacks, compromise a machine significantly and propagate through to other machines of the network. Also, a network-based IDS has difficulty dealing with attacks originating from inside the network. Therefore, an effective IDS should include an HIDS as a complement to the NIDS. Unfortunately, the existing host-based anomaly IDS has rarely been deployed in commercial environments due to two problems:

- The existing hosted-based anomaly IDS has a very high false alarm rate.
- The computing resource demand for the operations of an IDS is huge.

These two problems cause disruption to normal operations of the host even when an attack is not present.

Although the principles of the HIDS and NIDS are very similar in that intrusion detection is based on analyzing a collection of discrete time-sequenced events for patterns of attacks, their operations are quite different. The NIDS examines network packets or traffic, while the HIDS examines events such as what files were accessed, what applications were executed, and their associated activities. A number of techniques such as data mining, statistics, and genetic algorithms have been used for intrusion detection on the user and program activity levels individually. In [7] a novel framework, called Mining Audit Data for Automated Models for Intrusion Detection (MADAM ID), is proposed. This framework uses data mining algorithms to compute activity patterns from system audit data and extracts features from the patterns. Then machine-learning algorithms are applied to the audit records that are processed according to the feature definitions and generate intrusion detection rules. The data mining algorithms include meta-classification, association rules, and frequent episode algorithms. The test results in 1998 conducted by the Defense Advanced Research Project Agency (DARPA) Intrusion Detection Evaluation showed that the model was one of the best performing of all the participating systems in offline mode. In order to detect user anomalies, the normal user activity profiles are created, and a similarity score range (upper and lower bound) is assigned to each user's normal pattern set. When in action, the system computes the similarity score of the current activity's patterns. If this score is not in the similarity score range, the activity is considered abnormal.

On the program activity level (micro-level), anomaly detection systems based on system calls have received growing attention from many researchers since the successful initiative of Forrest *et al.* [8]. Forrest and Longstaff [8] proposed to define a normal profile by short-range correlations in process system calls. Their results show that short sequences of system

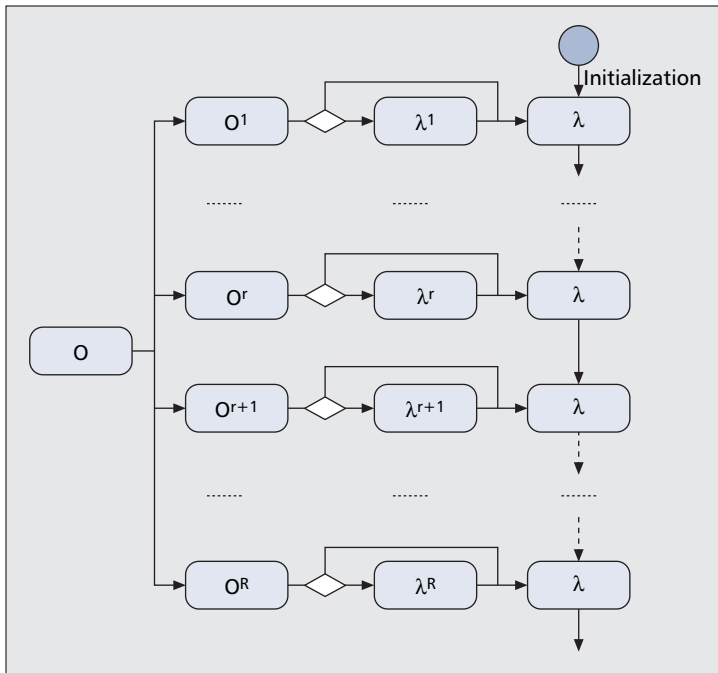
calls define a stable signature that can detect some common sources of anomalous behavior in the system event stream. Within this framework, each system generates its own normal database, based on software and hardware configuration and usage patterns. These normal databases will be compared against abnormal events. Since then, a certain number of novel schemes have been further discovered. In [9] Warrender *et al.* have pointed out that a number of machine-learning approaches such as rule induction and HMM can be used to learn the concise and generalizable representation of the "self" identity of a system program by relying on the program's runtime system calls. The learned models were shown to be able to accurately detect anomalies caused by attacks on the system programs. In [10] we implemented a user anomaly detection system based on the intrusion detection framework [7] and conducted several intrusion detection experiments by analyzing both macro-level and micro-level activities. User behavior is modeled by using data mining, and the frequent episode algorithms are used to build the user's normal profiles. The experimental results demonstrated that user anomalies and changes in the user's normal working patterns can be detected effectively.

In this article development in host-based anomaly intrusion detection is reported with emphasis on powerful HMM-based anomaly intrusion detection schemes. Based on our previous work [3, 4, 11], this article proposes an efficient HMM training scheme for system-call-based anomaly intrusion detection. The rest of this article is organized as follows. The next section provides an overview of HMM-based anomaly intrusion detection. We then propose a new incremental HMM training scheme. The following section presents experimental results and analysis, followed by the conclusions provided in the final section.

Related Works

HMM is a double stochastic process. The upper layer is a Markov process whose states are not observable. The lower layer is a normal Markov process where emitted outputs can be observed. HMM is a powerful tool in modeling and analyzing complicated stochastic process. For example, in weather forecast we may observe that a cohort of many frogs' songs may lead to a rainy tomorrow. However, this conclusion is based on a probabilistic sense. Also, there is no direct link between observed frogs' singing and tomorrow's rainy state. Experience tells us that there is a hidden link. HMM can be used in such scenarios. HMM has been widely used for protein sequence analysis and speech recognition. In an effort to find the best modeling method for normal program behavior using system calls, Warrender *et al.* [9] have investigated various modeling techniques through extensive experiments. They used the normal database method, frequency-based method, data mining, and HMM to construct detection models from the same normal traces of system calls. Their experimental results have shown that the HMM method can generate the most accurate results on average, although the training cost of the HMM method is very high. Therefore, for practical applications it is essential to reduce this training cost.

Rabiner [12] discussed the basic idea of estimation of HMM parameters from multiple observation sequences. The independence assumption of observation sequences is hard to achieve in practice. Improved estimation of HMM parameters from multiple observations was proposed by Davis *et al.* [13]. In this scheme multiple observation sequences have been used to train corresponding multiple HMM submodels. Various weighting methods have been proposed to combine sub-HMM



■ Figure 1. Proposed architecture of the enhanced incremental HMM training scheme.

models that have been individually trained by individual observation sequences. Elementary Monte Carlo techniques have been used to evaluate these weighting methods. Gotoh *et al.* [14] discussed alternatives to the usual EM algorithm for estimation of HMM parameters. They proposed two efficient HMM training approaches, incremental ML estimation and incremental MAP estimation. It was experimentally verified that the training of the incremental algorithms is substantially faster than the conventional method and suffers no loss of recognition performance.

A Simple and Efficient HMM Training Scheme

Based on our previous works [3, 11], a simple and efficient HMM anomaly intrusion algorithm is proposed as follows. Assume that an HMM model parameter is $\lambda = \{A, B, \pi\}$. $A = \{a_{ij}\}$ represents the probability of being in state j at time $t + 1$, given that we were in state i at time t ; $B = \{b_j^{(k)}\}$ represents the probability of observing symbol v_k at state j ; and $\pi = \{\pi_i\}$ is the probability of being at state i at time $t = 1$. The overall observation sequence data is O . An enhanced incremental HMM training scheme is proposed in a diagram in Fig. 1.

In this scheme a long training data set is partitioned into a number of subsequences. A general procedure of subsequence partition is first to divide the long training data set into R equally sized subsequences. Each subsequence must be long enough to train a sub-HMM model. In practice, it is normally more than three times the size of the sub-HMM model. Next, find the maximum number of subsequences that are similar (i.e., each pair's correlation index is above the preset correlation threshold) by adjusting the parameter R . Then each subsequence is used to train a submodel, and the trained submodel is incrementally merged into the final model using the weighting average algorithm proposed in our prior work [3]. Compared with Davis *et al.*'s method [13], this approach incrementally merges the submodel into the final model rather than merging all submodels after they have been completely trained. Because data preprocessing has been intro-

duced in Fig. 2, we can skip highly similar subsequences, which can effectively reduce the number of submodels during the training process. In order to identify similar subsequences, a correlation threshold needs to be determined. In general, the higher the threshold, the higher the correlation the two subsequences involved will be. However, a higher threshold will lead to fewer similar subsequences, which will gain less cost savings. If a lower value threshold is used, more subsequences will be identified as similar. This can gain more cost savings but at the price of losing more useful information which leads to the degradation of intrusion detection rate performance. A balance can be determined experimentally. Intuitively, the same programs operating in similar conditions will generate similar system call sequences. The volume of redundant information can be huge because IDS training data are normally collected over a long period of time. If we consider that a subsequence is generated by an operational condition, there is indeed a mapping between operation condition similarity and subsequence similarity. In an extreme case, the same operational condition will generate the same subsequence. Therefore, it is reasonable to identify similar subsequences through standard correlation methods such as the Correlation Matrix. The overall operational flow chart is shown in Fig. 2.

Experiments

Experiment Setup

Training and attack data are from a public database [15] provided by the University of New Mexico (UNM) and the Massachusetts Institute of Technology (MIT) Artificial Intelligence (AI) Laboratory. They contain sendmail, inetd, and lpr programs, which are popular programs used in everyday life. They have also been chosen in many papers for testing system-call-based IDS schemes. Sendmail is the most popular UNIX-based implementation of the Simple Mail Transfer Protocol (SMTP) for transmitting emails. Inetd is a UNIX function that manages common TCP/IP networking services such as FTP and Telnet. Lpr is UNIX software that sends print requests to printers. The performance metric is the false positive rate, defined as [11]

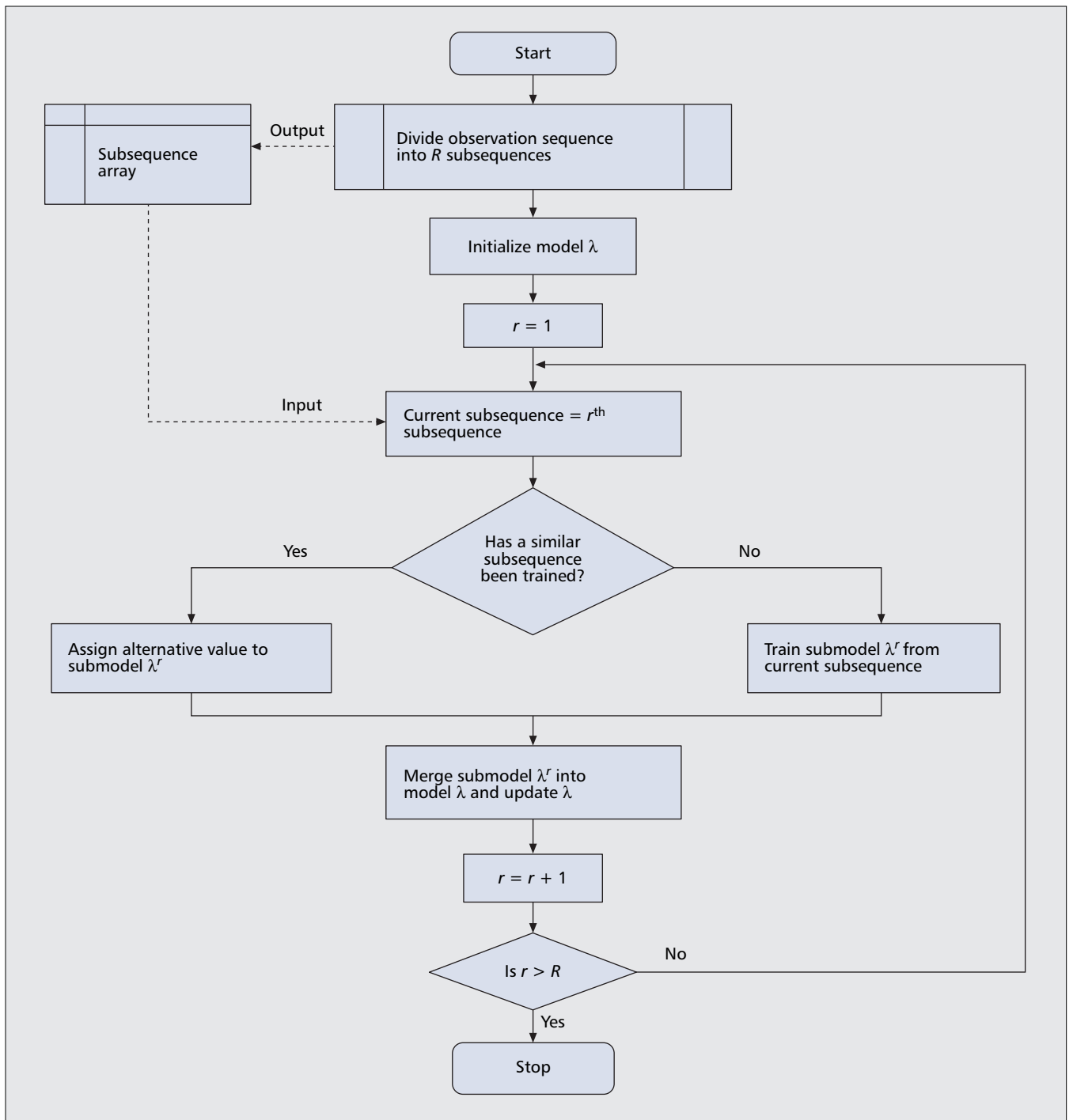
$$FPR = \frac{FP}{FP + TN},$$

where FP is the number of false positives and TN is the number of true negatives. Three different datasets shown in Table 1 are used for the test [3, 11]. The specifications of the machine where experiments have been conducted are given as follows:

- Processor and memory:
 - 2.4 GHz Intel Core 2 Duo processor with 3 Mbytes on-chip shared L2 cache
 - 1066 MHz frontside bus
 - Two 1-Gbyte SO-DIMMs
- Storage: 250-Gbyte hard drive

Unless stated otherwise, the following are descriptions of training data and attack data taken from [15].

Sendmail Dataset — The synthetic sendmail is the main dataset. The sendmail data were collected at UNM on Sun SPARC stations running unpatched SunOS 4.1.1 and 4.1.4 with the sendmail program. The sendmail dataset contains



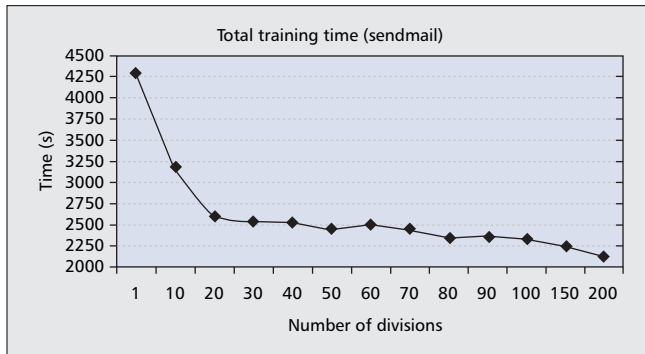
■ Figure 2. Operational flow chart of the proposed efficient incremental HMM training scheme.

19,526 system calls and has 47 distinct system calls. The sendmail program is sufficiently complex for testing [3, 11]. We use traces that contain syslog intrusion as testing data. The syslogd attack uses the syslog interface to overflow a buffer in sendmail. A message is sent to the sendmail on the victim machine, causing it to log a very long error message. The log entry overflows a buffer in sendmail, replacing part of sendmail's running image with the attacker's machine code. The new code is then executed, causing the standard I/O of a root-owned shell to be attached to a port. The attacker may then attach to this port at ease. This attack can be run either locally or remotely. Both modes will be used for the test.

UNM Synthetic lpr Dataset — The synthetic data for lpr were collected on Sun SPARC stations running unpatched SunOS 4.1.4 with the lpr program. This data set contains 2398 system calls and 37 distinct system calls. The attack against lpr used for testing is iprcp intrusion. The lprcp attack script uses lpr to replace the contents of an arbitrary file with others. The attack exploits the fact that older versions of lpr use only 1000 different names for printer queue files, and do not remove the old queue files before resuming them. The attack produces 1001 traces. In the first trace lpr places a symbolic link to the victim file in the queue. The subsequent traces advance lpr's counter, until on the last trace the victim file can be overwritten with the attacker's own material.

Programs	Intruding trace		Normal trace	
	System calls	Distinct calls	System calls	Distinct calls
sendmail	3090	50	19,526	47
inetd	8371	35	541	35
UNM lpr	164,232	38	2398	37

■ Table 1. Summary of training data and attack data.



■ Figure 3. Training time vs. number of submodels.

Inetd Dataset — The inetd program data were traced on a UNM computer running a modified Linux 2.0.35 kernel that allows us to collect system call traces. This data set contains 541 system calls and 35 distinct system calls. The inetd program is typically started as a foreground process, which initiates a daemon process to run in the background and then exits. The intrusion against the inetd program is a DoS attack that extensively consumes the network connection resources. As the attack progresses, more of the system calls requesting resources return abnormally and are re-issued. The intrusion data collected include a startup process, a daemon process, and several child processes, but only the daemon process is expected to show any deviation from normal behavior.

Experimental Results and Analysis

For convenience, the following discussion is under the context of 100 percent detection rate. First, we consider the large sendmail database. A common practice in choosing the size of an HMM model is to make it equal to or greater than the number of distinct system calls [3]. In this example, the size of the HMM model is chosen as 47. Using the proposed enhanced incremental HMM training scheme shown in Fig. 2, the following experimental results have been obtained, and are shown in Fig. 3 and Table 2.

Figure 3 depicts the trend of total training time against the number of subsequences. As expected, the speed of convergence is improved when more subsequences are partitioned. A significant improvement of training time cost saving, up to 30 min, which is about 50 percent training cost saving, has been observed at the point of 90 subsequences.

Table 2 shows the false alarm rate of batch training and incremental training vs. the number of subsequences in the sendmail case. As the table indicates, batch training has the lowest false alarm rate. In general, the false alarm rate increases as the length of subsequence decreases. However, the degradation of intrusion detection performance is very minor until the number of subsequences reaches 100 or more. When the number of subsequences reaches 91, 22 subse-

quences are identified as removable with the correlation threshold set as 0.9. The training time required for our proposed scheme is only about half of the batch training time, as shown in Fig. 3. The resulting FPR is around 0.098 percent, which is still close to the batch training result (i.e., within the same order of magnitude). The data reduction via proposed similarity data preprocessing for the sendmail program is shown in Fig. 4. Apparently, the number of redundant subsequences is a nonlinear function of the number of subsequences partitioned, although their general overall trends seem to be in synchronization. The maximum data reduction of 58 percent occurs at about the point of 121 subsequences. Caution must be taken not to generate too many subsequences as it can lead to too short length of the individual subsequences. It is well known [3] that the length of each training observation subsequence must be much greater than the number of hidden states in the corresponding HMM model in order to obtain reliable results. In our experiment the subsequence length is roughly four or more times longer than 47, which is the number of distinct system calls. The HMM performance will downgrade significantly if the subsequence is smaller than this threshold.

For the inetd data set, as it contains only 541 system calls, very few subsequences can be produced, which leads to no similar subsequences found. The lpr data set contains 2398 system calls but is still considered small. With minimum subsequence length constraint, the number of possible subsequences is very limited. In our experiment, only four similar subsequences can be eliminated, which produces a minor cost saving. This is acceptable as the cost of batch HMM training using small data size is already small.

Conclusion

In this article development of host-based anomaly intrusion detection has been studied with emphasis placed on system-call-based HMM training. An enhanced incremental HMM training framework has been proposed that incorporates a simple data pre-processing method for identifying and removing similar sub-sequences of system calls. This data processing approach can reduce the number of HMM submodels required in our prior incremental HMM training framework. Three popular public databases have been used to test the proposed algorithm in detecting anomaly intrusions. The experiment demonstrated that up to 50 percent (compared to batch HMM) training cost saving can be achieved for a large data set without noticeable degradation of intrusion detection performance. More than 58 percent data reduction has also been observed with a higher but still reasonable false alarm rate in terms of the same order of magnitude as achieved by the batch training method. This result will provide a promising mechanism toward making host-based anomaly IDS feasible in real life applications.

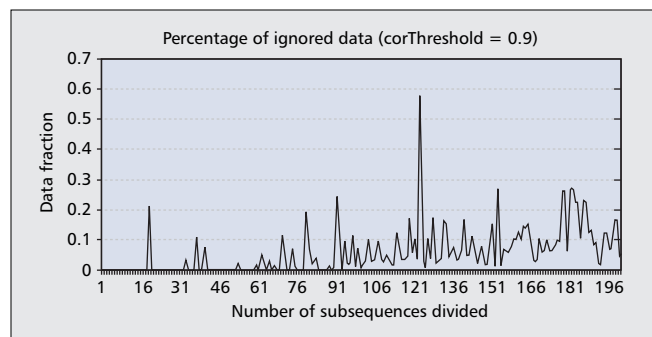
References

- [1] D. Denning, "An Intrusion-Detection Model," *Proc. 1986 EEE Symp. Sec. Privacy*, Apr. 7-9, 1986, pp. 118-31.
- [2] J. Hu, P. Bertok, and Z. Tari, "Taxonomy and Framework for Integrating Dependability and Security," in *Information Assurance: Dependability and Security in Networked Systems*, Y. Qian et al., Eds., Elsevier, 2008, pp. 149-70.
- [3] X. D. Hoang and J. Hu, "An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls," *IEEE Int'l. Conf. Net. '04*, Singapore, Nov. 16-19, 2004, vol. 2, pp. 470-74.
- [4] X. D. Hoang, J. Hu, and P. Bertok, "A Multi-Layer Model for Anomaly Intrusion Detection using Program Sequences of System Calls," *Proc. 11th IEEE Int'l. Conf. Net.*, Sydney, Australia, Sept. 28-Oct. 1, 2003, pp. 531-36.
- [5] A. Patcha and J. Park, "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends," *Comp. Networks*, vol. 51, Aug. 2007, pp. 3448-70.

Program sendmail (47 hidden states)

No. of subsequences	Subsequence length	FPR (%)	No. of subsequences	Subsequence length	FPR (%)
1	19526	0.069	10	1953	0.074
30	651	0.077	50	391	0.082
70	279	0.093	90	217	.098
100	195	0.134	150	130	0.443

■ Table 2. False positive rate of sendmail program.



■ Figure 4. Percentage of data reduction vs. number of subsequences partitioned in sendmail program experiment.

[6] Sophos, "Breaking News: Worm Attacks CNN, ABC, Financial Times, and The New York Times," Aug. 16, 2005, retrieved Oct. 26, 2005; <http://www.sophos.com/virusinfo/articles/breakingnews.html>

[7] W. Lee and S. I. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," *ACM Trans. Info. Sys. Sec.*, vol. 3, no. 4, Nov. 2000, pp. 227–61.

[8] S. Forrest, S. A. Hofmeyr, and A. Somdajji, "Intrusion Detection Using Sequences of System Calls," *J. Comp. Sec.*, vol. 6, 1998, pp. 151–80.

[9] C. Warrender, S. Forrest, and B. Perlmutter, "Detecting Intrusions Using System Calls: Alternative Data Models," *Proc. 1999 IEEE Comp. Soc. Symp. Research Sec. Privacy*, Berkeley, CA, May 1999, pp. 133–45.

[10] D. Hoang, J. Hu, and P. Bertok, "Intrusion Detection Based on Data Mining," *5th Int'l. Conf. Enterprise Info. Sys.*, Angers, France, vol. 3, Apr. 23–26, 2003, pp. 341–46.

[11] D. Qiu, "Efficient Training for HMM-based Anomaly Detection System Using Correlation Method," *Honors Thesis*, RMIT University, 2008.

[12] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, 1989, pp. 257–86.

[13] R. I. A. Davis, B. C. Lovell, and T. Caelli, "Improved Estimation of Hidden Markov Model Parameters from Multiple Observation Sequences" *Proc. 16th Int'l. Conf. Pattern Recognition*, vol. 2, 2002, pp. 168–71.

[14] Y. Gotoh, M. M. Hochberg, and H. F. Silverman, "Efficient Training Algorithm for HMM's Using Incremental Estimation," *Speech Audio Processing*, vol. 6, no. 6, 1998, pp. 539–48.

[15] University of New Mexico's Computer Systems Project, Oct. 24, 2008; <http://www.cs.unm.edu/immsec/systemcalls.htm>

Biographies

JIANKUN HU is an associate professor at the School of Computer Science and Information Technology, RMIT University, Australia. His major research interest is in computer networking and computer security, especially biometric security. He has been awarded three Australia Research Council grants. He served or is serving as Security Symposium Co-Chair for IEEE GLOBECOM '08 and IEEE ICC '09. He was Program Co-Chair of the 2008 International Symposium on Computer Science and Its Applications. He is an Associate Editor of the following journals: *Journal of Network and Computer Applications*, Elsevier; *Journal of Security and Communication Networks*, Wiley; and *Journal of Wireless Communication and Mobile Computing*, Wiley. He is the lead Guest Editor of a 2009 special issue on biometric security for mobile computing, *Journal of Security and*

Communication Networks, Wiley. He received a Bachelor's degree in industrial automation in 1983 from Hunan University, P.R. China, a Ph.D. degree in engineering in 1993 from the Harbin Institute of Technology, P.R. China, and a Master's degree for research in computer science and software engineering from Monash University, Australia, in 2000. In 1995 he completed his postdoctoral fellow work in the Department of Electrical and Electronic Engineering, Harbin Shipbuilding College, P.R. China. He was a research fellow of the Alexander von Humboldt Foundation in the Department of Electrical and Electronic Engineering, Ruhr University, Germany, during 1995–1997. He worked as a research fellow in the Department of Electrical and Electronic Engineering, Delft University of Technology, the Netherlands, in 1997. Before he moved to RMIT University Australia, he was a research fellow in the Department of Electrical and Electronic Engineering, University of Melbourne, Australia.

DONG QIU (dong.qiu@hotmail.com) He completed his Bachelor's degree with honors at RMIT University, Melbourne, and is now a software engineer. He is interested in the research area of network security, especially intrusion detection. He began his career in late 2007 working with SAI Global and has been involved in GRC system development for one year. He currently lives in Melbourne, Australia.

HSIAO-HWA CHEN (hshwchen@ieee.org) is currently a full professor in the Department of Engineering Science, National Cheng Kung University, Taiwan, and was the founding director of the Institute of Communications Engineering of the National Sun Yat-Sen University, Taiwan. He received B.Sc. and M.Sc. degrees from Zhejiang University, China, and a Ph.D. degree from the University of Oulu, Finland, in 1982, 1985, and 1990, respectively, all in electrical engineering. He has authored or co-authored over 300 technical papers in major international journals and conferences, five books, and several book chapters in the areas of communications, including the books titled *Next Generation Wireless Systems and Networks* and *The Next Generation CDMA Technologies* (Wiley, 2005 and 2007). He has been an active volunteer for various IEEE technical activities for over 20 years. Currently, he is serving as Chair of the IEEE ComSoc Radio Communications Committee and Vice Chair of the IEEE ComSoc Communications & Information Security Technical Committee. He served or is serving as symposium chair/co-chair of many major IEEE conferences, including VTC, ICC, GLOBECOM, and WCNC. He served or is serving as Associate Editor and/or Guest Editor of numerous important technical journals in communications. He is serving as Chief Editor (Asia and Pacific) for Wiley's *Wireless Communications and Mobile Computing* journal and Wiley's *International Journal of Communication Systems*. He is the founding Editor-in-Chief of Wiley's *Security and Communication Networks Journal* (<http://www.interscience.wiley.com/journal/security>). He is also an adjunct professor of Zhejiang University, China, and Shanghai Jiao Tong University, China. He is a recipient of the Best Paper Award at IEEE WCNC 2008.

XINGHUO YU [M'92, SM'98, F'08] received B.Eng. and M.Eng. degrees from the University of Science and Technology of China in 1982 and 1984, and a Ph.D. degree from South-East University, China, in 1988. He is currently with RMIT University, Melbourne Australia, where he is director of the RMIT Platform Technologies Research Institute and professor of information systems engineering. His research interests include variable structure and nonlinear control, complex and intelligent systems, and industrial information technologies. He has published over 300 refereed papers in journals, books, and conference proceedings, as well as co-edited 10 research books. He has served as an Associate Editor of three *IEEE Transactions (Circuits and Systems Part I (2001–2004), Industrial Informatics (2005–Present), and Industrial Electronics (2007–Present)*, and several other scholarly journals. He was the sole recipient of the 1995 Central Queensland University Vice Chancellor's Award for Research. He is a Fellow of the Institution of Engineers Australia and was made Emeritus Professor of Central Queensland University Australia in 2002 for his long-term contributions.