Anomaly Detection based on Probabilistic Properties of Hidden Markov Models*

Eunyoung Lee[1], Chan-Kyu Han[2], and Hyoung-Kee Choi[3]
[1 2 3] Dept. of Computer Engineering, Sungkyunkwan University
300 Chunchun-dong, Jangan-gu, Suwon, Gyeonggi-do, South Korea (440-746)
E-mail: {eylee[1], hedwig[2], hkchoi[3]}@ece.skku.ac.kr

Abstract: Due to increasing use of the Internet, there is a trend of increasing attacks over networks. Therefore, we have need of study for network anomaly detection and measurement scheme to network state. In this research, we propose a scheme for anomaly detection based on the traffic behavior of Hidden Markov Models. The proposed scheme detects anomalies in traffic using a time series. We decide whether or not anomaly detection is a network anomaly via an anomaly decision process using Hidden Markov Models. These processes are implemented in the Perl programming language, and decisions are made using a real-world trace containing de facto attacks. Despite the fact that the results are not clear-cut, we conclude that this does not invalidate this study, because this result is caused by an insufficient learning process using real-world traffic. On the contrary, assuming real-world states, increases the ability to detect and make decisions about attacks, because the manager is involved in decisions about access or application. We expect that this research will be applicable for determining real-time states of networks, detection and classification of new types of attack from networks.

1. Introduction

The increasing demand for Internet applications creates enormous economic wealth and industry. As this is further manifested, various threats such as system intrusion, malicious code injection, and network attacks in vulnerable networks have increased. The detection, diagnosis and analysis of network attacks are the focus of attention as an important research area. Here, the field of network attack detection is the predominant research area. The detection of network attacks can be classified into two types: rule-based detection and anomaly-based detection.

Rule-based detection focuses on detection after a network attack has already occurred. It is difficult to prevent an attack which is a variant of previous attacks or a new type of attack, using rule-based detection. On the other hand, anomaly-based detection focuses on pre-detection (prevention) of vulnerabilities or intrusions. However, the problem of the high rate of false positives and false negatives was unsolved in previous studies.

Consequently, reducing the ratio of false positives and false negatives is a high priority in anomaly-based detection. In this paper, we proposed an anomaly detection scheme based on probabilistic properties of Hidden Markov Models (HMMs). We both adapt HMMs for network anomaly detection, and provide details of implementation. The remainder of this paper is organized as follows. In Section 2, we illustrate the process of symptom derivation, which is a preliminary stage in our detection scheme. In Section 3, we explain how anomaly detection is feasible by probabilistic properties of HMMs. We explain the method of implementation and traffic data used for verification in Section 4. We analyze experimental results of the proposed scheme in Section 5. We conclude this paper in Section 6.

2. Related work

A study of network intrusion detection can be classified: signature-based and anomaly-based. A signature-based detection inspects traffic for known attacks. An anomaly-based detection usually uses the distribution of IP addresses and ports. There is an advantage that no ru1es need to be written, and that it can detect new attacks. [1]

There are three kinds of scheme used in anomaly detection: statistical anomaly detection scheme, machine learning based anomaly detection scheme and data mining based anomaly detection scheme. [2]

Some researches have been conducted on statistical anomaly detection scheme [3]. It uses descriptive statistics to model user behavior and also to model acceptable behavior for a generic user within a particular user group. In [4], the authors found correlations in fixed length sequences of system calls in the UNIX operating system, and use them to build a normal profile for anomaly detection. It used machine learning based anomaly detection scheme. In [5], the authors used data mining based anomaly detection scheme, which uses inductive rule generation to make rules for important, yet infrequent events.

3. Anomaly Detection

A symptom can be defined as a manifestation of an anomaly itself or its result. In this paper, we aim to detect anomalies by adapting HMMs to a series of symptoms. However, it is very difficult to observe anomalies directly, due to the fact that they are hidden in network traffic. In this research, we detect anomalies using time series forecasting.

Time series forecasting predicts a time series model sequence according to regularities of network traffic. Then, it regards out-of-threshold data as an anomalous event. In this paper, we utilize 12 types of time series data in network

* "This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-C1090-0801-0028)

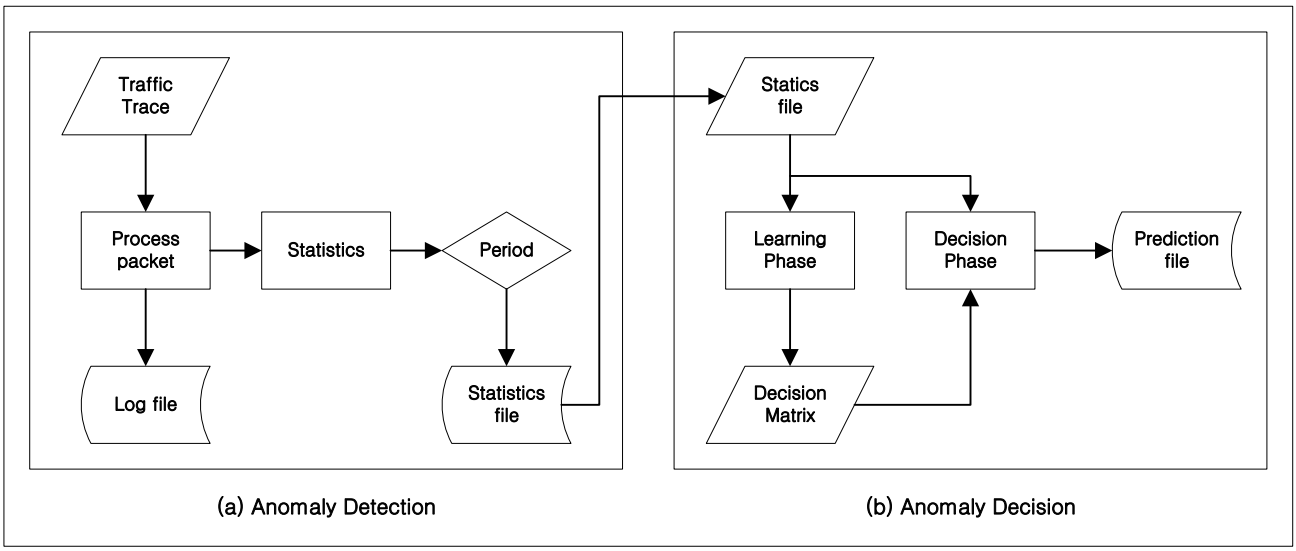

Figure. 1 The two part of program implemented using Perl

traffic. These are; destination IP usage, number of open ports, packet arrival rate, payload size, payload size variation per packet, port number variation, session time, payload size per session, number of sessions per port, source, destination IP ratio, destination IP distribution and reflection traffic.

Symptoms that are derived from time series data are as follows. The symptoms list is selected empirically. Modification of this list is allowed.

Port Scanning: This involves seeking vulnerable ports of a destination system using a port scanner. The adversary checks vulnerable ports via port scanning.

IP Scanning: In a manner similar to port scanning, the adversary uses this method to find victims. IP scanning makes a list of vulnerable IP address.

IP Spoofing: This means that the adversary creates packets which appear to be transmitted from different IP addresses which are not that of the adversary.

Packet Spoofing: The adversary modifies a packet header or payload of the packet in order to generate a protocol error.

Sudden Increase: It is not clear that an attack is occurring in cases of increased requests or responses. This can mean the adversary is collecting information for malicious purposes using port scanning etc.

Reflection: If a reflected packet, i.e. TCP RST packet, ICMP unreachable packet, is detected, we suspect a network attack.

4. Probabilistic Properties of HMMs

In this paper, we adapt HMMs to symptoms, in order to determine the probability of network attacks. We define the set of network attacks for which a direct observation is not viable as the *hidden state set*. The set of network attacks ($\mathcal{N}$) consists of four elements, as shown in Eq.1.

$$\mathcal{N} = \{DoS/DDoS, Worm, Unknown, Clean\} \quad (1)$$

We define the set of symptoms which is directly observable as the *observable state set*, then, Eq.2 denotes ($\mathcal{M}$):

$$\mathcal{M} = \begin{Bmatrix} Port\ scanning, & IP\ scanning \\ IP\ spoofing & Packet\ spoofing \\ Sudden\ increase & Reflection \end{Bmatrix} \quad (2)$$

In the previous section, we utilized time series forecasting, in order to decide whether or not a symptom has manifested. We collect a real-world traffic trace and derive the following; *state translation matrix* ($\mathcal{S}$), *emission distribution matrix* ($\mathcal{T}$) and *initial vector* (Π). Detailed information about $\mathcal{S}$, $\mathcal{T}$ and Π is available in [6]. The HMM model comprises five elements; *hidden state set*, *observable state set*, *state transition matrix*, *emission distribution matrix*, *initial vector*. The sequence prediction of $\mathcal{N}$ is achieved by applying the Viterbi algorithm to the HMM model [7], [8].

5. Implementation

We implement the proposed anomaly detection method based on HMM using Practical Extraction and Report Language (Perl) programming. We install the Net::Pcap module for monitoring network traffic and the Net::Packet module for analyzing it. The implemented program consists of two components: *anomaly detection process* and *anomaly decision process*.

Figure 1 (a) shows that anomaly detection involves reading a real-world trace file, processing each packet and creating log files. We derive various time series statistical data for each packet and produce statistical data each second. As shown in Figure 1 (b), anomaly decision utilizes time series statistical data, then, generates one symptom per second. The symptom sequence constructs the *observable state set*

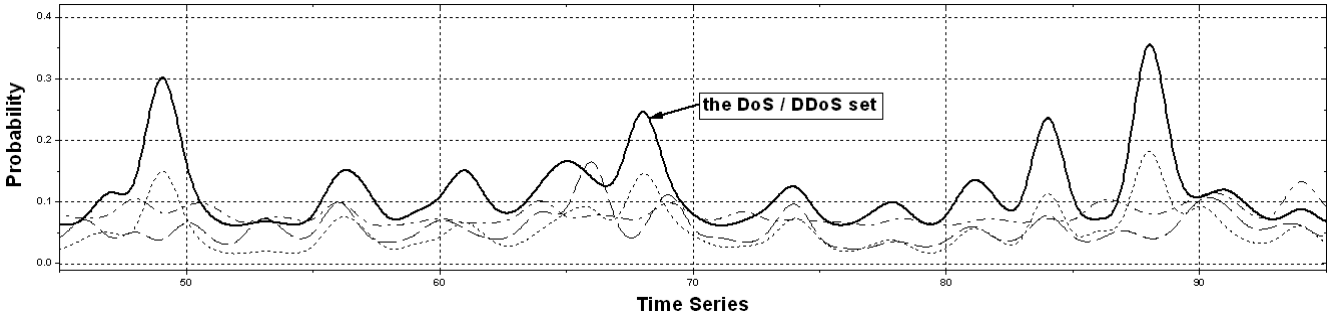

Figure. 3 Result of the attack probability time series using DoS/DDoS traffic trace

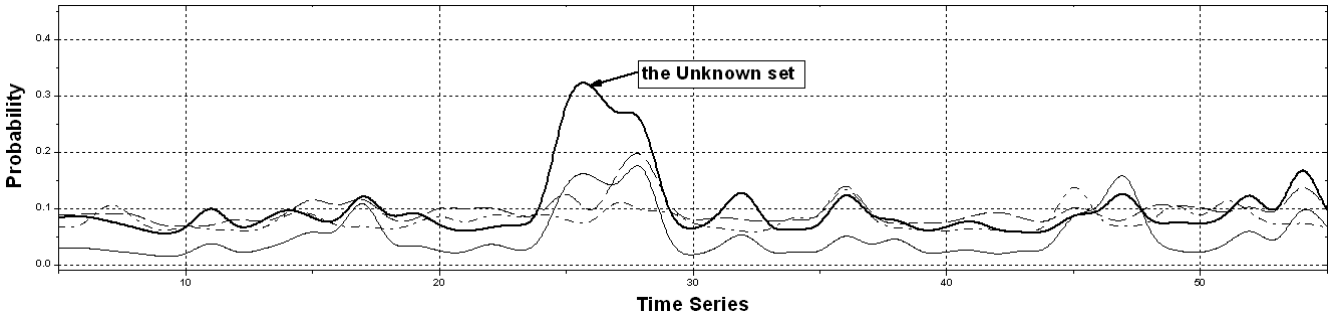

Figure. 4 Result of the attack probability time series using Codered worm traffic trace

in the HMM model, and this can enable us to calculate the *hidden state set* path, the probability of each anomaly, using the Viterbi algorithm.[7]

For verification of the proposed scheme, we experiment using five real-world traffic data sets including a particular network attack; i.e. Slammer Worm, Witty Worm, Codered Worm, Denial of Service (DoS), and Distributed DoS (DDoS). We confirm an anomalous event via a well-known intrusion detection system such as Snort and Bro, beforehand.

6. Experimental Results

We categorize network attacks as four types; *Worm*, *DoS/DDoS*, *Unknown*, *Clear* (free of attack) set. These are equivalent to the *hidden state set* in the HMM model.

Figure 2 depicts the probability of attacks using a *DoS/DDoS* traffic trace. In the case of a *DoS/DDoS* traffic trace; it is clear that the *DoS/DDoS* set has a high probability.

The result using a Codered worm traffic trace is shown in Figure 3. The probability of the *Unknown* set increases to a maximum of 33.02%, between 25 and 30 seconds, as shown in Figure 3. The reason is that IP Scanning in the *observable state set*, which is very common in the Codered worm, was learned as the *Unknown* set (the prior). The *Worm* set has a maximum probability of 22.49%, which occurs between the equivalent times in the *Unknown* set.

Probabilistic attack prediction using the Slammer worm traffic trace is shown in Figure 4. In case of the *Worm* set, the maximum probability of 33.9% is reached at 20 seconds in the time series. There is a fluctuation indicating that the initial state is unstable, because Reflection was learned as the *observable state set*, which is consecutively generated in the traffic trace used.

As shown in Figure 5, the *Worm* set has a high probability between 12 and 42 seconds. The maximum probability in the *Worm* set is 33.9%, which occurs within this period. The probability in the *Unknown* set reaches the maximum of 51.3%, at 47 seconds. This is because *Unknown* was learned as the *observable state set*.

The disadvantage of the proposed mechanism is indicated by the result that the overall detection probability is slightly reduced. This is due to the increased difficulty of detection resulting from the combination of anomalous traffic with normal traffic. Furthermore, the learning trace set is not sufficient. We plan to make improvements to overcome this weakness, and publish these improvements in a complete paper.

7. Conclusion

In this paper, we proposed a network anomaly detection scheme adopting probabilistic properties of HMM, derived symptoms for verification of the proposed scheme, then, and defined network attacks. We determined the probability of network attack in a time series via learning and decision phases of HMM.

Our contribution is summarized in two parts. First, our proposed scheme enables an administrator to determine whether or not a particular attack is occurring, via representation of the probability as a time series, such as in

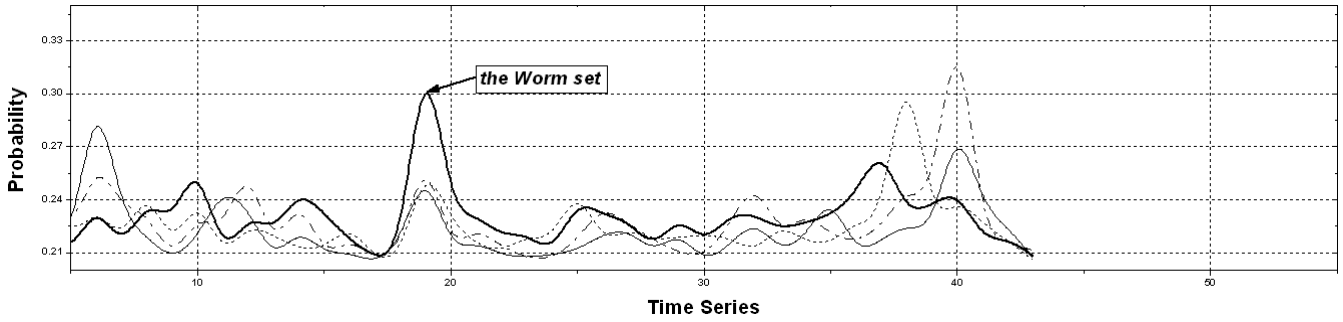

Figure. 5 Result of the attack probability time series using Slammer worm traffic trace

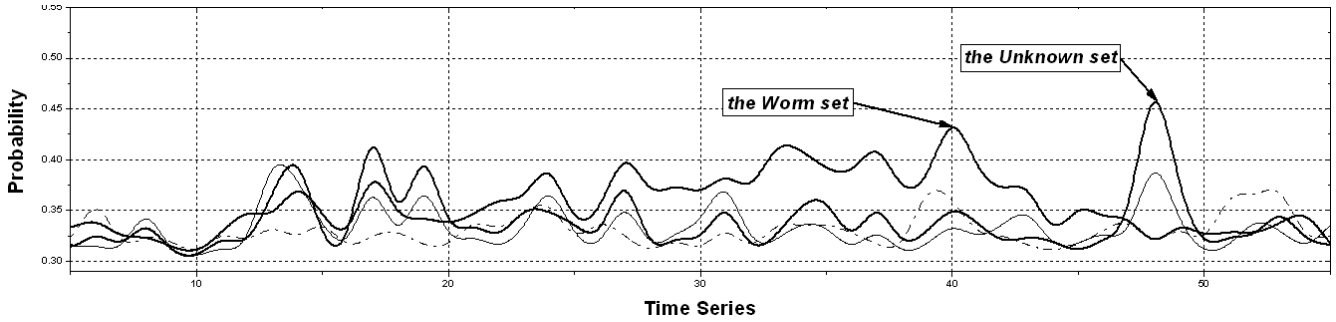

Figure. 6 Result of the attack probability time series using Witty worm traffic trace

weather forecasting. It is an epoch-making method, in contrast to attempting to reduce the rate of false positives and false negatives to a sufficiently low value, which is highly impractical. Second, we adapt a property of HMM to the detection of network anomalies which are not directly observable.

The results of our research can be used to; (1) define network symptoms, (2) monitor network traffic in real-time, (3) detect network attacks, and (most importantly) (4) learn of an unknown attack.

References

[1] Matthew V. M., "Network traffic anomaly detection based on packet bytes", *Proceedings of the 2003 ACM symposium on Applied computing*, Melbourne, Florida, March, 2003.

[2] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: existing solutions and latest technological trends," *Elsevier Computer Networks*, Vol. 51, Issue 12, pp. 3448–3470, February 2007.

[3] S. E. Smaha, "Haystack: An Intrusion Detection System," *IEEE Fourth Aerospace Computer Security Applications Conference*, pp. 37 - 44, Orlando, FL, December 1988

[4] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A Sense of Self for Unix Processes," *IEEE Symposium on Research in Security and Privacy*, pp. 120-128, Oakland, CA, USA, May 1996.

[5] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," *7th USENIX Security Symposium (SECURITY-98)*, pp. 79-94. , Berkeley, CA, USA, January 1998

[6] N. Ye, "A Markov Chain Model of Temporal Behavior for Anomaly Detection," *In Workshop on Information Assurance and Security*, New York, USA, June 2000.

[7] Rabiner L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of IEEE*, Vol. 77, No. 2, February 1989.

[8] Leek T. R., "Information Extraction using Hidden Markov Models," *Master's thesis*, UC San Diego, March 1996.