Towards an Immunity-Based Anomaly Detection System for Network Traffic

Takeshi Okamoto[1] and Yoshiteru Ishida[2]

[1] Department of Network Engineering, Kanagawa Institute of Technology,
1030, Shimo-ogino, Atsugi, Kanagawa, 243-0292 Japan
take4@nw.kanagawa-it.ac.jp

[2] Department of Knowledge-Based Information Engineering,
Toyohashi University of Technology,
1-1, Tempaku, Toyohashi, Aichi, 441-8580 Japan
ishida@tutkie.tut.ac.jp

Abstract. We have applied our previous immunity-based system to anomaly detection for network traffic, and confirmed that our system outperformed the single-profile method. For internal masquerader detection, the missed alarm rate was 11.21% with no false alarms. For worm detection, four random-scanning worms and the simulated *metaserver worm* were detected with no missed alarms and no false alarms, while a simulated *passive worm* was detected with a missed alarm rate of 80.57%.

1 Introduction

Anti-virus systems protect computers and networks from malicious programs, such as computer viruses and worms, by discriminating between malicious programs and harmless programs, and by removing only the former. Therefore, anti-virus systems can be considered as the computer's immune system.

An innovative method, a "*virus throttle*," was proposed by Williamson in 2002 [1]. The *virus throttle* slows and halts high-speed worms without affecting normal network traffic. In our previous study [2], we proposed a "*worm filter*" for preventing both slow- and high-speed worms from spreading. The *worm filter* limits the number of unacknowledged requests, rather than the rate of connections to new computers. However, the *worm filter* cannot stop *metaserver worms*, *topological worms, or passive worms* [3]. Not only do these worms attempt to connect to active servers that return a reply, but also their network traffic is similar to that of an actual user. In other studies [4,5], we proposed the immunity-based anomaly detection system for a UNIX command sequence that can discriminate between a legitimate user and a masquerader who abuses someone else's account. This system would be expected to detect the above worms, because the difference between a user and a worm is greater than those between different users.

We have applied this system to anomaly detection for network traffic to detect worms. The algorithm of the immunity-based anomaly detection system is described in detail in section 2. In section 3, we describe our experimental data and evaluation model. Section 4 presents a comparison of the performance evaluation between our immunity-based method and the conventional method. Sections

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part II, LNAI 4252, pp. 123–130, 2006.

5 and 6 present performance evaluations against simulated masqueraders, real worms, and some simulated worms.

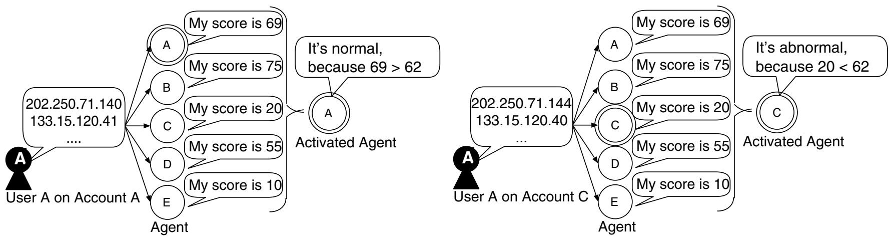
2 Immunity-Based Anomaly Detection for Network Traffic

At the heart of the immune system is the ability to distinguish between “self” (the body’s own molecules, cells, and tissues) and “nonself” (foreign substances, such as viruses or bacteria). We define an operation sequence of legitimate users on their own account as “self,” and all other operation sequences produced by masqueraders, worms, *etc.*, as “nonself.”

The immunity-based anomaly detection system (IADS) uses multiple user-specific agents. Each agent has a unique profile that is expressed by a parameter of the hidden Markov model (HMM) $\lambda = [\pi, A, B]$, as our previous study [6] indicated that it performs well. The parameters of the HMM, using the Baum-Welch algorithm, is estimated from sequences of each legitimate user. The agent computes a likelihood $P(O|\lambda)$ of the sequence O with the profile λ. The agent computes a high score (*i.e.*, a high likelihood) for only the sequences of the user corresponding to the agent. In this way, the agent is specialized so as to recognize the user.

In every operation, all agents compute their own score for a new operation sequence. The agent associated with that account is activated and compares it with the scores of all other agents. If the user of the account is the owner of the account, the score will be relatively high, but not necessarily the highest score compared with the scores of the other agents. Thus, we set a threshold value (Th), which is a percentage of the difference between the minimum (Min) and maximum scores (Max). If the activated agent computes a score higher than the threshold, obtained by the equation $Min + (Max - Min) \times Th$, the activated agent classifies the operation sequence as normal (*i.e.*, “self”). Otherwise, the agent classifies the operation sequence as abnormal (*i.e.*, “nonself”), and it raises an alarm. Furthermore, provided that all the scores are equal to the minimum value of all the computable $P(O|\lambda)$, the sequence is regarded as abnormal. Conversely, if all the scores are equal to the maximum value of all the computable $P(O|\lambda)$, the operation sequence is regarded as normal. Examples of discrimination between “self” and “nonself” are shown in Fig. 1.

In this study, the IADS monitored network traffic, more specifically outgoing TCP SYN packets. The IADS makes a sequence of destination IP addresses of these packets. However, the IP address space is too large to allow construction of a profile. Hence, pre-processing of the sequence is done to scale down the IP address space. The number of different IP addresses to which a user transmits a packet more than once is very small. Hence, the IADS assigns a unique number v to each IP address to which the user transmits a packet more than once, where v begins with 0. The assignment table is created at profile construction, and all the sequences are replaced according to this table. If the IP address does not exist in this table, the IP address is assigned a unique number $v_{max} + 1$, where $v_{max} + 1$



(a) "Self" (normal)

(b) "Nonself" (abnormal)

Fig. 1. Discrimination between "self" (a) and "nonself" (b). If we set the threshold value to 80%, and the different agents compute 10, 20, 55, 69, and 75, the effective threshold value is calculated to be 62 (= 10 + (75 − 10) × 0.80). In the case of (a), if user A browses a website on his/her own account, agent A that is specialized to recognize user A is activated, and decides that the operation sequence is normal. In the case of (b), if user A browses a website on the account of user C, the agent C that is specialized to recognize user C is activated, and it decides that the operation sequence is abnormal.

is equal to the number of all IP addresses included in the table. Empirically, we assign a value of 100 ± 20 to v_{max}.

3 Experimental Data and Evaluation Model

To evaluate detection performance, we captured network traffic from 12 users for about one month. We focused on web traffic, as this accounts for the majority of network traffic. Hence, we extracted only outgoing TCP SYN packets with destination port 80. The web traffic of each user contains more than 3,000 requests. The first 500 requests for each user are used as training data to allow construction of a profile. The next 1,000 requests are test data to evaluate the detection performance. The test for the sequence is performed at every request. Each sequence length is set to 400, so that the total number of tests for each user is 601.

For evaluation of user traffic, we simulate anomalous behavior by testing one user's request sequence against another user's profile, as our data do not include anomalous behavior. This simulation corresponds to the evaluation of masquerader detection.

Anomaly detection is important to reduce false alarms, because too many false alarms can cause the "cry-wolf syndrome." Hence, we evaluate the missed alarm rate with no false alarms.

4 Immunity-Based Method vs. Single-Profile Method

We compared our immunity-based method, which uses multiple profiles, with the conventional single-profile method using only one profile (*e.g.*, [6,7]). Each evaluation was performed for 12 users as described in section 3.

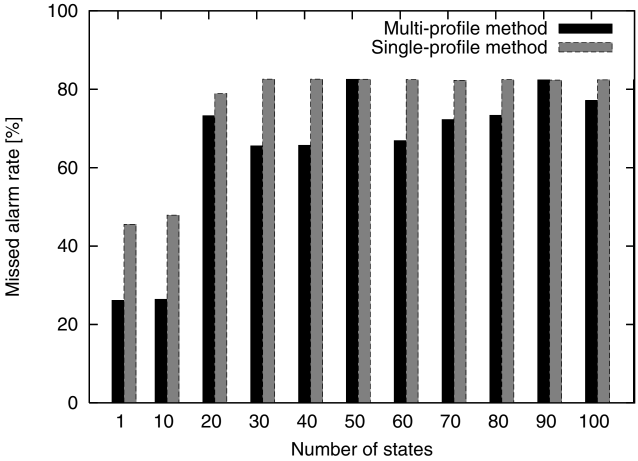


Fig. 2. Missed alarm rate with no false alarms for the single- and multi-profile methods as a function of the number of states in the HMM

Figure 2 shows the missed alarm rate with no false alarms for the single- and multi-profile methods as a function of the number of hidden states in the HMM, where the hidden states are assumed to be working states, such as searching, e-learning, blogging, *etc.* For all numbers of states, the immunity-based method outperformed the single-profile method. As shown in Fig. 2, the HMM for which the number of states = 1 showed the best performance. This HMM depends on only the request frequency of different websites. In addition, we confirmed that this HMM largely surpassed a method based on Markov chains. That is, the frequency property rather than the ordering property dominates the characteristics of request sequences. Therefore, we set the number of states to one in all performance evaluations after this section.

In our previous study [6], we confirmed that the missed alarm rate was inversely proportional to the number of states. In contrast, the missed alarm rate was proportional in Fig. 2. The reasons for this discrepancy are currently under investigation.

5 Performance Evaluation of User Traffic

Our approach may be less well-suited for detecting external masqueraders than for detecting internal masqueraders because our agents have no profiles for external masqueraders. Hence, we evaluated the performance of our detection scheme for internal and external masqueraders separately. We should evaluate all combinations of internal users among the 12 users because the performance of anomaly detection would depend on the combinations. We conducted 1,000 combinations chosen at random from among all possible combinations. For each combination, the first 6 users were assumed to be external masqueraders, while the remaining users were assumed to be internal users.

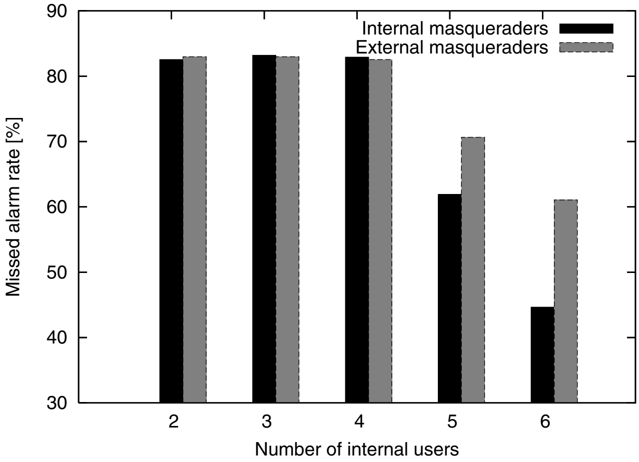


Fig. 3. Missed alarm rate with no false alarms for internal and external masqueraders. Each missed alarm rate is an average over 1,000 combinations.

Figure 3 shows the missed alarm rate with no false alarms for internal and external masqueraders, with varying numbers of internal users from two to six. As expected, the large difference in the missed alarm rates between internal and external masquerader detection was confirmed in the results with five and six internal users. In addition, both of the missed alarm rates were very high for small numbers of internal users. However, as the missed alarm rate seems to decrease with increasing number of internal users, the addition of internal users may reduce the missed alarm rate. In addition, as our previous study confirmed that the addition of diverse agents decreased the missed alarm rate [4,5], such diverse agents may reduce the number of missed alarms.

All the above evaluations were performed with only one threshold for all users to make evaluation easier. As fluctuations between agents' scores for each sequence were very large, we set a different threshold with no false alarms for each user. The results are shown in Fig. 4, in which there are no external users. As expected, the missed alarm rates decreased in the case with different thresholds. The average missed alarm rate was 11.21%. It is noteworthy that there were no missed alarms for the following users: E, F, I, K, and L.

6 Performance Evaluation of Worm Traffic

We evaluated worms, setting the threshold of the IADS to a different value with no false alarms for each user.

We have evaluated four random-scanning worms in the wild: `CodeRedv2`, `CodeRedII`, `Slammer`, and `Blaster`. Although `Slammer` and `Blaster` do not send a packet to TCP port 80, we assumed that they sent to this port. As a result, there were no missed alarms and no false alarms on any of the accounts for all the

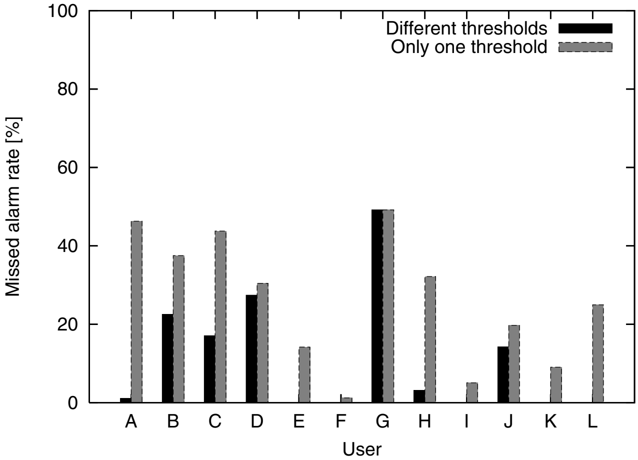


Fig. 4. Missed alarm rate with no false alarms for different thresholds and only one threshold. For the former, the thresholds were: 99.9%, 37.6%, 68.0%, 29.6%, 83.2%, 99.9%, 24.6%, 99.9%, 99.9%, 38.2%, 95.2%, and 89.6%. For the latter, the threshold was 24.6%, which coincided with the threshold of user G.

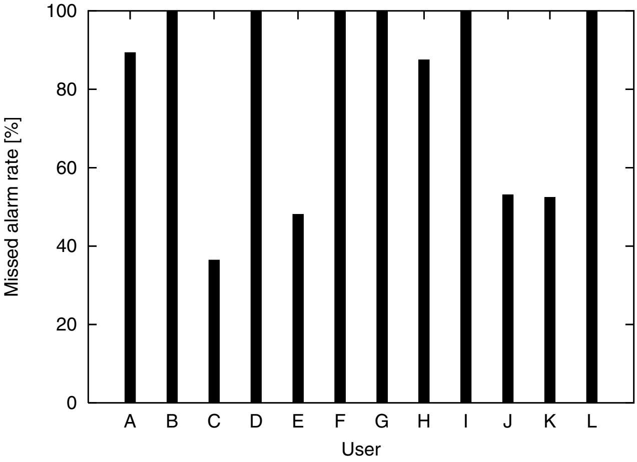


Fig. 5. Missed alarm rate with no false alarms for a simulated *passive worm*. The average missed alarm rate was 80.57%. User C showed the best missed alarm rate of 36.44%.

worms examined, because none of the users had ever connected to IP addresses generated randomly by these worms.

Three worms, a *metaserver worm*, a *topological worm*, and a *passive worm* [3], escaped our previous method (*i.e.*, worm filter) [2] by propagating to only the active servers that return a reply.

The IADS would be expected to detect the *metaserver worm* and the *topological worm*, because both of these worms are difficult to connect to IP addresses to which the user has ever connected. We simulated a *metaserver worm*, such as `Santy` worm, which attempts to propagate to IP addresses in the search results provided by Google^{TM} (`www.google.com`). The traffic of the simulated worm was evaluated and the results indicated that there were no false alarms and no missed alarms for all accounts.

The *passive worm*, which either waits for target computers to visit or follows user's requests into target computers, is more difficult for an anomaly detection system to detect, because its behavior is similar to that of the user. We simulated the *passive worm*, which propagates to servers to which the user has connected, immediately after the connection. For example, if a user browses websites: A → B → B → C in this order, the infected computer attempts to connect to A → A → B → B → B → B → C → C. The traffic of the simulated worm was evaluated and the results are shown in Fig. 5. The average missed alarm rate was 80.37%. Although this rate was not good, it is notable that the worm was detected on six accounts. Investigation of request sequences on the six accounts indicated that the frequency of $v_{max} + 1$ (*i.e.*, the frequency of browsing new websites not included in the assignment table of websites) is relatively high. Hence, the detection of this worm would depend on the frequency of $v_{max} + 1$.

7 Conclusions

We applied our previous immunity-based system to anomaly detection for network traffic. The results of this study confirmed that our system outperformed the single-profile method. For internal masquerader detection, the missed alarm rate was 11.21% with no false alarms. For worm detection, four random-scanning worms and the simulated *metaserver worm* were detected with no missed alarms and no false alarms, while the simulated *passive worm* was detected with a missed alarm rate of 80.57%. Further studies will require the generation of diverse agents to reduce the missed alarm rate. Inspired by the mechanism of adaptation in the immune system, methods should be developed to update the user's profile so that each agent can adapt to recent behavior of the user.

Acknowledgements

This work was supported in part by Grants-in-Aid for Scientific Research (B) 16300067, 2004. This work was partly supported by the 21st Century COE Program "Intelligent Human Sensing" from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

References

1. Williamson, M.M.: Throttling viruses: restricting propagation to defeat malicious mobile code. In: ACSAC Security Conference 2002. (2002) 61–68
2. Okamoto, T.: A worm filter based on the number of unacknowledged requests. In: KES 2005, LNAI 3682 (2005) 93–99

3. Weaver, N., Paxson, V., Staniford, S., Cunningham, R.: A taxonomy of computer worms. In: The 2003 ACM Workshop on Rapid Malcode, ACM Press (2003) 11–18
4. Okamoto, T., Watanabe, T., Ishida, Y.: Towards an immunity-based system for detecting masqueraders. In: KES 2003, LNAI 2774 (2003) 488–495
5. Okamoto, T., Watanabe, T., Ishida, Y.: Mechanism for generating immunity-based agents that detect masqueraders. In: KES 2004, LNAI 3214 (2004) 534–540
6. Okamoto, T., Watanabe, Y., Ishida, Y.: Test statistics for a masquerader detection system – a comparison between hidden markov model and other probabilistic models. Transactions of the ISCIE **16**(2) (2003) 61–69
7. Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., Vardi, Y.: Computer intrusion: Detecting masquerades. Statistical Science **16**(1) (2001) 58–74