Evaluation of Secular Changes in Statistical Features of Traffic for the Purpose of Malware Detection

Kenji Kawamoto, Masatsugu Ichino, Mitsuhiro Hatada, Yusuke Otsuki, Hiroshi Yoshiura, and Jiro Katto

Abstract. Applications and malware affecting them are dramatically changing. It isn't certain whether the currently used features can classify normal traffic or malware traffic correctly. In this paper, we evaluated the features used in previous studies while taking into account secular changes to classify normal traffic into the normal category and anomalous traffic into the anomalous category correctly. A secular change in this study is a difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. The evaluation is based on the Euclidean distance between the normal codebook or anomalous codebook made by vector quantization and the test data. We report on what causes these secular changes and which features with little or no secular change are effective for malware detection.

1 Introduction

The threat of malware is increasing. Malware is the word made from "malicious" and "software" and this sort of software compromises the security of or hijacks computers. A certain web site [1] claimed about 4,000 malware incidents occured in the first half of 2011 in Japan. The threat of stealth botnets and infections through

Kenji Kawamoto · Jiro Katto
Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan
e-mail: kawamoto@kom.comm.waseda.ac.jp, katto@waseda.jp

Masatsugu Ichino · Yusuke Otsuki · Hiroshi Yoshiura
University of Electoro-Communications, Japan
e-mail: ichino@inf.uec.ac.jp, otsuki@uec.ac.jp, yoshiura@hc.uec.ac.jp

Mitsuhiro Hatada
NTT Communications Corporation, Tokyo, Japan
e-mail: m.hatada@ntt.com

R. Lee (Ed.): Software Engineering, Artificial Intelligence, Networking, SCI 443, pp. 1–11.
springerlink.com

web sites is especially increasing. In addition, new kinds of malware are appearing. Malware detection has thus becomes important for the safety of Internet usage.

Fujiwara [2] categorized research on detecting malware and found that it tended to focus on detecting known malware: methods of detecting unknown malware have not been discussed sufficiently. In this paper, we focus on detecting unknown malware by using traffic data because we suppose that normal traffic is quite different from anomalous traffic data. Moreover, we thought that malware might be easier to detect if we treated traffic as a time series signal. For example, there are numerous biometric recognition algorithms that work for lip movements, etc, and Ichino [3] showed that the accuracy of algorithms that use images streams is better than those that use static-image matching.

There are a lot of malware detection methods using packet payload information in previous research. For example, Karamcheti [4] used the inverse distributions of packet contents. However, it is impossible to detect malware in encrypted communication and to maintain privacy. Therefore, we focus on the packet header on the Internet in this research. After extracting the features of these headers, we classified the traffic into normal or anomalous.

Features used in malware detection have not been thoroughly evaluated. In this study, we tried to determine ones that would be effective for classifying normal or anomalous traffic by using CCCDATAset2009, 2010, 2011 [5] (we refer to these sets as CCC2009, CCC2010, CCC2011 later in this paper) as the anomalous traffic data and traffic data captured in an intranet as normal traffic data. We studied secular changes that occur over the course of three years worth of data. A secular change is difference in a feature between the date the training data were caputred and the date the test data were captured in the same circumstance. It is important to take into account secular changes because traffic data may dramatically change in a year. Features for which discrimination rates vary greatly from year to year aren't effective for malware detection. Therefore, secular changes are important factor for the evaluation of features.

This paper is organized as follows. In section 2, we describe the previous research and utilized features. Section 3 explains our experiment, and section 4 discusses accurate features for detecting malware. Section 5 is the conclusion.

2 Related Works

Here, we describe the features used in the previous research on malware detection and network intrusion detection.

Sato [6] discussed a network intrusion detection system that incorporated detection modules based on timeslot and flow count analysis. The timeslot method extracts features at fixed time intervals by refering to the frequency of TCP header flags and the number of TCP, UDP, and ICMP packets. The flow count method, on the other hand, extracts features from every flow. A flow is a group of packets that have the same five-tuple of protocol type, source address, source port, destination address, and destination port. Fragmented packets and the inverse of the same port

number frequency are used in flow count methods. In the field of malware detection, it is important to detect malware traffic quickly in order to prevent malware from spreading thourgh the network. However, detecting malware in real time by using flow count method is hard because feature extraction finishes when all the same flow packets are captured. Thus, we shall use the timeslot method in this study.

Hiramatsu [7] studied a clustering method for defining multiple normal states from network traffic data. The normalization numbers of ICMP, SYN, FIN, UDP and TCP except SYN, and FIN packets extracted every 60 minutes are used to define multiple normal states.

Kugisaki [8] focused on the host's transmission intervals as a feature and confirmed that there is a difference in transmission interval between traffic originating from human and botnet.

The above studies show that the number of packets, transmission interval, TCP flags, and port number is often used in the field of malware detection and network anomalous detection.

3 Evaluation Experiment

3.1 Evaluation Eature

We use the existing research as a guide to extract features from the packet header and compiled statistics about the header information. Table 1 shows the 36 types of features evaluated in this paper.

Table 1 36 types of features

number	feature [unit]
1	number of packets
2	sum of packet sizes [byte]
3	mean packet size [byte]
4	minimum packet size [byte]
5	maximum packet size [byte]
6	standard deviation of packet size [byte]
7	mean transmission interval [seconds]
8	minimum transmission interval [seconds]
9	maximum transmission interval [seconds]
10	standard deviation of transmission interval [seconds]
11	number of SYN packets
12	number of FIN packets
13	number of PSH packets
14	number of ACK packets
15	number of RST packets
16	number of URG packets
17	number of SYN/ACK packets
18	number of FIN/ACK packets

number	feature [unit]
19	number of PSH/ACK packets
20	number of RST/ACK packets
21	ratio of SYN packets to TCP packets
22	ratio of FIN packets to TCP packets
23	ratio of PSH packets to TCP packets
24	ratio of ACK packets to TCP packets
25	ratio of RST packets to TCP packets
26	ratio of URG packets to TCP packets
27	ratio of SYN/ACK packets to TCP packets
28	ratio of FIN/ACK packets to TCP packets
29	ratio of PSH/ACK packets to TCP packets
30	ratio of RST/ACK packets to TCP packets
31	number of ICMP packets
32	number of UDP packets
33	number of 69/UDP port packets
34	number of 80/TCP port packets
35	number of 110/TCP port packets
36	number of 443/TCP port packets

3.2 Methods Used in the Experiment

1. Evaluation method
   The method to classify the test traffic into the normal or anomalous is a follows. First, we prepared a normal codebook and an anomalous codebook by separately

using normal traffic data and malware traffic data as training data. The codebooks were made by vector quantization. Each codebook have one dimension to evaluate one individual feature. The timeslot interval for extracting features is 0.1, 1, 10, or 100 seconds, the vector quantization algorithm is LBG and splitting and vector quantization level is 2, 4, 8, 16, or 32. Vecotor quantization level means how many codebooks are made by the vector quantization. For the parameters(set of features, timeslot interval and vector quantization level), we discriminated on the basis of the Euclidean distance in the feature space between the labeled test data and the normal or anomalous codebook. If the distance between the test data and the normal codebook is shorter than the distance between the test data and the anomalous codebook, the test data is classified into normal traffic. If not, the test data is classified into malware traffic.

As evaluation indicators, we used the true negative rate (TNR), i.e, the rate at which normal traffic is correctly classified into normal category, and the true positive rate (TPR), i.e, the rate at which malware traffic is correctly classified into anomalous category. For each features and parameters, we calculated TNR and TPR by using traffic data from 2009, 2010 and 2011 in every timeslot.

2. Experimental data

We used CCC2009 for the anomalous codebook and normal traffic data captured on Mar 13, 14, 15, 2009 as the normal codebook. The test data for the malware traffic is CCC2009, CCC2010, and CCC2011 and the test data of the normal traffic is from 2009, 2010, and 2011. The CCCDATAset was captured in a honeypot and the normal traffic was captured in an intranet. The normal traffic and malware traffic data were captured on the same dates.

It would have been desirable to use normal and malware traffic data captured in the same circumstance for the experiment. However, resources on malware traffic are rather limited. In addition, normal traffic data captured in honeypot would not be realistic because nobody generates traffic in a honeypot. To handle this problem, the normal traffic data needs to be preprocessed to imitate the capture circumstances of malware traffic.

- Preprocessing for normal traffic

The normal traffic data was preprocessed to meet the following requirements.

a. Generated from one host.
   It is necessary to imitate the capture circumstances of malware traffic.
b. Generated by normal users.
   If the host is infected with malware, it will download or update new malware or try to connect to the Internet. However, such transmissions are normal in terms of their behavior. In this research area, it is important to be able to distinguish malware transmissions and behavior of human with no malicious intent. Hence, the normal traffic generated by a normal user must be used.

- Preprocessing for malware traffic

In this experiment, we used honeypot traffic data from CCC2009, CCC2010, and CCC2011, which includes scan traffic, exploit traffic, and infected traffic. This

means it includes non-infected traffic data. However, it is essential for us to use only infected traffic data in the evaluation experiment. Hence, we did preprocessing to extract the malware traffic from the other attacking traffic data. The procedure for doing so is as follows.

a. Cut out control packets generated only in the honeypot circumstance.
b. Divide the pcap data in the OS reset interval of the honeypot.
c. Check whether traffic is truly infected by refering to the malware collection log provided in the CCCDATAset and look for the first packet of the malware transmission.
d. Extract the traffic data after the first packet of the malware transmission.

4 Experimental Results and Analyses

Here, we summarize the experimental results and analyze which features are effective at detecting malware thourgh secular changes in TNR and TPR, and we classify the features into two categories, one is the case that the secular change is big, the other in which the secular change is small. Then, we determine also which timeslots and vector quantization levels are effective. Finally, we summarize which features overall are the most effective.

First, we looked at the changes in the TNR and TPR over the course of three years. Table 2 shows the discrimination rates of TNR and TPR in 2009, 2010, and 2011. The average TNR or TPR is the mean of the corresponding values calculated for each feature types, timeslot length, and number of vector quantizattion levels.

Table 2 Discrimination rates of TNR and TPR

year	2009	2010	2011
average(TNR)	36.1%	35.2%	40.7%
average(TPR)	57.0%	54.1%	51.2%

The average TNR in 2011 is the highest, while the average TPR in 2011 is the lowest. From this result, it is clear that the secular change in the test data affects the discrimination rate.

4.1 Secular Change

1. TNR
   Figure 1 shows features for which the average TNR is higher than 50% during the three years.

   - Features with large secular change
     Features 2, 3, 9, 14, 17,18, 19, 20, 24, 34, and 36(these numbers match those in Table 1) show large secular changes in TNR. Except for feature 9, the average TNR are the highest in 2011. Figure 2 shows why the average TNR of these

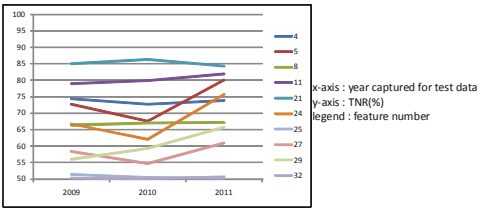

Fig. 1 Change in features for which the average TNR is higher than 50% during three years

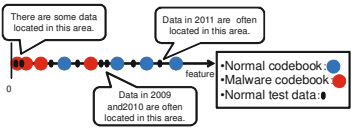

Fig. 2 Why TNR is the highest in 2011

features is the highest in 2011. In terms of the above features, almost all of the normal test data in 2011 can be classified as normal because the feature values are too high and very close to the normal codebooks. However, the normal test data in 2009 and 2010 are often classified as malware tarffic. This is why the average TNR is the highest in 2011. This situation arises from the differnce in the number of packets in the normal test data. Table 3 shows how many packets there are in each year. The unit of the average is packets per slot.

Table 3 Number of packets of normal test data for each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	14.3	10.8	5.51
standard deviation	33.8	23.1	9.5
timeslot 1 seconds			
year	2009	2010	2011
average	189.9	33.2	30.9
standard deviation	580.4	61.3	43.3
timeslot 10 seconds			
year	2009	2010	2011
average	340.8	157.0	1039.7
standard deviation	1561.7	451.1	1176.7
timeslot 100 seconds			
year	2009	2010	2011
average	1656.4	6060.0	9225.8
standard deviation	4520.9	1362.3	10603.3

Table 4 TNRs over 90% for three years for minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	4	98.4%	92.6%	90.4%
0.1 seconds	8	98.2%	92.6%	93.3%
1 seconds	4	99.8%	98.7%	100%
1 seconds	8	100%	98.7%	100%
1 seconds	16	98.0%	99.1%	100%
10 seconds	2	100%	100%	100%
10 seconds	4	100%	100%	100%
10 seconds	8	100%	100%	100%
10 seconds	16	100%	100%	100%
100 seconds	2	99.3%	100%	100%
100 seconds	4	100%	100%	100%
100 seconds	8	100%	100%	100%
100 seconds	16	100%	100%	100%

Table 3 shows that the number of test data packets in 2011 is the largest. The contents of traffic is similar for each year. Hence, the number of test data packets significantly affects the secular change.

- Features with small secular changes
  Features 4, 7, 8, 11, 21, 25, 28, 30, 31, 32, and 35(these number match those of Table 1) show little secular change in TNR. These features are typically ratios(for example, ratio of SYN packets to TCP packets). Therefore, it is effective to use such features for suppressing drops in discrimination rates caused by secular changes.

Among these features, features 4 (minimum packet size), 11 (number of SYN packets), and 21 (ratio of SYN packets to TCP packets) have average TNRs higher than 75%.

– Minimum packet size
  Table 4 shows TNRs over 90% for three years for the minimum packet size.

  Table 5 shows the average and standard deviation of the minimum packet size in the normal and anomalous test traffic data. The unit of the average is byte per slot.

Table 5 Average and standard deviation of minimum packet size in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	79.5	81.0	93.7	68.6	84.7	114.4
standard deviation	95.6	113.1	134.7	32.4	89.1	174.6
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.1	60.8	60.0	70.7	73.3	101.1
standard deviation	1.3	7	0	31.1	34.8	83.1
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.9	60.1	60	67.8	71.2	102.2
standard deviation	0.2	3	0	28.2	40.4	113.9
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	60.4	60	60	69.3	66.8	84.2
standard deviation	2.8	0	0	44.9	40.4	60.4

Table 6 Number of packets of anomalous test data in each year

timeslot 0.1 seconds			
year	2009	2010	2011
average	42.3	11.6	3.3
standard deviation	78.6	11.7	4.3
timeslot 1 seconds			
year	2009	2010	2011
average	139.6	166.6	6.6
standard deviation	121.2	140.1	13.6
timeslot 10 seconds			
year	2009	2010	2011
average	781.6	1439.3	23.4
standard deviation	584.6	1161.6	54.2
timeslot 100 seconds			
year	2009	2010	2011
average	3350.4	7578.9	348.7
standard deviation	5212.5	9580.1	1711.7

Table 5 shows that the minimum packet size of normal traffic is almost always 60 bytes if the timeslot interval is larger than 1 seconds. On the other hand, the minimum packet size of anomalous traffic varies. There is an enormous difference between the standard deviation of the minimum packet size of normal traffic and that of anomalous traffic. In normal traffic, the standard deviation is almost always 0, in contrast, it is much larger than zero for anomalous traffic. We suppose that this difference would be effective for malware detection. That is, we think that both of the minimum packet size and its standard deviation are useful and efficient features for detecting malware.

– Number of SYN packets, ratio of SYN packets to TCP packets
  TNR is very high when the number of SYN packets or ratio of SYN packets to TCP packets is used. This is because anomalous traffic data tends

to behave like a SYN scan. Because of this, the values in the anomalous codebook are much larger than those of the normal codebook. Moreover, normal test traffic data doesn't have a lot of SYN packets. Therefore, almost all of the normal traffic data are classified in the normal category. That is why the TNR is very high. However, malware traffic doesn't always have SYN scans. Although it is difficult to use these features for classifying whether traffic is normal or malware, it would be effective for predicting or detecting attack.

2. TPR
   Figure 3 shows for which the TPR is higher than 70% over the course of three years.

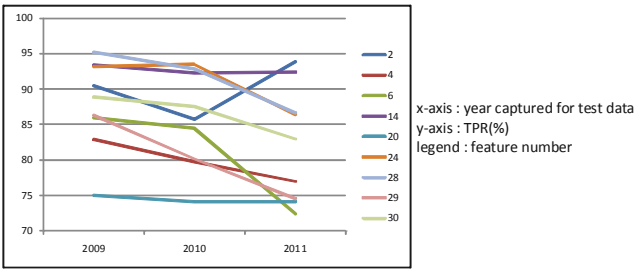

Fig. 3 Changes in TPR higher than 70% over the course of three years

- Features with large secular change
  Features 1, 3, 5, 6, 7, 9, 10, 11, 21, and 29 have large secular changes in TPR. Except feature 1 and 9, the average TPR is lower in 2011 than in 2009 and 2010. We consider there are two reasons why TPR is the lowest in 2011. The first reason is that the anomalous traffic data in 2011 has fewer SYN scans than the anomalous traffic data in 2009 and 2010. If traffic data doesn't have a lot of SYN scans, the average packet size is large. The behavior is close to that of normal traffic data. That's why the average TPR is the lowest in 2011 for features 3, 11, and 21. The second reason is that the number of packets in the anomalous test data in 2011 is fewer than in 2009 or 2010. Table 6 shows the number of packets in the anomalous test data. The unit of the average is packets per slot.

  It is clear that the number of packets in the anomalous test data is the fewer in 2011 than in 2009 or 2010. There isn't a large year-to-year difference in the anomalous test data as regards the number of PSH/ACK packets. However, the ratio of PSH/ACK packets to TCP packets is the highest in the anomalous test data in 2011 and close to the ratio of the normal test data. That's why the average TPR for feature 29 is the lowest in 2011.

- Features with small secular change
  Features 4, 14, 17, 18, 19, 20, 25, 30, 31, 32, 34, and 36 have small changes in TPR. Among these features, those that have average TPRs higher than 80% are 4 (minimum packet size), 14 (number of ACK packets), 21 (ratio of RST/ACK packets to TCP packets).

- Minimum packet size
  Table 7 shows TPRs over 90% over the course of three years for the minimum packet size.

Table 7 TPRs over 90% for three years for the minimum packet size

timeslot	vector quantization level	2009	2010	2011
0.1 seconds	32	99.8%	94.6%	90.2%
1 seconds	32	100%	99.6%	95.8%
10 seconds	32	94.0%	92.0%	92.4%

Table 8 Average number of ACK packets in normal and anomalous test traffic

timeslot 0.1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	10.9	6.9	3.3	2.6	0.6	2.3
timeslot 1 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	174.1	20.0	19.0	3.1	1.3	3.4
timeslot 10 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	289.1	103.2	787.2	10.0	10.9	11.8
timeslot 100 seconds						
	normal			anomalous		
year	2009	2010	2011	2009	2010	2011
average	1305.4	432.8	6669.8	40.7	73.5	37.5

In terms of TPR, the minimum packet size and its standard deviation are effective for malware detection just as they are for the TNR.

- Number of ACK packets
  Table 8 shows the average number of ACK packets in normal and anomalous traffic for three years.

  From Tables 3 and 6, we can see that there is no large difference in the number of packets between normal and anomalous test traffic. Moreover, the number of ACK packets in the anomalous test traffic is very few in comparison with that in the normal test traffic. Therefore, the number of ACK packets is an effective feature to classify traffic data into normal or anomalous.

- Ratio of RST/ACK packets to TCP packets
  In terms of the ratio of RST/ACK packets to TCP packets, TPR itself is high and the secular change is small. However, both sorts of test traffic have a lot of timeslot intervals in which the ratio of RST/ACK packets to TCP packets is 0. Moreover, the anomalous codebook is closer to 0 than the normal codebook is. Therefore, the ratio of RST/ACK packets to TCP packets can't detect malware correctly. This type of feature is not useful for malware detection.

4.2 Timeslot Length

Table 9 shows the average TNR and TPR in each timeslot throughout three years.

Table 9 Average TNR and TPR in each timeslot

timeslot	0.1 seconds	1 seconds	10 seconds	100 seconds
Average TNR	31.0%	33.6%	36.2%	48.5%
Average TPR	48.2%	56.3%	57.2%	54.5%

Table 10 Average TNR and TPR at each vector quantization level

VQ level	2	4	8	16	32
Average TNR	45.4%	41.6%	40.4%	38.2%	38.0%
Average TPR	52.3%	52.3%	51.4%	51.5%	48.6%

It is obvious that 0.1 seconds is too short a period for detecting malware traffic. Moreover, considering actual circumstances and the need for real time detection of malware, 100 seconds interval would be long. We, hence, suppose that it would be better to use 1 or 10 seconds for extracting features.

4.3 Vector Quantization Level

Table 10 shows the average TNR and TPR at each vector quantization level throughout three years.

In the case of using one feature, level 32 is too high for detecting malware, and level 2 or level 4 is effective in this experiment. We will study a malware detection method combining two or three features in the near future. In such a situation, we think the level 8 or 16 level may be best.

4.4 Effective Features for Malware Detection

The effective features for malware detection are ones with small secular changes and simultaneously high TNR and TPR. Features with either high TNR or high TPR may also be effective. The above analysis shows that the most effective features for malware detection are the minimum packet size(and/or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. In addition, 1 or 10 seconds is a good time interval for extracting these features, and level 2 or 4 is an effective for vector quantization level in the case of using one feature.

5 Conclusion

In this paper, we looked at how well the features used in the previous research can classify normal and malware traffic and discussed which of them are actually effective at malware detection. Our analysis showed that secular changes significantly affect the discrimination rate. We guessed that there are two main reasons for secular changes. First, if there are large differences between each test data, the

discrimination rate dramatically changes. Second, if some test data have a particular behavior, for example, SYN scan, the features in test data dramatically change.

Considering such secular changes, we concluded that four features are especially effective for malware detection, the minimum packet size(or its standard deviation), the number of SYN packets, the ratio of SYN packets to TCP packets, and the number of ACK packets. The best time interval for extracting features is 1 or 10 seconds and 2 or 4 may be the best level of vector quantization in case of using one feature.

In our research, we have three subjects of future work. First, we should discuss how to combine features so as to improve the discrimination rate.

Second, we should discuss what types of traffic data we should use for training data in order to enhance the discrimination rate. We have found that the number of packets and certain behaviors especially affect it. Therefore, we should look at training data that would emphasise these points.

Third, we should look into the capture circumstances of normal traffic. In this experiment, the normal traffic data was captured in an intranet while the anomalous traffic was captured in a honeypot circumstance. Although it is valid to use normal traffic after it has been preprocessed in the above circumstance, the malware circumstance is much different from the normal traffic circumstance. Therefore, it is important to research normal traffic circumstances in order to perfome a more reliable experiment.

References

1. Internetthreatmonthlyreport (May 2011), `http://ip.trendmicro.com/jp/threat/security_news/monthlyreport/article/20110602082147.html`
2. Fujiwara, M., Terada, M., Abe, T., Kikuchi, H.: Study for the classification of malware by infectionactivities. In: IPSJCSEC, vol. 21, pp. 177–182 (March 2008) (in Japanese)
3. Ichino, M., Sakano, H., Komatsu, N.: Speaker Recognition Using Kernel Mutual Sbuspace Method, Thetransactions of the Institute of Electronics. Information and Communication Engineers 88(8), 1331–1338 (2005)
4. Karamcheti, V., Geiger, D., Kedem, Z., Muthukrishan, S.M.: Detecting malicious network traffic usinginverse distributions of packet contents. In: The ACM SIGCOMM Workshop on Mining Network Data, pp. 165–170 (2005)
5. Hatada, M., Nakatsuru, I., Akiyama, M.: Datasets for Anti-Malware Resarrch-MWS2011Datases-, MWS2011 (October 2011) (in Japanese)
6. Sato, Y., Waizumi, Y., Nemoto, Y.: Improving Accuracy of Network-basedanomalous Detection Using Multiple Detection Modules. Technical Commiteeon Network Systems (2004) (in Japanese)
7. Hiramatsu, N., Waizumi, Y., Tsunoda, H., Nemoto, Y.: Using Multiple Normal States for Network Anomaly Detection. In: IEICE (2006) (in Japanese)
8. Kugisaki, Y., Kasahara, Y., Hori, Y., Sakurai, K.: Study for botnet detection based on behavior observation of datatransmission interval. In: SCIS (2009) (in Japanese)