Botnet Traffic Detection Techniques by C&C Session Classification Using SVM

Satoshi Kondo[1] and Naoshi Sato[2]

[1] Trend Micro Incorporated, Shinjuku MAYNDS TOWER 27F 2-1-1 Yoyogi, Shibuya-ku, Tokyo, Japan
satoshi_kondo@trendmicro.co.jp
[2] Institute of Information Security, 2-14-1 Tsuruya-chou, Kanagawa-ku, Yokohama, Japan
sato@iisec.ac.jp

Abstract. Bots, which are new malignant programs are hard to detect by signature based pattern matching techniques.

In this research, we focused on a unique function of the bots the remote control channel (C&C session). We clarified that the C&C session has unique characteristics that come from the behavior of bot programs. Accordingly, we propose an alternative technique to identify computers compromised by the bot program for the classification of the C&C session from the traffic data using a machine learning algorithm support vector machine (SVM). Our evaluation resulted in 95% accuracy in the identification of the C&C session by using SVM. We evaluated that the packet histogram vector of the session is better than the other vector definitions for the classification of the bot C&C session.

1 Introduction

We are facing a new type of malware called bot[1] whose attacks are increasing. The bots spread widely and infect many computers through the Internet. They are designed for distributed attacks such as DDoS and port scanning, for reconnaissance of networks and computers, and as a SPAM mail dispatcher. These bot activities are different from existing malware such as viruses and worms. Their main activities do not focus on attacking the infecting host; instead, the hosts are compromised to use them in a distributed attack platform.

The bot program creates a communication and control channel to the attacker and it is used to conduct over then thousands of the compromised computers by them. This communication and control channel is called the command and control (C&C) channel. The attacker controls bots using this channel to execute distributed attacks and other activities. That is called botnet, and it is causing serious security problems. Until the bot receives a command from the attacker, it stays dormant in the compromised host. This makes bots harder to detect than other viruses and worms.

The other peculiar feature of bots is that they have generated many variants in a short time, like an explosion. Attacker communities share the source code

A. Miyaji, H. Kikuchi, and K. Rannenberg (Eds.): IWSEC 2007, LNCS 4752, pp. 91–104, 2007.

of bots and breed them to be more powerful. It is easy to generate more bots by sharing the source codes. However, the attackers do not have many programming skills, so they find and use the bot program development kit to make their own bots.

Furthermore, the bot program uses the packer techniques to encrypt and archived itself with a self-extraction and decryption function to generate unique binary files from a single source. This combination generates many kinds of unique binary files of the bot program, which are spread throuhgh the Internet.

1.1 Related Works

There are several approaches to detect bots and computers compromised by them. Here, we summarize them and address the related issues.

Signature- and Rule-based Detection. For bot detection, the most popular method is signature-based pattern matching. This approach has been used typically for malware detection. In this approach, it is necessary to prepare signature data for each malware if its binary file is different. Therefore, signature-based pattern matching cannot find the malware program if the signature is not available. As described above, bots are generated with many short-term valiants and each variant is not widely propagated to the Internet. Therefore, it is hard for a security vendor to capture entire bot programs. As a result of not capturing entire bot programs, signature-based pattern matching technique has a low bot detection rate[2].

On the other hand, a different approach based on IDS/IPS such as snort[3] is used to detect the bots. This alternative method uses packet-matching rules to detect those security attacks using the well-known security vulnerabilities. This approach can detect any variant of the bot when it uses the same security vulnerability for infection. This approach depends on the knowledge of the security vulnerability and exploits; therefore, it cannot detect any new security attack, called a zero day attack. We cannot prepare for the zero day attacks until we have analyzed and identified conceptual or real attacks.

Behavior and Activity Detection. James et al.[4] proposed an algorithm to detect bot communication channels on an IRC[5] server using the work weight of the channel members. They defined the work weight as the ratio of the count of TCP control packets to all TCP packets. The TCP control packet is enabled by some TCP flag bit such as RST. If the compromised computer performs a port scan or exhibits some network attacking behavior, the work weight increases. Therefore, the scanning activities by malwares such as bots are identified. Based on this work weight, James et al. monitored an IRC server channel and detected which IRC channel is used by bots to communicate. However, the compromised computer cannot be detected until the bot is activated by the attacker to execute network attacking activities. The infection activities of bots uses not only security vulnerabilities but other methods as well, such as email, P2P file sharing, uploading to web sites and instant messenger services.

Anirudh et al.[6] used the activities to lookup a DNS-based black list (DNSBL) by the botnet owner. One of the purposes of the bots is to dispatch SPAM mail. The DNSBL is commonly used to block the server that relays SPAM. The mail server refers the list to identify the SPAM relay server and rejects mail transfer from it. Once it is listed in the DNSBL, the mail server does not transfer the mail to the other mail relay server and hence blocks incoming SPAM mail from the mail relay. The study found that the bot owners periodically look up the DNSBL to check if their own compromised computer is listed. The botnet owners can confirm, whether the compromised computer still has the capability of sending SPAM mail by checking the DNSBL. This unique activity of checking the DNSBL is useful in identifing the compromised computer which is used for dispatching SPAM mail. However, it could not be applied to detect the compromised computer infected by any other kind of bot.

C&C Session Detection. Christopher[7] report the bots C&C session detection using snort. This approach analysis IRC packet payload and define detection rule for specific bot command strings. It can detect well known bots, but cannot detect any variant of bots that uses different command strings. The attacker easy to decieve these detection rules to modify their own bot programs.

Carl et al.[8] attempted two stage identification approach to detect the bots C&C session using machine learning algorythms . The first stage, they split the IRC session from whole traffic data. The second stage, identify the bots C&C session from classified IRC session data. They achieved the low false-positive rate (7.89%) for the bots IRC session detection using Naïve Bayes. However, they prepared single kind of bot program to collect the bots C&C sessions for training data.

1.2 Motivation

We addressed the issue of bot detection in the previous section. The motivation for this research is to find a generic approach to detect the computer compromised by any kind of bot; one that does not depend on any single instance of the bot program and specific activities. Therefore, we focused on the communication channel used by the attacker to control compromised computers remotely. This is a generic activity of bots. We can identify the computer compromised by a bot from the source IP address of the C&C sessions, if we can identify the C&C sessions from the entier network traffic.

2 Analysis of Bot C&C Session

First, we explain the collection of the malware dataset which we used to learn the characteristics of the bot C&C session traffic. Then, we discuss the characteristics of the bot C&C session learned from our analysis.

2.1 Dataset

We used the collection of malware data set which is captured by our honeypot system. The honeypot system uses nepenthes[9] collect bots from the Internet. It is operated by the bot analysis team of the Institute of Information Security. In this research, we used 2161 unique binary files that were captured by this system. The malware collection was scanned by ClamAV[10] antivirus tool (version 0.88.2 and signature file number 2416) and the result are shown in Table 1.

The antivirus program identified 1473 files as bots and 366 unknown files. ("Unknown" means ClamAV does not have a signature to identify the bot.) Exploit.DCOM.Gen is detected as generic DCOM exploit code in the file. It cannot identify any existing malware, but most of them are a viruses, worms, or new bot programs that are not identified exactly by the signature.

We captured all IP packets during execution of each malware on the sandbox environment for traffic analysis. We executed a malware under Windows XP (with no service pack applied) in VMware[11] for 3 minutes. All packets from/to this sandbox environment were captured and stored as files in tcpdump format.

We found that some of the malwares were corrupted. Some of them tried to connect to an IRC server as the C&C server; they gave the error that the server was shoutdown or not reachable. In other cases, the domain name of the server was deleted from the DNS entry.

Table 2 shows the result of malware execution. By hand-analysis, we identified 957 active bots and 1229 C&C server sessions. Those bots accessed to 97 unique servers. Some of them were running on the same machine. That machine was compromised by attackers and multiple C&C servers were installed and configured. We used these 1229 sessions as bot C&C sessions for analysis of characteristics and used for classification. More details of their analysis and examination of the classification are given in the following section.

Table 1. Contents of Malware Specimen Collection

malware type	sub type	num.	percentage
SDBot	123	598	27.67%
MyBot	197	539	24.94%
PoeBot	19	243	11.24%
IRCBot	18	93	4.30%
Others	84	205	9.49%
Exploit.DCOM.Gen		117	5.41%
Unknown		366	16.94%
total		2161	100.00%

2.2 Analysis of Bot C&C Session Characteristics

We analyzed the captured packet data to identify specific characteristics of C&C sessions. In this analysis, the payload of the packets and protocol header information such as protocol types, source and destination IP addresses, and port

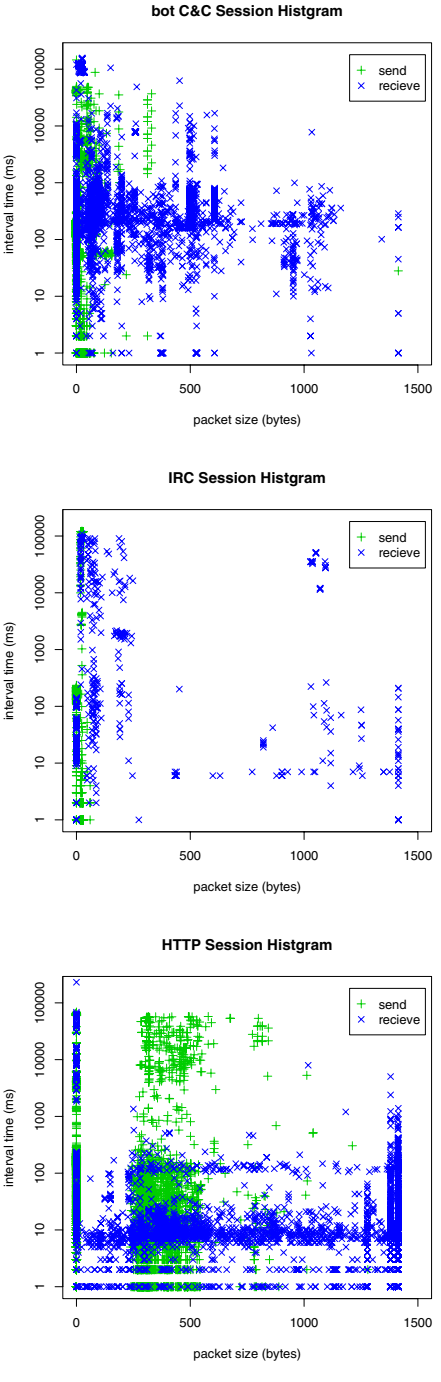

Fig. 1. Packet Histogram of the Session

Table 2. Detection of Active Bots and C&C Servers

Active bot programs (C&C session detected)	957
Unknown (ClamAV not detected as a malware)	67 (7%)
Total C&C Sessions	1229
Unique Servers (pair of IP addresses & port number)	97
Unique IP addresses	71

number was not inspected. However, the protocol identification that checks some unique protocol signature such as command strings is easy to invalidate by encryption or modification of the protocol. In addition, there are privacy issues if we apply this technique to an ISP or some backbone network to check the packet payload data. Therefore, we focused on minimum information such as the packet size and packet interval time.

The traffic management research field contains many reports on the classification of sessions and traffic. Andrew et al.[12] used Bayesian analysis to classify Internet traffic. They categorized applications in several groups by their traffic characteristics. They achieved 95% accuracy to classify the traffic to categorized groups by 248 discriminators per flow data. Laurent et al.[13] proposed a technique that relies on the observation of the first five packets of a TCP connection to identify the application. It has a better detection rate even when using minimal discriminators such as first five packets.

Based on these reports, we used background information such as packet payload size and packet interval time. These discriminators should be as minimal as possible and easy to collect on the network. Our challenge was to find any identical characteristics between the C&C session and the IRC chat session. Both sessions are based on same protocol with minimum differences.

Fig.1 shows the packet histogram plot with respect to the packet payload data size and the interval time from the previous packet.

We can see unique characteristics of the C&C sessions that are compared with HTTP sessions and IRC chat sessions; however, these are derived from simple information. For example, in the Web session plot, the send packet size was plotted around 200 through 500 bytes. It is represented that request command of HTTP protocol from the client. The receive packet size was plotted and had maximum packet payload size near 1500 bytes. Most packet interval times where short.

Even though both sessions were using same protocol, the IRC protocol, we can see some differences between them. This means some information and characteristics do not depend on the protocol but on usage and behavior.

The application behavior and functionality appear in that information.

In contrast, the bot program uses the IRC servers installed into the compromised computers as the C&C servers. Therefore the compromised computer has lower bandwidth and processing performance than typical open relay IRC servers. This also effect the interval time of the packet sequence. The attackers maintain their own IRC servers for bot control to prevent the operator from shuting down the communication channels to block the bot activities. Attackers

need to use some customized version of the IRC server that is optimized for handling to over 10,000 bots as an IRC client. There are many functional trade-offs and requirements to support some specific features of the bot activities. Those characteristics are difficult to modify and mimic for other applications. Some functionality and architecture of the specific activities of the bots appear similar. It is the unique characteristics of the session for bot programs that have similar functionality.

Therefore, we can use this information for the classification and identification of the bot C&C session.

3 Experiment of C&C Session Classification

3.1 Definition of the Feature Vector

To evaluate the effectiveness and accuracy of the session classification using the vector definition as discriminators, we defined three kinds of vectors for session classification (Fig. 2). These are session information vector, packet sequence vector, and packet histogram vector.

Session Information Vector. The vector is defined as total receive packet numbers, total receive packet data size, total send packet numbers, total send packet data size and session time. This session information vector is generated using these five values for each session from the packet capture data file. It is the most simple vector definition.

Packet Sequence Vector. The packet sequence vector consists of the packet size, and packet interval time of the first 16 packets from the session established. It does not include packet payload size of 0 bytes such as control packets like SYN packets.

The difference in protocol appears at the beginning after the session established. For most protocols, the first few packets represent the protocol negotiation stage, therefore there are unique characteristics shown by each protocols.

Packet Histogram Vector. The packet histogram vector is the histogram data by packet payload size and packet interval time in the session. We divided the packet payload size in 16 ranges of 100 bytes each, and the packet interval time in 10 ranges on a log scale. Send and receive packets are separated by count histogram data. The send packets were defined as packets outgoing from the sandbox environment and the receive packets are defined as incoming packets.

This vector has the 320 vector values; 160 vectors for the send packets histogram and the rest for the receive packets histogram.

3.2 Dataset for Experimentation

We prepared 4 datasets for the C&C session classification. We used them as a set to create three experimental vector data.

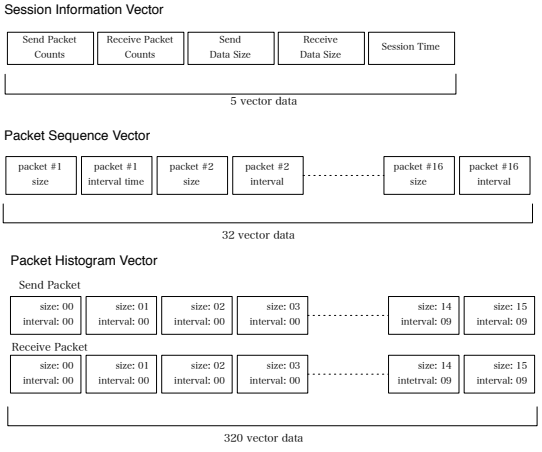

Fig. 2. Feature Vector Definitions

Training Dataset. We used 1029 sessions for the training dataset as the bot C&C sessions. Also, we used 6581 sessions as a background noise which were not identified as the C&C sessions during execution of malware on the sandbox environment. We labeled them as 'bot' and 'other' for training of the classifier.

Testing Dataset. This dataset includes 200 sessions of the C&C session and 800 sessions from background sessions.

IRC Chat Dataset. We prepared a typical IRC chat sessions dataset to verify the classification functionality between a C&C session and a normal IRC chat session.

HTTP Dataset. This dataset includes 132 HTTP sessions which are generated by normal web browsing operation by us within 1 hour. We used this dataset as typical session data of the network traffic.

3.3 Support Vector Machine

The support vector machine (SVM)[14] is known as a better machine learning algorithm for two class discrimination. It shows good accuracy in the classification of voice dictation, image recognition, and other pattern-matching applications. The SVM has better classification functionality and processes the high dimension of the vector data well for training and classification. Therefore, we used the SVM for the C&C session classification. In the experimentats, we evaluated the performance of the SVM and compared it with the other algorithms such as Naïve Bayes and k-Nearest Neighbor (k-NN)[15].

3.4 Experiment Environment

In this experimentats, we used R[16] to verify the classification functionality by SVM. The SVM module is provided as the e1071 package[17] for R; it is based on LIBSVM[18] implementation.

4 Results

In this section, we show the result of the C&C session classification by SVM using each vector definition.

4.1 Session Information Vector

Table 3 shows the result of C&C session classification by the session information vector. It correctry detects 863 sessions as bot C&C session from 1029 sessions in the training data set and detects 162 sessions as bot C&C session from 200 sessions in the testing data set. In other words, the detection rate for C&C sessions of the training dataset is 83.87% and 81.00% for the test dataset. That was a good classification for the bot C&C session using simple vector data to represent the session characteristics. However, the false-positive rate is higher (46.15%) for the IRC chat session. It misclassified the normal IRC chat session as the C&C session. The session information vector data does not reveal the differences between the C&C and normal IRC chat session, because they are using the same protocol.

4.2 Packet Sequence Vector

Table 4 shows the classification result using the packet sequence vector. In this result, all of the C&C sessions in the training dataset were correctly identified as such. However, there is an 82.00% false-negative results for classification of the C&C sessions in the testing dataset. That is, it does not have better classification for the non-training dataset.

4.3 Packet Histogram Vector

Table 5 shows the classification result using the packet histogram vector. The result was better than the other two vector definitions. It classified the C&C session in the training dataset and testing dataset well. The false-positive rate is 0.03% in the training dataset; the other data had no false-positive result. The false-negative rate is 2.62% in the training dataset and 5.00% in the testing dataset.

Comparing with the signature based detection, this has the better false-negative rate result (7% in the Table 2).

Table 3. M_I : The Detection Rate of the Session Data Vector

result	training set		testing set		IRC	HTTP
	bot	other	bot	other		
bot	863	4	162	2	12	1
other	166	6577	38	798	14	131
total sessions	1029	6581	200	800	26	132
detection rate	83.87%		81.00%			
false-positive		0.06%		0.25%	46.15%	0.76%
false-negative	16.13%		19.00%			

Table 4. M_S : The Detection Rate of the Packet Sequence Vector

result	training set		testing set		IRC	HTTP
	bot	other	bot	other		
bot	1029	0	36	0	0	0
other	0	6581	164	800	26	132
total sessions	1029	6581	200	800	26	132
detection rate	100.00%		18.00%			
false-positive		0.00%		0.00%	0.00%	0.00%
false-negative	0.00%		82.00%			

4.4 Comparison of SVM with Other Classification Algorithms

We compared the result of C&C session classification using SVM with other classification algorithms. We used the same dataset and a three feature vector definition data format for Naïve Bayes and k-Nearest Neighbor (k-NN).

Fig. 3 shows that we recognized that SVM is better than the other algorithms for classifying the ession information. SVM provides a better result than the other algorithms for the false-positive rate (Fig. 3). The Naïve Bayes misclassified all sessions as the C&C session in all feature vector data format. The result by k-NN is similar to SVM except for the false-positive rate.

As Table 6 shows, the processing speed of the session classification using SVM was faster than other algorithms. The testing machine was Linux kernel 2.6.15 running on Intel Pentium M 2GHz CPU and 1Gbyte of memory. The classification processing by SVM took less than 1 ms per session. On the other hand, Naïve Bayes and k-NN took more than ten times longer to classify a session with the packet histogram vector. In particular, the processing took a long time at the session sequence vector and the packet histogram vector. In contrast, the feature vector does not significantly affect the processing speed by SVM. The result shows better scalability for high dimension vector data such as the packet histogram vector.

The training processing cost was expensive in the SVM. However, SVM needs to be trained only at the beginning and it can be preprocessed. The total processing of the SVM was faster than other algorithms. k-NN does not need training for the classification; therefore, it uses all training data for the classification to

Table 5. M_H : The Detection Rate of the Packet Histogram Vector

result	training set		testing set		IRC	HTTP
	bot	other	bot	other		
bot	1002	2	190	0	0	0
other	27	6579	10	800	26	132
total sessions	1029	6581	200	800	26	132
detection rate	97.38%		95.00%			
false-positive		0.03%		0.00%	0.00%	0.00%
false-negative	2.62%		5.00%			

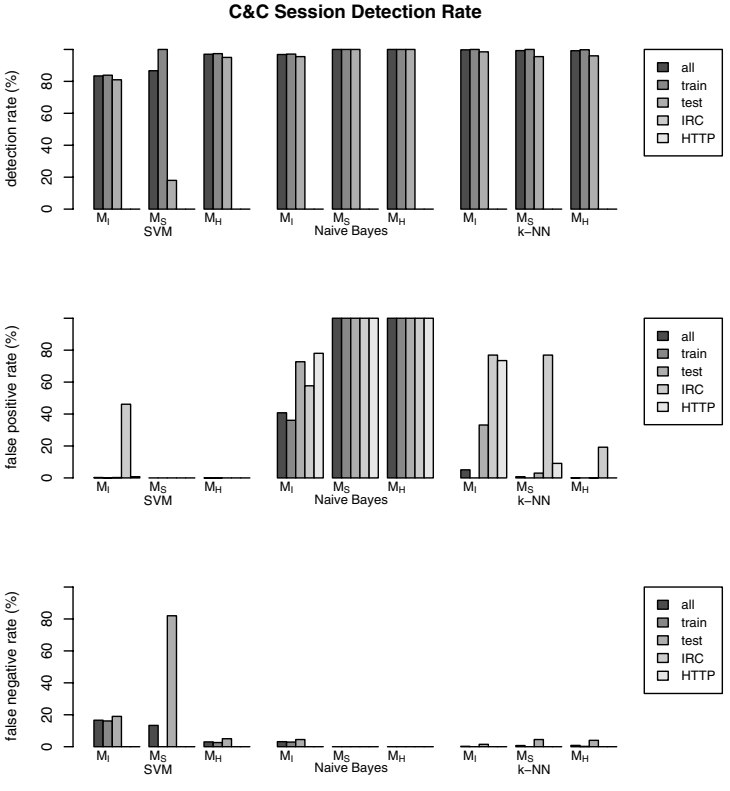

Fig. 3. Comparison of the C&C Session Detection Rate

calculate the distance between the target session vector data and training sessions vector datas for each session. The k-NN needs to optimize the training dataset to minimize the processing cost, but for unknown session data, it is difficult to simultaneously reduce the processing cost and maintain better accuracy of the classification.

Table 6. Processing Time for Training and Classification

Vector Definitions	SVM	Naïve Bayes	k-NN
Session Information Vector			
Total (s)	3.03	6.85	1.62
Training (s)	2.15	0.04	–
Classification (ms/session)	0.10	0.78	0.18
Session Sequence Vector			
Total (s)	47.46	50.66	88.36
Training (s)	38.60	0.72	–
Classification (ms/session)	1.01	5.70	10.08
Packet Histogram Vector			
Total (s)	6.73	159.42	393.91
Training (s)	5.41	1.99	–
Classification (ms/session)	0.15	17.96	44.93

5 Discussion

We discuss here the contribution of this research and the issue, to be verified in future work.

5.1 Training Dataset

We prepared the sample dataset generated from the sandbox environment. It does not include any other sessions belonging to user activities that appear in the real network traffic data. Those session data are effective for C&C session classification accuracy in providing a contrast between the characteristics of C&C sessions and other sessions. We predict that those session data provide better optimization of the support vector after training on SVM. However, it can have a negative effect on detection rate if those sessions have characteristics similar to the C&C session.

5.2 Feature Vector Definition

The detection result was strongly affected by feature vector definition from session information. In particular, it creates false-positive rates for a non-C&C session. This means we have to choose better definitions to represent characteristic differentiation between C&C sessions and other sessions.

In our research, we identified that the packet histogram vector provides a better result than the others. There may be some better vectors for the session classification.

5.3 Deceiving the Bots C&C Session Characteristics

The attacker can modify the C&C session characteristics to deceive our detection technique. For example, bots can emulate regular chat sessions and messages and

mixed control command and messages into them. However, the attacker installs C&C server on the computer compromised by bots to construct the botnets to avoid to identified and shutdown. There are differences network behavior between open IRC server and these installed bots C&C servers for example band width and system performance. It will affect the packet histgram data and still we have a chance to detect them. However, we have to consider these counter approach to compromised our detection technique.

5.4 Application to the Various of C&C Session Protocols

In this research, we targeted the C&C sessions that are based on IRC protocols. In the facts, most of bots use IRC protocols, and some of them use modified protocol commands and encrypt the command and control messages.

Our approach in applicable to these bots, but some bots using other protocols for command and control exist.

These different protocol-based C&C sessions appear to have different traffic characteristics. However, They would have similar behavior that comes from the objectives and motivation of bots and functionality of botnet.

6 Conclusion

We conclude that our alternative approach is valid for detecting the computer compromised by bots. The difference in characteristics of the bot C&C sessions are useful for session classification, even though we used limited information such as the packet size and packet interval time. This difference was related to the functionality of the bot program; therefore it appeared for all bot programs.

We showed that the feature vector definition greatly influenced the identification accuracy, and using the packet histogram among the definitions of three feature vectors, we obtained a 95% detection rate for the non-training dataset. In particular, it had minimum false-positives and false-negatives in classifying C&C sessions and normal IRC chat sessions, which both use the same IRC protocol. Compared with other algorithms, SVM showed better accuracy in the classification of the C&C session and better scalable performance.

However, there are several issues need to be considered in the above. In particular, the counter approach of our proposed detection technique and targeting the non-IRC protocol based C&C sessions are required to concern in the next step from this research.

Acknowledgment

In this research, we used the collection of malware dataset by the bot research team in Tanaka Laboratory in the Institute of Information Security. We express our deep thanks to the team members.

References

1. Barford, P., Yagneswaran, V.: An Inside Look at Botnets. In: Special Workshop on Malware Detection, Advances in Information Security, Springer, Heidelberg (2006)
2. Nepenthes Development Team: `http://nepenthes.mwcollect.org/stats:scannertest` Available from `http://nepenthes.mwcollect.org/stats:scannertest`
3. M. Roesch: Snort: Lightweight intrusion detection for networks. In: 13th Systems Administration Conference (LISA'99), pp. 229–238. USENIX Associations (1999)
4. Binkley, J.R., Singh, S.: An Algorithm for Anomaly-based Botnet Detection. In: Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), pp. 43–48 (July 2006)
5. Oikarinen, J., Reed, D.: Internet Relay Chat Protocol. RFC1459, Internet Engineering Task Force (1993)
6. Ramachandran, A., Feamster, N., Dagon, D.: Revealing Botnet Membership Using DNSBL Counter-Intelligence. In: Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), pp. 49–54 (July 2006)
7. C.W. Hanna: Using Snort to Detect Rogue IRC Bot Programs. Technical report, (October 2004)
8. Livadas, C., Walsh, B., Lapsley, D., Strayer, T.: Using Machine Learning Techniques to identify botnet traffic. In: Proceedings of 2nd IEEE LCN Workshop on Network Security (November 2006)
9. Nepenthes Development Team: Nepenthes - Finest Collection. Available from `http://nepenthes.mwcollect.org/`
10. ClamAV project: ClamAV. Available from `http://www.clamav.net/`
11. VMware Inc.: VMware workstation. Software available at `http://www.vmware.com/`
12. Moore, A.W., Zuev, D.: Internet Traffic Classification using Bayesian Analysis Techniques. In: SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 50–60 (2005)
13. Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: SIGCOMM Comput. Commun. Rev. Number 36(2), 23–26 (2006)
14. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
15. Fix, E., Hodges, J.: Discriminatory analysis: Nonparametric Discrimination: Consistency Properties. Technical report 21-49-004, USAF School of Aviation Medicine (1951)
16. R Development Core Team: R: A Language and Environment for Statistical Computing. (2005) `http://www.R-project.org` ISBN 3-900051-07-0
17. Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A.: A: The e1071 Package (2006), Available at `http://cran.r-project.org/src/contrib/Descriptions/e1071.html`
18. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. Software (2001), available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`