2011 IEEE/IPSJ International Symposium on Applications and the Internet

Detection of Attackers in Services Using Anomalous Host Behavior based on Traffic Flow Statistics

Yukiko Sawaya, Ayumu Kubota, and Yutaka Miyake
KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino City, Saitama Prefecture 356-8502 JAPAN
{yu-sawaya, kubota, miyake}@kddilabs.jp

Abstract— **Flow-based attacker detection is a common way to detect malicious hosts at a router on a high-traffic network with fewer computing resources. The most challenging aspect is to detect attackers that traverse well-known ports such as TCP ports 21, 25, 80, 443, etc. Although various methods have been studied, they cannot accurately detect such attackers. We propose a new flow-based attacker detection method that achieves a high detection rate using traffic flow statistics obtained by NetFlow, sFlow, etc. The proposed method focuses on the characteristics of attackers who send flows to both the object port and generally closed port in the global network. Our method accurately identifies hosts sending flows to object port as attackers, without any deep packet inspection. We evaluated our method using actually collected NetFlow data. The results show that it detects 90.0% of attackers, with few misidentifications of legitimate hosts.**

Keywords - flow-based attacker detection; NetFlow; spam mail sending hosts; DDoS attack; botnet;

I. INTRODUCTION

As the traffic volume from attackers on the Internet rises, it is becoming increasingly important for Internet Service Providers (ISPs) to detect them and shut them off to prevent their customers' computers from coming under attack.

One of the major methods focuses on detecting attackers from the traffic at routers. This attacker detection is divided into two categories: packet-based and flow-based attacker detection.

Packet-based attacker detection analyzes packet contents and it is effective in detecting attackers because malicious hosts include features in the contents of the packets they send. However, all the packets traversing the router have to be exhaustively checked, meaning high computing resources are required for accurate detection when the traffic volume increases in a high traffic network, such as ISP backbone networks.

To overcome this problem, flow-based attacker detection uses the statistics of traffic flow data to detect attackers instead of analyzing the contents of each packet. Some flow-based methods focus on the volume of traffic from specific IP addresses. This is an easy way to detect DoS attackers because traffic statistics from DoS attackers differ significantly from those of legitimate hosts. Conversely, detecting attackers such as DDoS attackers, spam mail sending hosts, or P2P based attackers are more difficult because their flows are hardly distinguished from those of the legitimate hosts. To solve these problems, there are some approaches. For example, the entropy-based DDoS attack detection method can detect the occurrence of a DDoS attack but cannot specify the attackers. Other methods using the supervised machine learning method can detect even spam mail sending hosts which are hard to distinguish from legitimate traffic, but the blacklist and whitelist of hosts must be predefined for the training process and the detection rate is insufficient.

In this paper, we propose a new flow-based attacker detection method which can accurately detect attackers called *"nuisance attackers"*, whose traffic was previously hard to differentiate from those that are legitimate without predefined blacklist and whitelist. One of the key ideas of our method is collecting sample data of the attackers for detection without any predefined blacklist/whitelist by taking advantage of specific behavior of attackers. The behavior we focus on is that some malicious hosts attack not only such ports that are opened in the global network and receive flows from both attackers and legitimate hosts (e.g. TCP ports 25 and 80, defined as *"object ports"*) but also those that are generally closed in the global network (e.g. TCP/UDP ports 135, 139 and 445, defined as *"decoy ports"*) to exploit their vulnerabilities, since the attackers are often controlled by botnets' command and control servers and forced to attempt multiple attacks. In this paper, the attackers that attack both object and decoy ports are called *"obvious attackers"*.

The proposed method first collects and analyzes traffic flows related to the object port from obvious attackers as their flow samples. Next, it determines the attackers targeting the object ports by analyzing the similarity between the features of samples and hosts destined for object ports.

We evaluated the detection accuracy of our method using approximately 60,000 NetFlow records with respect to 9,000 hosts related to TCP port 25 (SMTP) actually collected from the routers in a corporate network. We achieved an attacker detection accuracy of 90.0%, with few misidentifications of legitimate hosts.

II. RELATED WORKS

Packet capturing is an effective means of detecting malicious behavior and suspicious hosts in networks because malicious hosts have common features in terms of the contents of the packets they send [1-3].

978-0-7695-4423-6/11 $26.00 © 2011 IEEE
DOI 10.1109/SAINT.2011.68

Esquivel *et al*. capture packets destined for TCP port 25 at the router and analyze the TCP headers to develop fingerprint signatures based on the specific operating system and version from which the email is sent [1]. These signatures are then used to identify spam mail sending hosts. Although these approaches analyze the packets in detail and can accurately detect attacks, they are impractical for high volume traffic networks such as ISP backbone networks because analyzing all the packets requires a high level of processing resources. To detect the attacks from the huge amount of traffic, flow data analysis is studied [4-7]. Some DoS attack detection methods effectively detect attackers. This is based on the change of traffic volume from a specific IP address range, whereby attacks are identified when the traffic volume exceeds the threshold. The detection of such attackers is easier because traffic statistics of those attackers differ significantly from those of legitimate hosts.

In contrast, detecting attackers such as DDoS attackers, spam mail sending hosts, etc. is more difficult because related traffic flows are hardly distinguishable from those of legitimate hosts. To solve these problems, there are some approaches [8-10]. Wang *et al*. propose a system model with an explicit algorithm to perform on-line traffic analysis [8]. In this scheme, they first make use of degree distributions to effectively profile traffic features, and then use the entropy to determine and report changes of degree distributions, whereby changes of entropy values can accurately differentiate a massive network event, normal or anomalous via an adaptive threshold. They can detect the occurrence of a DDoS attack but attackers cannot be distinguished from legitimate hosts.

Ehrlich *et al*. identify spam mail sending hosts to port 25 via the supervised machine learning method [10]. This method analyzes the flow data statistics of hosts of the training datasets, and makes classifiers for the detection. Although this method can detect attackers from flows destined for object port 25, the attacker detection accuracy is low because it is designed to find the bot network and its controller. Moreover, both the predefined blacklist and whitelist of the hosts are necessary for training.

To solve the problems described above, we propose a new detection method to accurately identify the attackers without requiring high computing resources and a predefined blacklist/whitelist of the hosts.

III. PROPOSED METHOD

To achieve accurate detection, we use traffic flow data such as NetFlow [11] and sFlow [12] retrieved from routers. In our approach, the following information retrieved at the routers is used:

- Source IP addresses
- Destination port
- Protocol type
- TCP flags
- Data size (packets, bytes)

Figure 1 shows an overview of the attackers' model. Let *P* and *Q* be the port number of the object port and the decoy port defined in section I, respectively. There are three types of attackers when we focus on port *P* and *Q*: "*simple attackers*", "*obvious attackers*", and "*nuisance attackers*". The simple attackers send traffic flows to only port *Q*, while the obvious attackers send flows to both port *P* and *Q*. The nuisance attackers do not send flows to the port *Q* but send flows to only port *P*. Simple attackers and obvious attackers defined as "*weird hosts*" are easily detectable because malware-infected hosts, such as Blaster [13], W32.Zotob.E [14], etc., tend to send flows to such closed ports to exploit vulnerabilities but legitimate hosts hardly do so. Conversely, the nuisance attackers is hardly distinguishable from legitimate hosts because they do not send flows to the decoy port. Our method detects such nuisance attackers using similarities with the obvious attackers.

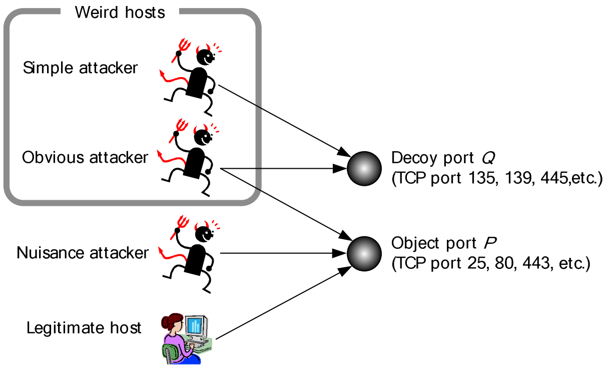


Figure 1. Overview of the attackers' model.

Figure 2 shows the process of our method, which includes three essential steps:

- Step A: Weird host list generation.

We first generate a weird host list (WHL) that consists of the amount of malware infected host IP addresses in order to determine the obvious attackers for an arbitrary object port. Note that the purpose of generating the WHL is not to detect attackers targeting the decoy port, but to find the obvious attackers in order to obtain the flow sample of the attackers' behavior in the next step.

- Step B: The flow data analysis of the obvious attackers.

Based on the WHL, we determine the obvious attacker for the specific object port *P* and calculate the flow statistics for the obvious attackers to obtain a sample of their tendency used for detecting nuisance attackers.

- Step C: Identification of the nuisance attackers.

The nuisance attackers are identified based on the similarity between the features of each of the hosts sending flows to port *P* and the samples.

A detailed algorithm of each step is described in the following subsection.

A. Weird host list generation.

Most TCP ports receive various types of flags including SYN, ACK, PSH, RST, URG, and FIN flags regularly, whereas the decoy ports receive flows containing solely SYN flags more frequently because these ports are usually closed. Based on this behavior, decoy port *Q* is discovered and the WHL is generated as follows:

(1) Enumerate the destination TCP ports and sort them based on the number of flows received. Next, find the port that receive the flows the most and including solely SYN flags, at a percentage exceeding 99% of all flows, indicating almost all of the packets are related to vulnerability scanning. Then define the port as decoy port *Q*.

(2) Extract the flows destined for the decoy port *Q* and collect source IP addresses to generate the WHL.

B. The flow data analysis of the obvious attackers.

The flows destined for the object port *P* from the hosts listed in WHL (i.e. obvious attackers) show certain specific features. We performed a preliminary analysis of the flows destined for port *P* retrieved in a day: the distribution ratio of the packets (*pkt*), bytes per packet (*bpp*), and packets per flow (*ppf*) regarding obvious attackers and all hosts. Figure 3 shows examples of the feature differences between obvious attackers (WHL-listed) and all the hosts (All) retrieved during one day. In this analysis, ports *P* and *Q* are assigned as follows: port *P* = 21 (FTP server), 25 (SMTP server), and 80 (Web server), and *Q* = 445. WHL was determined from past 20 days of flows. Here, α is defined as the number of the obvious attackers to all of the hosts sending flows to port *P*. In Figure 3, certain peaks of distribution differ between obvious attackers and all hosts (pointed by arrow). This result indicates that we can potentially identify nuisance attackers by comparing the features with those obtained from the obvious attackers' flow.

The following two steps show the scheme in detail for analyzing the obvious attackers' features to identify nuisance attackers.

Step 1: The flow data extraction.

The feature vector of host *i* of *N* obvious attackers, $\boldsymbol{f}_i$, is defined as

$$\begin{aligned}
&\boldsymbol{f}_i = (FL_i, PK_i, PYT_i, PPF_i, BPF_i, BPP_i)\\
&FL_i = \log(fl_i)\\
&PK_i = \log(pkt_i)\\
&BYT_i = \log(byt_i)\\
&PPF_i = \log(pkt_i / fl_i)\\
&BPF_i = \log(byt_i / fl_i)\\
&BPP_i = \log(byt_i / pkt_i) \quad ,
\end{aligned} \tag{1}$$

where fl_i, pkt_i, and byt_i are the number of the flows, the number of the packets, and the number of the bytes of host *i* retrieved during certain duration of the time, respectively. The logarithm are taken for each value for the purpose of reducing the order of the values.

After extracting vector $\boldsymbol{f}_i$ for each host *N*, each feature vector element is normalized so that the minimum and

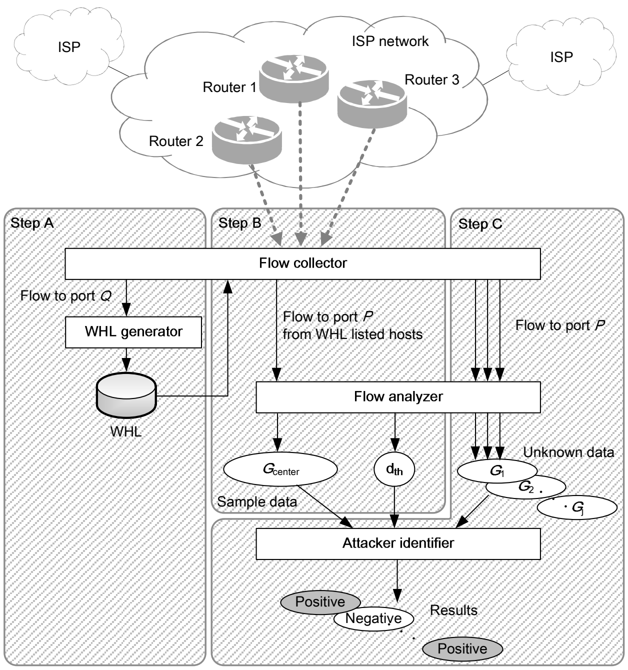
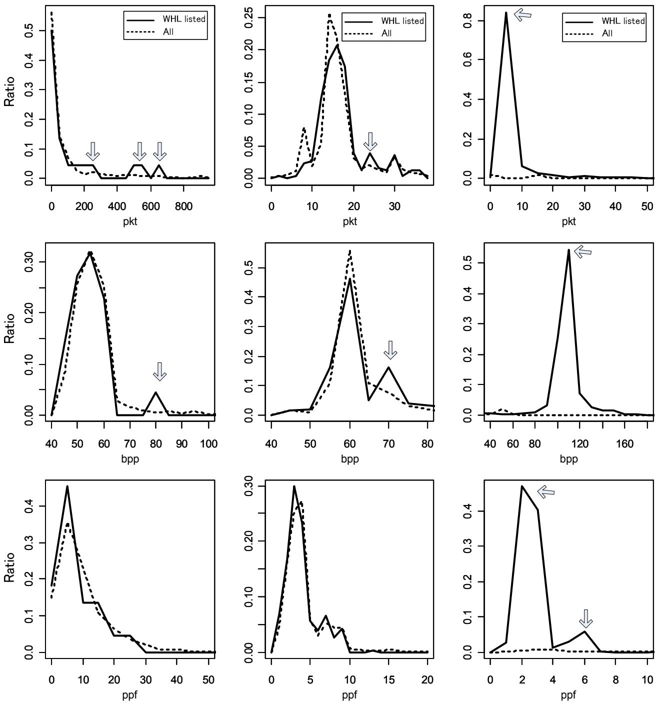


Figure 2. System architecture of the proposed method.



(a) *P*=21
α= 22/1,261

(b) *P*=25
α=305/9,519

(c) *P*=80
α=4,731/342,416

Figure 3. Distribution ratio of *pkt*, *bpp*, and *ppf* of each obvious attackers and all hosts.

maximum values of each element are then converted to 0 and 1, respectively, by a common normalization method to absorb the unit differences among the vector elements. The normalized feature vector $\boldsymbol{f'_i}$ is given in the following form:

$$\boldsymbol{f'_i} = \left(FL'_i, PKT'_i, BYT'_i, PPF'_i, BPF'_i, BPP'_i\right) \tag{2}$$

Step 2: Calculation of the parameters.

There is an optimal combination of feature vector elements that represent the obvious attackers' flow characteristics.

To determine the combination mentioned above, the following schema is applied.

(1) First, select the feature vector element that is the most symbolic feature of the vector. One element whose variance is the minimum is chosen from among six feature vector elements and denoted by x to represent the common denominator of attackers.

(2) Next, find the feature vector element that is not highly correlated to x. n ($0 \le n \le 5$) elements are chosen from the feature element set $Y = \{FL', PKT', BYT', PPF', BPF', BPP'\}$ as C_y based on the correlation coefficients as follows.

$$C_y = \{y \in Y \mid \mathrm{r}(x, y) \le \mathrm{R} \wedge y \ne x\} \ , \tag{3}$$

where r(x,y) is correlation coefficient between x and y, and R is a threshold value. We use R=0.3 according to [15].

Subsequently, we define feature vector $\boldsymbol{G}$ consist of feature element selected in C_y as follows:

$$\boldsymbol{G} = (x,\ C_y) = (x,\ y_1, \ldots, y_n) \quad (0 \le n \le 5) \tag{4}$$

The representative feature vector $\boldsymbol{G_{center}}$ is created from N obvious attackers' feature vectors as follows.

$$\boldsymbol{G_{center}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{G_i} = \left(\frac{1}{N}\sum_{i=1}^{N}x_i, \frac{1}{N}\sum_{i=1}^{N}y_{1i}, \ldots, \frac{1}{N}\sum_{i=1}^{N}y_{ni}\right) \tag{5}$$

Next, the parameter for detecting nuisance attackers is defined by the Euclidian distance between $\boldsymbol{G_{center}}$ and the feature vector of each obvious attacker. The distance d_i beteen $\boldsymbol{G_{center}}$ and the feature vector of host i, $\boldsymbol{G_i}$, is given by the following:

$$\begin{aligned} d_i &= \left|\boldsymbol{G}_i - \boldsymbol{G}_{center}\right| \\ &= \sqrt{\left(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i\right)^2 + \left(y_{1i} - \frac{1}{N}\sum_{i=1}^{N}y_{1i}\right)^2 + \ldots + \left(y_{ni} - \frac{1}{N}\sum_{i=1}^{N}y_{ni}\right)^2} \end{aligned} \tag{6}$$

The nuisance attacker is detected in step C based on whether distance d is smaller than specific threshold $\mathrm{d_e}$ or not. The threshold $\mathrm{d_e}$ is determined so that the following equation is satisfied.

$$\frac{\sum_{i=1}^{N}\begin{Bmatrix} 1, (\text{if} \quad d_i < \mathrm{d_e}) \\ 0, (\text{Otherwise}) \end{Bmatrix}}{N} = \mathrm{e} \tag{7}$$

The value e means the ratio of the number of feature vectors within a hypersphere whose radius is $\mathrm{d_e}$ to N. e is a constant value which depends on the environment or condition of network.

C. Identification of the nuisance attackers.

The host j is identified as the nuisance attacker by calculating the vector $\boldsymbol{G}$ according to formula (4) and the distance d_j calculated by formula (6) in the previous subsection. The nuisance attacker detection rule is defined as the following:

$$\text{result } (j) = \begin{cases} \text{positive} & (d_j \le \mathrm{d_e}) \\ \text{negative} & (\text{Otherwise}) \end{cases} \tag{8}$$

IV. Evaluation

We evaluated each step of our method using the NetFlow data collected in two routers of a corporate network. Via these routers, we could collect approximately 20,000,000 flow records on a weekday and 13,000,000 flow records on a holiday on average.

In this evaluation, we defined the object port as P = 25. This port is useful to evaluate whether the detection using our method is correct or incorrect because information on actual attackers (i.e. spam mail sending hosts) is available as DNS Black Lists (DNSBLs) [16-23].

A. Result of WHL generation

Figure 4 shows the top five ports that received flows the most during one day (on 25 July, 2010). On this day, port 80 received 2,853,257 flows. However, only 5.2% of packets contained the SYN flag solely. Conversely, port 445, ranked second, received 320,163 flows and 99.8% of them contained the SYN flag solely.

According to this result, we defined the decoy port Q = 445. We collected 20 days (6 July, 2010 - 25 July, 2010) of unique IP addresses in order to generate WHL. As a result, a WHL comprising 1,524,573 IP addresses was compiled during the period.

B. Result of flow analysis of obvious attackerts

We evaluated whether we could detect nuisance attackers from the traffic destined for specific object port P.

We used a dataset of the flows retrieved in one day (on 25 July, 2010) for this evaluation. On this day, we obtained 9,519 hosts sending flows to port P, 305 of which were listed in the WHL (α=305/9,519). The remainder of the 9,414 hosts were either nuisance attackers or legitimate hosts.

First, we analyzed the flow data from N = 305 obvious attackers. As a result, variances of the vector element *FL', PKT', BYT', PPF', BPF',* and *BPP'*, were 0.027, 0.010, 0.012, 0.030, 0.016, and 0.011, respectively. From this result, *PKT'* was selected as x. The correlation coefficients between x and *FL', PKT', BYT', PPF', BPF',* and *BPP',* were 0.67, 1.00, 0.92, 0.10, 0.14, and 0.13, respectively. Therefore, *PPF'*, *BPF'*, and *BPP'* were set as y_1, y_2, and y_3, respectively. From these selections of the feature vector elements, we made the feature vector $\boldsymbol{G}$ and $\boldsymbol{G}_{center}$ as follows:

$$\boldsymbol{G} = \left(PKT', PPF', BPF', BPP'\right) \tag{9}$$

$$\boldsymbol{G}_{center} = \left(\frac{1}{N}\sum_{i=1}^{N} PKT'_i, \frac{1}{N}\sum_{i=1}^{N} PPF'_i, \frac{1}{N}\sum_{i=1}^{N} BPF'_i, \frac{1}{N}\sum_{i=1}^{N} BPP'_i\right) \tag{10}$$

Next we determined the threshold distance d_e for detecting the nuisance attackers. We calculated multiple d_es obtained from the values e = 0.10, 0.20, 0.25, 0.50, 0.90, 0.95 and 1.00 as follows:

(e, d_e) = (0.10, 0.058), (0.20, 0.079), (0.25, 0.101), (0.50, 0.162), (0.90, 0.397), (0.95, 0.495), (1.00, 1.000)

C. *Result of detecting nuisance attackers*

We calculated $\boldsymbol{G_j}$ and d_j ($1 \le j \le 9{,}414$) for each host.

Here, we referred to eight DNSBLs [16-23] to verify the detection accuracy of the proposed method. We consider the hosts listed in each of the eight DNSBLs to be nuisance attackers. We also used the server logs of one of the SMTP servers whose flow was traced in the router to ensure the nuisance attackers and legitimate hosts and obtain the actual attacker's information.

Figure 5 shows the distribution of d_j regarding nuisance attackers and legitimate hosts. The smaller the distance, the more nuisance attackers are included, while the legitimate hosts' distribution is concentrated at a higher distance.

Figure 6 shows the detection accuracy of the nuisance attackers and legitimate hosts for each value e. The following factors were used to evaluate the accuracy:

- True positives: the nuisance attackers correctly detected.
- False positives: the hosts incorrectly detected as nuisance attackers.

At the condition e = 0.20, 28.8% (2,580 hosts) of all nuisance attackers were correctly detected without any misidentification of legitimate hosts. 90.0% (7,961 hosts) of all nuisance attackers were detected with 6.6% (24 hosts) of legitimate hosts misidentified as nuisance attackers at the condition e = 0.90.

V. Discussion

A. *Effectiveness of the WHL*

In the previous section, we described how we could generate 1.5 million-WHL in 20 days by collecting flows destined for port 445.

Port 445 is a well-known port that is allocated to the windows server named for direct hosting SMB services, and is not commonly used in the global network. However, malware-infected hosts try to find vulnerabilities to that port. As we successfully found a decoy port and collected hosts sending flows to the port, we could generate the WHL, including a huge amount of hosts.

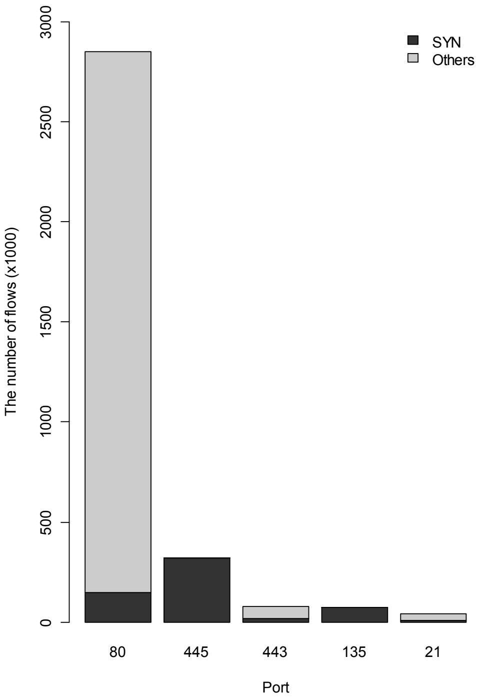


Figure 4. The number of received flows of top five ports.

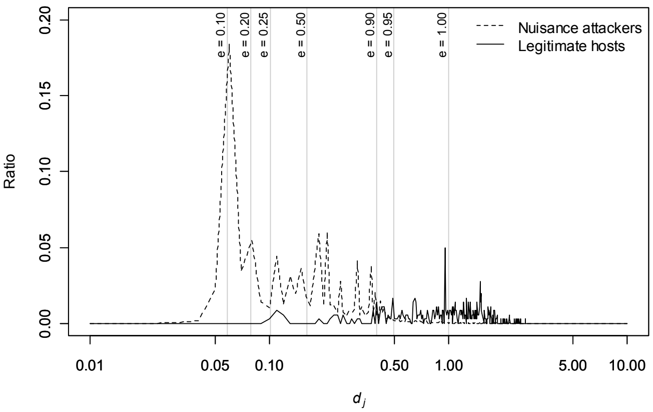


Figure 5. Distribution ratio as a function of the distance d_j.

All N = 305 obvious attackers of port P were listed in at least one of the DNSBLs. This result means that WHL is effective way to collect attackers' samples.

B. *Detection accuracy*

We compared the detection accuracy with the previous method [10]. Their detection rate scored a maximum of

88.7%, with false positive rate of over 13.6%, despite using such complicated method as supervised machine learning method. Conversely, our method detected 90.0% of nuisance attackers with 6.6 % false positives.

We also compared the accuracy of our technique and packet based technique [1], which can use more rich information. Their technique can detect approximately 90% of spam messages with false positive rates of 2 - 8 %.

Although the conditions and dataset differ from those adopted in their methods, we reduced the number of false positives by more than 1/2 with higher true positives compared with the previous flow based technique in [10] and showed detection accuracy at similar value with the packet based detection in [1].

With this highly accurate method, we consider that our method is promising the solution to predict the time-series behavior of the malicious hosts and predict whether the attacks will increase or decrease based on past behavior.

C. Application for blacklist generation

Our method correctly detected 2,580 nuisance attackers without any false positives using just N = 305 obvious attackers listed in WHL. We evaluated WHL with the other object ports, port 21 and 80. As a result, we obtained N = 22 from 1,261 hosts and N = 4,731 from 342,416 hosts, respectively. Therefore, we expect that our method can detect nuisance attackers without any false positives even for other object ports by using WHL.

Moreover, evaluation of the nuisance attackers presented in this paper is given by using the data collected in one day, thus if we ran the proposed method for longer, we would be able to collect even more nuisance attackers without any false positives.

Today, the DNSBLs are available as a blacklist for port 25, but there are few blacklist for the other ports. We believe that our method can be applied to create a blacklist automatically for arbitrary ports other than port 25.

D. The effectiveness of finding the combination of the feature vector element

According to the study [24], the content length of spam mail messages and the frequency of sending email were similar among the botnet-controlled attackers. From this study, we consider that the sizes of the packets contained in the flows from attackers in one day, i.e. the feature vector element *PKT'*, were similar. In our evaluation, as *PKT'* was selected as the most representative feature vector x, the result of our method was quite reasonable.

Here, we assumed feature vectors $\boldsymbol{f'}$ to $\boldsymbol{G}$, which means that all the feature vector elements in $\boldsymbol{f'}$ were used to represent feature vector $\boldsymbol{G}$, to verify whether we could select the optimal combination of the feature vector element. As a result, the detection rate was 90.0% at the condition e = 0.90, while false positive rate was 7.1%. The accuracy of the proposed method with the determination of the combination element in the feature vector was the same as that without such determination with less false positives. Moreover, in general, the fewer the feature vector elements, the fewer resources is required for the nuisance attacker detection process. Our method is able to select the feature vector elements effectively.

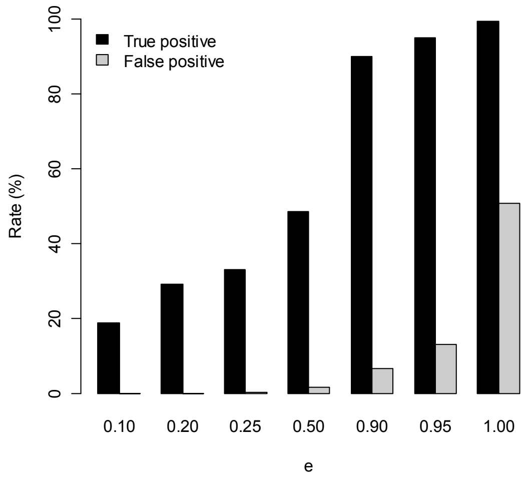


Figure 6. Detection accuracy of the proposed method.

VI. Conclusion

In this paper, we proposed a new flow-based attacker detection method using only limited statistics of packets instead of deep packet inspection.

We could detect such nuisance attackers that previous methods have difficulty to detect because their flows are hardly distinguished from those of the legitimate hosts.

With this method, we have a possibility to detect attackers in a high traffic network, such as ISP backbone networks without high computing resources.

Further experiments and evaluations regarding various kinds of object and decoy ports, and comparisons with other existing methods are future work.

References

[1] H. Esquivel, T. Mori, A. Akella, “Router Level Spam Filtering Using TCP Fingerprints: Architecture and Measurement Based Evaluation,” in Proc. of Sixth Conference on Email and Anti-Spam, Mountain View, CA, USA

[2] J. J. Flores, A. Antolino, J. M. Garcia, “Evolving HMMs For Network Anomaly Detection – Learning Throug Evolutionary Computation,” in Proc. of Sixth International Conference on Networking and Services, 2010, pp. 271 – 276.

[3] T. Limmer, F. Dressler, “Dialog-based Payload Aggregation for Intrusion Detection,” *Proceedings of the 17th ACM conference on Computer and communications security,* Chicago, Illinois, USA, *Oct.* 2010.

[4] G. Miinz, G. Carle, “Real-time Analysis of Flow Data for Network Attack Detection,” in Proc. of 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 100 – 108.

[5] Lakhina, M. Crovella, and C. Diot, “Characterization of Network-Wide Anomalies in Traffic Flows,” in Proc. of 4th ACM SIGCOMM Conference on Internet Measurement. Taormina, Sicily, Italy: ACMPress, Oct. 2004, pp. 201-206.

[6] Lakhina, M. Crovella, and C. Diot, “Mining Anomalies Using Traffic Feature Distributions,” in Proc. of ACM SIGCOMM Conference, Philadelphia, PA, USA, Aug. 2005, pp. 217-228.

[7] J. Terrell, K. Jeffay, F. D. Smith, L. Zhang, H. Shen, Z. Zhu, and A. Nobel, “Multivariate SVD Analyses For Network Anomaly Detection,” in Proc. of ACM SIGCOMM Conference, Poster Session, Philadelphia, PA,USA, Aug. 2005.

[8] W. Wang, W. Wu, “Online Detection of Network Traffic Anomalies Using Degree Distributions,” Int. J. Communications, Network and System Sciences, 2010, Vol. 3, pp. 177-182.

[9] L. Feinstein, D. Schnackenberg, R. Balupari, D. Kindred, “Statistical Approaches to DDoS Attack Detection and Response,” Proceedings of the DARPA Information Survivability Conference and Exposition, Washington, DC, USA, 2003, vol. 1, pp. 303.

[10] W. K. Ehrlich, “Detection of Spam Hosts and Spam Bots Using Network Flow Traffic Modeling,” in Proc. of 3rd USENIX workshop on Large-Scale Exploits and Emergent Threats, San Jose, 2010.

[11] “RFC3954: Cisco Systems NetFlow Services Export Version 9,” http://tools.ietf.org/html/rfc3954

[12] “RFC3176: InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks,” http://tools.ietf.org/html/rfc3176

[13] “Blaster Worm TFTP Backdoor (UDP): Attack Signature - Symantec Corp.,”http://www.symantec.com/business/security_response/attacksignatures/detail.jsp?asid=20098

[14] “W32.Zotob.E | Symantec,”
http://www.symantec.com/security_response/writeup.jsp?docid=2005-081615-4443-99

[15] “Correlation,”
http://www.sjsu.edu/faculty/gerstman/StatPrimer/correlation.pdf

[16] “UCEPROTECT®-Network - Germanys first Spam protection database,” http://www.uceprotect.net/en/index.php

[17] “The Spamhaus Project,” http://www.spamhaus.org/

[18] “BarracudaCentral.org - Technical Insight for Security Pros,
”http://www.barracudacentral.org/

[19] “DNSBL, RBL, SURBL, DNSWL Services, FREE Real Time Black & White List Services, RBL, DNSBL, SURBL, DNSWL, Spam Protection, Junk Email Filter, nsZones.com,” http://nszones.com/

[20] “Passive Spam Block List,” http://psbl.surriel.com/

[21] “no-more-funn: dr. Joslash;rgen Mash's dnsbl,”
http://www.moensted.dk/spam/no-more-funn/

[22] “SORBS (Spam and Open-Relay Blocking System),” http://www.au.sorbs.net/

[23] “SpamCannibal,” http://www.spamcannibal.org/cannibal.cgi

[24] H. Husna, S. Phithakkitnukoon, S. Palla, R. Dantu, “Behavior analysis of spam botnets,” in Proc. of the third International Conference on communication system software and middleware, 2008, Bangalore, India, pp. 246-253