IEICE TRANS. INF. & SYST., VOL.E103–D, NO.12 DECEMBER 2020

PAPER

A Lightweight Detection Using Bloom Filter against Flooding DDoS Attack

Sanghun CHOI[†a)], **Yichen AN**[†b)], ***Student Members***, ***and*** **Iwao SASASE**[†c)], ***Fellow***

SUMMARY The flooding DDoS attack is a serious problem these days. In order to detect the flooding DDoS attack, the survival approaches and the mitigation approaches have been investigated. Since the survival approach occurs the burden on the victims, the mitigation approach is mainly studied. As for the mitigation approaches, to detect the flooding DDoS attack, the conventional schemes using the bloom filter, machine learning, and pattern analyzation have been investigated. However, those schemes are not effective to ensure the high accuracy (ACC), the high true positive rate (TPR), and the low false positive rate (FPR). In addition, the data size and calculation time are high. Moreover, the performance is not effective from the fluctuant attack packet per second (pps). In order to effectively detect the flooding DDoS attack, we propose the lightweight detection using bloom filter against flooding DDoS attack. To detect the flooding DDoS attack and ensure the high accuracy, the high true positive rate, and the low false positive rate, the dec-all (decrement-all) operation and the checkpoint are flexibly changed from the fluctuant pps in the bloom filter. Since we only consider the IP address, all kinds of flooding attacks can be detected without the blacklist and whitelist. Moreover, there is no complexity to recognize the attack. By the computer simulation with the datasets, we show our scheme achieves an accuracy of 97.5%. True positive rate and false positive rate show 97.8% and 6.3%, respectively. The data size for processing is much small as 280bytes. Furthermore, our scheme can detect the flooding DDoS attack at once in 11.1sec calculation time.

key words: *DDoS attack, flooding attack, bloom filter*

1. Introduction

The DDoS (Distributed Denial of Service) attack [1] is a major threat to Internet services. The DDoS attack is a type of DoS attack [2] by the multiple distributed compromised machines which are called bots [3]. Recently, many Internet services have hugely damaged by the DDoS attack. For example, the US web-based hosting service GitHub was damaged by the DDoS attack [4]. In addition, the attacker demanded bitcoin to the private enterprise services such as food delivery to stop attack [5]. According to Akamai's state of Internet [6], on April 22, 2019, the DDoS attack occurred against the Internet service of the established banks. The attack reached a peak of 160 Gbps and 32 million packet per second (pps). Therefore, the DDoS attack is a serious problem for Internet service these days. The DDoS attack can be classified [7], [8] into three types: brute-force attack [9], spoofing attacks [10], flooding attacks [11]. In those types, the flooding attack is mainly used for the DDoS attack [12]. The flooding attacks focus on disrupting the connectivity of victims, which are the legitimate user's and the server, by exhausting victim network's bandwidth, consuming excess amounts of the victim's resources as TCP/SYN flood, ICMP flood, etc. [13]. To protect the user and the server from the flooding attack, two types of approaches are studied as shown in Fig. 1: survival approaches, mitigation approaches. In the survival approaches as shown in Fig. 1 (a), the victims need to detect the flooding DDoS attack by the self-defense program [14] and firewall [15]. Since the survival approaches are focused to protect the single targeted victim, it is not suitable to protect multi-targeted victims from the flooding DDoS attack. Moreover, since the victims need to prepare its resource at their devices to defense from the flooding DDoS attack by victim self, it would be a burden to victims. As shown in Fig. 1 (b), the mitigation approaches are studied to detect the flooding DDoS attack in the router and networks between the attacker and victims before the flooding DDoS attack reaches to victims. Since the attack packet could be blocked by analyzing the attack features before it reaches to victim, the mitigation approaches are suitable for multi-targeted victims. Moreover, because the burden for victims is lower than the survival approaches, the mitigation approaches are mainly investigated. To detect the flooding DDoS attack as mitigation approaches, Aborujilah et al. find the attack feature by arranging attack packet into the covariance matrix [16]. Kalkan et al. try to classify the entropy of traffic to get attack feature [17]. Xiaoyong et al. proposed the machine learning to learn pat-

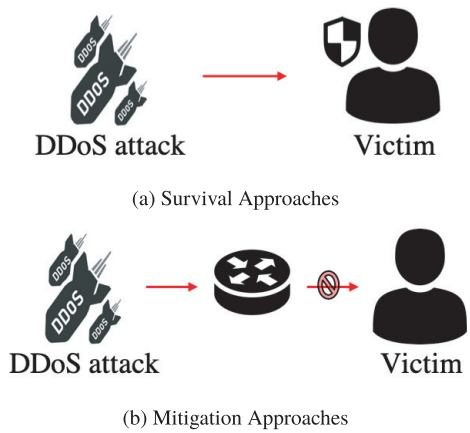


Fig. 1 Structure of the DDoS attack detection approaches

Manuscript received May 26, 2020.
Manuscript revised August 11, 2020.
Manuscript publicized September 14, 2020.

[†]The authors are with Dept. of Information and Computer Science, Keio University, Yokohama-shi, 223–8522 Japan.
a) E-mail: choi@sasase.ics.keio.ac.jp
b) E-mail: anyichen@sasase.ics.keio.ac.jp
c) E-mail: sasase@ics.keio.ac.jp

DOI: 10.1587/transinf.2020EDP7115

terns from sequences of network traffic and trace network attack activities [18]. In addition, the methods which are the getting packet feature from transmission between the attack and bots, are also investigated [19], [20]. However, those methods have high complexity and high calculation time to get attack feature. Moreover, the large memory space for analyzing attack features is needed to execute processes. In order to defect the flooding DDoS attack at once with lightweight space for process, a bloom filter [21], [22] is utilized for the detection of the flooding DDoS attack [23]–[25]. The flooding DDoS attack is detected by recording the IP address into the bloom filter. The highly inputted IP address into the bloom filter is blocked as the attacker's IP address. However, those methods occur a high false positive rate since the performance is not effective from the fluctuant pps.

We assume that if the problems mentioned above can be overcome, it will be suitable for detecting the flooding DDoS attack. Therefore, we propose the lightweight detection using bloom filter against flooding DDoS attack. In order to check the attack feature at once with low complexity, low calculation, and small memory space for analyzing the feature, we use the bloom filter. To reduce the false positive rate from bloom filter, the packet will be flexibly deleted, and the checkpoint to detect attack IP address is also flexibly changed by ramdoly selected to avoid attacker's expectation as the fluctuant pps. Moreover, since there are no blacklist and whitelist in the proposed scheme, we can save more memory space for analyzing features. The evaluations show that the lower calculation time and smaller memory space to detect the flooding DDoS attack in the proposed scheme are smaller than the conventional schemes. Furthermore, the proposed scheme shows better performance for the rates of detection of the attacker and false positive. The rest of this paper is constructed as follows. The related work and attack model are explained in Sect. 2. The proposed scheme is described in Sect. 3. The results and evaluations are shown in Sect. 4. The discussion and limitation are described in Sect. 5. Finally, we conclude our research in Sect. 6.

2. Related Work and Attack Model

2.1 Survival Approaches

Survival approaches enlarge the resources of the single-targeted victim which is the user or server during the flooding DDoS attack. This means that the victim needs to detect the flooding DDoS attack by the victim self. He et al. proposed the machine learning from the victim side to defect the flooding DDoS attack by learning the features of attacks [14]. However, this method needs many times to learn the attack features. Moreover, it cannot detect the unlearned flooding DDoS attack at once. Jeyanth et al. suggested a virtual firewall by continuously monitoring the incoming packets [15]. Since the method requires Ram and Storage to analyze the packet, it would be a burden for the victim's side. Those survival approaches focus on a single-targeted victim. In fact, the resourceful side can win in an arms race between the attacker and the victim. The attacker can usually win from the arms race. This is because the attacker can easily gain more resources by recruiting more agents such as bots. This is the reason that the survival approaches are not suitable to detect the flooding DDoS attack.

2.2 Mitigation Approaches

Mitigation approaches focus on how to detect the flooding DDoS before the attack reaches the victim. Since the flooding DDoS attack would be blocked from in the mid-network between the attack and victim, the multi-targeted victims could be protected. Aborujilah et al. [16] proposed the covariance matrix to the flooding DDoS attack for the cloud server. The covariance matrixes of normal and attack traffic are calculated. The covariance matrixes of network traffic are matched with the expected matrixes. Although the covariance matrix could detect the flooding DDoS attack, there is a high calculation time via the analyzing the attack traffic by the complexities of the matrix. Kalkan et al. [17] suggested an entropy-based detection by using destination IP address entropy and TCP layer attributes to detect the flooding DDoS attacks. This method combines nominal, preparatory, and active mitigation stages. However, since the detection process is performed using a single topology, it cannot detect the new attack traffic at once. Moreover, there is high calculation time to get attack feature. Yuan et al. [18] focused on how to detect the flooding DDoS attack by learning the attack feature by composing fully connected layers. They extracted the attack features by a sliding time window for analyzing packets. However, this method's performance highly depends on the training of the model. In addition, the large space of memory is needed for processing. Yuan-H et al. [19]. and Yichen et al. [20] studied to get the attack features between the attacker and bots before the victim is attacked by the flooding DDoS attack. They analyzed the packet patterns between the attacker and bots from the Internet provider. By calculating the ranks of the normalized attack feature, the attacker's bots can be detected. However, those methods take a lot of time to analyze the attack feature and gather packets between the attacker and bots. Moreover, the large space of memory is also needed for processing. To reduce the calculation time and the space of memory to detect the flooding DDoS attack, fthe detection methods using the bloom filter [21], [22] have been investigated [23]–[25]. The bloom filter was first designed by Burton Howard Bloom in 1970 [21] and has since then become a standard and widely used algorithm for a quick set with large sets and/or low memory conditions [22]. Since the bloom filter can reduce disk access frequency, it has been successfully used in various network-related tasks. Song et al. [23] proposed the detection method which is based on the source IP into the bloom filter. Basically it can detect all kinds of flooding DDoS attacks. However, since it is not effective from the fluctuant pps, there is a high false positive rate. This is because the legitimate user's packets are

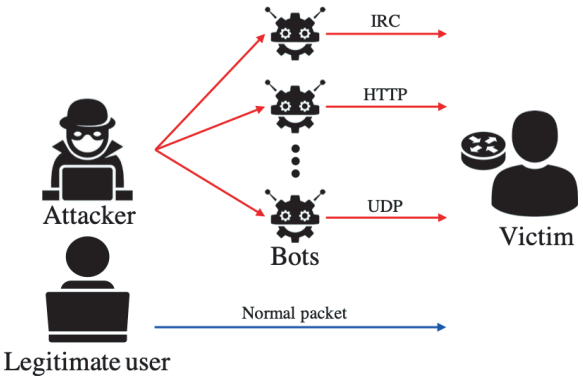


Fig. 2 Attack Model

incorrectly recognized as the attacker's packet. Although Roh et al. [24] proposed the whitelist for the bloom filter and Bogdanoski et al. [25] suggested the blacklist for bloom filter to reduce the false positive rate, the many legitimate users are also blocked as the attacker. There are two reasons why legitimate users are detected as the attacker. As for the first reason, the checkpoint to detect the attacker's IP address in the Bloom filter is fixed. Since the checkpoint is fixed and it only considers the pps to detect, the checkpoint detects the legitimate users as the attacker when legitimate users flow into the checkpoint like the attacker's pps. This is because the legitimate user's IP address reaches to the checkpoint before the attacker's IP address is detected. As for the second reason, although the blacklist and white list are suggested to reduce the false-positive, the legitimate users are still detected as the attacker. If the new legitimate users have the pps like the attacker, the new legitimate users are recorded as the attacker in the blacklist when they reach the checkpoint. So, they will be blocked. In the whitelist, only the legitimate users who do not reach the checkpoint are recorded. So, the new legitimate users who have pps like the attacker cannot be recorded as the normal legitimate user in the white list. Therefore, the conventional schemes using the bloom filter with the blacklist and whitelist are not effective for guaranteeing the low false positive.

2.3 Attack Model

We assume that the attacker tries to run out the victim's network resources. As shown in Fig. 2, to cause the large amounts of traffic on the victim side, the flood DDoS attacks by using the botnet including IRC (Internet Relay Chat), HTTP (Hyper Text Transfer Protocol), P2P (Peer to Peer), SMTP (Simple Message Transfer Protocol), and UDP (User Datagram Protocol), etc., are used. The botnet's packets flow into the router with the legitimate user's packets to attack the victim.

3. Proposed Scheme

In order to detect the flooding DDoS attack at once with the low calculation time, the less false positive rate, and the lightweight memory space, we propose the lightweight detection using bloom filter against flooding DDoS attack. The main idea of our scheme is that the checkpoint for detecting the flooding DDoS attack and the dec-all (decrement-all) operation for reducing the false positive rate in the bloom filter, are flexibly changed as the fluctuant pps. Although there is no complexity in the proposed scheme, the flooding DDoS is detected and blocked. Moreover, there are no blacklist and whitelist. This is a new approach for the detection of the flooding DDoS attack scheme using the bloom filter. The flooding DDoS attack can be detected before the victim is attacked as the mitigation approach. This section is classified into four categories as 3.1 Way to cope with the fluctuant pps, 3.2. Separation of IP address into the bloom filter, 3.3. Counting IP address on the bloom filter, and 3.4. Detection of the attack IP address as the fluctuant pps.

3.1 Way to Cope with the Fluctuant Pps

Here, we describe the way to cope with the fluctuant pps in the proposed scheme. In the conventional schemes using the bloom filter [23]–[25], the performance for detecting the flooding attack is not effective when the fluctuant pps has flowed. Only the higher pss attack IP address than the legitimate IP address can be detected. This means that the low pps attack IP address is hard to be detected. In addition, the high pps legitimate IP address is misinterpreted as the attack IP address since the IP address, which is reached the fixed checkpoint while increasing its values in the bloom filter, is recognized as the attack. Here, to avoid this problem, we suggest the changing value deletion and the changing checkpoint by computing for the bloom filter from the fluctuant pps.

$$R_b \begin{cases} G(10,20) & \text{if } 1 \le P \le 30 \\ G(20,30) & \text{if } 30 \le P \le 60 \\ G(30,40) & \text{if } 60 \le P \le 90 \\ G(40,50) & \text{if } 90 \le P \le 120 \\ G(50,60) & \text{if } 120 \le P \le 130 \\ G(60,70) & \text{if } 150 \le P \end{cases}, \tag{1}$$

$$R_t \begin{cases} G(60,70) & \text{if } 1 \le P \le 30 \\ G(50,60) & \text{if } 30 \le P \le 60 \\ G(40,50) & \text{if } 60 \le P \le 90 \\ G(30,40) & \text{if } 90 \le P \le 120 \\ G(20,30) & \text{if } 120 \le P \le 130 \\ G(10,20) & \text{if } 150 \le P \end{cases}, \tag{2}$$

$$R_c \begin{cases} G(5,10) & \text{if } 1 \le P \le 30 \\ G(10,15) & \text{if } 30 \le P \le 60 \\ G(15,20) & \text{if } 60 \le P \le 90 \\ G(20,25) & \text{if } 90 \le P \le 120 \\ G(25,30) & \text{if } 120 \le P \le 130 \\ G(30,35) & \text{if } 150 \le P \end{cases}. \tag{3}$$

In order to ensure the effective performance when the

Algorithm 1 Proposed scheme

Separation of the flowing IP address;
Input: IP address.
Output: Separated IP address s_i as 4 parts.
a = IP address 000,000,000,000;
$f_s(a) = s_i = s_1, s_2, s_3, s_4$.
Mod operation;
Input: s_i.
Output: K_j counting number.
R_b = The range of bits of bloom filter.
P = pps.
Select $b[R_b]$ as P.
$K_j = s_i$ - (R_b * int(s_i/R_b)).
Count K_j on the bloom filter;
Input: R_b, K_j.
Output: Counted bloom filter.
b = bloom filter.
Set $b = b[R_b] = b[K_j]+1$.
Deletion the values in the bloom filter;
Input: b.
Output: Deleted values in $b[R_b]$.
R_t = the deletion value point.
Select R_t as P.
f(dec-all(b)) as R_t.
Changing checkpoint;
Input: b.
Output: Detected values.
R_c = the rate of checkpoint.
Select R_c as P.
Detect reached values on R_c in $b[R_b]$.

fluctuant pps has flowed, R_b which is the range of bits of the bloom filter, R_t which is the deletion value point in the bloom filter, and R_c which is rate of the checkpoint to detect the attack IP address are randomly selected, respectively. There are two reason for why R_b, R_t, and R_c are randomly selected in the fluctuant pps. Firstly, even if the fluctuant pps has flowed, there is the range in the fluctuant pps. This range can be defined the number. Therefore, by using the number of ranges the fluctuant pps, R_b, R_t, and R_c are can be set as shown in (1), (2), (3). G is the formula that makes the random setting number. We categorize the range of fluctuant pps from 1 minimum pps to 30 maximum pps increasing 30 pps as shown in (1), (2), (3). However, from 150 pps, the range is not increased. This is because the bloom filter ensures the effective performance in the high rate pps from 150 pps. As for the second reason, it is necessary to set the random range to avoid the attacker's expectation. If the range is fixed, the attacker can extract the detection features during the flooding DDoS attack. In addition, since the ranges of the bit and checkpoint are fixed for bloom filter in the conventional schemes, the performance is not effective to detect the attack IP address. Moreover, even if the range is randomly chosen, the suitable random range is needed to deal with the fluctuant pps. Therefore, R_b is set as from 10 minimum bits to 20 maximum bits increasing 10 bits in (1) for the scalability of the detection in the fluctuant pps. We describe the performance for the range of bits in the bloom filter in Sect. 4.2. R_t is set as from 60 minimum msec to 70 maximum msec decreasing 10 msec in (2) to reduce the false positive rate. R_c is set as from 5 minimum msec to 10 maximum msec increasing 5 msec in (3) to detect the IP address without the misrecognition for the legitimate IP address.

3.2 Separation of IP Address into the Bloom Filter

For the detection of the flooding DDoS attack, we notice the IP address. This is because, according to [26], even though the IP address of bot is randomly chosen at the initiation of attack by the attacker, the IP address is not changed during the attack. Therefore, the IP address can be utilized to detect the flooding DDoS attack. Moreover, since we only consider the IP address to detect, we can deal with all kinds of the flooding DDoS attack. In addition, there are no analyzations for the models of traffic and entropy to find the attack features. There are two steps in this section. As for the first step, the IP address is stored into two memories. In the first memory, the IP address is separated via the mod operation as shown in the separation of IP address in Fig. 3. Since the original IP address cannot be restored after converted by the mod operation in the first memory, the copied IP address is stored in the second memory. There are two procedures for the second memory. As for the first procedure, the second memory is just used as the storage to keep the IP address. This is because the IP address is converted via the first memory, it cannot be restored the original IP address. So, there is no way to check to detect the attacker's original IP address from the first memory. When the IP address is detected as the attacker, it is necessary to check the attacker's original IP address. Therefore, as for the second procedure, the second memory is used for checking the attacker's original IP address. Since the second memory keeps the attacker's original IP address, although the attacker's original IP address is converted via the first memory, it can be detected via the second memory. As shown in the algorithm 1, the IP address is converted via the mod operation as s_i the separated IP address into four parts to input the IP address into the bloom filter. For example, if the IP address is 192.164.107.19, it is separated as s_1 192, s_2 164, s_3 107, and s_4 19. R_b is randomly selected as P. R_b is used for the mod operation. To explain the mod operation, R_b is set as 10 in the mod operation in Fig. 3. Then, the bloom filter is generated as 10 bits array. As for the second step, through the R_b mod operation in the first memory, 192, 164, 107, and 19 are converted to be K_j which is the counting number as 2, 4, 7, and 9.

3.3 Counting Values on the Bloom Filter

To defect the attack IP address, all of the flowing IP addresses are counted on the bloom filter. As shown in Counting values on the bloom filter in Fig. 3, each value, which is converted from the IP address via the mod operation, is counted on the array of the b bloom filter. The b is set as $b[R_b]$ via the algorithm 1. If the R_b is 10, $b[R_b]$ is set as the 10bits bloom filter. As the more same IP address flows, the more same K_j counting number increases at $b[K_j]$ which is the number of the array in the bloom filter as shown in the

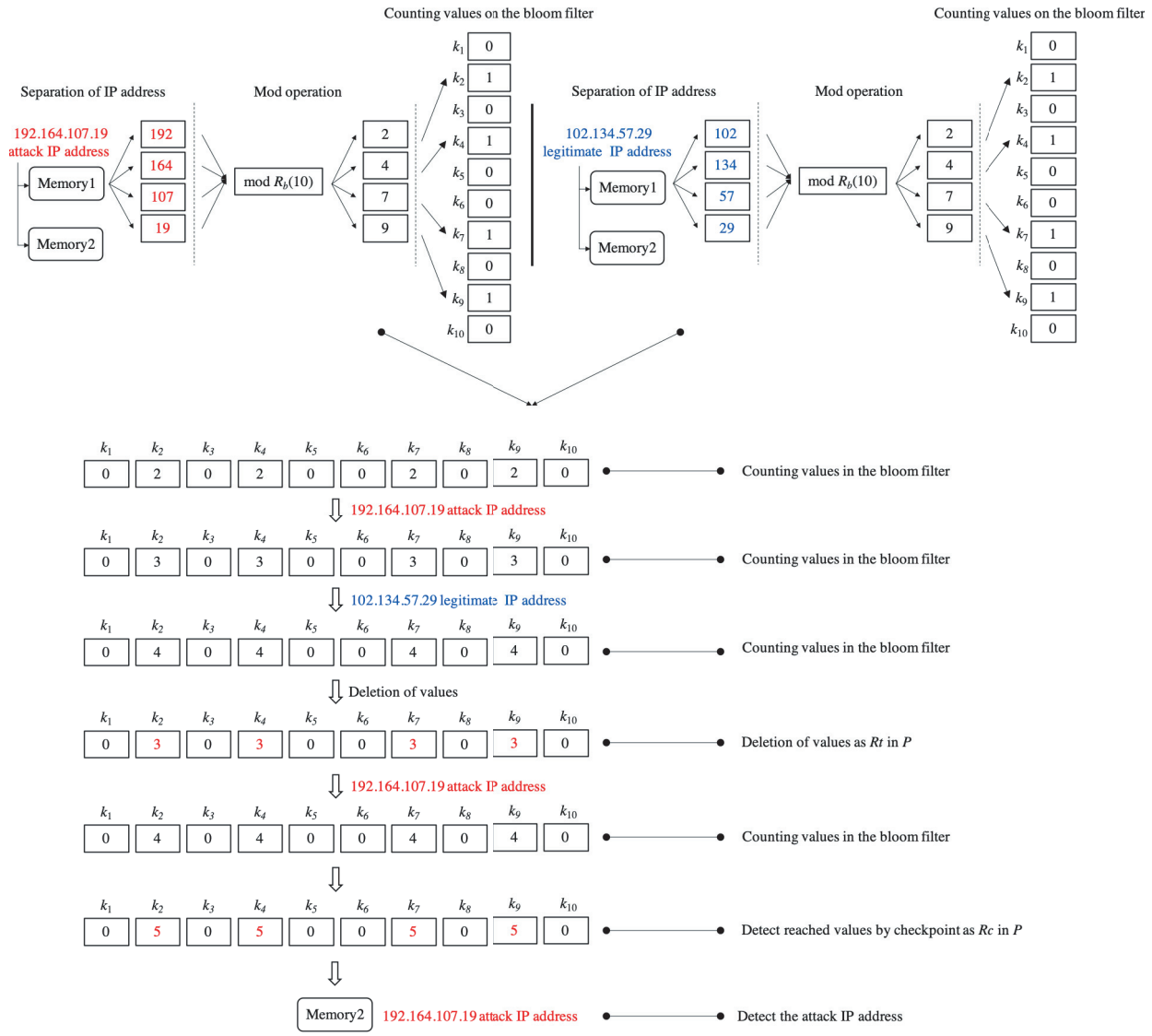


Fig. 3 The processes of the proposed scheme

counting values on the bloom filter in Fig. 3. However, the attack IP address cannot be detected yet. This is because if the converted legitimate IP address is the same as the attack IP address, the values are counted on the same array as shown in Fig. 3. This is a serious problem that occurs the false positive rate in the conventional schemes using the bloom filter. We describe how to solve the false positive rate in 3.4.

3.4 Detection of the Attack IP Address as the Fluctuant Pps

The attacker tries to exhaust the victim's network resource by the flooding DDoS attack. For this, attacker has to send the high rate packet then the legitimate packet. In addition, if the attacker can send the low rate packet to the victim, the packets of attacker can flow with the legitimate packet into the victim. The conventional schemes using the bloom filter are not flexible for the fluctuant pps. This means that the low rate attackers' packet cannot be detected. Here, we describe how to detect the attack IP address as the fluctuant pps. This section is classified into two steps as 3.4.1 and 3.4.2.

3.4.1 Deletion of Value in the Bloom Filter via Dec-all Operation

As the same packets are flowing into the router, the rate values of the bloom filter are increasing as shown the counting values in the bloom filter in Fig. 3. This means that if the values of legitimate IP address via the mod operation is the same with the values of attack IP address, it will be counted as the same value on the bloom filter. This is the main reason to occur the false positive rate in the conventional schemes using the bloom filter. To explain the reason that the false positive rate has occurred, the checkpoint is set as 5 to detect the attack IP address. If the 4 values in the bloom filter are

reached to 4 by only the attack IP address, and the legitimate IP address is added one time, the total values are reached to 5 in the bloom filter. Then, the legitimate IP address will be blocked as the attack IP address. Moreover, if the low rate attack IP address does not reach the checkpoint, the attacker's packet will flow into the victim. This is because the values of attack IP address are not counted on the bloom filter. To reduce the false positive rate and detect the low attack IP address, the values in the bloom filter are deleted via dec-all operation as the fluctuant pps. As shown algorithm 1, R_t is randomly selected as P to avoid the attacker's expectation when the packet is deleted and detected as the fluctuant pps. The values in the array of the bloom filter are decreased by the dec-all operation as shown in the deletion of values in R_t as P in Fig. 3. Since the values in the bloom filter are deleted before they reach the checkpoint, the values of the legitimate IP address which occurs the false positive rate can be reduced.

3.4.2 Changing the Checkpoint as the Fluctuant Pps

Although the false positive rate can be reduced, the low rate attack IP address cannot be detected. This is because the values of the low rate attack IP address is deleted before it reaches to the checkpoint. Therefore, the checkpoint should be flexible. If the checkpoint is fixed, it is difficult to detect attack IP address when the attacker and legitimate have the high or low pps. The checkpoint can be changeable as the fluctuant pps in the proposed scheme. As shown algorithm 1, R_c the rate of the checkpoint to detect the reached IP address is randomly selected as P. Whatever the attack pps is high and low, it can be detected via the changing the checkpoint as the fluctuant pps.

Through the 3.4.1 and 3.4.2, the attack IP address can be detected regardless of the fluctuant pps. In addition, the legitimate IP address cannot be blocked as the attacker. This means that the false positive rate will be reduced in the proposed scheme. There are no complicated processes which are the analyzing features by gathering and learning the packets in the proposed scheme. By the preparation for detecting the attack IP address, the proposed scheme can detect the attack IP address at once. After the attack IP address is detected, the memory 1 and memory 2 are reset to check the new flowing IP address.

4. Results and Evaluations

We compare the performances of the proposed scheme with the conventional detection schemes using the bloom filter [23]–[25], the machine learning scheme [18], the packet pattern analyzation schemes [19], [20] in terms of the accuracy, the overhead. We evaluate the performances on datasets [27], [28] in the environment of the experiments as shown in Table 1. In the conventional schemes [18]–[20], only the ISCX dataset is used. To ensure that the proposed scheme can detect the attack IP address from the various botnet flooding DDoS attack, we used three kinds of datasets which are the ISOT dataset, the botnet traffic generated by the malware capture facility project, and the ISCX dataset [27], [28]. According to [28], the ISOT dataset has been created by merging different available datasets. It contains both malicious and non-malicious traffic. The botnet traffic generated by the malware capture facility project is a research project with the purpose of generating and capturing botnet traces in the long term. The ISCX has been generated in a physical testbed implementation using real devices that generate real traffic that mimics users' behavior. Table 2 shows the description of datasets. The portion of flows means that how many the bots access to the victim from the datasets. There are 16 kinds of bots and two kinds of fluctuant pps as the attack pps and the legitimate pps for each bot in the datasets. The range of the pps is not fixed for each bot. This means that there is the fluctuant pps in each bot. There are three protocols as the IRC, HTTP, and P2P. The accuracy part shows the comparison of the ACC (Accuracy), the TPR (True Positive Rate), the FPR (False Positive Rate), and the ACC for bits of the bloom filter in the proposed scheme. The overhead part shows the comparison of the data size and the calculation time to detect the attack IP address.

Table 1 Experiments environment

Items	Parameter value
Operating System	Ubuntu Linux 16.04.3 LTS
Programming language	Python
CPU	Intel i9-7900X 3.38Ghz
Memory	64GB
datasets	[27], [28]

Table 2 Description of datasets

Bot (Protocol) type	Portion of flows	Attack PPS	Legitimate PPS
Neris (IRC)	25967	0 ~ 1620	0 ~ 1240
Rbot (IRC)	83	0 ~ 430	0 ~ 213
Menti (IRC)	2878	0 ~ 392	0 ~ 545
Sogou (HTTP)	89	0 ~ 285	0 ~ 261
SMTP Spam (P2P)	21633	0 ~ 822	0 ~ 548
UDP Storm (P2P)	44062	0 ~ 1213	0 ~ 1076
Tbot (IRC)	1296	0 ~ 380	0 ~ 224
Zero Access (P2P)	1011	0 ~ 173	0 ~ 166
Murlo (IRC)	4881	0 ~ 697	0 ~ 587
Virut (HTTP)	58576	0 ~ 1785	0 ~ 1648
NSIS (P2P)	757	0 ~ 339	0 ~ 328
Zeus (P2P)	502	0 ~ 273	0 ~ 263
Weasel (P2P)	42313	0 ~ 1463	0 ~ 1263
Smoke bot (P2P)	78	0 ~ 195	0 ~ 174
Zeus Control (P2P)	31	0 ~ 175	0 ~ 143
ISCX IRC bot (P2P)	1816	0 ~ 492	0 ~ 363

- PPS: Packet Per Second

4.1 Characteristic of Bot in Datasets

We need to know what is the bot's purpose in the datasets to understand each characteristic. Therefore, we describe the characteristic of each bot as below:

- Neris generates the high failed ratio and performs scanning technique by the traffic content pertaining.
- Rbot causes traffic content pertaining to IRC based attacks.
- Menti causes traffic content pertaining to identity theft and login credentials.
- Sogou causes traffic content pertaining to spam and popup adware to collect personal information.
- SMTP Spam is primarily used for spreading spam content on the web via SMTP.
- UDP Storm is camouflaged to look like legitimate P2P communication to gather data on the user.
- Tbot is the malware that uses the Tor network to communicate with its C&C (Command and Control) server.
- Zero access is the malware which uses an unstructured P2P architecture. It regularly queries their neighbors for new malware payloads.
- Murlo causes traffic content pertaining to the use of scanning activities and proprietary mechanisms for establishing C&C.
- Virut causes traffic content pertaining to spam, fraud, and data theft attacks by using a fast-flux.
- NSIS causes traffic content pertaining to identity theft and login credentials by using extra payloads.
- Zeus is intended to perform malicious activities on the victim's computer.
- Weasel is designed for implementing secure communication between bots and the botmaster.
- Smoke bot is a modular loader where attackers can select any payload to be installed on the victim. It can be used to load other malware.
- Zeus Control is intended to perform malicious activities on the victim's computer with its C&C server.
- ISCX IRC bot is the set of scripts or an independent program that connects to IRC as a client, and so appears to other IRC users as another user.

4.2 Accuracy

In order to compare the effectiveness for detecting the flooding DDoS attack, the ACC, the TPR, and the FPR are calculated as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4}$$

$$TPR = \frac{TP}{TP + FN}, \tag{5}$$

$$FPR = \frac{FP}{FP + FN}. \tag{6}$$

Where the TP, the TN, the FP, and the FN denote the number of True Positive, True Negative, False Positive, and False Negative, respectively. Figure 4 shows the total fluctuant pps of flooding DDoS attack in the datasets. To evaluate the performances between the proposed scheme and the conventional schemes, all of the attack IP address and the legitimate IP address from each bot flow into each scheme at the same time from 0sec to 70ksec as the fluctuant pps. Since each bot has its the fluctuant pps as the attack pps and legitimate pps as shown in Table 2, the victim side cannot know whether the attack IP address flows or not via the fluctuant pps. Figure 4 shows the total fluctuant pps of flooding DDoS attack in the datasets. The total fluctuant pps reaches from 0 pps to 1,785 pps. In the conventional schemes, many legitimate IP addresses are detected as the attack IP address. This is because the separation to know the attacker and legitimate user is not effective when the legitimate IP address has the fluctuant pps like the attack pps in the conventional schemes. However, in the proposed scheme, though the legitimate pps is similar to the attack pps, the legitimate pps is not detected as the attack. This is because the changeable dec-all operation and the checkpoint as the fluctuant pps make to correctly detect the attack pps regardless of the range of the fluctuant pps in the datasets. This is the advantage of the proposed scheme for detecting the attack IP address from the fluctuant pps.

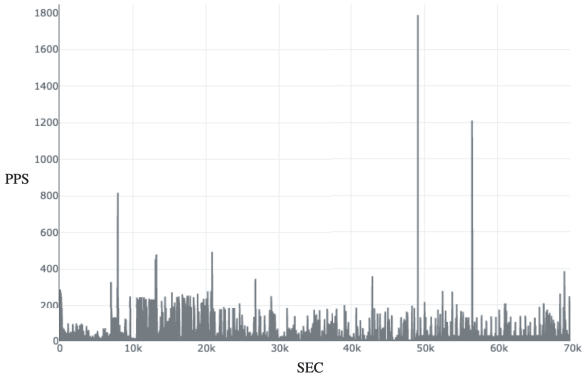
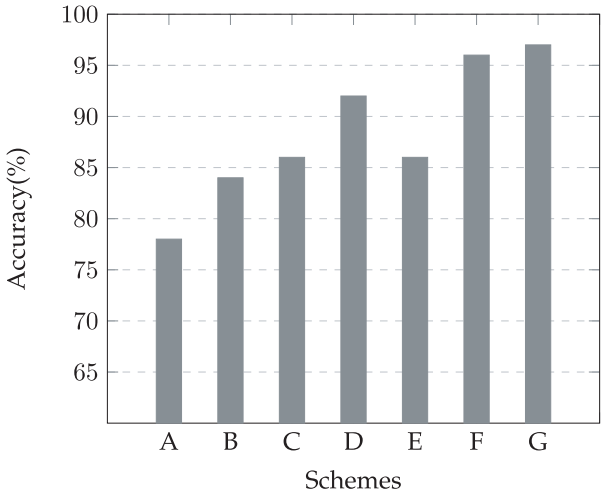


Fig. 4 Total fluctuant pps of flooding DDoS attack in the datasets



A: Song et al. B: Roh et al. C: Bogdanoski et al. D: Yuan et al.
E: Yuan-H el al F: Yichen el al. G: Proposed

Fig. 5 Comparison of the ACC

Figure 5 shows the comparison of the ACC. Where A, B, C, D, E, F, and G denote the scheme using bloom filter [23], the scheme using bloom filter with the

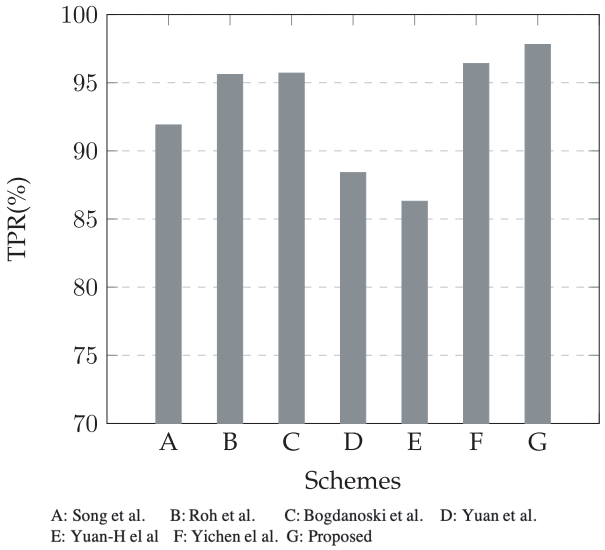


Fig. 6 Comparison of the TPR

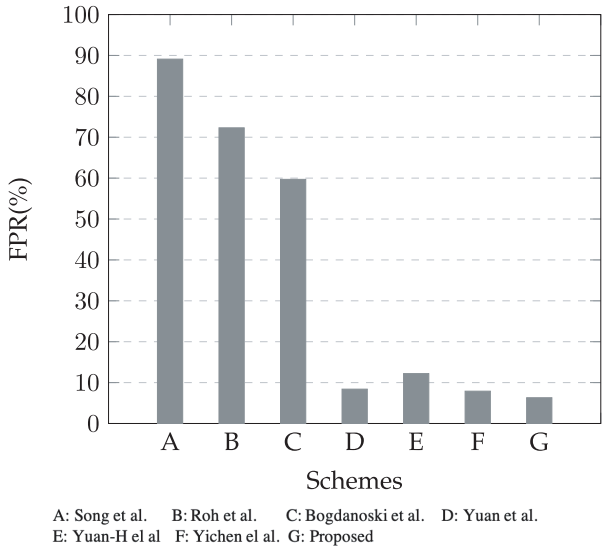


Fig. 7 Comparison of the FPR

blacklist [24], the scheme using bloom filter with whitelist [25], the machine learning scheme [18], the pattern analyzation scheme [19], the pattern analyzation scheme by ranking [20], and the proposed scheme, respectively. The ACC is 78.1% in the conventional scheme using the bloom filter [23]. This is because the attack IP address cannot be detected and the many legitimate users are blocked as the attacker. Since it is difficult to classify which IP address is bad or not, the ACC is low. By using the blacklist with bloom filter [24] and whitelist with bloom filter [25], the ACC rates are reached 84.1% and 86.8%, respectively. However, since those schemes cannot detect the low rate attack IP address, the ACC is low. In the machine learning [18], the ACC is 92%. Since the attack feature cannot be learned at one, the TP is low. In the pattern analyzation scheme [19], if there are unanalyzed packets between the attack and the bots, the attack IP address cannot be detected. So, the ACC just shows 86.9%. Although there are unanalyzed packets, since the attack feature can be extracted by the ranking of the packet, Yichen et al. [20] shows the 96.3% ACC. In the proposed scheme, since the TP is high and FP is low, the ACC is reached 97.5%. This is because the attack IP address can be detected regardless of the fluctuant pps. Moreover, the legitimate IP address is not considered as the attack. The ACC of the proposed scheme is much higher than the conventional schemes using the bloom filter. In addition, the ACC is 1.2% higher than Yichen et al. [20].

Figure 6 shows the comparison of the TPR. The conventional schemes using bloom filter [23]–[25] show 91.9% TPR, 94.6% TRR, and 95.7% TRP, respectively. This is because there is the attacker's packet which cannot be detected since those schemes are not flexible to detect as the fluctuant pps. In addition, the legitimate packet interrupts the detect the attack IP address by increasing its values before the values of attack IP address is checked in the bloom filter. [18], [19] show 88.4% TPR and 86.3% TRP, respectively. This is because the attack packets cannot be detected at once since the attack packets are not learned and analyzed. [20] shows the high TPR as 96.4% among the conventional schemes. This method can detect the attack features by the ranking of packets even if it is not analyzed. The proposed scheme shows 97.8% TPR. This is 1.4% higher than [20]. This is because the checking point in the bloom filter is flexibly changed from the fluctuant pps. In addition, the values in the bloom filter are deleted before the legitimate packet interrupts the detect the attack IP address. Therefore, the TPR in the proposed scheme is highest.

Figure 7 shows the comparison of the FPR. The conventional schemes using bloom filter [23] shows 89.1% FPR. This is the highest rate among the comparison schemes. This is because the legitimate user is recognized as the attacker when the legitimate user's packet is reached the checkpoint before the attack packet is reached. The FPR in [24] also shows high FPR as 72.3%. By using the blacklist, the FPR is a bit reduced. However, it is still high. Even though the FPR is reduced as 59.7% in [25] by using the whitelist, it is not enough for guaranteeing the better performance. [18] shows 8.4% FPR since there are the unlearned attack features via the machine learning. [19] shows 12.2% FPR. This is because it cannot recognize which packets are the attack or not before analyzing the packet between the attack and the bots. [20] shows 7.9% FRP since if the attacker sends the same packet with the legitimate user, it is hard to extract the attack features. In the proposed scheme, when the legitimate user sends many packets like the flooding attack, it will be recognized as the attack. Therefore, the FPR is 6.3% in the proposed scheme. However, this is the lowest FPR among the comparison schemes.

Figure 8 shows the ACC as the range of bits for the bloom filter in the proposed scheme. In order to detect the flooding DDoS attack, we randomly set the range of bits for the bloom filter. The ACC is 93.6% from 10 bits bloom

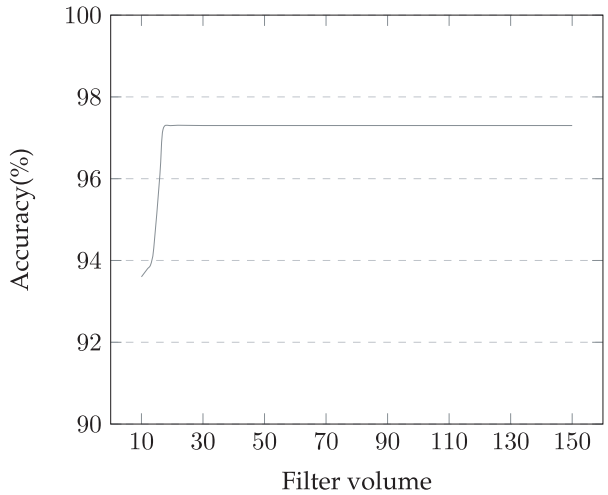


Fig. 8 Range of bits for the bloom filter

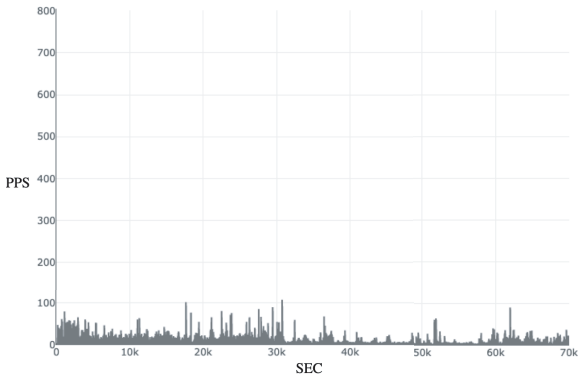


Fig. 9 The pps via the proposed scheme

filter. However, as shown in Fig. 8, the ACC is the same as 97.8% from 20 bits to 150 bits even if the calculation time is a bit increased. Therefore, the range from 20 bits does not make much difference for the ACC in the proposed scheme. However, although the effect from the range of bits is low, the bits for the bloom filter can be changed for the scalability reason to cope with the fluctuant pps.

Figure 9 shows how much the pps rate from the flooding DDoS attack can be reduced via the proposed scheme. The packet rate is 92% reduced compared to the attack packet as shown in Fig. 4. Therefore, although there is no complex process to detect the flooding DDoS attack, we can know that the proposed scheme has the high ACC, the high TPR, and the low FPR for detecting the flooding DDoS attack.

4.3 Overhead

4.3.1 Comparison of the Data Size

This section describes the comparison of the data size which is generated while detecting the attack IP address on the datasets from each scheme size to detect the attack IP address. Figure 10 shows the comparison of data size. In [23], the data size is 409 bytes since the bloom filter just

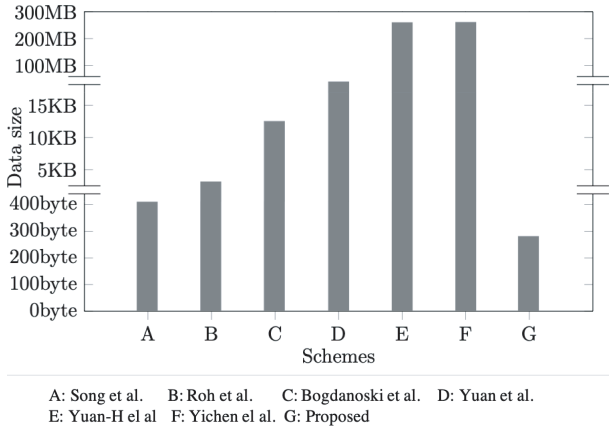


Fig. 10 Comparison of the data size

uses the IP address to detect the attack. For the same reasons, in [24], [25] show the low data size as the 3.1Kbytes and 12.5Kbytes, respectively. Since the blacklist and the whitelist has the difference size, the data size to detect is a bit different. However, the data size extremely increases in [18]–[20]. In [18], since the machine learning scheme needs to learn by storing the attack feature for the analyzation, the data size is 39.2Mbytes. The data size is much highest in [19], [20] as 259Mbytes and 260Mbytes, respectively. This is because both schemes store all of the packets between the attack and bots to extract the attack patterns. This means that the more data size will be increased when the new types of flooding DDoS attack flow. Moreover, the highest data size might be the burden to detect the attack. However, the proposed scheme just shows a much smaller data size as 280bytes than the conventional schemes. This is because the values in the bloom filter are deleted from the fluctuant pps. Moreover, there are no blacklist and whitelist in the proposed scheme to detect the attack. In addition, after the attack is detected, the first memory and second memory are reset. Therefore, the proposed scheme has the advantage as lightweight detection.

4.3.2 Comparison of the Calculation Time

Here, we evaluate the calculation time for processing to detect attack IP address on the datasets from each scheme. Figure 11 shows the comparison of the calculation time. The calculation time is 10.9sec in [23]. Since the bloom filter has no complexity, it can quickly detect the attack. However, in [24], [25], The calculation times are increased as 45.2sec and 45.3sec, receptively. This is because it takes time to recognize the attack via the blacklist and the whitelist. In [18], the calculation time shows 37.9sec since the machine learning needs to learn the attack features while taking the attack. The calculation times in [19], [20] show 17.2sec and 16.2sec, respectively. The attack can be detected after the attack patterns between the attack and bot are analyzed. The proposed scheme shows 11.1sec. This is 0.2sec slower than [23]. This is because the dec-all operation and the checkpoint are changed as the fluctuant pps. It increases a bit of

Table 3 The comparison of quantitative evaluation data

Bot	Neris	Rbot	Menti	Sogou	SMTP Spam	UDP Storm	Tbot	Zero Access
Number of dec-all	9088	29	1007	31	7571	15421	453	353
Number of checkpoint	25967	83	2878	89	21633	44062	1296	1011
Protocol	IRC	IRC	IRC	HTTP	P2P	P2P	IRC	P2P

Bot	Murlo	Virut	NSIS	Zeus	Weasel	Smoke bot	Zeus Control	ISCX IRC bot
Number of dec-all	1708	20501	264	175	14809	27	10	635
Number of checkpoint	4881	58576	757	502	42313	78	31	1816
Protocol	IRC	HTTP	P2P	P2P	P2P	P2P	P2P	P2P

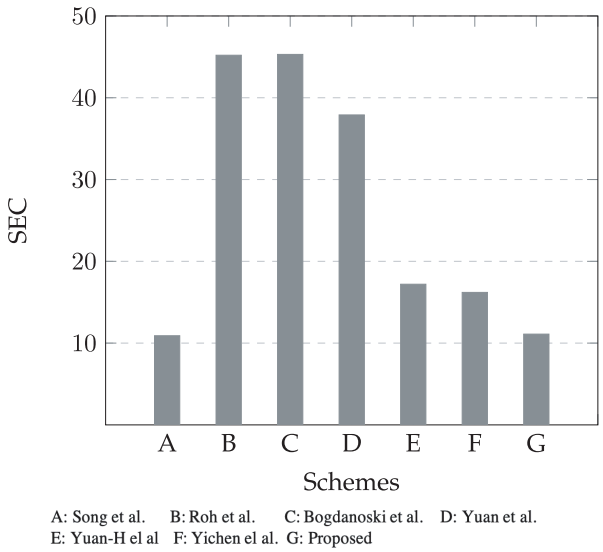


Fig. 11 Comparison of the calculation time

time to detect the attack in the proposed scheme. However, our proposed scheme also can quickly detect the attack.

5. Discussion and Limitation

We describe the quantitative evaluation data for dec-all operation and the checkpoint as shown in Table 3. The number of dec-all and checkpoint denote how many the dec-all operation and checkpoint are conducted on the dataset in the proposed scheme. The number of dec-all operation shows 30~40% ranges on the portion of flows in the dataset. Since the most bots in the dataset flow as 150 over pps, the range of dec-all is not frequently changed. The checkpoint is steady to deal with 150 over pps in the dataset. In addition, the number of checkpoint is the same as the number of the portion of flows in the dataset. This is because the checkpoint is changed when the IP address flows as the fluctuant pps.

Since we only consider the IP address to detect the flooding DDoS attack, if the bot's IP address is converted via the IP spoofing, it is not able to track back in the proposed scheme. However, although the bot's IP cannot be traced, the IP address is not changed during the attack as we mentioned in Sect. 3. Therefore, the attack IP address will be detected at once even if there is the bot's attack by using the different IP addresses in the proposed scheme. In addition, since we focused on how to reduce the data size and calculation time with the high ACC, the high TPR, and the low FPR without the complexity, the attack features cannot be extracted as the entropy, the pattern by the analyzation via the machine learning, etc. For future work, we will research how to extract the attack feature from the bloom filter via machine learning.

6. Conclusion

Since the conventional schemes are not effective to detect the flooding DDoS attack, we have proposed the lightweight detection using bloom filter against flooding DDoS attack. The evaluations show that our proposed scheme is improved as for the accuracy and the overhead. By using the dec-all operation and the checkpoint which are flexibly changed as the fluctuant pps, the proposed scheme ensures the high ACC, TPR, and the low FRP for the accuracy. In addition, the data size and calculation time are much lower for the overhead than conventional schemes. The flooding DDoS attack can be detected without the complex process in the proposed scheme.

Acknowledgments

This work is partly supported by the Grant in Aid for Scientific Research (No.17K06440) from Japan Society for Promotion of Science (JSPS).

References

[1] C. Buragohain, M. Jyoti, S. Singh, and K. Dhruba, "Anomaly Based DDoS Attack Detection," International Journal of Computer Applications, vol.123, no.17, pp.35–40, 2015.
[2] J. Mirkovic, A. Hussain, S. Fahmy, P. Reiher, and R.K. Thomas, "Accurately measuring denial of service in simulation and testbed experiments," IEEE Trans. Depend. Secure Comput., vol.6, no.2, pp.81–95, April-June 2009.
[3] N. Hoque, D.K. Bhattacharyya, and J.K. Kalita, "Botnet in DDoS attacks: Trends and challenges," Commun. Surveys Tuts., vol.17, no.4, pp.2242–2270, 4th Quart. 2015.
[4] A. Chadd, "DDoS Attacks: Past, Present and Future," Network Security, vol.2018, no.7, pp.13–15, 2018.
[5] https://www.cisomag.com/attackers-launch-ddos-attack-on-food-delivery-startup-liefrando/ [Internet]
[6] https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-financial-services-hostile-takeover-attempts-report-2020.pdf [Internet]
[7] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "Analysis of DDoS-Capable IoT Malwares," Proc. 2017 Federated Conference on Computer Science and Information Systems, FedCSIS

2017, vol.11, pp.807–816, 2017.
[8] A.M.R.K. Munivara Prasad and K. Rao, "Dos and ddos attacks: Defense, detection and traceback mechanisms - a survey," Global Journal of Computer Science and Technology, vol.14, 2014.
[9] C. Yao, X. Luo, and A.N. Zincir-Heywood, "Data Analytics for Modeling and Visualizing Attack Behaviors: A Case Study on SSH Brute Force Attacks," 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proc. 2018-January, pp.1–8, 2018.
[10] Y.-J. Lee, N.-K. Baik, C. Kim, and C.-N. Yang, "Study of Detection Method for Spoofed IP against DDoS Attacks," Personal and Ubiquitous Computing, vol.22, no.1, pp.35–44, 2018.
[11] X. Jing, Z. Yan, X. Jiang, and W. Pedrycz, "Network Traffic Fusion and Analysis against DDoS Flooding Attacks with a Novel Reversible Sketch," Information Fusion, vol.51, no.November 2018, pp.100–113, 2019.
[12] A. Furfaro, P. Pace, and A. Parise, "Facing DDoS Bandwidth Flooding Attacks," Simulation Modelling Practice and Theory, vol.98, no.September 2019, 101984, 2020.
[13] M.T. Manavi, "Defense Mechanisms against Distributed Denial of Service Attacks: A Survey," Computers and Electrical Engineering, vol.72, no.4, pp.26–38, 2013.
[14] Z. He, T. Zhang, and R.B. Lee, "Machine Learning Based DDoS Attack Detection from Source Side in Cloud," Proc. - 4th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2017 and 3rd IEEE International Conference of Scalable and Smart Cloud, SSC 2017, pp.114–20, 2017.
[15] N. Jeyanthi and P.C. Mogankumar, "A Virtual Firewall Mechanism Using Army Nodes to Protect Cloud Infrastructure from DDoS Attacks," Cybernetics and Information Technologies, vol.14, no.3, pp.71–85, 2014.
[16] A. Aborujilah and S. Musa, "Cloud-Based DDoS HTTP Attack Detection Using Covariance Matrix Approach," Journal of Computer Networks and Communications, vol.2017, 2017.
[17] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "JESS: Joint Entropy-Based DDoS Defense Scheme in SDN," IEEE Journal on Selected Areas in Communications, vol.36, no.10, pp.2358–2372, 2018.
[18] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," 2017 IEEE International Conference on Smart Computing, SMARTCOMP 2017, pp.1–8, 2017.
[19] Y.-H. Su, A. Rezapour, and W.-G. Tzeng, "The forward-backward string: A new robust feature for botnet detection," IEEE Conference on Dependable and Secure Computing, 2017.
[20] Y. An, S. Haruta, S. Choi, and I. Sasase, "Traffic Feature-based Botnet Detection Scheme Emphasizing the Importance of Long Patterns," IEICE Commun. Express, vol.9, no.1, pp.7–12, Jan. 2020.
[21] B.H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," Communication of the ACM, vol.13, no.7, pp.422–426, July 1970.
[22] I. Voras and M. Žagar, "Adapting the Bloom Filter to Multithreaded Environments," Proc. Mediterranean Electrotechnical Conference - MELECON, no.2, pp.1488–93, 2010.
[23] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, "Fast Hash Table Lookup Using Extended Bloom Filter: An Aid to Network Processing," Computer Communication Review, vol.35, no.4, pp.181–192, 2005.
[24] B.-H. Roh, J.W. Kim, K.-Y. Ryu, and J.-T. Ryu, "A Whitelist-Based Countermeasure Scheme Using a Bloom Filter against SIP Flooding Attacks," Computers and Security, vol.37, pp.46–61, 2013.
[25] M. Bogdanoski, A. Toshevski, D. Bogatinov, and M. Bogdanoski, "A Novel Approach for Mitigating the Effects of the TCP SYN Flood DDoS Attacks," World Journal of Modelling and Simulation, 12, no.3, pp.217–30, 2016.
[26] X. Huang, X. Du, and B. Song, "An Effective DDoS Defense Scheme for SDN," IEEE International Conference on Communications, 2017.
[27] https://www.unb.ca/cic/datasets/botnet.html [Internet]
[28] E.B. Beigi, H.H. Jazi, N. Stakhanova, and A.A. Ghorbani, "Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches," 2014 IEEE Conference on Communications and Network Security, CNS 2014, pp.247–55, 2014.

Sanghun Choi received his M.Sc. degrees from Keio University in 2018. He is a Ph.D. student at Keio University. His research interests are the security and privacy for the location system, the DDoS detection, and IoT. He is a member of IEICE and IEEE.

Yichen An received his B.Sc. degrees from Beijing Information Science and Technology University in 2017. received his M.Sc. degrees from Keio University in 2020. His research interest is an information security. He is a member of IEICE.

Iwao Sasase was born in Osaka, Japan in 1956. He received the B.E., M.E., and D.Eng. degrees in Electrical Engineering from Keio University, Yokohama, Japan, in 1979, 1981 and 1984, respectively. From 1984 to 1986, he was a Post Doctoral Fellow and Lecturer of Electrical Engineering at the University of Ottawa, ON, Canada. He is currently a Professor of Information and Computer Science at Keio University, Yokohama, Japan. His research interests include modulation and coding, broadband mobile and wireless communications, optical communications, communication networks and information theory. He has authored more than 297 journal papers and 445 international conference papers. He granted 45 Ph.D. degrees to his students in the above field. Dr. Sasase received the 1984 IEEE Communications Society (ComSoc) Student Paper Award (Region 10), 1986 Inoue Memorial Young Engineer Award, 1988 Hiroshi Ando Memorial Young Engineer Award, 1988 Shinohara Memorial Young Engineer Award, 1996 Institute of Electronics, Information, and Communication Engineers (IEICE) of Japan Switching System Technical Group Best Paper Award, and WPMC 2008 Best Paper Award. He is now serving as a President of IEICE. He served as President of the IEICE Communications Society (2012-2014). He was Board of Governors Member-at-Large (2010-2012), Japan Chapter Chair (2011-2012), Director of the Asia Pacific Region (2004-2005), Chair of the Satellite and Space Communications Technical Committee (2000-2002) of IEEE ComSoc., Vice President of the Communications Society (2004-2006), Chair of the Network System Technical Committee (2004-2006), Chair of the Communication System Technical Committee (2002-2004) of the IEICE Communications Society, Director of the Society of Information Theory and Its Applications in Japan (2001- 2002). He is Fellow of IEICE, and Senior Member of IEEE, Member of the Information Processing Society of Japan.