Using a History-based Profile to Detect and Respond to DDoS Attacks*

Negar Mosharraf[1], Anura P. Jayasumana[1] and Indrakshi Ray[2]
[1]*Dept. of Elec. and Comp. Eng., Colorado State University, Fort Collins, CO, U.S.A.*
[2]*Dept. of Computer Sc., Colorado State University, Fort Collins, CO, U.S.A.*

Keywords: Distributed Denials of Service Attack, Flooding Attack, Network Security, Bloom Filter.

Abstract: Distributed Denial-of-Service (DDoS) attack has been identified among the most serious threat to service availability on the Internet. It prevents legitimate users from accessing the victim node by flooding and consuming network resources. In order to combat such attacks, a distributed defense mechanism is needed that will thwart the attack traffic in real time. We propose one such mechanism that when deployed is able to filter out malicious and allow legitimate traffic to the extent possible during the attack period. We characterize the network traffic and introduce a new history-based profile algorithm that filters potential attack traffic and aims to maximize the flow of legitimate traffic in the event of an attack. We investigate the features of network traffic that can be used to do such filtration. We use a Bloom filter based mechanism to efficiently implement the history-based profile model which serves to reduce the communication and computation costs. We evaluate our scheme using simulations on recent real-world traffic datasets. The experimental results demonstrate the effectiveness of our scheme. In order to improve communication and computation costs even further, we propose using three octets of the IP address instead of the whole address. We also demonstrate how using three octets of IP address impacts the accuracy, efficiency, and memory requirement of the filtering mechanism.

1 INTRODUCTION

Internet services often constitute critical infrastructure and they must be protected from Denial of Service (DoS) attacks. In recent years, DDoS attacks on networks have been responsible for large number of network infrastructure and service outages (Steinberger et al., 2005), (Munivara Prasad et al., 2014). On February 9, 2000, Yahoo, eBay, Amazon.com, E*trade, ZDnet, buy.com, the FBI and several other websites fell victim to DDoS attacks resulting in millions of dollars in damages (Gil and Poletto, 2001), (Waikato Applied Network Dynamics Research Group, 2016). Despite significant research focusing on countermeasures, DDoS attacks still remain a major threat (Chen and Park, 2005). Recent examples include a record 400 Gbit/s DDoS attack against CloudFlare, a rate about 100 Gbit/s more than the largest previously seen DDoS attack (Mathew J. Schwartz, 2014). The frequencies and the impact of DDoS attacks have motivated researchers in the Internet security community to provide techniques for preventing, detecting, and surviving such attacks. In fact, the flooding traffic is large enough to crash the victim by communication buffer overflow, disk exhaustion, or connection link saturation (Chen et al., 2007), and most approaches have failed to provide service availability in the presence of DDoS attacks.

A plethora of DDoS defense and response mechanisms have been suggested in the past, including preventive techniques (Yaar et al., 2003), (Wang et al., 2007), packet filtering (Kim et al., 2004), flood pushback (Ioannidis and Bellovin, 2002), DDoS detection mechanism (Gil and Poletto, 2001), (Mirkovic et al., 2002) and distributed defense mechanism (Mahajan et al., 2002), (Papadopoulos et al., 2003), (Francois et al., 2012), (Aghaei Foroushani, 2014). Preventive techniques aim to solve the problem of IP spoofing but this is a hard problem because attackers can compromise large number of computers to create zombies. Other works, which we discuss in Section 2, focus on detecting and mitigating the attack. However, these techniques often penalize legitimate traffic because it is unable to distinguish between attack and

*This work was partially supported by NSF I/UCRC Award Number 1650573 and funding from CableLabs. The views and conclusions contained in this document are those of the authors and should not be automatically interpreted as representing the official policies, either expressed or implied of NSF and CableLabs.

Mosharraf, N., Jayasumana, A. and Ray, I.
Using a History-based Profile to Detect and Respond to DDoS Attacks.
DOI: 10.5220/0006435801750186
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 175-186
ISBN: 978-989-758-259-2

legitimate traffic. We propose an alternative solution to this problem.

In order to continue to provide services under DDoS attacks, it is essential to distinguish attack traffic from legitimate traffic. The challenge is how to detect attack traffic without misclassifying legitimate traffic. An unusually high traffic volume may not by itself be a good indicator of a DDoS attack, as it can occur due to flash crowds. Thus, other features that help distinguish attacks from normal traffic have to be considered. We look into multiple features of DDoS attacks and normal traffic to extract characteristics that give information about the occurrence of the DDoS attack. These features and their correlations are used to establish high confidence IP address history that forms a normal traffic profile. Anything outside the normal traffic profile constitutes an attack. The IP address history that forms the normal traffic profile must be propagated to the upstream routers so that the attack traffic can be blocked early on close to the source of the attack. However, propagating this IP address history for normal traffic during an attack introduces communication overhead on already congested routes. Towards this end, we demonstrate how a Bloom filter can be used to store the normal IP address history profile which is propagated to the routers close to the attack nodes. The use of Bloom filter reduces the communication cost and also minimizes the storage cost at the routers. The effectiveness of our approach is validated using the real dataset collected from Colorado State University (CSU) 2015 as well as Center for Applied Internet Data Analysis (CAIDA) 2007 attack dataset. In order to minimize the size of the Bloom filter, we use three octets of the IP address instead of four octets while creating the normal profile. We demonstrate through experiments how this impacts the size of the Bloom filter, the protection it provides, and the amount of legitimate traffic that is prohibited from reaching the destination.

The rest of this paper is structured as follows. Section 2 describes some related work in this area. Section 3 describes how we distinguish attack from normal traffic. Section 4 validates our model using CSU 2015 and CAIDA 2007 dataset. Section 5 concludes the paper with pointers to future directions.

2 RELATED WORK

Several works appear in detecting and responding to them. Pushback (Mahajan et al., 2002) is one of the earlier techniques that mitigates DDoS attacks using a cooperative mechanism. When a link is heavily utilized, information is sent to the upstream routers to curtail some of the downstream traffic. This approach requires deployment of the mechanism in all the routers; this may be unacceptable to some routers because of the high computational and memory overheads. It also penalizes legitimate traffic.

Attack source identification and responsive techniques actively try to mitigate DDoS attacks by filtering or limiting the rate of suspicious flow (Gil and Poletto, 2001), (Chen and Park, 2005), (Mirkovic et al., 2002). Such schemes have two components, namely, attack detection and packet filtering. The characteristic of attack packets, such as source of IP address or marked IP header values (Yaar et al., 2003), (Sung and Xu, 2003), (Yaar et al., 2004), are often used to detect and identify attack traffic. These characteristics are used for packet filtering. Note that packet filtering can be applied either close to the attacking node (Gil and Poletto, 2001), (Mirkovic et al., 2002) or close to the victim node (Yaar et al., 2003), (Mirkovic et al., 2002), (Sung and Xu, 2003), (Yaar et al., 2004); however, once the attacker knows features that are of interest to the detection mechanism, he can develop strategies to bypass it.

Some researchers use change-point detection theory to detect abnormal Internet traffic caused by DDoS attacks where the scheme is based on abrupt change (Chen et al., 2007), (Peng et al., 2004), (Wang et al., 2004), (Manikopoulos and Papavassiliou, 2002), (Noh et al., 2008). In this approach, the attacker can bypass the detection mechanism by sending out attack flow to change the statistics of the traffic. Moreover, we often do not have accurate statistics to describe the pre-change and post-change traffic distribution.

Another type of approach is based on flow dissymmetry (Gil and Poletto, 2001), (Mirkovic and Reiher, 2005), (Wang et al., 2002) where the attacker may use the random spoofing source IP address and send out the same amount of SYN packets, and FIN/RST packets that can go unnoticed when compared with legitimate traffic flows. Moreover, discriminating flash crowd traffic from DDoS attack traffic is a major drawback of the proposed approaches.

An efficient approach (Peng et al., 2003) called a History-based IP Filtering (HIF) was proposed to discriminate good traffic from malicious traffic. This approach is based on monitoring the number of the new source IP addresses instead of the volume of the traffic. HIF keeps a history of the legitimate IP addresses that have appeared before and applies filters in the edge router based on this history. However, an adversary can bypass this mechanism by starting to send packets with its IP address prior to conducting the attack. Therefore, we need a more robust and

efficient identification mechanism for discriminating attack traffic while allowing legitimate traffic to apply during the attack period.

3 ATTACK IDENTIFICATION MECHANISM

In general DDoS attacks send large volumes of packets and consume critical resources in a network that makes the service unavailable for legitimate use. The attacker usually uses spoofed IP addresses in order to make a trace back more difficult, thereby thwarting the discovery of the real source of the attack. We scope our work on flooding attacks (e.g., spoofed/non spoofed UDP flood, ICMP flood, TCP SYN flood, DNS flood, VoIP flood, etc. (Peng et al., 2007), (RioRey, Inc., 2012) that focus on exhausting bandwidth of the victim's network. In this section, we propose our approach for characterizing and classifying normal and malicious traffic. In order to successfully respond to the attack, the approach must accurately detect the attacks and respond without penalizing legitimate traffic. It should also have low computation, communication, and storage overheads.

In order to continue to provide services under DDoS attacks, we need to distinguish attack traffic from legitimate traffic. An unusually high traffic volume may not be a good indicator of a DDoS attack. We need to consider other features that help distinguish DDoS attacks from normal traffic. Since many of the previous approaches depend on monitoring the volume of the traffic, detecting distributed flooding attacks is hard. During bandwidth attacks, most source IP addresses are new to the victim, whereas most IP addresses in a flash crowd have appeared at the victim before. Jung et al. (Jung et al., 2002) mentions that around 82.9% of all IP addresses involved in flash crowd events have sent prior requests. Peng et al. (Peng et al., 2003) advocate the use of network connection history to distinguish good packets from malicious ones. Many enterprises, such as universities and banks have a group of users that access their services on a regular basis and they have persistent characteristics. Although the user base fluctuates with new additions, deletions, etc., in general such a base changes at a much slower time scale compared to attacks and disruptions. These observations often form the basis of mechanisms that filter out the attack traffic at the victim. Examples include approaches that use the IP address history that have appeared at the victim node to distinguish between bad and good packet. However, the adversary can bypass this mechanism by starting to request and communicate with a victim node prior to conducting the attack. We address this problem by using multiple features that help distinguish normal from malicious traffic to generate accurate normal traffic signatures. Using this high confidence IP address history, we can defend against DDoS attacks that maximally preserve the service availability and minimize the attack impact.

3.1 Identification Features

In this section, we enumerate the features that we use to help distinguish attack and normal traffic. We use the parameters mentioned by Lee et al. (Lee et al., 2007) for detecting DDoS attacks. These include source/destination IP address, port number, and packet type (ICMP, TCP, UDP). In addition to these parameters we also use packet size as one feature. Recall that Jung et al (Jung et al., 2002) mentioned that most source IP addresses are new to the victim during bandwidth attacks but in flash crowd traffic previously viewed source IP addresses are most common. Thus, we also use the frequency of an IP address as a feature that may help distinguish an attack from normal traffic. Since we create a history based on legitimate and valid IP addresses, we consider only those IP addresses with a successful TCP handshake. Note that, a spoofed IP address will not have a complete three-way handshake (Peng et al., 2003). The attackers are therefore forced to use legitimate IPs and establish a three-way handshake. This limits the number of IP addresses that an attacker can use and the attack can be identified by monitoring for abrupt change in the traffic volume during the attack time. Our method uses a much more comprehensive set of features compared to existing identification approaches and we use them in an integrated manner to create filters as described in the next section.

3.2 Metrics

Our parameter set, denoted by $\mathcal{P}$, for establishing the model for historical IP addresses consists of the following features.

P_1: Source IP address

P_2: Port number

P_3: Size of packet

The number of features is denoted by K. In our model, $K = 3$. In addition, we also consider the packet type, namely, ICMP, UDP, and TCP. Thus, if the packet type is TCP, we require a three-way handshake to demonstrate legitimate traffic. For each parameter P_i, where $1 \leq i \leq K$, we maintain its frequency of

occurrence in a given time window and its cumulative distribution function (CDF). Let the parameter P_i taken on M different values. The frequency and CDF within a given time window is defined as follows.

Frequency Let f_{ij} be the number of packets for which the parameter P_i takes on the value of j in a given time window. Let N be the total number of packets. Thus,

$$N = \sum_j f_{ij}$$

Consider the case of the source IP address which corresponds to $i = 1$, there are 1000 different IP addresses within the time window making $M = 1000$. If there are a total of 5000 packets and a particular IP address j occur 30 times, then $f_{1j} = 30$ and $N = 5000$.

Cumulative Distribution Function (CDF) The CDF $C_X(x)$ measures the probability that the variable X takes on a value less than or equal to x. Now consider the parameter i. Let F_i be defined as the random variable representing f_{ij} for some j, CDF of F_i is given below.

$$C_{F_i}(x) = P(F_i \leq x)$$

For example, $C_{F_1}(10)$ demonstrates the probability that source IP address frequency is less than or equal to 10.

We use these metrics and features for generating the IP address history, which is addressed next.

3.3 History-based Profile Creation

Our goal is to define a good signature to make the IP address database accurate and robust, and to make it hard to be bypassed by an attacker. Our approach overcomes some of the deficiencies of existing approaches, such as use of only the IP address field as a feature useful for distinguishing attack and normal traffic. A key observation that can be used for defense against DDoS attack is that the DDoS attacks tend to use randomly spoofed IP addresses (Peng et al., 2007) and the other packet features, such as, port number and size of packet, are also selected randomly. Moreover, the interaction of these features also exhibits some anomaly when compared to that of the normal traffic. Therefore, we make use of the individual features, and also the interactions and correlations, in defining the signature. The signature is based on the CDF of each parameter's frequency during the training period. This signature assists us in selecting reliable IP addresses during the training period and later filters traffic based on those IP addresses. The signature for each feature determines which values occur more frequently during normal traffic conditions.

The next step is assigning scores to IP addresses in the training period and generating the source IP address history. The score value for each IP address depends on the frequencies and the signatures of the selected features. Frequency threshold α_l, which is a measure of reliability in our model, and the corresponding scores are presented in Table 1. For each selected feature value, if it is more than α_l in the related signature, it is assigned a score b_t indicating the confidence level. Here four levels are defined, denoted by b_1, b_2, b_3, and b_4, that assign different reliabilities to various IP addresses based on how frequently different feature values occur. In our model, α_l selected from 70%, 50%, and 30% indicates how the selected feature follows the signature of normal traffic conditions. This method allocates highest weight to the top 30% of the IP addresses ($i = 1$), port numbers ($i = 2$) and sizes of packets ($i = 3$) that occur most frequently. Note the value of frequency threshold α_l can be adjusted dynamically for each victim node based on the frequency and the signatures of selected feature for that point. Our selection of particular values for α_l is based on experience with different datasets, but it may be fine-tuned as needed. According to our model, when the selected feature occurs more frequently in normal traffic, as indicated by the signature of normal traffic, we give higher weight for it by assigning score b_4. In contrast, when α_l has value less than 30%, we assign the lowest confidence level factor b_1 to it. Low confidence measure indicates that such a feature value is not a common occurrence in normal traffic and thus dropping such packets will have less of an effect on the normal traffic. Frequent occurrence of such a value during an attack therefore is also considered as a potential threat.

For example, consider source IP address 129.1.1.1 with frequency 10000. When $C_{F_1}(10000) \geq 70\%$, it means according to IP address signature in training window 70% of IP addresses occur at frequency less than 10000. Thus, this is based on top 30% of occurrence rates of IP addresses and we give a high score for this feature. To summarize, the above procedure allocates weight from the set $b_1, ..., b_4$ to each IP address, each packet size, and each port number. Let the weight b_x, b_y, b_z be allocated for IP address β, port number j, packet size k respectively. We assign a net score S_β for IP address β. The equation below gives the formula for calculating S_β.

$$S_\beta = (f_{1\beta} * b_x + Max_{j,k}(f_{2j} * b_y + f_{3k} * b_z))/N \quad (1)$$

S_β is defined as the score value for each IP address β and is determined according to the frequency f_{ij} and confidence degree b_j in Eq. 1. Thus S_β consists of three components. For the above example, the

Table 1: The Score Manager.

Case	Frequency	Conclusion	Score
1	$\geq \alpha_1 (= 70\%)$	High Confidence	$b_4 = 4$
2	$\geq \alpha_2 (= 50\%)$	Medium	$b_3 = 3$
3	$\geq \alpha_3 (= 30\%)$	Low	$b_2 = 2$
4	$\prec \alpha_4 (= 30\%)$	Potential threat	$b_1 = 1$

first component of $S_{129.1.1.1}$ is computed for IP address 129.1.1.1 based on $f_{1129.1.1.1}$ and b_4 that is 4. The second and third components are selected by taking the maximum value of the sum of corresponding values for port number and size of packet over different packets with this IP address. The IP addresses that have an overall score S_β higher than a threshold ν are selected as legitimate IP addresses for our history. As noted, for TCP connections we consider one additional condition – only IP addresses with a successful TCP handshake are classified as valid. Therefore, for TCP connections, source IP addresses which have established a three-way handshake and have a score higher than ν are selected. This helps us create a signature-based IP address history and this history can be used through routers to perform filtering for the victim node.

3.4 Bloom Filter Mechanism

Bloom filter is a space efficient probabilistic data structure for checking whether an element is a member of a set or not. A Bloom filter (Wang et al., 2004) is a data structure that maps a set $S = \{s_1, s_2, ..., s_n\}$ of n elements into an an array of m bits, initially all is set to 0. A Bloom filter uses k independent hash functions $h_1, ..., h_k$ which when applied to each member of the set S returns a value in the range $\{1, ..., m\}$ (Mitzenmacher, 2002). For each element $s \in S$, the bits $h_i(s)$ in the array are set to 1 for $1 \leq i \leq k$. Note that, a position in the array can be set to 1 multiple times by the various hash functions, but only the first change has an effect. To check if $x \in S$, we check whether all $h_i(x)$ are set to 1. If at least one bit is not 1, then $x \notin S$. If all $h_i(x)$ are set to 1, we assume that $x \in S$, although there is a probability of a false positive, where it suggests that an element x is in S even though it is not. This is acceptable for many applications if the probability of a false positive is sufficiently small. The probability of a false positive for an element not in the set, or, the false positive rate, can be calculated in a straightforward fashion. After all the elements of S are hashed into the Bloom filter, the probability p that a specific bit is still 0 is given below.

$$p = (1 - \frac{1}{m})^{kn} \approx e^{-k\frac{n}{m}} \quad (2)$$

We let $p = e^{-k\frac{n}{m}}$. The probability of a false positive f is then:

$$f = (1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{-k\frac{n}{m}})^k = (1 - p)^k \quad (3)$$

Note that there are three fundamental performance metrics for Bloom filters: the probability of error (corresponding to the false positive rate f), size of the Bloom filter array (corresponding to the array size m) and the number of hash functions k. Bloom filters are highly efficient even if $m = cn$ for a small constant c. Although Bloom filters introduce false positives, their use is justified because of the reduction in the network traffic and overhead. In our analysis, we limit the false positive rate to 0.1. In order to achieve this false positive rate, we should set c to 5 in our mechanism. In other words, the Bloom filter array is 5 times the number of IP addresses kept in the history. Moreover, according to Eq. 3, we need 4 hash functions for the Bloom filter that will store the IP address history.

4 MODEL VALIDATION

Our objective is to evaluate the accuracy and robustness of our filter to protect against DDoS attacks. To illustrate the effect of history-based profile to distinguish attack and normal traffic we examined the CSU 2015 and CAIDA 2007 attack dataset. The CSU packet trace contains 4 week daily Argus files with flows on a 1Gb/s link from Feb 1st to Feb 28th 2015 where the total compressed data is about 20GB for each day (Impact Cyber Trust, 2015). CAIDA 2007 attack dataset contains approximately one hour of anonymized traffic traces from a DDoS attack on August 4, 2007 that contains 359,656,205 attack packets from 9,066 unique IP addresses (Center for Applied Internet Data Analysis, 2007).

4.1 Experiment Setup

Our approach consists of two steps as described below.

[Step 1:] Create IP address history from the training dataset using the generated signatures and the overall scores.

[Step 2:] Construct the Bloom filter using the IP address history created.

In the first step, the signatures for port number, the size of the packet, and IP address are created separately. For TCP traffic, IP addresses with successful TCP handshake are considered as valid IP address. For UDP traffic, we ignore those incoming packets that just send a single packet during the day. This helps us to create a more reliable signature-based IP address history. In this part of the experiment we set the length of the window to be one week. For each day we create a signature-based IP address history based on all IP addresses that come to CSU during this one week period. We build the IP address history using data trace from Feb 1st to Feb 21st and compare it to the trace taken from Feb 8th to Feb 28th. We will discuss about the effectiveness of history accuracy for different window size in Section 3.3. Figure 1 shows the signature of IP address's frequency for the first week of CSU traffic. According to this figure, IP address with an occurrence rate above 500 can achieve the highest score as $C_{F_1(500)} \geq 70\%$ and IP address with an occurrence rate below 20 get the lowest score. From Figure 1 we also see that around 10% of IP addresses have a frequency above 2000 and maximum frequency was around 12000 for a few IP addresses. Similar analysis exist for port number and packet size signature. The IP address history is created based on the overall score and value of threshold ν. In our experiment, ν is determined to be 0.36. In other words, the signature of at least two selected features should have α_l more than 50% and one should have it more than 30% in training time as per Table 1 and Eq. 1. However, this value can be fine-tuned as necessary. Determining the value of ν is a trade-off between history size, history accuracy, and how much protection we have against attack traffic. When we increase the threshold of ν, we are better able to filter attack traffic and the size of history decreases. However, increasing ν also blocks legitimate traffic and so the accuracy of the history for allowing legitimate traffic decreases.

4.2 Results

History-based profile algorithm is evaluated on the basis of these four parameters: (i) history accuracy, (ii) attack traffic detection rate, (iii) normal traffic detection rate, and (iv) size of Bloom filter. We performed a number of experiments to see the efficacy of our techniques. We experimented with using the entire IP address and the first three octets of the IP address to create the profile. We also investigated the impact of the window size. These results are reported in the following subsections.

4.2.1 Using First Three Octets of IP Address

In this section, we demonstrate how using first three octets of IP address instead of the entire IP address impacts the size of the Bloom filter and the history accuracy. The IP address history is created based on method described in Section 2.3. Our experiment was conducted based on the entire 4-octet IP address and also with the first three octets of IP addressed.

As shown in Table 2, the number of unique IP addresses that were retained in the history is around 3,000,000 for each day. By accepting 10% false positive rate the size of Bloom filter would be between 1.4MB to 2.2MB for each day as shown in Table 2 and Figure 2. Note that, as per equation of the Bloom filter, it is possible to set lower false positive rates but at the cost of increasing the size of the Bloom filter. We accept the false positive rate as 0.1 and $5n$ for the length of Bloom filter as reasonable values for error rate and Bloom filter size, where n is the number of inserted elements in Bloom filter, which are the number of IP addresses kept in the history.

Our objective is to provide a smaller Bloom filter while preserving the accuracy of the history. By creating the history with the first three octets of IP address we achieve significant reduction for the number of unique IP address to be kept in the history and consequently it decreases the size of Bloom filter as shown in Table 2 and Figure 2. The number of unique addresses in history drop down to around 1,500,000 by using the first three octets of IP address and it shows about a 50% reduction of total IP addresses needed to keep in the history. As a result, the size of the Bloom filter reduces to around 0.9 MB for each day as shown in Figure 2. Furthermore, the result shows that at the same time the accuracy of IP address history increases due to this change.

The next step is to evaluate the accuracy of the history, that is, the percentage of traffic for each day that have already appeared in the history. As shown in Figure 3, the history accuracy is about 60% to 80% for each day and this confirms that most of the IP addresses that appear in the CSU network under normal conditions have previously visited and follow the signature of each day's network traffic. The highest history accuracy is for Feb 21st with 80% and the lowest one is for Feb 12th with 60%. Table 3 shows the number of unique IP addresses of packets visiting CSU and the number of IP addresses that match with history for each corresponding day. For instance, on Feb 8th the total number of unique IP addresses on packets coming to CSU is 2,709,558 and 1,790,443 of those IP addresses match with the history. We also show the effectiveness of using the first three octets of IP address in Figure 3. In this case, the accuracy of IP

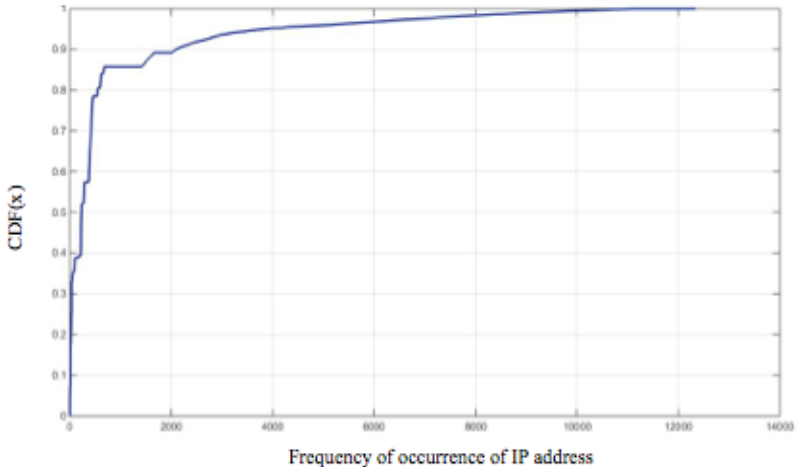

Figure 1: Frequency of Occurrence of IP Addresses.

Table 2: Number of IP Addresses in History.

Date	Unique IP addresses	Bloom filter size (MB)	Unique first three-octets of IP addresses	Bloom filter size (MB)
8-Feb	3567433	2.22	1549329	0.96
9-Feb	2898726	1.81	1540879	0.96
10-Feb	2993910	1.87	1533835	0.95
11-Feb	3251485	2.03	1573583	0.98
12-Feb	3253085	2.03	1542783	0.96
13-Feb	3214944	2.00	1543112	0.96
14-Feb	3367433	2.10	1506123	0.94
15-Feb	3130830	1.95	1478050	0.92
16-Feb	2986955	1.86	1488892	0.93
17-Feb	3077742	1.92	1465505	0.91
18-Feb	3181456	1.98	1493287	0.93
19-Feb	3086945	1.92	1461922	0.91
20-Feb	3131923	1.95	1486039	0.92
21-Feb	3077169	1.92	1484376	0.92
22-Feb	2957518	1.84	1450000	0.90
23-Feb	3181719	1.98	1522926	0.95
24-Feb	3095821	1.93	1562730	0.97
25-Feb	3095821	1.93	1510018	0.94
26-Feb	2940737	1.83	1541806	0.96
27-Feb	2915739	1.82	1547367	0.96
28-Feb	2303440	1.43	1440457	0.90

address history improves to around 75% for most of the days while the size of Bloom filter reduces to 50% as shown in Figure 2.

The other important parameter to evaluate is how much of the traffic volumes can pass through the Bloom filter and reach the end nodes. As shown in Figure 4, the normal traffic detection rate varies from 68% to 82% using the entire IP address. This demonstrates that the performance of signature-based IP address history is highly reliable for identifying normal traffic and withholding insignificant amount of legitimate traffic. Normal traffic detection rate increases from 80% to 90% if the first three octets of IP addresses is used to generate IP address history. In this part we investigate the effectiveness of history accuracy and measure how many of packet traffic can pass through the Bloom filter. The results are shown in Figure 6.

The x-axis is the traffic volume based on number of packets. From this figure we observe the total number of traffic received is varying from 160 million to 220 million packets in each day. We denote the external traffic also increase suddenly on Feb 10th and Feb 16th where the portion of normal traffic that can pass

Figure 2: Size of the Bloom Filter.

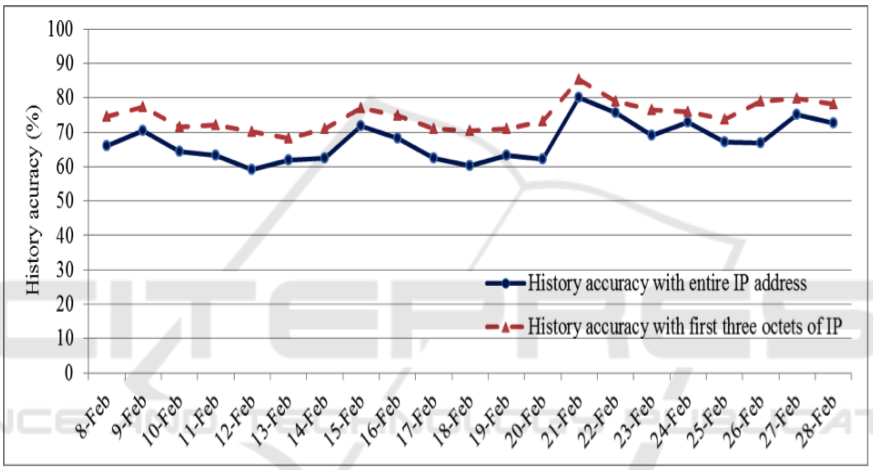

Figure 3: History Accuracy.

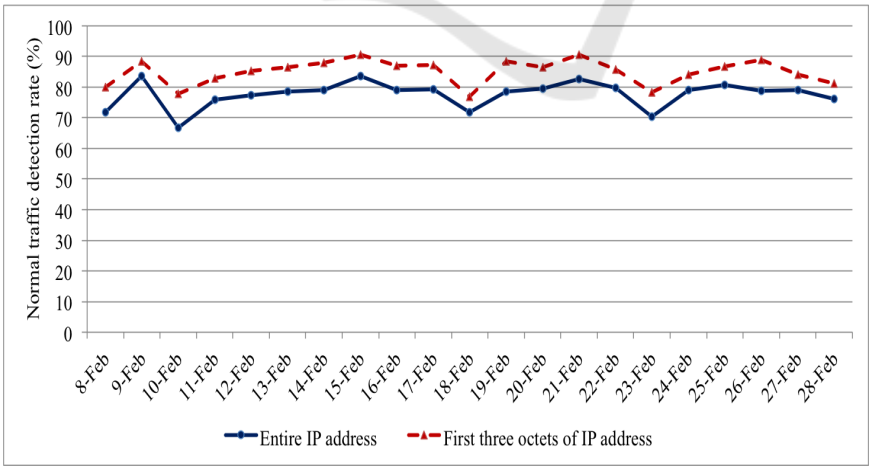

Figure 4: Normal Traffic Detection Rate.

the history does not change too much. It means for those days part of the incoming traffic to CSU does not appear before and it was not included in the history. For almost all the other days the normal traffic that can pass the history has similar behavior. Furthermore, the result shows the signature-based IP address

Table 3: Number of Unique IP Addresses of Packets Arriving to CSU and Their Match with History.

Date	IP match	Total IP in history
8-Feb	1790443	2709558
9-Feb	1627396	2309466
10-Feb	1898679	3009246
11-Feb	1981443	3129533
12-Feb	1957757	3316582
13-Feb	1984888	3201160
14-Feb	1995127	3191604
15-Feb	1750964	2440758
16-Feb	1775144	2598016
17-Feb	1998038	3206285
18-Feb	1989359	3309144
19-Feb	1808770	2864580
20-Feb	3131923	2900503
21-Feb	1684604	2370058
22-Feb	1335560	1767686
23-Feb	1362394	1698466
24-Feb	1424731	1957384
25-Feb	1654167	2459654
26-Feb	1412483	2115752
27-Feb	1652258	2197652
28-Feb	1395530	1922479

history with the first three octets can perform better than the history with entire IP address as expected. We can therefore deduce that by using the first three octets of IP address history, the normal traffic detection rate increases. However, there is a tradeoff between the protection rate and the normal traffic detection rate. When we are using the first three octets of IP address there is some possibility that malicious traffic also share the first three octets of IP addresses that exist in the history and the signature-based IP address history could not filter them. Therefore the protection rate reduces by using three octets of IP address. On the other hand by using first three octets of IP address the size of Bloom filter reduces and normal traffic detection rate increases. Therefore, selecting an appropriate approach is based on the size of Bloom filter, history accuracy and protection rate. In general, using the first three octets of IP address gives better history accuracy, normal detection rate, and it reduces the size of Bloom filter which results in decrease of protection rate. The other important result of this experiment is the attack detection rate shown in Fig. 5. CAIDA attack 2007 dataset is used to evaluate this step. The result shows the Bloom filter with first three octets can perform approximately as well as the Bloom filter with the entire IP addresses. According to Fig. 5, we observe almost 95% success with attack detection rate. In fact our experiment shows that 70% of legitimate traffic can be preserved while filtering out 95% of attack traffic.

4.2.2 Window Size

We studied the impact of window size by creating history for four different window sizes and measuring the history accuracy. We set the window size to 14, 7, 10 and 3 days respectively to create the history and then calculated history accuracy during the period from Feb 15th to Feb 28th as shown in Figure 7. The number of unique IP addresses that appeared in the history is shown in Table 4. Figure 7 shows that the history accuracy improves as long as we increase the length of window size from 3 to 14 days; however, the interesting observation is that history accuracy does not change significantly after 10 days. In other words, most of the legitimate IP addresses have appeared in the history in the past 10 days and very few other IP addresses have visited CSU only prior to 10 days. The other observation in Figure 7 is that the largest gap is between window size 3 and 7 days. The history accuracy is around 50% for 3 days window size while this value improves to 65% for 7 days. In fact this demonstrates that 3 days window size is too short a period to create a history with high accuracy.

The other important parameter that impacts history accuracy is size of the Bloom filter. As shown in Table 4 and Figure 8, enlarging the window size increases the size of history and also improves the history accuracy. According to Figure 8, size of history increases about 0.7MB in average when we change the window size from 10 days to 14 days but as shown in Figure 7 history accuracy is almost similar. Consequently, in choosing between 10 days and 14 days

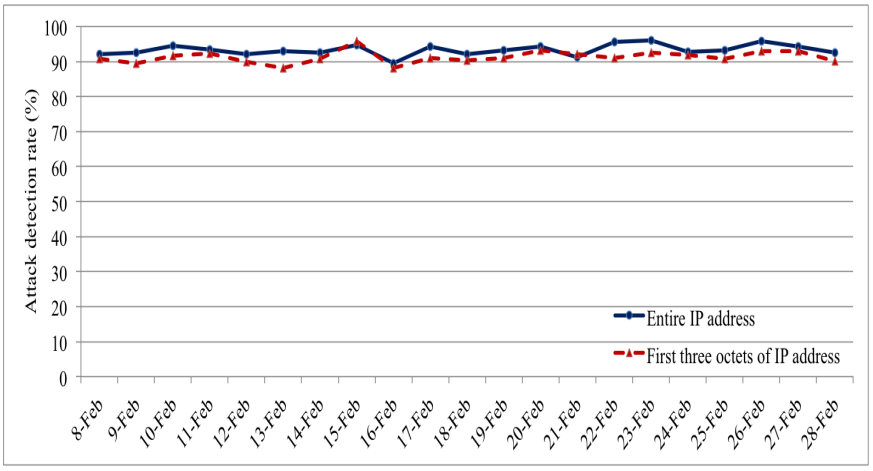

Figure 5: Attack Traffic Detection Rate.

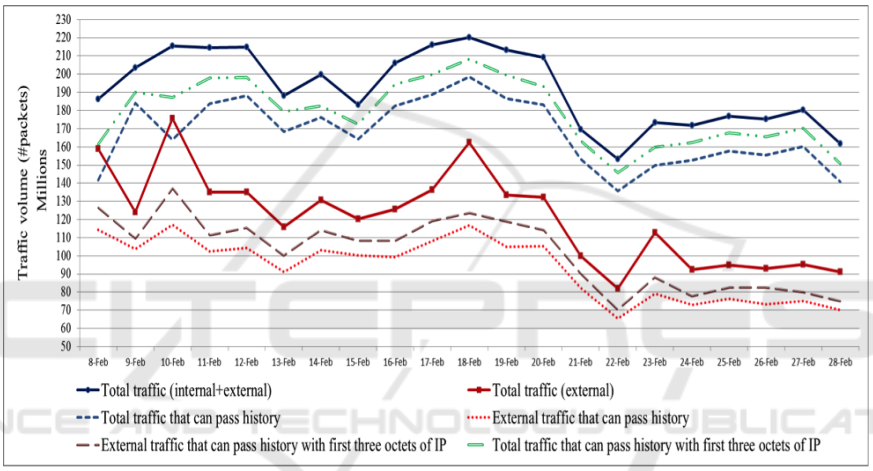

Figure 6: Traffic Volume and Number of Packets that Can Pass History.

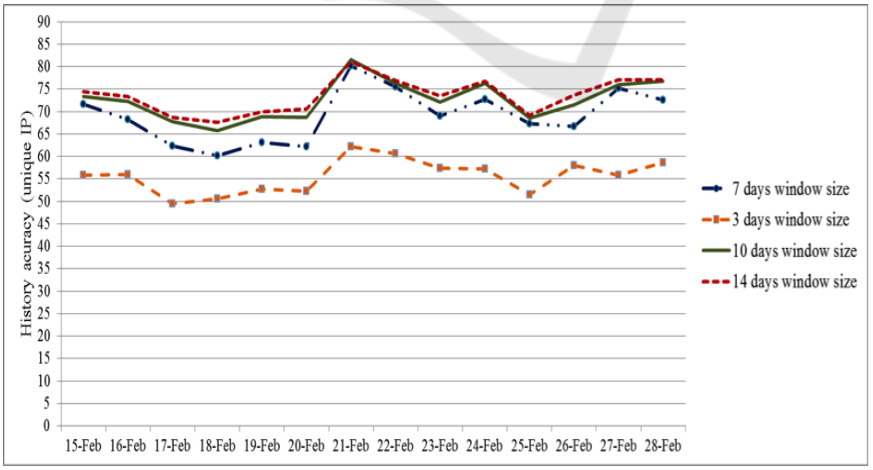

Figure 7: History Accuracy for Different Window Sizes.

window size, 10 days window size would be a more appropriate selection. Furthermore, from Figure 8 we also see that size of Bloom filter reduces by 1MB in average when the window size changes from 10 days to 7 days but the history accuracy decreases by less than 5% because of this reduction. Thus, the 7 days

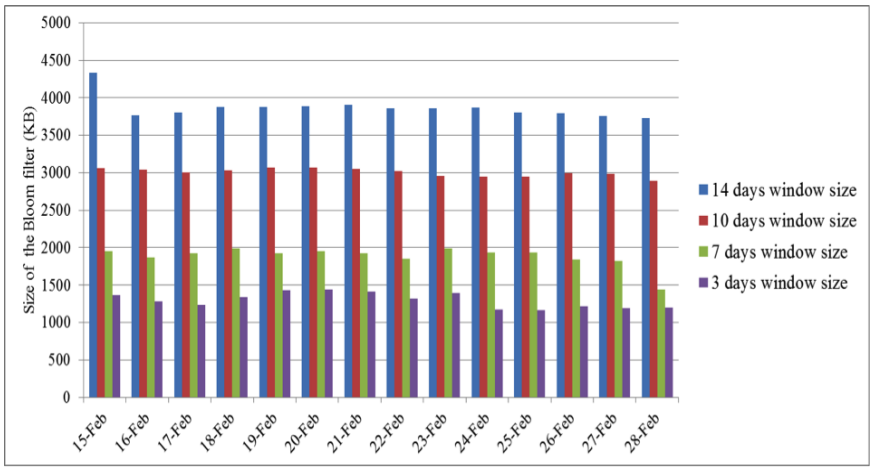

Figure 8: Size of Bloom Filter for Different Window Sizes.

Table 4: Number of Unique IP Addresses in History with Different Window Sizes.

Window size	14 days	10 days	7 days	3 days
15-Feb	6930826	4889103	3130830	2189643
16-Feb	6019868	4858096	2986955	2047126
17-Feb	6080193	4803532	3077742	1987207
18-Feb	6206424	4853005	3181456	2141480
19-Feb	6198630	4910655	3086945	2289896
20-Feb	6224286	4911569	3131923	2308403
21-Feb	6244686	4887286	3077169	2259784
22-Feb	6178677	4832357	2957518	2118029
23-Feb	6166982	4735330	3181719	2239626
24-Feb	6192012	4714798	3095821	1874455
25-Feb	6091122	4717731	3095821	1865488
26-Feb	6069075	4785940	2940737	1944892
27-Feb	6007085	4775288	2915739	1911688
28-Feb	5969252	4628090	2303440	1917622

window size is the best choice where we can get good history accuracy with a reasonable size Bloom filter.

5 CONCLUSION

We provide a solution for defending against DDoS attacks that looks at the history and uses a rich set of header fields for discriminating attack from normal traffic. We demonstrated how such a differentiation can be done using a Bloom filter. The use of Bloom filter improves efficiency and reduces memory requirement in upstream routers during the attack time. The efficacy of our approach is validated using a recent dataset from CSU 2015. Our experiment indicates that our filtering model can protect the victim node from 95% of attack traffic while allowing 70% of normal legitimate traffic with a 2MB Bloom filter. In addition, we show that the size of the Bloom filter can be reduced significantly by creating an IP address history based on the first three octets of IP address where we present high history accuracy and attack detection rate. In short, we show very good results in successfully characterizing the network traffic and preserving good traffic with appropriate filtering size. The experiment results verify that our signature-based mechanism can be deployed in real networks. Our future work includes extending our scheme for IPv6. In contrast to IPv4, IPv6 has a vastly enlarged address space that makes it more difficult to address the specific characteristics related to identifying and filtering the attack traffic.

REFERENCES

Aghaei Foroushani, Z. H. (2014). TDFA: Traceback-based defense against DDoS flooding attacks. In *Proc. of AINA IEEE*, pages 710–715.

Center for Applied Internet Data Analysis (2007). The CAIDA "DDoS Attack 2007" dataset. http://www.caida.org/data/passive/ddos-20070804-dataset.xml. accessed 7-May-2016.

Chen, C. and Park, J.-M. (2005). Attack diagnosis: Throttling distributed denial-of-service attacks close to the attack sources. In *Proc. of IEEE ICCCN*, pages 275–280.

Chen, Y., Hwang, K., and Ku, W.-S. (2007). Collaborative detection of DDoS attacks over multiple network domains. *IEEE Trans. Parallel Distrib. Syst.*, 18(12):1649–1662.

Francois, J., Aib, I., and Boutaba, R. (2012). FireCol: A collaborative protection network for the detection of flooding DDoS attacks. *IEEE/ACM TON*, 20(6):1828–1841.

Gil, T. M. and Poletto, T. (2001). MULTOPS: A data-structure for bandwidth attack detection. In *IProc. of USENIX Security Symposium*.

Impact Cyber Trust (2015). Colorado state university dataset: "FRGPContinuousFlowData". https://www.impactcybertrust.org/dataset_view?idDataset=260. accessed 26-Oct-2016.

Ioannidis, J. and Bellovin, S. (2002). Implementing pushback: Router-based defense against DDoS attacks. In *Proc. of NDSS*.

Jung, J., Krishnamurthy, B., and Rabinovich, M. (2002). Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *Proc. of WWW conf.*, pages 293–304.

Kim, Y., Lau, W., Chuah, M., and *et al*. (2004). PacketScore: Statistics-based overload control against distributed denial of service attacks. In *Proc. of INFOCOM*, pages 141–155.

Lee, K., Kim, J., Kwon, K. H., and *et al*. (2007). DDoS attack detection method using cluster analysis. *Expert Syst. with Applications*, 34(3):1659–1665.

Mahajan, R., Bellovin, S. M., Floyd, S., and *et al*. (2002). Controlling high bandwidth aggregates in the network. *ACM SIGCOMM*, 32(3):62–73.

Manikopoulos, C. and Papavassiliou, S. (2002). Network intrusion and fault detection: A statistical anomaly approach. *IEEE Comm. Magazine*, 40(10):76–82.

Mathew J. Schwartz (2014). DDoS Attack hits 400 Gbit/s, Breaks Record. http://www.darkreading.com/attacks-and-breaches/ddos-attack-hits-400-gbit-s-breaks-record/d/d-id/1113787. accessed 2-Nov-2014.

Mirkovic, J., Prier, G., and Reiher, P. L. (2002). Attacking DDoS at the source. In *Proc. of IEEE ICNP*, pages 312–321.

Mirkovic, J. and Reiher, P. (2005). D-WARD: A Source-end defense against flooding denial-of-service attacks. *IEEE TDSC*, 2(3):216–232.

Mitzenmacher, M. (2002). Compressed bloom filters. *IEEE/ACM TON*, 10(5):604–612.

Munivara Prasad, K., Rama Mohan Reddy, A., and Venugopal Rao, K. (2014). DoS and DDoS attacks: Defense, detection and traceback mechanisms - a survey. *Journal of JCST*, 14(7-E):15–32.

Noh, S., Jung, G., Choi, K., and *et al*. (2008). Compiling network traffic into rules using soft computing methods for the detection of flooding attacks. *Journal of Applied Soft Comp.*, 8(3):1200–1210.

Papadopoulos, C., Lindell, R., Mehringer, J., and *et al*. (2003). COSSACK: Coordinated suppression of simultaneous attacks. In *Proc. of Discex III*, pages 94–96.

Peng, T., Leckie, C., and Ramamohanarao, K. (2003). Protection from distributed denial of service attack using history-based IP filtering. In *Proc. of IEEE ICC*, pages 482–486.

Peng, T., Leckie, C., and Ramamohanarao, K. (2004). Detecting distributed denial of service attacks using source IP address monitoring. In *Proc. of NETWORKS*.

Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, 39(1):1–42.

RioRey, Inc. (2012). Taxonomy DDoS attacks. http://www.riorey.com/xresources/2012/RioRe. accessed 24-Dec-2015.

Steinberger, J., Sperotto, A., and Baier, H. (2005). Collaborative attack mitigation and response: A survey. In *IFIP/IEEE Int. Symp. on IM*, pages 910–913.

Sung, M. and Xu, J. (2003). IP traceback-based intelligent packet filtering: A novel technique for defending against internet DDoS attacks. *IEEE TPDS*, 14(9):861–872.

Waikato Applied Network Dynamics Research Group (2016). Auckland university data traces. http://wand.cs.waikato.ac.nz/wand/wits/. accessed 12-March-2016.

Wang, H., Jin, C., and Shin, K. (2007). Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Trans. on Networking*, 15(1):40–53.

Wang, H., Zhang, D., and Shin, K. (2002). Detecting SYN flooding attacks. In *Proc of IEEE INFOCOM*, pages 530–1539.

Wang, H., Zhang, D., and Shin, K. (2004). Change-Point monitoring for the detection of DoS attacks. *IEEE TDSC*, 1(4):193–208.

Yaar, Y., Perrig, A., and Song, D. (2003). Pi: A path identification mechanism to defend against DDoS attacks. In *Proc. of IEEE S&P*, pages 93–107.

Yaar, Y., Perrig, A., and Song, D. (2004). SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks. In *Proc. of IEEE S&P*, pages 130–143.