This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Jing, Xuyang; Zhao, Jingjing; Zheng, Qinghua; Yan, Zheng; Pedrycz, Witold
A reversible sketch-based method for detecting and mitigating amplification attacks

Published in:
Journal of Network and Computer Applications

DOI:
[10.1016/j.jnca.2019.06.007](https://doi.org/10.1016/j.jnca.2019.06.007)

Published: 15/09/2019

Document Version
Peer reviewed version

Published under the following license:
CC BY-NC-ND

Please cite the original version:
Jing, X., Zhao, J., Zheng, Q., Yan, Z., & Pedrycz, W. (2019). A reversible sketch-based method for detecting and mitigating amplification attacks. *Journal of Network and Computer Applications*, 142, 15-24.
<https://doi.org/10.1016/j.jnca.2019.06.007>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

A Reversible Sketch-based Method for Detecting and Mitigating Amplification Attacks

Xuyang Jing¹, Jingjing Zhao¹, Qinghua Zheng², Zheng Yan^{1,3*}, Witold Pedrycz⁴

¹State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China

²Department of Computer Science and Technology, Xi'an Jiaotong University, China

³Department of Communications and Networking, Aalto University, Finland

⁴Department of Electrical and Computer Engineering, University of Alberta, Canada

Abstract Amplification attacks bring serious threats to network security due to their characteristics of anonymity and amplification. How to detect amplification attacks attracts more and more attention. However, as the age of networking for big data is coming, traditional amplification attack detection methods become inefficient due to the impact of big-volume network traffic that swamp significant signals of attacks. The premise of accurate effective attack detection is efficiently processing big-volume traffic. In this paper, we propose a meaningful work that applies sketch technique to detect and mitigate amplification attacks. This step enables the detection method to handle big-volume network traffic. We use a Chinese Remainder Theorem based Reversible Sketch to directly collect network traffic and then monitor the abrupt changes in one-to-one mapping between request packets and response packets to identify amplification attack traffic. The detection mechanism is robust and efficient since it does not need to record complicated traffic features and makes full use of the basic characteristic of amplification attacks. We examine the performance of our method through a series of experiments conducted on simulation and real world traffic. The results denote that the method can accurately detect and mitigate amplification attacks.

Keywords: Amplification attacks; Reversible Sketch; Bloom Filter; Chinese Remainder Theorem; Network security

1. Introduction

Distributed Reflection Denial of Service (DRDoS) attacks belong to the type of indirect Distributed Denial of Service (DDoS) flooding attacks have gained popularity and became serious threats to the Internet [1-3, 40-42]. For launching DRDoS flooding attacks, an attacker utilizes infected hosts to transmit request packets that spoof the source address as the address of victim to reflectors. Usually, the reflectors are servers, such as Domain Name System (DNS) server, Network Time Protocol (NTP) server and Trivial File Transfer Protocol (TFTP) server. The reflectors respond to the request packets and send the response packets directly to the intended victim. The large number of response packets drastically consumes victim's system resources and affect its normal network communications. Usually, we also call DRDoS as amplification attacks, since the number/size of response packets send from reflectors is many times larger than the number/size of request packets send from infected hosts [4]. Two unique properties make amplification attacks so popular: (i) anonymity: the location of an attacker can be hidden by employing spoofed source IP addresses, (ii) amplification: the volume of attack traffic

can be easily amplified with the help of the unsymmetrical size of response packets and a botnet. *Cloudflare*, which is the famous service providers in DDoS protection, states that the amplification is becoming more serious in DDoS flooding attacks [5].

Detecting amplification attacks can be realized at reflector-end and victim-end [1, 6]. At the reflector-end, the detection of spoofed addresses is the key technique against amplification attacks [7-9, 39]. The similar characteristics (e.g., packet sizes, packet rate, packet inter-arrival time) among spoofing request packets can also be utilized to detect amplification attacks [14]. But reflector-end detection methods have two drawbacks: (i) the large number of potential reflectors makes detection methods difficult to deploy in practice, and (ii) illegitimate requests with spoofed addresses are similar to legitimate requests. At the victim-end, the most effective detection method is to analyze the unbalanced relationship between outgoing request packets and incoming response packets [10, 11]. Moreover, traffic generated by reflectors may have sufficient similarity that it can be filtered out near the victim [12]. However, the ever-increasing network traffic poses great challenges on processing big-volume network traffic due to both computational overhead and memory requirements.

It is difficult to deploy an detection method for amplification attacks at reflector-end since numerous potential reflectors (legitimate servers) are deployed in the network. The detection methods for amplification attacks at victim-end are easier to deploy and apply than the detection methods at reflector-end under the premise of handling big-volume network traffic. However, a few of existing works have considered the impact of massive volume traffic on the performance of detection methods, especially efficiency. An efficient amplification attack detection method should be able to handle big-volume traffic, and also ensure a high accuracy. If not, the detection method will become infeasible or even useless.

With the purpose of extremely alleviating the impact of big-volume traffic on the capability of attack detection, this paper proposes a detection and mitigation method for amplification attacks by employing sketches, which are compact and constant-size data structures used to summary network traffic information. Our aim is to effectively collect network traffic, and thus we adopt a reversible sketch proposed in our previous work, called Chinese Remainder Theorem based Reversible Sketch (CRT-RS) [13]. Based on the traffic information recorded by CRT-RS, we monitor the abrupt changes in one-to-one mapping between request packets and response packets to detect amplification attacks. Such a mapping exists in many protocols, which are commonly used to conduct amplification attacks, such as DNS protocol, NTP protocol, Simple Network Management Protocol (SNMP) protocol, etc. Under normal network communications, the number of request packets that the host sends out equals to the number of response packets that the host receives. When an amplification attack happens, the victim will receive massive response packets without the corresponding request packets send out in previous. As a result, this unbalanced relationship must be immediately considered as suspicious. After the abnormal buckets in each row in CRT-RS are found out, the malicious source addresses of response packets are reversely calculated and added into a blacklist. A bloom filter that operated on blacklist is used to look up whether the source address of incoming traffic exists in the blacklist and if so, the traffic is filtered. By combining sketch technique and the basic characteristics of amplification attacks to perform detection, our method can process big-volume network traffic, support protocol independence, and detect different types of amplification attacks. We summary the major contributions of our paper as follows:

- We employ CRT-RS to effectively collect network traffic in a distributed manner and reversely reconstruct anomalous source addresses. This step can efficiently process big-volume network traffic at victim-end.
- We design an efficient and reliable detection method to detect amplification attacks that supports protocol independent detection based on the traffic information collected by CRT-RS. Our method is less complicated and can accurately detect and effectively mitigate amplification attacks since it does not need to record complicated traffic features and manage status information of network traffic. Moreover, the method is not sensitive to detection interval.
- We examine the performance of our method using both NS3 simulation and real world traffic with the purpose of showing its high advantages of *accuracy*, *protocol independability*, and *efficiency*.

The paper is structured as follows. Section 2 offers a brief review of related work. We introduce some preliminaries in Section 3. In section 4, we first describe CRT-RS and then elaborate on an overall process of the attack detection and mitigation method. We show experimental results in Section 5. Conclusions are offered in Section 6.

2. Related Work

Amplification attacks have become a serious problem in network security. Many excellent methods have been proposed for detecting amplification attacks. We give a review on existing related work about amplification attack detection performed at reflector-end and victim-end. Moreover, we take a comparison of our method with others in order to indicate the advantages.

Detect Amplification Attacks at Reflector-End: Cai et al. [14] used K-means algorithm to build a traffic pattern by clustering traffic into normal and abnormal categories. Three traffic features are selected, namely the number of DNS query packets, the size ratio of response packet to query packet and the number ratio of DNS response packets to the DNS query packets. The newly arrived datum is analyzed based on its distance to the central point of the clusters.

Meitei et al. [15] employed Decision Tree, Multilayer Perceptron, Naive Bayes and Support Vector Machine (SVM) to classify DNS request traffic that based on the following several traffic features such as the mean of packet arrival time from the same IP, the statistical information of packet sizes, and the probability of IP occurrence, etc. Attribute selection algorithms were also used to reduce redundant features.

If reflectors are able to detect whether the source addresses of request packets has been spoofed, it would be very easy to detect and prevent amplification attacks. Rudman and Irwin [16] found that the changes of source address and Time to Live (TTL) values can be used to detect NTP amplification attacks by analyzing two large datasets. If TTL value of the packets that come from the same address changes obviously, it indicates that the source address is the target of attack. This is because the attacking hosts are commonly existed in different subnets, there must be a significant difference between the final TTL values of the packets transmit to the same victim address.

Bottger et al. [17] used three traffic features, namely the size of request and response packet, the number of Internet Control Message Protocol (ICMP) port unreachable packets and TTL values to characterize amplification attacks and employed the features to identify spoofed attack requests from legitimate requests. The experimental results

showed that the protocol-agnostic method is effective against the attacks on static port numbers and also new amplification attacks.

Huistra [18] observed that it is feasible to detect DNS amplification attacks by analyzing the number and the size of DNS packets at reflector-end. They determined whether this flow is suspicious or normal by directly comparing the size of the DNS request inflow with a predefined threshold. In addition, they also compared the size of DNS response outflow and the average size of the response packets with a threshold as a second metric.

But these methods do not consider the problem of deployment and big-volume network traffic, which is well solved in this paper.

Detect Amplification Attacks at Victim-End: Wei et al. [12] pointed out that response flows come from the same reflectors have a linear relationship with others. They used the flows that consist of the packets to a victim through one router to measure the correlation coefficient between flow pairs by utilizing Spearman's rank correlation coefficient to distinguish amplification attack traffic from legitimate traffic.

Gao et al. [19] stated that the following features can be considered as attack indicators of DRDoS flooding attacks, such as the number of IP packets, the sum sizes of packets, the maximum number of the packets among all ports, and the difference between the packets received and sent out by the victim. Based on abrupt changes of the above features, they used a normalized polynomial kernel based Support Vector Machine to identify the attacking traffic.

Kambourakis et al. [11] built a four-column DNS request table that consist of source IP address, source port, destination IP address, and destination port. Similarly, they built a DNS response table with the same structure as request table based on the traffic information from the DNS response packets received by the victim. Through comparing these two tables, the response packet is considered to be a suspicious packet if there is no matched request packet that the victim previously sent within a given period. An alarm will be generated once the number of suspicious packets over a preset threshold.

However, the ever-increasing network traffic has a great impact on amplification detection methods deployed at victim-end, which will impact their accuracy and efficiency.

Sketch: Sketches employ data-oriented hashing technique for network traffic information summarization [20]. There are some popular existing sketches that have different update and estimation process, such as K -ary Sketch [21], Count Sketch [22], Count-Min (CM) Sketch [23], Augmented Sketch [24] and Pyramid Sketch [25].

Table I. Comparison of our Detection Method with other Existing Methods Deployed at Victim-end

Reference	Handle big-volume traffic	Identify malicious addresses	Protocol independability	Experiment environment
[10]	No	No	Yes	Real world traffic
[11]	No	No	No	Real world traffic
[12]	No	No	Yes	Simulation
[19]	No	No	Yes	Simulation
Our method	Yes	Yes	Yes	Simulation and real world traffic

A tabulated comparison of our detection and mitigation method with other existing methods for amplification attacks at victim-end is presented in Table I. Existing studies have not considered the impact of big-volume traffic on the performance of amplification

attack detection. However, newly proposed should have the ability of processing the ever-increasing network traffic in the era of networking for big data [43]. Otherwise, the detection method will become inefficient or even cannot operate once it is overwhelmed by huge network traffic. Moreover, identifying malicious source addresses is practically requested in attack mitigation stage. With the purpose of addressing the problem introduced here and in Section 1, we apply a scalable and distributed reversible sketch technology to firstly process and compress network traffic. Then based on the network traffic records that are generated with the sketch technology, the detection becomes more effective and accurate than existing methods as well. From Table I, we can see the advantages of our proposed method in terms of big-volume traffic handling, malicious source address tracking, protocol independability support and performance evaluation.

3. Preliminaries

In this section, we introduce the amplification attacks and the basic knowledge of sketch and bloom filter.

3.1. Amplification Attacks

Fig. 1. An illustration of amplification attack.

The detailed brief process of amplification attacks is shown in Fig. 1. An attacker commands bots to transmit spoofed request packets that set source address as the victim's address to reflectors. The reflectors reply the request with response packets and naturally send them to the victim. As a result, the victim is overwhelmed by reflected response packets. The attacker usually selects the protocols of TCP/IP model that support request-response relationships to launch amplification attacks. Typical types of request-response relationships are summarized in Table II [10]. Herein, we briefly introduce some popular types of amplification attacks.

(1) DNS amplification attack: DNS servers offer two kinds of reflection. The first is a DNS server answers DNS reply packets corresponding to the DNS request packets came from bots to the spoofed address. The second is DNS servers recursively query the victim which is a name server for a specified area to resolve DNS requests issued by an attacker. The DNS amplification attack is prevalent over the Internet due to a small DNS request packet could trigger a much larger DNS response packet [9, 26, 27].

(2) TFTP amplification attack: In a TFTP amplification attack, an attacker sends a specially adjusted TFTP Read Request packet with a spoofed source IP address for the file on a TFTP server, which generates a larger response packets directed to the victim [28].

(3) SNMP amplification attack: During a SNMP amplification attack, an attacker sends an SNMP request packet (such as GetBulkRequest) with a spoofed source IP address of the victim to a host, which has vulnerabilities in SNMP service [29]. The

Table II. A Summary of Request-Response Relationship

Request	Response
TCP SYN packets to open TCP ports	TCP SYN/ACK packets
DNS request packets	DNS response packets
NTP monlist request packets	NTP monlist response packets
SNMP request packets	SNMP response packets
TFTP request packets	TFTP response packets
ICMP request packets	ICMP response packets
UDP packets to closed TCP ports	ICMP port unreachable packets
TCP packets to closed TCP ports	TCP RST packets

Table III. The *Bandwidth Amplification Factors* of Some Known Protocols

Protocols	Amplification Factors	Vulnerable Commands
NTP	556.9	Monlist request
CharGEN	358.8	Character generation request
QOTD	140.3	Quote request
RIPv1	131.24	Malformed request
Quake Network Protocol	63.9	Server information exchange request
LDAP	46 to 55	Malformed request
DNS	28 to 54	Resolution request
SSDP	30.8	SEARCH request
Portmap	7 to 28	Malformed request
Kad	16.3	Peer list exchange request
Multicast DNS	2 to 10	Unicast query request
SNMP	6.3	GetBulk request
Steam protocol	5.5	Server information exchange request
BitTorrent	3.8	File search request
NetBIOS	3.8	Name resolution request

response packet that includes a list of SNMP identifiers generated from the host will be larger than the request packet and directed to the victim.

(4) ICMP amplification attack: The most popular way to elicit ICMP reflector replies is to utilize ICMP protocols designed as request/response (such as ICMP echo). An attacker sends an ICMP Echo request packet with the spoofed source IP address of the victim to the directed-broadcast address of a specific network. The packet is then

broadcast throughout the network. All the devices that receive the request packet send corresponding response packets to the source IP address of the packet (i.e., victim IP address). Therefore, a deliberately designed ICMP request packet could generate large amounts of attack traffic. Nowadays, ICMP amplification attack is always not feasible as routers block ICMP request packets sent to a broadcast address [4].

There are two amplification factors measure the impact of amplification attacks: *Packet Amplification Factor (PAF)* that denotes the number ratio of response packets sent to victim to the request packets sent to reflectors and *Bandwidth Amplification Factor (BAF)* that denotes the size ratio of the response packets sent to victim to the request packets sent to reflectors. Through a series of experiments, Rossow [5] and Kuhrer et al. [30] evaluated amplification potential of some vulnerable protocols that can be easily utilized to launch amplification attacks. They conducted a series of experiments to show that amplification attacks usually have high amplification factors when exploiting UDP protocols. A list of *BAF* of some known protocols is shown in Table III [31].

3.2. Sketch and Bloom Filter

Fig. 2 shows the structure of sketch that consists of H hash tables with the size of M . Sketch models incoming network traffic into $(key, value)$ format, where the *key* is selected as the attributes in packet header, and the *value* is selected as accumulative traffic information, such as the sum of packet sizes, the number of packets, etc. Each hash function h_i ($i=1, 2, \dots, H$) maps the incoming *keys* into the range of $(1, 2, \dots, M)$ to locate corresponding columns used to store the *values* in each row. An example is shown in Fig. 2, the *key* is hashed by $\{h_1, h_2, \dots, h_H\}$ and the *value* is added to the corresponding bucket in each row.

Fig. 2. A Diagram of a Sketch.

As shown in Fig. 3, a bloom filter is made up of an array of m bits and K independent hash functions $\{h_1, h_2, \dots, h_K\}$ with a space of $(0, 1, 2, \dots, m-1)$. The bits that stored the information of element x at the bucket denoted as $h_i(x)$ ($i=1, 2, \dots, K$) are set to 1. To look up whether the information of another element y has been stored in the bloom filter, we need to check the bits at the buckets denoted as $h_i(y)$ ($i=1, 2, \dots, K$). If all checked bits are equal to 1, the information of element y has been stored in the bloom filter S . As indicted by Fig. 3, y has not been stored because there exists some bits equal 0.

Fig. 3. A Diagram of a Bloom Filter.

4. Amplification Attack Detection and Mitigation Method

In this section, we first introduce the basic knowledge of CRT-RS that play a role of the fundamental theory in our proposed method. Then, we describe the proposed amplification attack detection and mitigation method.

4.1. Chinese Remainder Theorem Based Reversible Sketch

Fig. 4. The structure of CRT-RS [13].

The main problem of traditional sketch is the irreversibility. When performing attack detection based on traffic information recorded by traditional sketch, we only obtain the information about whether there were any attacks. It is difficult to get the original IP addresses of abnormal hosts because no address information is stored in the traditional sketch. However, obtaining the malicious addresses is essential for attack defense. In order to solving the irreversibility problem of traditional sketches, we proposed CRT-RS in our previous work [13], which allows us to reversibly recover abnormal *keys* with lower computation complexity. Fig. 4 shows the structure of CRT-RS. The hash functions utilized in CRT-RS is $h_i(x) \equiv x \bmod p_i, i = (1, 2, \dots, H)$, where $p_1, p_2, \dots, p_H$ are chosen from pairwise coprime integers and satisfy $p_1 > p_2 > \dots > p_H$. CRT-RS supports four basic operations: (1) UPDATE: When a new pairwise item (*key*, *value*) arrives, the *key* will be hashed *H* times and the corresponding bucket in each row will increase by adding the *value*. The update process is described in Fig. 4; (2) QUERY: Return a point estimation for a given *key*; (3) COMBINE: The COMBINE function is used to merge multiple CRT-RSs into a single one by an accumulation operation; (4) RECOVER: This function aims to reversely recover the original IP addresses that exhibit abnormal characteristics and have been hashed by CRT-RS with low computation consumption. Detailed description of these operations is introduced in [13].

4.2. Amplification Attack Detection and Mitigation

4.2.1. System Architecture

Fig. 5 shows the overall process of proposed method. First, we respectively record incoming and outgoing network traffic at edge routers using CRT-RS. Based on the linearity of the CRT-RSs, we aggregate multiple sketches into one sketch, and apply proposed detection method to locate anomalous buckets in each row of the aggregated sketch. The malicious source addresses of response packets are reversely calculated based on the reversible characteristic of CRT-RS and are stored in a blacklist. A bloom filter that operated on blacklist is used to look up whether the source address of incoming traffic exists in the blacklist and if so, the traffic is filtered. Detailed process is introduced below. Notably, our detection and filtering method both assume that the attacked hosts and the reflectors are outside of the protected Internet Service Provider (ISP) or organization.

Fig. 5. System Architecture.

4.2.2. Network Traffic Collection

Our method is based on the one-to-one mapping between request packets and response packets. Due to asymmetric routing, some routers may use load balancing technology to deliver packets so that some pairwise packets may pass different paths. We can deploy a traffic collector at all routers in a network domain and transport the traffic traced from one router to others. This method is obviously expensive. In order to gain a global perspective of network traffic of a specific network scenario, collecting traffic at edge routers seems reasonable. But massive volume of traffic still brings great challenges on collecting traffic at edge routers [1, 33-37]. Hence, we use dual efficient methods to reduce the system burden of collecting and recording traffic at edge routers: (1) With the purpose of reducing data collection cost, we aggregate packets into flow with NetFlow standard and then collect flow-level traffic at edge routers. (2) With the purpose of reducing collecting and storage cost, we use CRT-RS to record network traffic information at edge routers.

Two CRT-RSs are employed to respectively record outgoing request packets and incoming response packets at each edge router, denoted as $CRT-RS_{out}$ and $CRT-RS_{in}$. In $CRT-RS_{out}$, we set the *key* as $\{D_{IP}, D_P\}$ (destination IP address and destination port), and

the value is flow size (the number of packets in a flow). Each bucket in the CRT-RS is denoted as $RS[i][j], i = 1, 2, \dots, H, j = 0, 1, \dots, p_i - 1$. So the update process in $CRT-RS_{out}$ is $RS[i][h_i(D_{IP} \& D_P)] += flow\ size, i = (1, 2, \dots, H)$. In $CRT-RS_{in}$, we set the key as $\{S_{IP}, S_P\}$ (source IP address and source port), the value is flow size, and update process is $RS[i][h_i(S_{IP} \& S_P)] += flow\ size, i = (1, 2, \dots, H)$. Selecting the key as $\{S_{IP}, S_P\}$ in $CRT-RS_{out}$ and the key as $\{D_{IP}, D_P\}$ in $CRT-RS_{in}$ also satisfies our purpose to match the incoming response packets with the corresponding outgoing request packets that are previously sent out. But for further attack mitigation, we should get the addresses of reflectors in order to filter the traffic coming from them. So, we choose the key as $\{D_{IP}, D_P\}$ in $CRT-RS_{out}$ and the key as $\{S_{IP}, S_P\}$ in $CRT-RS_{in}$.

During the period of collection, $CRT-RS_{out}$ and $CRT-RS_{in}$ continuously record traffic in real-time. Traffic traces are delivered to a central processor at the end of a collection time.

4.2.3. Amplification Attack Detection

In central processor, we aggregate all CRT-RSs into one by applying COMBINE function. Aggregated CRT-RS has total traffic information of a given network domain as if all the traffic passes through the same router. We allocate the weight of $CRT-RS_{out}$ to -1 and the weight of $CRT-RS_{in}$ to +1. The combination formula is described as follows:

$$RS_{aggregated}[i][j] = \sum_{n=1}^N (RS_{in-n}[i][j] + (-1)RS_{out-n}[i][j]) + RS_{last} \quad (1)$$

where $RS_{aggregated}[i][j]$ represents the bucket in the aggregated $CRT-RS_{aggregated}$, $RS_{in-n}[i][j]$ and $RS_{out-n}[i][j]$ represent the buckets in $CRT-RS_{in}$ and $CRT-RS_{out}$, respectively, N is the number of edge routers ($n = 1, 2, \dots, N$), $i = (1, 2, \dots, H)$, $j = (1, 2, \dots, M)$. RS_{last} is the CRT-RS that used to record negative buckets in the previous detection cycle. As the matched response is some time later than the request, there would be some difference between the number of the requests and the one of the responses in a specific time interval. This situation could lead to high false alarm rate. Therefore, we use RS_{last} to record negative buckets of $CRT - RS_{aggregated}$ in the previous detection cycle.

When an amplification attack occurs, the victim receives huge volume of response packets that come from one or more reflectors. This directly causes to generate many large positive values of buckets in $CRT-RS_{aggregated}$. In order to catch the anomalies, for each bucket of $CRT-RS_{aggregated}$, $RS_{aggregated}[i][j]$, the value is compared with a predefined threshold τ to decide whether this bucket is anomalous. τ is a small positive number instead of zero. The reason is that usually one request packet would generate one response packet. But a large response packet would break up to several packets due to the IP fragment when its length exceeds the Maximum Transmission Unit (MTU).

4.2.4. Amplification Attack Mitigation

After estimating each bucket in $CRT-RS_{aggregated}$, the buckets that exhibit abnormal will be added in the lists used for storing column number. The element stored in the i -th list ($i = 1, 2, \dots, H-1$) is in the format of (column number _{i} , $flag$), where the $flag$ is used to record column number of the $i+1$ row. We take H lists as input to RECOVER function of CRT-RS for reversibly recovering malicious source addresses and add these malicious addresses in a blacklist. In attack mitigation stage, our intention is to verify whether the source address of new arriving traffic exists in the blacklist. If so, the attacking traffic

will be filtered. We adopt a bloom filter here to fulfill this function due to its high efficiency in querying set membership.

However, the reflectors are innocent although they send massive traffic to overwhelm the victim. They only fulfill their responsibilities that reply the request packets. In order to still allow normal network communications, we will periodically clean the blacklist stored in bloom filters with a long time interval (e.g., a day).

Algorithm 1 describes the detection and mitigation process of the proposed method.

Algorithm 1: Amplification Attack Detection and Mitigation

1. Bloom filters are deployed at each edge router to query whether the source address of incoming traffic exists in the blacklist and if so, the traffic is filtered. Otherwise, send the traffic to collectors operated on CRT-RS.
 2. Use $CRT-RS_{out}$ and $CRT-RS_{in}$ to record traffic information at each edge router.
 3. After a period of collection, aggregate all CRT-RSs in the final $CRT-RS_{aggregated}$ at a central processor using Equation (1).
 4. For each row in $CRT-RS_{aggregated}$, locate abnormal buckets compared with a predefined threshold.
 5. Compute abnormal addresses using CRT-RS and store them in the blacklist.
 6. Update the bloom filter with the new blacklist.
 7. Return step 1.
-

5. Experiments

In this section, we first offer an introduction about the configurations of parameters. Then, we introduce the evaluation criteria used in experiments and describe our experimental environment in detailed. At last, we perform a number of experiments to evaluate our detection and mitigation.

5.1. Parameter Configurations and Evaluation Criteria

In this subsection, we provide the selection process of parameters that highly affect the effectiveness of the proposed method.

5.1.1 Parameter Configurations

- (1) **The parameters of CRT-RS:** There are two parameters affect the performance of CRT-RS, namely the number of hash functions H that determines the operating time and the column number of each row ($p_1, p_2, \dots, p_H$) that determines the size of memory needed for executing the proposed method [13]. In this paper, we set $H=5$ as a default value [13, 38]. Moreover, the *key* is chosen as IP address and port. Its space size is 2^{48} . We select pairwise coprime integers around 2^{20} as the values of $p_1, p_2, \dots, p_H$ by referring [38].
- (2) **The parameters of bloom filter:** The number of hash functions K and the hash table size m determine the performance of bloom filter. How to choose proper values of K and m of a bloom filter is elaborately discussed in [38]. By referring

[38], we set $K = 15$ and $m = 2^{22}$.

- (3) **Detection time interval:** The collection time interval T affects the detection time of the system to attacks. At the end of a collection time, traffic traces recorded in CRT-RS are delivered to a central processor to aggregate and analyze. The proposed method only needs to analyze the traffic information in current time interval and tries to find out the unbalanced relationship between request and response packets. It does not need to gain any pre-defined network behaviors. Moreover, the application of RS_{last} in equation (1) also compensates the time difference between outgoing request and incoming response packets. Therefore, the performance of the proposed method is not affected by the value of T . We empirically set T as 20s.
- (4) **Decision threshold:** Decision threshold τ directly influences the performance of the proposed method. It is a small positive number used to decide whether a bucket in each row is legitimate or not. We set $\tau = 15$ as default value.

The default and experimental test values of the parameters are summarized in Table IV.

Table IV. Summary of the parameters

	Parameters	Description	Value
Sketch	H	Number of hash functions	5
	$p_1, p_2, \dots, p_5$	Column number of each row	Around 2^{20}
Bloom filter	K	Number of hash functions	15
	m	Hash table size	2^{22}
Detection time interval	T	Detection time interval	20s
Decision threshold	τ	Detection threshold	15

5.1.2 Evaluation Criteria

True Positive Rate (TPR) and False Positive Rate (FPR) are employed as evaluation criteria to measure the efficiency of the proposed method.

- (1) TPR: $TPR = \frac{TP}{TP+FN}$, where TP denotes the number of malicious addresses that are accurately detected, FN denotes the number of malicious addresses that are not reported.
- (2) FPR: $FPR = \frac{FP}{FP+TN}$, where FP denotes the number of legitimate addresses that are wrongly identified as abnormal, TN is the number of legitimate addresses that are accurately identified as normal.

5.2. Performance Evaluation of Attack and Mitigation Method

5.2.1 Evaluation on NS3 Simulation

(1) Experimental environment

The performance of the proposed method was conducted on the NS-3 simulation platform. We built an experimental network topology, as shown in Fig. 6. It contains one hundred servers and half of them are reflectors (the number of malicious addresses and normal addresses are both fifty), one victim, six edge routers, one attacker and one botnet. We adjusted the number of bots in the botnet from 2000 to 10000 for increasing attack intensity. For launching amplification attacks, attack traffic with a spoofed source IP address is sent to reflectors with different transmission rate, which includes many request packets for TCP and UDP protocols (the red lines in Fig. 6). We do not consider ICMP protocol because ICMP amplification attack can be easily defended. The reflectors send the corresponding response packets to the victim through different edge routers. Meanwhile, the victim also conducts normal communication with servers/reflectors (the green lines in Fig. 6). The duration of simulation is 200 seconds. During this period, we launched amplification attacks twice, one began at 60th second and ends at 100th second and the other began at 140th second and ends at 160th.

Fig. 6. Network topology simulated in NS3.

(2) Evaluation Results

1) TPR and FPR

Fig. 7 (a) and (b) show the experimental results of TPR and FPR with different number of attacking hosts. We observe that our method achieves a good performance as TPR keeps 100% and FPR keeps 0%. The reason is that the proposed method utilizes the unbalanced mapping relationship between request packets and response packets to detect amplification attacks, which is the basic attack characteristic of amplification attacks. When amplification attacks occur, the number of response packets that victim receives is many times more than the number of request packets. Any abrupt changes detected in $CRT-RS_{aggregated}$ are considered as anomalies.

We also selected two state-of-the-art victim-end methods that both evaluate their schemes using attack simulation, namely traffic correlation based detection method [12] and machine learning based method [19], to conduct a qualitative comparison in the same simulation environment. Their TPR and FPR are indicated in Fig.7 (a) and (b), respectively. As the number of bots increase, the volume of attack traffic also gets large. Both the correlation based and machine learning based methods become ineffective since they have no means to deal with big-volume traffic. The comparison results denote that the proposed method can correctly and reliably detect amplification attacks with low FPR and high TPR.

Fig. 7. Effect of number of bots (a) TPR; (b) FPR

2) Analysis on Attack Mitigation

After detecting abnormal hosts, bloom filters are used to verify whether the source address of incoming traffic is legitimate. If not, the traffic will be filtered. Fig. 8 shows the variation of the number of packets the victim receives (some parts of packets generated by 2000 attacking hosts). We launched two amplification attacks at 60th second and 140th second. The drastic increase of the number of packets can be seen at the end of 80s, 100s and 160s. After employing our method to mitigate attacks, the attack traffic is accurately filtered, as the red line shown in Fig.8 (a). However, there is still some attack traffic arrived at victim between 100s and 120s because processing traffic takes some time.

3) Analysis on Protocol Independability

Many potential protocols can be utilized for launching amplification attacks, as shown in Table II and III. Any type of amplification attacks will occur whenever and wherever possible. Therefore, a protocol independent detection method that takes full advantages of basic amplification attack characteristic is worthy of study. In this study, we utilize the attack characteristic of amplification attacks that cause a abrupt deviation between the number of request packets that the victim sends and the number of response packets that the victim receives to perform attack detection and mitigation. This method can support detecting amplification attacks under different protocols. Fig. 8 (b) and (c) show the number changes of TCP and UDP packets that the victim receives before and after deploying our method (some parts of packets generated by 2000 attacking hosts). The

experimental results indicate that the method is a generic solution that supports protocol-independent detection.

Fig. 8. The packet number changes before and after deploying our method in the NS3 link (2000 bots). (a) total packets. (b) TCP packets. (c) UDP packets.

4) Analysis on Efficiency

Fig. 9. The relationship between processing time of the data collected in a detection interval and number of bots

The core step of our method is to employ CRT-RS to collect network traffic in a distributed manner. In our previous work, we have taken some experiments to verify that CRT-RS can deal with big-volume network traffic [13]. The average processing time of the data collected in a detection interval is also tested with the changes of the number of

bots and show the testing result in Fig. 9. Our method keeps a stable performance of traffic analysis while other two methods take a long time to analyze and even cause a delay in attack detection. Overall, our scheme can efficiently process traffic and perform attack detection and mitigation.

5.2.2 Performance Evaluation on Real World Traffic

To ensure our method can be deployed in real world, we also conducted a series of experiments on a public amplification attack dataset [44], which is provided by the LANDER project. This dataset includes malicious traffic generated by DNS amplification attack and normal communications between two sites. There are six recursive DNS servers acted as reflectors, an attacker and a victim in an attack process. The dataset lasts for about ten minutes and is divided into several subtraces. We selected some subtraces containing attack traffic and then combined them into one. We got 1637 normal source addresses and 6 malicious source addresses (the source addresses of reflectors).

Fig. 10. Effect of number of hash functions on TPR and FPR

Fig. 11. Effect of number of hash functions on processing time per detection interval

First, we tested the effect of the number of hash functions on the performance of our method, as shown in Fig. 10 and Fig. 11. As the number of hash functions increasing, we obtain higher TPR and lower FPR, but more processing time per detection interval (refer to Fig. 11). More rows in CRT-RS can help achieving higher accuracy of reversible

calculation and less hash collision. Hence, the detection accuracy of our method can observably improve at the cost of spending more processing time.

Second, we also conducted a qualitative comparison with the table-based method proposed in [11] on the same public dataset. Both our method and the table-based method utilize the one-to-one mapping relationship between request packets and response packets to perform amplification attack detection. Two main differences exist in these two methods: (i) we employ a reversible sketch to record and analyze network traffic, which makes our method has the ability of handling big-volume traffic and also performing attack mitigation; (ii) we consider the issue of packet delay and make a corresponding measure, as indicated in Equation (1). From the experimental results shown in Table V, we can see that our method outperforms with regard to TPR, FPR and processing time.

Table V. Comparison of experimental results with other test on real world traffic

	TPR (%)	FPR (%)	Processing time per detection interval (s)
Our method	100	2.56	0.08161
Table-based method [11]	66.7	13.5	27.759

6. Conclusions

This paper proposed an effective and efficient amplification attack detection and mitigation method at victim-end. We employed the sketch technique to record traffic information in a distributed manner and monitored abrupt changes in a one-to-one mapping between the request packets that the victim sends and the response packets that the victim receives. The results drawn from a series experiments conducted based on both NS3 simulations and real world traffic indicate that our method is much more accurate and efficient than existing methods, with additional suport on protocol independability. Moreover, the proposed method can accurately filter attack traffic by taking the advantage of CRT-RS that is capable of reversibly recovering the malicious source addresses.

Acknowledgments

This work is sponsored by the National Key Research and Development Program of China (grant 2016YFB0800700), the NSFC (grants 61672410 and U1536202), the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016ZDJC-06), Academy of Finland (grant No. 308087), and the 111 project (grants B08038 and B16037).

References

- [1] X. Y. Jing, Z. Yan, W. Pedrycz, Security Data Collection and Data Analytics in the Internet: A Survey, IEEE Communications Surveys and Tutorials, 21 (1) (2019) 586-617.

- [2] S. T. Zargar, J. Joshi, and D. Tipper, A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, *IEEE Communications Surveys and Tutorials*, 15 (4) (2013) 2046-2069.
- [3] T. Peng, C. Leckie, and K. Ramamohanarao, Survey of network-based defense mechanisms countering the DoS and DDoS problems, *ACM Computing Surveys*, 39 (1) (2007) .
- [4] C. Rossow, Amplification Hell: Revisiting Network Protocols for DDoS Abuse, in *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, 2014, pp. 1-15.
- [5] Matthew Prince, Deep Inside a DNS Amplification DDoS Attack, Accessed: 11-07-2017, [Online], Available: <https://blog.cloudflare.com/deep-inside-a-dns-amplification-ddos-attack/>.
- [6] F. J. Ryba, M. Orlinski, M. Wahlisch, C. Rossow, and T.C. Schmidt, Amplification and DRDoS Attack Defense - A Survey and New Perspectives, arXiv: 1505.07892, 2015, [Online], Available: <https://arxiv.org/abs/1505.07892>.
- [7] M. Backes¹, T. Holz, C. Rossow, T. Rytlahti, M. Simeonovski, and B. Stock, On the Feasibility of TTL-Based Filtering for DRDoS Mitigation, in *Proceedings of International Symposium on Research in Attacks, Intrusions, and Defenses*, 2016, pp. 303-322.
- [8] L. Kavisankar, C. Chellappan, P. Sivasankar, A. Karthi, and A. Srinivas, A pioneer scheme in the detection and defense of DRDoS attack involving spoofed flooding packets, *KSII Transactions on Internet and Information Systems*, 8 (5) (2014) 1726-1743.
- [9] J. Krupp, M. Backes, and C. Rossow, Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks, in *Proceedings of ACM SIGSAC Conference on Computer & Communications Security (CCS)*, 2016, pp. 1426-1437.
- [10] H. Tsunoda, K. Ohta, A. Yamamoto, N. Ansari, Y. J. Waizumi, and Y. Nemoto, Detecting DRDoS attacks by a simple response packet confirmation mechanism, *Computer Communications*, 31 (14) (2008) 3299-3306.
- [11] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, Detecting DNS Amplification Attacks, in *Proceedings of International Workshop on Critical Information Infrastructures Security*, 2007, pp. 186-196.
- [12] W. Wei, F. Chen, Y. J. Xia, and G. Jin, A Rank Correlation Based Detection against Distributed Reflection DoS Attacks, *IEEE Communications Letters*, 17 (1) (2013) 173-175.
- [13] X. Y. Jing, Z. Yan, X. Q. Liang, W. Pedrycz, Network Traffic Fusion and Analysis against DDoS Flooding Attacks with a Novel Reversible Sketch, *Information Fusion*, 51 (2019) 110-113.
- [14] L. Z. Cai, Y. K. Feng, J. Kawamoto, and K. Sakurai, A Behavior-Based Method for Detecting DNS Amplification Attacks, in *Proceedings of International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2016, pp. 608-613.
- [15] I. L. Meitei, K. J. Singh, and T. De, Detection of DDoS DNS Amplification Attack Using Classification Algorithm, in *Proceedings of International Conference on Informatics and Analytics*, 2016.
- [16] L. Rudman and B. Irwin, Characterization and Analysis of NTP Amplification Based DDoS Attacks, in *Proceedings of Information Security for South Africa*, 2015, pp. 1-5.
- [17] T. Bottger, L. Braun, O. Gasser, F. V. Eye, H. Reiser, and G. Carle, DoS Amplification Attacks – Protocol-Agnostic Detection of Service Abuse in Amplifier Networks, in *Proceedings of International Workshop on Traffic Monitoring and Analysis*, 2015, pp. 205-218.
- [18] D. Huistra, Detecting Reflection Attacks in DNS Flows, in *Proceedings of Twente Student Conference on IT*, 2013.
- [19] Y. X. Gao, Y. K. Feng, J. Kawamoto, and K. Sakurai, A Machine Learning Based Approach for Detecting DRDoS Attacks and Its Performance Evaluation, in *Proceedings of Asia Joint Conference on Information Security*, 2016, pp. 80-86.
- [20] L. H. Chi, X. Q. Zhu, Hashing Techniques, *ACM Computing Surveys*, 50 (1) (2017) 1-36.
- [21] B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, Sketch-based Change Detection: Methods, Evaluation, and Applications, in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003, pp. 234-247.
- [22] M. Charikar, K. Chen, M. F. Colton. Finding frequent items in data streams, in *Proceedings of International colloquium on Automata, Languages and Programming*, 2002, pp. 693-703.
- [23] G. Cormode and S. Muthukrishnan, An improved data stream summary: the count-min sketch and its applications, *Journal of Algorithms*, 55 (1) (2005) 58-75.
- [24] P. Roy, A. Khan, G. Alonso, Augmented sketch: Faster and more accurate stream processing, in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 1449-1463.
- [25] T. Yang, Y. Zhou, H. Jin, S. G. Chen , X. M. Li, Pyramid Sketch: a Sketch Framework for Frequency Estimation of Data Streams, *Proceedings of the VLDB Endowment*, 10 (11) (2018) 1442-1453.
- [26] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, DNS amplification attack revisited, *Computers & Security*, 39 (2013) 475-485.
- [27] D. C. MacFarland, C. A. Shue, and A. J. Kalafutb, The best bang for the byte: Characterizing the potential of DNS amplification attacks, *Computer Networks*, 116 (2017) 12-21.
- [28] B. Siekik, R. Macfarlane, and W. J. Buchanan, Evaluation of TFTP DDoS amplification attack, *Computers & Security*, 59 (2016) 67-92.
- [29] Prolexic, An analysis of DRDoS SNMP/NTP/CHARGEN reflection attacks part II of the DrDoS white paper series, Accessed: 11-07-2017, [Online], Available:

<https://www.akamai.com/us/en/about/news/press/2014-press/akamai-plxsert-warns-of-threat-sntp-reflection-ddos-attacks.jsp>.

- [30] M. Kuhrer, T. Hupperich, C. Rossow, and T. Holz, Exit from Hell? Reducing the Impact of Amplification DDoS Attacks, in Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), 2014.
- [31] Alert (TA14-017A) UDP-Based Amplification Attacks, Accessed: 11-07-2017, [Online], Available: <https://www.us-cert.gov/ncas/alerts/TA14-017A>.
- [32] D. C. MacFarland¹, C. A. Shue¹, and A. J. Kalafut, Characterizing Optimal DNS Amplification Attacks and Effective Mitigation, in Proceedings of International Conference on Passive and Active Network Measurement, 2015, pp. 15-27.
- [33] G. Q. Li, Z. Yan, Y. L. Fu, and H. L. Chen, Data Fusion for Network Intrusion Detection: A review, Security and Communication Networks, 2018 (2018).
- [34] D. H. Zhou, Z. Yan, Y. L. Fu, Z. Yao, A Survey on Network Data Collection, Journal of Network and Computer Applications, 116 (2018) 9-23.
- [35] H. Q. Lin, Z. Yan, Y. Chen, L.F. Zhang, A Survey on Network Security-Related Data Collection Technologies, IEEE Access, 6 (1) (2018) 18345-18365.
- [36] G. Liu, Z. Yan, W. Pedrycz, Data Collection for Attack Detection and Security Measurement in Mobile Ad Hoc Networks: A Survey, Journal of Network and Computer Applications, 105 (2018) 105-122.
- [37] L. M. He, Z. Yan, M. Atiquzzaman, LTE/LTE-A Network Security Data Collection and Analysis for Security Measurement: A Survey, IEEE Access, 6 (1) (2018) 4220-4242.
- [38] C. X. Wang, T. T. Miu, X. P. Luo, J. H. Wang, SkyShield: A Sketch-Based Defense System Against Application Layer DDoS Attacks, IEEE Transactions on Information Forensics and Security, 13 (3) (2018) 559-573.
- [39] E. Biagioni, Preventing UDP Flooding Amplification Attacks with Weak Authentication, in Proceedings of International Conference on Computing, Networking and Communications, 2019, pp. 78-82.
- [40] N. Sun, J. Zhang, P. Rimba, S. Gao, Y. Xiang, Data-driven Cybersecurity Incident Prediction and Discovery: A Survey, IEEE Communications Surveys and Tutorials, DOI: 10.1109/COMST.2018.2885561.
- [41] L. Liu, O. D. Vel, Q. L. Han, J. Zhang, Y. Xiang, Detecting and Preventing Cyber Insider Threats: A Survey, IEEE Communications Surveys and Tutorials, 2 (2) (2018) 1397-1417.
- [42] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, "Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies," IEEE Communications Surveys and Tutorials, 19 (1) (2017) 465-481.
- [43] S. Yu, M. Liu, W. C. Dou, X. T. Liu, S. M. Zhou, Networking for big data, IEEE Communications Surveys and Tutorials, 19 (1) (2017) 531-549.
- [44] Scrambled Internet Trace Measurement dataset, IMPACT ID: USC-LANDER/DoS_DNS amplification-20130617/rev5529. Traces taken 2013-06-17 to 2013-06-17. Provided by the USC/LANDER project. Accessed: 5-01-2019, [Online], Available: <http://www.isi.edu/ant/lander>.