

A Privacy-Aware Collaborative DDoS Defence Network

Carol Fung, Yadunandan Pillai
Virginia Commonwealth University, Richmond, VA
{cfung, pillaiyt}@vcu.edu

Abstract—Distributed denial of service (DDoS) attacks can bring tremendous damage to online services and ISPs. Existing adopted mitigation methods either require the victim to have a sufficient number of resources for traffic filtering or to pay a third party cloud service to filter the traffic. In our previous work we proposed CoFence, a collaborative network that allows member domains to help each other in terms of DDoS traffic handling. In that network, victim servers facing a DDoS attack can redirect excessive connection requests to other helping servers in different domains for filtering. Only filtered traffic will continue to interact with the victim server. However, sending traffic to third party servers brings up the issue of privacy: specifically leaked client source IP addresses. In this work we propose a privacy protection mechanism for defense so that the helping servers will not be able to see the IP address of the client traffic while it has minimum impact to the data filtering function. We implemented the design through a test bed to demonstrate the feasibility of the proposed design.

I. INTRODUCTION

Distributed Denial of Service (DDoS) attacks are one of the top threats to enterprise networks and ISPs. They flood their target with bogus traffic which exhausts its resources, consequently resulting in slowing down the victim's response speed or taking it down entirely. As a result, businesses targeted by such attacks suffer massive financial damage. For example, the average financial loss due to DDoS attacks rose to over 2.5 million in 2017 for each businesses, 84% of which experienced at least one DDoS attack in that year [1]. Some recent incidents showed that DDoS attacks significantly increased in strength. For example, the Spamhaus attack in 2013 [2] generated traffic reaching 300 Gbps. This number constantly increased each year, reaching 1.7 Tbps by 2018 [3]. DDoS attacks are motivated by financial gains and are significantly easier to launch now due to the booming of dark markets and available network stressing tools such as *Booters* [4] and *Webstresser* [5]. The cost of launching a median scale DDoS attack can be as low as \$5 [6].

Among the DDoS attacks, a large portion of the attack traffic is *IP spoofing attacks*, where the source IP addresses of the attack packets are fabricated. A significant example is SYN flood attacks [7], where a large number of spoofed SYN packets are sent to the victim to overrun its resources. DDoS attacks can also be launched via real source IP addresses. For example, reflected traffic [8] can be used where the attack traffic is the response of spoofed requests from attackers.

Both attacks can be implemented via *bot nodes attacks* where compromised zombie devices controlled by a bot master are instructed to launch simultaneous attacks on one or more target.

Traditional methods of DDoS defense demands large computation capacity on involved network functions, such as firewalls and routers. Upgrading or adding new hardware brings high cost. On the other hand, utilizing third party cloud-based DDoS filters such as CloudFlare [9], [10] not only introduces high delay by directing traffic to a remote third party, but also brings high cost since they are third-party commercial services. To address those challenges, in our previous work [11], [12], we proposed CoFence, a collaborative DDoS mitigation framework by utilizing peer-to-peer help from third party alliance networks to mitigate DDoS attacks. The main idea is to have the alliance networks create DDoS traffic filtering agents to handle excessive external attack traffic. When one domain is under DDoS attack, it can direct the excessive traffic to the alliance domains for traffic filtering. However, the privacy concern is the main issue in this collaborative network.

To preserve the privacy of the participating domains. In this work we propose a privacy protection design for CoFence that can protect the privacy when sharing traffic with other domains. Our design provides an agile and robust mechanism design which adds a layer of data privacy while brings little overhead on packets processing.

II. RELATED WORK

In the last few years, there have been several efforts to enhance the robustness of networks and domains against DDoS attacks. In this section, we briefly review some of the existing solutions on DDoS mitigation including our proposed CoFence framework.

The approach proposed in [13] utilizes a DDoS blocking application that runs over the SDN controller while using the standard OpenFlow interface. However, one weakness is that a large amount of traffic between the application and the controller must be protected. This dependency on their cooperation causes high latency for client traffic.

[14] proposed a cloud-based and NFV-based firewall service in which the virtualization capabilities of the system help to protect networks from DDoS attacks. The designed cloud architecture acts as an intermediary to filter the attack traffic for the customer, and transmits only legitimate traffic.

Some other works [15], [16] make use of NFV and SDN to implement a DDoS defense system, Bohatei. The flexibility of SDN and NFV technology introduces a new way of DDoS defense through cooperation among virtual network functions. However, the internal resource may be quickly exhausted when the attack volume is high enough.

Our previous work CoFence further improves the defense capability by utilizing external resources through a collaboration network. CoFence [11], [12] is designed to address DDoS solutions' obstacles due to their high cost and low flexibility. CoFence, using NFV technology, facilitates a collaboration framework among NFV-based peer domain networks. It allows domain networks help each other handle large volumes of DDoS attacks through resource sharing. Specifically, we focus on the resource allocation problem in the collaboration framework. Through CoFence, a domain network decides the amount of resources to share with other peers based on a reciprocal-based utility function.

In contrast to the mentioned related works, the focus of this work is to add a layer of protection for the privacy of each participating domains. By using encryption and packets modification our proposed mechanism can hide the source IP address of the packets while still keep the collaborative filtering function working.

III. THE BACKGROUND OF COFENCE

In this section, we will briefly describe CoFence, our previously proposed solution on DDoS defense based on collaborative networks. We will start with the basic concept of traffic filtering delegation and then describe the design of CoFence.

A. DDoS Defense Using Delegated Handshaking

Synflood attack is a top DDoS attack where attackers send a large volume of bogus SYN packets to overwhelm the recipients. To defend against the SYN flood attack, the key concept of CoFence is to allow external agents to delegate the handshaking process. As shown in Figure 1(a), an intrusion prevention system (IPS) performs delegated TCP handshaking by responding to the SYN packets directly without forwarding to the server under attack. It sends back SYNACK back to the source and verifies the ACK packets if the source is legitimate and sends back an ACK packet. Once the source IP is verified, the IPS performs a spoofed handshaking with the attacked server and connects the source with the server.

Figure 1(b) shows the scenario that an IPS handles SYN-flood traffic with spoofed source IP addresses. We can see that the SYN flood will not be able to reach the server since they cannot finish the handshaking process with the IPS.

One major issue with using internal IPS for handshaking delegation is that when the SYN flood traffic is high, the internal IPS can be overwhelmed as well and will be able to handle all the traffic.

In Our previous work CoFence [12], [17], [18], [11], [19], we proposed to use external resource to handle SYN flood attacks. External resources can be domains who volunteer to

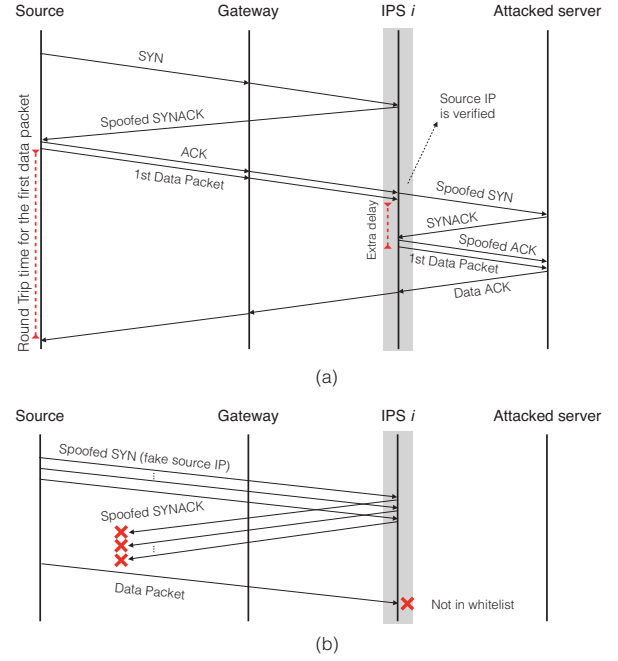


Fig. 1. SYN Flood Attack overview: (a) a scenario on how the IPS handles traffic with authentic source IP addresses; (b) a scenario how the IPS blocks DDoS attacks with spoofed source IP addresses.

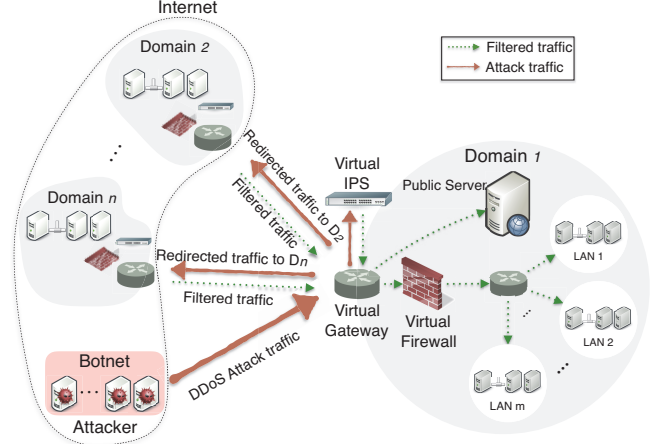


Fig. 2. CoFence Framework Overview

provide resource to help others in the federation network. The architecture of CoFence is described in the next subsection.

B. CoFence Architecture

CoFence is a federated network that consists of a number of domain networks who want to help each other to handle DDoS traffic to increase the resistance of the domain against DDoS attacks. Each domain contains a virtual gateway and a virtual IPS. The purpose of the virtual IPS is to detect and filter DDoS attacks. Due to the flexibility of NFV, a virtual IPS can be created and its capacity configured dynamically based on need. When a domain joins CoFence, the domain can choose whether to share its IPS with other trusted domains or not, and configure the maximum external traffic it is willing to handle for other domains.

Figure 2 illustrates a case study of CoFence. When the attacker launches a DDoS attack against the public server in domain 1 and the attack traffic volume exceeds the maximum capacity of the local IPS, some incoming traffic can be redirected to its collaborator domains for filtering, for example by updating the forwarding tables in the gateway. The SYN flood will be filtered remotely and only the filtered traffic is forwarded back to domain 1.

C. Privacy Concerns about CoFence

CoFence is intended to be an open network that allows domain networks to help each other in terms of DDoS mitigation service. However, redirecting TCP handshaking packets to other domains may bring privacy concern problem. For example, the source IP addresses in the SYN packets may leak the client information of the victim network. For example, the malicious domains may collect the source IP addresses of the victim's clients when the handshaking packets are redirected to them. Therefore, CoFence must be enhanced where it can protect the privacy concerns of the victim server.

IV. PRIVACY PROTECTION PROTOCOL DESIGN FOR COFENCE

In this section, we introduces the privacy protection protocol design for CoFence. We start with the overall flow design and then focus on the function design of the key component - the Victim gateway.

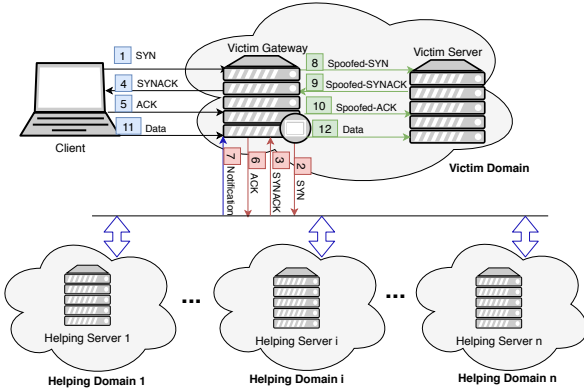


Fig. 3. CoFence Protocol Design

A. Handshake Delegation Process

Figure 3 shows the overall handshake delegation process of the CoFence system. Under the DDoS mitigation mode, when a SYN packet from a client to the victim server arrives from the victim domain gateway, it is redirected (step 1,2) to one of the helping domains after substituting the source and destination IP addresses (see next subsection for details). The helping server responds to the SYN packet with a SYNACK packet (step 3). The SYNACK packet is redirected to the client after substituting the source and destination IP addresses (see next subsection for details). When the ACK packet from the client arrives, it will be redirected to the same helping domain

(step 5, 6). When the ACK packet is accepted by the helping server, a notification will be sent to the gateway (step 7). After that, the victim domain processes a spoofed handshake with the victim server using the same SYN packet from the client (step 8,9,10). After that, a connection between the client and the victim server is established and the data packets will be directed to the victim server (step 11,12).

B. Privacy-Aware Packet Modification

To redirect the handshaking packets to external handshaking delegates, the victim gateway modifies the handshake packets so that they are rerouted by the underlying system. For the SYN packet, the original source IP address will be encrypted using a lightweight encryption method (see next subsection) and stored as a meta data in the modified packet (Figure 4). The source IP is modified to be the victim server IP and the destination IP is modified to be the helper server IP. The modified SYN packet is then redirected based on the new header information. Note that this process is to prevent the helper server from seeing the client IP address. The helper server only sees the encrypted client IP, which is not valuable for collection.

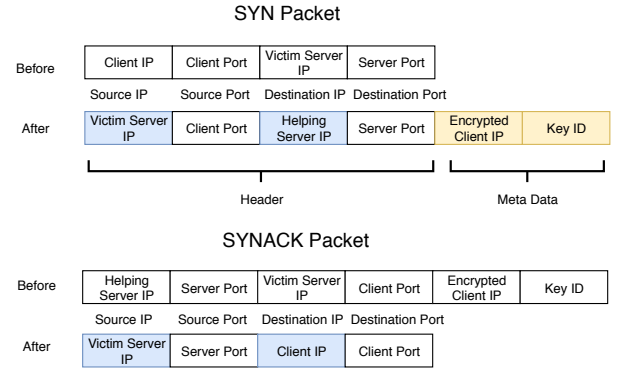


Fig. 4. Modification of a SYN Packet on the Victim Gateway

Similarly, the SYNACK packet from the helper server is also modified at the victim server gateway so that the source IP address is changed back to the victim server IP address and the destination IP address is back to the client IP address. Note that the client IP address is obtained through decrypting the cipher text in the meta data.

The process of ACK packet is similar to SYN packet except that the modified ACK packet will have two versions: one using the current key chain and the other using backup key chain (see next subsection) in case the key chain has been refreshed during the waiting period.

C. Light-weight Encryption and Key Chain Management

To avoid excessive delay in packet processing, we will use a light-weight encryption mechanism to encode the source IP addresses of the clients' packets. For this purpose, we introduce the concept of *key chain*. A key chain is a data structure that consists of a sequence of randomly generated bits to stores a sequence of encryption keys identified by key ID,

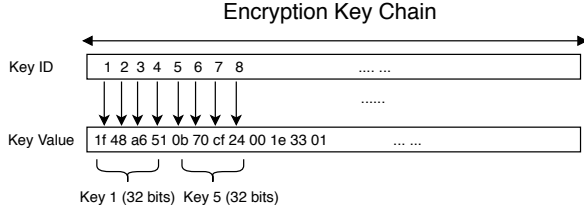


Fig. 5. The Design of the Encryption Key Chain

which is also the index of the first byte of the key (Figure 5). Given a key ID, an encryption key is obtained by reading 32 consecutive bits (4 bytes) on the belt starting from the key ID. The random bits on the belt should be updated periodically before the adversary obtains sufficient information on the key chain. As we mentioned earlier the system maintain Two key chains (C and P) which represent the CURRENT key chain and the PREVIOUS key chain used in the last time window. Each CURRENT key chain expires after time t and it will be stored as a PREVIOUS key chain. A new CURRENT key chain will be generated periodically by filling the belt with new random values.

The selection of key ID is determined by the source IP to be encrypted. We can use a simple hash function H to obtain a corresponding key in the chain. i.e., $ID = H(IP)$, where IP is the IP address to be encrypted and ID is the corresponding key ID. Let the function $Key(C, ID)$ denote the corresponding key value given a key chain C and a key ID . Then we have and $K = Key(C, ID)$.

For an accelerated encryption/decryption process, The IP address encryption and decryption operations are done through a simple XOR operation $\oplus$. Let $E(.)$ denote the encryption function and $D(.)$ denote the decryption function, then we have the $E(IP) = IP \oplus K$. On the other hand, the decryption function $D(EIP) = EIP \oplus K$, where EIP is the encrypted IP address and K is the key value.

D. The Connection Table

The gateway maintains a connection table which records all active and established flows connected to the victim domain. When an ACK packet is accepted by the helping server, then the source IP is validated. The helping server sends a notification packet to the victim gateway, which contains the encrypted IP address of the client (EIP), the key ID (ID), the port number of the source (P_s), the IP address of the source (IP_s), the port number of the destination (P_d), the sequence number (S_q) and the ack number (A_q) of the ACK packet, denoted by tuple

$$\langle EIP, ID, IP_s, P_s, P_d, S_q, A_q \rangle$$

. After that, the gateway kicks off a three-way spoofed handshake with the victim server immediately (step 8-10 in Figure 3) using the client's credentials, a.k.a., the same SYN packet as the client, which can be constructed using the tuple above. We denote the sequence number and the ack number of the spoofed ACK packet using S'_q and A'_q respectively.

When a successful spoofed handshaking with the victim server, the gateway adds a new entry to its white list connection table which stores all active flows and the offset between the sequence number negotiated with the helping server and the sequence number negotiated with the victim server. The new entry inserted can be expressed using tuple

$$\langle EIP \oplus Key(ID), P_s, IP_s, P_d, S_q, A_q - A'_q \rangle$$

and the offset will be applied to each incoming data packet accordingly. All data packets associated with this established connection are now directly forwarded to victim server, with the sequence numbers being modified according to the offset in the connection table.

V. CONCLUSION

In this work we propose a privacy protection protocol for collaborative DDoS mitigation networks. We propose to use a light-weight encryption function based on a dynamic key chains to reduce the overhead on the gateway. We implemented our solution using virtual machines to test the feasibility of the proposed protocol. In our future work, we plan to finish the prototyping and provide a comprehensive performance evaluation of the proposed solution.

REFERENCES

- [1] Charlie Osborne. The average ddos attack cost for businesses rises to over \$2.5 million. <https://www.zdnet.com/article/the-average-ddos-attack-cost-for-businesses-rises-to-over-2-5m>.
- [2] John Leyden. Biggest ddos attack in history hammers spamhaus. <https://www.theregister.co.uk/2013/03/27/spamhaus-ddos-megaaflood>.
- [3] Famous ddos attacks. <https://www.cloudflare.com/learning/ddos/famous-ddos-attacks>.
- [4] José Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras. Booters?an analysis of ddos-as-a-service attacks. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 243–251. IEEE, 2015.
- [5] The webstresser homepage. Webstresser.org.
- [6] Brigid O’Gorman, Candid Wueest, Dick O’Brien, Gillian Cleary, Hon Lau, John-Paul Power, Mayee Corpin, Orla Cox, Paul Wood, Scott Wallace, et al. Internet security threat report, 2019.
- [7] Wesley M Eddy. Syn flood attack. In *Encyclopedia of Cryptography and Security*, pages 1273–1274. Springer, 2011.
- [8] Saman Taghavi Zargar, James Joshi, and David Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys and tutorials*, 15(4):2046–2069, 2013.
- [9] Prolexic routed. <https://www.akamai.com/us/en/solutions/products/cloud-security/prolexic-routed.jsp>.
- [10] Cloudflare. <https://www.cloudflare.com>.
- [11] Bahman Rashidi and Carol Fung. Cofence: A collaborative ddos defence using network function virtualization. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 160–166. IEEE, 2016.
- [12] Bahman Rashidi, Carol Fung, and Elisa Bertino. A collaborative ddos defence framework using network function virtualization. *IEEE Transactions on Information Forensics and Security*, 12(10):2483–2497, 2017.
- [13] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In *6th Ubiquitous and Future Networks (ICUFN)*, pages 63–68, 2014.
- [14] Fouad Guenane, Michele Nogueira, and Ahmed Serhrouchni. DDoS mitigation cloud-based service. In *7th IEEE UbiSafe Computing*, pages 1363–1368, 2015.

- [15] Seyed K. Fayaz, Yoshiaki Tobioka, Vyas Sekar, and Michael Bailey. Bohatei: Flexible and elastic ddos defense. In *24th USENIX Security Symposium*, pages 817–832. USENIX, 2015.
- [16] Carol J Fung and Bill McCormick. Vguard: A distributed denial of service attack mitigation method using network function virtualization. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 64–70. IEEE, 2015.
- [17] AHM Jakaria, Wei Yang, Bahman Rashidi, Carol Fung, and M Ashiqur Rahman. Vfence: A defense against distributed denial of service attacks using network function virtualization. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 431–436. IEEE, 2016.
- [18] Jelena Misic, Jun Fung, and Vojislav B Misic. Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis. In *8th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'05)*, pages 8–pp. IEEE, 2005.
- [19] Carol Fung and Raouf Boutaba. *Intrusion Detection Networks: A Key to Collaborative Security*. Auerbach Publications, 2013.