Defending DDoS Attacks Using Traffic Differentiation and Distributed Deployment

Rohan Patil [†‡], Aditya Kumat[†‡], Karan Bulbule[†‡], Maitreya Natu[‡]
[†]Student author, College of Engineering, Pune, India
[‡]Tata Research Development and Design Centre, Pune, India

***Abstract*—In this paper we present a defense against distributed denial of service attacks. We propose a computationally light-weight approach to differentiate legitimate traffic from the attack traffic and perform appropriate rate-limiting. We then present initial ideas about systematic selection of locations to deploy defense nodes. We implemented the proposed approach as a loadable Linux kernel module and performed live-traffic experiments on Emulab testbed. We present experimental evaluation of the proposed algorithms to demonstrate their effectiveness in different attack scenarios.**

I. Introduction

Distributed denial of service (DDoS) attacks remain an unmitigated threat to today's networks in spite of various academic and commercial attempts to build an effective defense. This unmitigated threat can be attributed to various factors such as the large number of potential attackers, continuously evolving types of attacks, ability of the attackers to hide their identity through techniques such as spoofing, reflector attacks, zombie attacks, etc. Previous attempts on DDoS defense provide two main insights: (a) Wide deployment is a necessary condition for a DDoS defense. Many smart defenses, when deployed sparsely, can be defeated by the attacker by either being too diffused to bypass unnoticed, or being too large to overwhelm the defense itself. (b) There is a critical need for a sophisticated mechanism to differentiate legitimate traffic from attack traffic. However, the wide variety of attacks, the large scale of attacks, together with limited resources at the defense node, etc. make it very difficult to accurately differentiate legitimate traffic from attack traffic.

In this paper, we present a defense against DDoS attacks using the above two insights. We propose a mechanism for traffic differentiation and deployment of defense nodes. The objective of the proposed solution is to defend a victim network from DDoS attacks.

A. Distributed deployment

Single point deployment cannot achieve successful defense. The DDoS defense demands a distributed solution where defense nodes located throughout the network cooperate to achieve an overall effective defense.

A DDoS defense requires three vital functionalities namely (a) Attack detection: quickly detecting the presence of an attack; (b) Traffic differentiation: differentiating legitimate traffic from the attack traffic; and (c) Rate limiting: providing preferential treatment of the shared resource to the legitimate traffic. These functionalities are best met at different network locations such as client network, core network, or the victim network.

B. Traffic differentiation

The defense must be able to differentiate legitimate traffic from attack traffic. Thus, the defense should then reduce attack flows to manageable flows and ensure good service to legitimate traffic even during the attack.

While meeting this objective, the defense should have following properties: (1) Defense should be light-weight to support fast packet processing and to prevent itself from being overwhelmed by the attack. (2) In order to deal with large scale attacks, defense should be capable of multi-node deployment and each defense node should be able to cooperate with other defense nodes. (3) One way of differentiating legitimate client from attacker is to assign a reputation score to each client. One-time computed reputation can be exploited by attackers by behaving good for some time to earn good reputation and then turning bad. Hence, the reputation of a client should be periodically evaluated.

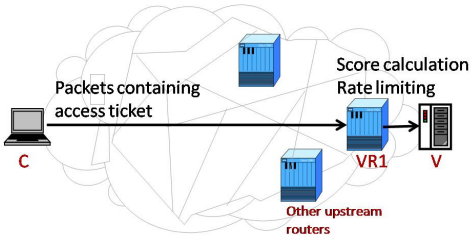

Fig. 1. Setup for the proposed approach.

The reputation should be non-binary with multiple levels of confidence in the legitimacy of the client. This can provide a fine-grained evaluation of the client behavior.

C. Contributions:

In this paper, we primarily focus on presenting an algorithm for differentiation of legitimate traffic from attack traffic by assigning scores to clients based on their past behavior and compliance to TCP. We are different from the past attempts in reputation-based defense in the following manner. (a) Capability-based techniques such as SIFF [4] and TVA [5] provide efficient mechanisms for generation and verification of access tickets, but they lack an automated mechanism for granting or denying these access tickets. We propose an automated mechanism to grant or deny access to the shared resource. (b) In many past attempts, the access to shared resource is binary in nature providing full or no access to clients. Such solutions are vulnerable to attackers that first act legitimate to obtain the access and then turn malicious. We grant non-binary signatures providing different access levels to the shared resource. (c) Instead of maintaining only legitimate clients (DWARD [3]) or only attack clients (Pushback [1]), we propose to maintain both profiles for better traffic policing. (d) Unlike techniques like SOS [2] that demand architectural changes, the proposed solution does not demand any such changes and allows incremental deployment.

The main contributions of this paper are as follows: (1) We present a computationally light-weight yet effective defense mechanism to defend DDoS attacks. (2) We present experimental evaluation in different attack scenarios to demonstrate the effectiveness of the proposed defense. (3) We present initial ideas on systematic selection of minimal set of nodes to deploy defense.

II. Proposed approach

It is very difficult to model an attack behavior due to the wide variety of attacks, similarity of packet content of legitimate and attack traffic, and the continuously evolving nature of the attacks. However, the legitimate TCP client can be predictable. We propose to model a legitimate traffic pattern and present an algorithm to compute score of a client traffic based on its compliance to the TCP behavior. The scores are assigned such that legitimate TCP compliant traffic gets higher score while the non-TCP traffic gets lower score. However, a client behavior needs to be captured in a computationally light-weight fashion. Thus the problem demands an effective but computationally light-weight solution. As the identification of the legitimacy of a client traffic is not always ensured in black and white, there is a need for intelligent rate-limiting of the client traffic based on the inferred reputation of the client such that *more* legitimate client gets a larger share of the resource than the *less* legitimate client and the critical resource is well-utilized.

In this section, we present a mechanism for traffic differentiation and rate-limiting. The objective of the defense mechanism is to prevent DDoS attacks by (1) differentiating legitimate clients from attackers, and (2) appropriate rate-limiting of the shared resource based on the reputation of the client traffic.

Figure 1 represents the setup involved in the proposed approach.

1. A traffic coming from client C to the victim node V is monitored at a verification node VR1. First-time client undergoes a verification handshake with the verification server and is granted an access ticket. The access ticket is encrypted in each packet is passed by the client in the subsequent communication.
2. The node VR1 monitors the client behavior to congestion and compliance to allocated bandwidth share and assigns a score to the client. Unlike previous solutions [4], the score is non-binary in nature. The confidence in the legitimacy of a client is periodically evaluated and the score is updated accordingly.
3. The access to the shared resource is based on the client score. High score clients are indicative of legitimate clients and are given privilege over unknown and the low score clients.

The functionality of the verification node VR1

can be distributed across multiple nodes. In Section III, we discuss techniques for systematic selection of such nodes. In this section, we present details of the access ticket, score computation, and rate limiting.

A. Access ticket

Access tickets are bound to each client IP and are unique for each client IP. The tickets are short-lived and are frequently updated. The access ticket prevents IP spoofing and thus prevents attackers from using the access ticket of a legitimate client. Various techniques proposed in the past [2] present ways for performing a secure handshake of such tickets between a client and a verification server to ensure the credibility of an IP address. Due to lack of space, we do not discuss further details of the access ticket mechanism.

B. Score computation

Unlike the previous work on granting computing binary capabilities where a client is either considered legitimate or attacker, we propose non-binary capabilities by assigning scores to the clients. We compute client scores by exploiting the backing-off property of the TCP behavior in the presence of congestion.

The score has following properties. (1) Scores can be zero, positive, or negative. (2) There is a limit (MAX_SCORE) on maximum positive score a client can be awarded. This prevents a long-term well-behaving client from achieving a status where it can create damage by turning bad. (3) There is a limit (MIN_SCORE) on maximum negative score. A client reaching the maximum negative score frequently is entered into the list of blacklisted clients. Blacklisted clients are always denied the resource.

Each active client is assigned a fair-share of the critical resource (e.g. bottleneck bandwidth). We later discuss the resource share allocation while explaining the rate-limiting process. Based on the compliance to fair-share, the client score is periodically changed as follows.

1. Each new client is assigned a small positive score to allow the client some time to establish its reputation.

2. *Additive increase:* When a client traffic is within its allocated share, the client is rewarded with a fixed increase α in the score. We reward in a conservative manner by performing an *additive* increase in the client score.

$$Score_{new} = min(Score_{old} + \alpha, MAX\_SCORE) \quad (1)$$

3. *Weighted subtractive decrease:* When a client traffic exceeds its allocated share, the client is penalized with a decrease in the score. We penalize in an aggressive manner by performing a *weighted* decrease in the client score. We compute a penalty p for each client as a function of (a) the past penalty p_{old} and (b) the extent e by which the client violates the allocated share. The score of the client is decreased by the value computed by the penalty.

$$Score_{new} = max(Score_{old} - (p_{old} * e), MIN\_SCORE) \quad (2)$$

We perform a weighted subtractive decrease till the score reaches a negative threshold. Beyond the threshold, we start performing a flat subtractive decrease. This is done to ensure that any incorrectly penalized client gets a chance to improve its reputation in finite time. With the additive increase and weighted subtractive decrease we ensure that a client behaving bad for some time would have to behave good for longer time to regain trust.

4. *Subtractive decrease:* When a client stays idle for more than certain amount of time, we start performing a subtractive decrease β in its score. This is done to (a) improve resource utilization, (b) to prevent a long-term idle client from misusing its high score established in the past. The score of idle clients is not decreased beyond zero to distinguish idle clients from attackers. Clients behaving idle for a long time receive a score of zero and are treated like new clients.

$$Score_{new} = min(Score_{old} - \beta, 0) \quad (3)$$

C. Rate limiting

We propose a priority-queue based mechanism to do the desired rate-limiting. We implement a priority queue where each priority level is assigned a certain share of the resource. Clients are ranked on their score and priority levels are assigned to clients such that a fixed percentage of active clients are assigned to each priority level. Higher priority clients are assigned larger share of the bandwidth than the lower priority clients. This policy ensures (a) preferential treatment to clients with higher score, (b) effective resource utilization, (c) prevention of over-provisioning of the available resource.

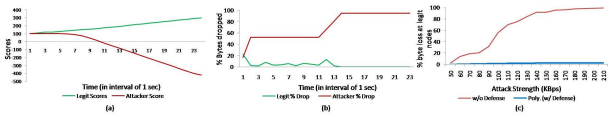

Fig. 2. (a) Scores computed for legitimate and attack clients, (b) Percent bytes dropped for legitimate and attack clients, (c) Percent packet loss at legitimate client with and without defense deployment.

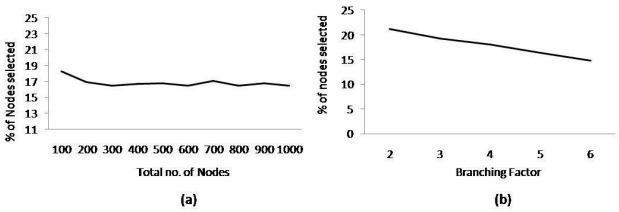

Fig. 3. Percentage of total nodes selected as defense nodes for (a) different network sizes, (b) different average node degree.

III. Distributed deployment

Single-point defense can lead to inaccuracy in traffic differentiation causing high collateral damage and single point of failure. On the other hand, the other extreme of deploy-everywhere is discouraged due to economic concerns and lack of deployment incentive at locations away from victim network. The selection of defense nodes should be done such that defense nodes are evenly distributed throughout the network and none of the defense nodes is overwhelmed with traffic from a large number of nodes. Furthermore, a hierarchical deployment of defense nodes is required such that defense nodes are deployed at strategic locations in the client, core, and victim networks.

A. Problem description

We propose to model the network as a graph and build a victim-rooted traffic tree. The root node of this tree is the victim node. The paths from all nodes to the victim node form this tree. For clarity, we assume static-single path between a node and the victim node. The defense deployment problem is then to select appropriate nodes on this tree as defense nodes such that (a) all nodes in the tree are covered, (b) the number of nodes to be defended by the defense nodes is uniformly distributed across all defense nodes.

Given a tree *T(V,E)*, for a node $v \in V$ we define *SuccessorNodes(v)* as the nodes in the successor graph rooted at node *v*. For each node $v \in V$, we define *MonitoredNodes(v)* where *MonitoredNodes(v)* $\subseteq$ *SuccessorNodes(v)* and *MonitoredNodes(v)* consists of the nodes in *SuccessorNodes(v)* that do not belong to *MonitoredNodes(w)* where $w \in \{V - v\}$. The problem of defense node selection is to select the smallest number of defense nodes such that each node is monitored by one or more defense nodes and for each selected defense node v, $|MonitoredNodes(v)| < k$. A dual to this problem is to select k number of defense nodes such that *MonitoredNodes(v)* for each v in defense nodes is uniformly distributed.

B. Proposed approach

The best solution to the above problem involves a combinatorial approach and can be proved to be NP-Complete. However, an approximation algorithm can be built by using multiple graph traversals to obtain a close to optimal (minimal number of defense nodes) solution. The key idea behind the proposed approach is to perform multiple bottom-up and top-down traversals of the tree to achieve a subtree of the desired cluster size. The root of the subtree is selected as a defense node and the subtree thus formed is pruned from the original tree. The process is performed until all nodes in the graph are covered and the root of the tree is reached. We do not present further details of the proposed approach due to lack of space.

IV. Experimental evaluation

We performed various experiments to evaluate the proposed algorithms. We next present experimental evaluation of the proposed defense and deployment algorithms.

A. Defense mechanism

We implemented the proposed reputation computation mechanism in a Linux software router as a loadable kernel module. We performed live-traffic experiments in the Emulab test-bed. Victim node V is connected to the rest of the network via a bottleneck link of 512Kbps. The rest of the topology consists of several client nodes each of which is connected to 100Mbps link. We generate legitimate TCP traffic by performing a SFTP file transfer between client and the victim node. We generate attack traffic by using raw sockets to send TCP packets at a specified rate.

We present the scenario where the legitimate traffic consists of an SFTP file transfer, and the attacker sends traffic at the rate of 40KBps. From Figure 2a it can be seen that the score of the legitimate TCP client steadily increases due to the property of additive increase. There is a rapid decrease in the score of the attacker due to the property of weighted subtractive decrease. Beyond a threshold the attacker score starts decreasing in a flat subtractive manner. From Figure 2b it can be seen that the bytes dropped for legitimate client is close of zero, while that the attacker reaches above to 90%. The defense involves a learning period of few seconds to evaluate a client behavior and take appropriate action.

We computed the time taken to perform a 832KB file transfer in scenarios with and without defense deployment. In case of an attack, the file transfer time without defense deployment is 214 sec. On the other hand, the file transfer time with the defense deployment is only 13 sec.

We next present how the proposed defense mechanism responds to attacks of different strengths. We ran a legitimate traffic and performed attacks of different strengths from 50KBps to 200KBps over a bottleneck link of 150KBps. From Figure 2c it can be seen that the defense successfully transmits the legitimate traffic even in the presence of heavy DDoS attack. In the absence of the proposed defense, the heavy attacks can otherwise lead to close to 100% packet loss of the legitimate traffic.

B. Distributed deployment

To evaluate the proposed deployment mechanism, we generated various realistic network topologies using BRITE topology generator. In each topology we randomly chose a defense node and computed the location for defense nodes using the proposed deployment mechanism. We demonstrate the effect of network parameters such as number of nodes and average node degree on the required number of defense nodes. Each graph plotted is an average of 10 runs. We also plot the 95% confidence intervals.

We first fix the average node degree to 3 and generate topologies by changing the number of nodes from 100 to 1000 in the network. It can be seen from Figure 3a that the percentage of total number of nodes selected as defense nodes stays constant (close to 20%) irrespective of network sizes. The number of selected defense nodes primarily depends on the user-specified cluster size of a defense node.

We next fix the number of nodes to 1000 and change the average node degree from 2 to 6. It can be seen from Figure 3b that the required number of defense nodes decreases with increasing average node degree. With larger average node degree the network becomes denser. Thus all nodes in the network are covered with a fewer number of defense nodes.

V. Conclusion and future work

In this paper we present a defense against distributed denial of service attacks. We propose a computationally light-weight approach to differentiate legitimate traffic from the attack traffic and maintain score for each client. We perform traffic policing using these scores to maintain appropriate rate-limiting. We then present initial ideas about systematic selection of locations to deploy defense nodes. We present experimental evaluation of the proposed algorithms using simulation experiments as well as live-traffic experiments. As part of future work, we plan to exploit strengths of different locations in the network to build different classes of the proposed defense. For instance, defense at source networks can perform sophisticated traffic differentiation due more computing resources and less traffic. On the other hand, defense victim network demands a very light-weight traffic differentiation mechanism.

References

[1] J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against ddos attacks. In *In Proceedings of Network and Distributed System Security Symposium*, 2002.

[2] A. D. Keromytis, V. Misra, and D. Rubenstein. Sos: Secure overlay services. In *In Proceedings of ACM SIGCOMM*, pages 61–72, 2002.

[3] Jelena Mirkovic, Max Robinson, and Peter Reiher. Alliance formation for ddos defense. In *In Proc. New Security Paradigms Workshop, ACM SIGSAC*, pages 11–18, 2003.

[4] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. In *In IEEE Symposium on Security and Privacy*, pages 130–143, 2004.

[5] X. Yang, D. Wetherall, and T. Anderson. A dos-limiting network architecture. In *In Proceedings of ACM SIGCOMM*, pages 241–252. ACM Press, 2005.