Poster: Protecting Campus Networks with Cost-effective DDoS Defense

Wei-Hsuan Chiang, Shu-Po Tung, Yu-Su Wang, I-Jen Hsiao, Hsu-Chun Hsiao
National Taiwan University, Taipei, Taiwan
{b04902077, b04902003, b04902014, b05902005, hchsiao}@csie.ntu.edu.tw

I. Introduction

Distributed Denial of Service (DDoS) attacks have been a serious threat to many public services [1], including those running on campus networks [2][3]. On November 1st, 2018, our department's mail server was attacked and the service was disrupted for several hours, causing widespread inconvenience. Despite sending low-volume attack traffic, the attack quickly saturated the mail server's connection state by exploiting an application-layer bottleneck [4][1].

However, protecting campus networks from DDoS is challenging due to two main reasons: limited budget and diverse applications managed by individual departments or laboratories. Since campus networks are often operated under tight budget, it might be infeasible to deploy DDoS defense solutions that require substantial upgrade of the infrastructure (e.g., upgrading the infrastructure to support software-defined networking) or that depend on expensive cloud-based or on-premise appliances. In addition, because each department or laboratory may host its own public services, it is difficult for the security operations center of the school or the upstream provider to detect attacks targeting application-layer bottlenecks.

In this work, we propose a cost-effective solution that is designed to mitigate application-layer DDoS attacks on campus networks. We observe that because campus networks (at the department or laboratory level particularly) are smaller and simpler than enterprise networks, we can achieve similar results of lightweight on-demand defenses [5], [6] without needing to change the infrastructure. Specifically, our proposed solution dynamically reroutes potential attack traffic for further inspection by **modifying ARP tables**, which can be reliably performed by administrators who have permission to control layer 2 switching.

Our preliminary results show that the proposed solution can successfully block and recover from the attack in 10 seconds, which we believe is sufficient for most department- or lab-operated services. We plan to deploy the proposed solution to protect critical services in our department. Additionally, we would like to incorporate other differentiation mechanisms (e.g., CAPTCHAs) in addition to whitelisting, and evaluate the effectiveness of the proposed solution using real attack traces we previously collected.

[1]We use Dovecot, an open source mail server in our department. Its default high-security mode limits the number of concurrent logged-in users to 100, which is stated to be sufficient for a small site.

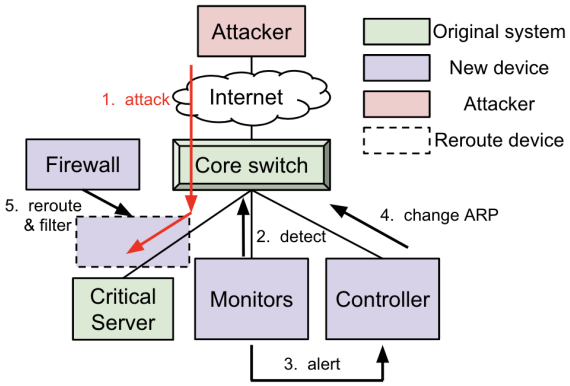

Fig. 1. System Structure

II. Threat Model

We consider non-volumetric DDoS attacks that exploit application-layer bottlenecks, or application-layer DDoS. This means that 1) network links are not congested, and 2) the upstream providers that are agnostic about the victim application may fail to detect the attack. We consider an external adversary who has no control over our infrastructure and can access the critical server only through public IPs.

III. System Design

Our goal is to protect critical services hosted on a campus network. These services can be reached externally via an L3 core switch. To protect these services, our system introduces three additional components, a **monitor**, a **switch controller**, and a **firewall**, as Fig. 1 illustrates. All the components can be implemented using commodity computers, which can be easily acquired and maintained.

When the monitor detects the critical service is unavailable, it alerts the controller of potential attacks. In response, the controller changes the ARP table on the core switch such that traffic to the critical service can be routed through the firewall for cleaning. The detailed workflow is as follows.

Detect & Alert: The monitor checks the availability of each critical service through both passive and active measures, and reports observed anomalies to the controller. For example, the monitor can keep track of the bandwidth usage, CPU usage, and the number of connections and send an alert if any of the values exceed a certain threshold. The monitor can also periodically ping the service and observe the response time.

Change ARP: Once receiving an alert, the controller will reroute all traffic originally bound for the critical server to the firewall by modifying the core switch's ARP table. That is, the ARP entry of the server IP (which was the server's MAC address) will be replaced with the firewall's MAC

address, such that the core switch will treat the firewall as the critical server, and traffic originally destinated to the critical server will be re-directed to the firewall. Compared with SDN-based re-direction approaches, our solution does not require infrastructure upgrade but is hard to scale beyond small-scaled networks. Compared with changing L3 routing, changing a single L2 ARP entry is relatively safe because misconfiguration affects only a single host rather than the entire subnet. Nevertheless, changing L2 switching requires the permission to access the corresponding switch, which may be inapplicable in some cases (e.g., the infrastructure is managed by another IT team).

Reroute & Filter: Finally, the firewall will be enabled to filter out potential attack traffic based on a certain filtering strategy. The filtering strategy can be pre-defined by the administrator or dynamically selected by the monitor according to the type of anomalies and the attacked service. As the critical server is now hidden behind the firewall and has no public address, the remaining traffic will be sent to the server via an isolated internal Virtual Local Area Network (VLAN). Since the firewall is dynamically deployed and can be shared among many services, it does not introduce extra latency during peace time, and the cost is lower than having a constantly working firewall.

IV. Evaluation

We have built a proof-of-concept system to evaluate our proposal. In our experiment, we use *pfSense* [7] as the firewall to filter out potential attack traffic. [2] We use *Zabbix* [8] to implement the network monitor, which detects whether the inbound traffic to critical servers is abnormal. If the inbound traffic exceeds a pre-defined threshold, *Zabbix* will trigger the controller to reroute the traffic. To simulate a burst of traffic like in DDoS, we use iPerf [9] to generate many concurrent flows. The inbound traffic of the critical server retrieved from *Zabbix* are shown in Fig. 2. [3]

There are three types of delay in our experiment: monitor delay, alert delay and reroute delay. Monitor delay is the reaction time for monitors to detect abnormal traffic, which is mostly caused by the false-positive prevention set on the monitor. Only if the average traffic of the last 5 seconds exceeds the threshold will the alert go off. After detecting vicious events, the time before the controller finishes the script is called alert delay, which is mainly caused by the connection and authentication delay between controller and core switch. Finally, changing the ARP table will temporarily disconnect normal users from the critical server. The time to re-connect to the server after the network modification is defined as reroute delay, which is almost negligible. Duration of each delay type is shown in TABLE I.

As Fig. 2 illustrates, the overall delay from the start of attack to service recovery takes about 10 seconds only, and most of them are monitor delay. Moreover, the reroute delay takes less than 0.1 seconds, implying that the process is nearly transparent to benign users, and after the recovery, malicious traffic is fully blocked as expected. The preliminary result shows the efficiency and effectiveness of the proposed system.

[2]In experiment, we treat traffic from IP located in the same organization with the critical server as benign, and others as malicious.

[3]We set the update interval of *Zabbix* to one second, the minimum value.

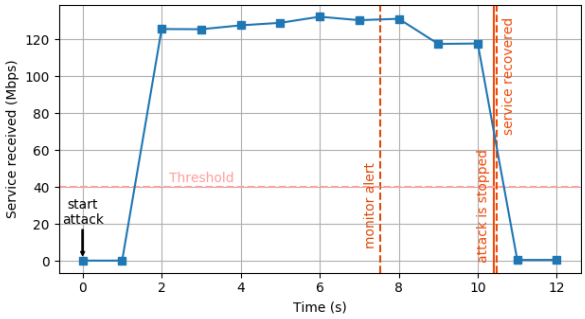

Fig. 2. Reaction against malicious traffic

TABLE I
Duration of Each Delay Type

	Monitor Delay	Alert Delay	Reroute Delay
Time (sec)	8.905	2.682	< 0.1

V. Conclusion & Future Work

Inspired by a real DDoS attack incident occurred in our department, we proposed a cost-effective, light-weight, and dynamic DDoS defense system that can be applied on small-scaled campus networks. With the proposed system, the attack can be mitigated by the server-side defense with little delay when the upstream fails to handle non-volumetric DDoS traffic. Thanks to its simplicity (no infrastructure change) and low cost, the proposed DDoS mitigation system can be easily deployed to campus networks, particularly at the department or laboratory level. The preliminary evaluation shows the feasibility of our proposal. Future work includes 1) further evaluation based on real attack traces, 2) deployment to the production environment, and 3) providing flexible interfaces to plug-in additional differentiation and filtering strategies for distinguishing benign and malicious traffic.

Acknowledgment

This work was supported by the Ministry of Science and Technology of Taiwan under grant MOST 108-2636-E-002-008.

References

[1] "Corero Half Year 2018 DDoS Trends Report" [Online]. Available: http://info.corero.com/rs/258-JCF-941/images/H1-2018-Corero-Trends-Report-Final.pdf
[2] "University of Albany Targeted with DDoS Attacks" [Online]. Available: http://www6.campuslifesecurity.com/Articles/2019/03/01/University-of-Albany-Targeted-with-DDoS-Attacks.aspx
[3] "Infinite Campus DDoS attack impedes access to student data" [Online]. Available: https://www.zdnet.com/article/infinite-campus-ddos-attack-impedes-access-to-student-data/
[4] "Dovecot v2.x Official Wiki - LoginProcess" [Online]. Available: https://wiki.dovecot.org/LoginProcess
[5] Seyed K. Fayaz, Yoshiaki Tobioka, Vyas Sekar and Michael Bailey, "Bohatei: Flexible and Elastic DDoS Defense", Usenix Security, 2015.
[6] L. Zhou, H. Guo, "Applying NFV/SDN in mitigating DDoS attacks", TENCON 2017 IEEE Region 10 Conference, pp. 2061-2066, 2017.
[7] CM Buechler and J. Pingle, "pfsense: The definitive guide", Reed Media Services, 2009.
[8] "Zabbix" [Online]. Available: https://www.zabbix.com/
[9] "iPerf 2 user documentation" [Online]. Available: https://iperf.fr/iperf-doc.php