Poster: DDoS Detection at an ISP

Rajat Tandon, Jelena Mirkovic
USC Information Sciences Institute
Email: tandon, sunshine@isi.edu

***Abstract*—Distributed denial-of-service (DDoS) detection and signature generation at an ISP level must be both light on resources, and accurate enough to minimize collateral damage to legitimate traffic. This is challenging because ISPs often serve many diverse clients, and cannot afford to build and maintain an accurate profile for each client.**

We propose Asym – a scalable, low-impact detection tool, which models information about volume and asymmetry of flows per destination. To save memory, this information is stored in a fixed-size hash map. Asym detects an attack when both volume and asymmetry measures exceed their historical values. We show how this approach helps detect both high-rate and low-rate attacks, and filter them with high accuracy.

I. Introduction

In high-volume distributed denial of service (DDoS) attacks many attackers generate excessive traffic that interferes with the victim's legitimate operation. The traffic often creates congestion at or near the victim or its ISP, which leads to legitimate traffic drops and delays.

DDoS attacks are very frequent (more than 140 per day) and some are also high-volume (e.g., 1.2 Tbps attack on Dyn in 2016). While they may last for days most are short-lived, lasting up to five minutes.

ISPs need mechanisms to quickly detect and profile for DDoS attacks, to generate accurate signatures for their filtering. The attacks' short duration and high frequency requires an automated detection and response. Yet, DDoS detection at an ISP is hard. An ISP may transit traffic for millions of customer devices. Each of these devices may become overwhelmed by attacks of different rate and type, depending on the device's own capabilities, and on its network's bandwidth. A typical approach to DDoS detection and profiling, which builds each destination's inbound traffic's profile and applies anomaly detection, would be prohibitively expensive at an ISP level. A coarser grained approach, though, may miss some attacks or may arrive at inaccurate signatures.

In this poster we present our approach to low-impact, scalable and accurate DDoS detection and profiling, called Asym. Inspired by AMON [1], we use per timestamp and per-destination matrix to store statistics about both inbound and outbound traffic, and we use hashing of destination addresses to limit memory cost. This structure is known as "AMON databrick". Our work improves on AMON, which uses only inbound traffic volume and packet counts to detect attacks. Our statistics include flow volume and flow asymmetry, where we define an asymmetric flow as unidirectional UDP or TCP flow, which lacks its pair. We regard asymmetric flows as signals of unwanted traffic, which may indicate unresponsive destination, scanning or DDoS attacks. We detect a DDoS attack when both inbound traffic volume and inbound flow asymmetry exceed their historical values. We then profile a limited number of asymmetric flows for the attack's target to devise attack signature.

We show how this approach helps us detect both low-volume and high-volume attacks, and produce limited number of alerts per day. We also show how our signatures help accurately filter attack traffic, with low collateral damage.

II. Methodology

In this section we detail our attack detection and profiling approach.

Data Structures. We use time series of *AMON databricks* [1] to store statistics about traffic, which we need to detect DDoS attacks. Each databrick contains information about volume and asymmetry of inbound traffic flows for the given time interval. A databrick is divided into *bins*, with multiple destinations being hashed into a single bin. We also store information about historical mean and standard deviation of volume and asymmetry metrics for each bin, over some past time interval (in our evaluation we use 1 hour).

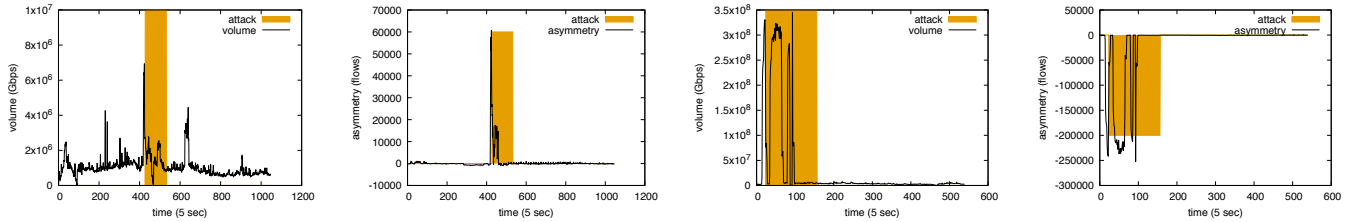
When we suspect an attack within a given bin, we sample flows that match the suspected attack in a limited-size per-bin array, called *samples*.

We devise *signatures* for these samples and store the best signature per destination bin. We then collect limited inbound and outbound flows matching this signature, and use them to estimate collateral damage.

Statistics Collection. We assume that we have either sampled or all flows that traverse a given ISP, where a flow is a standard Netflow (or Argus or SFlow) record containing information about source and destination IP addresses and ports, the transport protocol, TCP flags if any, number of packets and number of bytes. We use each flow to update our statistics in the following way.

We hash the source and destination addresses into two bins, using some standard hash algorithm. We experimented with

Network and Distributed Systems Security (NDSS) Symposium 2018
18-21 February 2018, San Diego, CA, USA
ISBN 1-1891562-49-5
http://dx.doi.org/10.054722/ndss.2018.23xxx
www.ndss-symposium.org



(a) Volume - SYN flood (b) Asymmetry - SYN flood (c) Volume - NTP refl. (d) Asymmetry - NTP refl

Fig. 1. Two attacks detected by Asym.

SHA-256 but converged on simpler and cheaper algorithm, where we combine second and third octet of the address and calculate modulus with the number of bins in a databrick.

We then add the flow's volume to the volume of the destination bin. If the flow is TCP flow with the PUSH bit set we do no update to asymmetry measure in the bin, i.e., TCP flows with PUSH bit set are always considered as wanted traffic. Otherwise, if it is a TCP flow or a UDP flow, we say its *asymmetry factor* is 1 if it is sent from a high-numbered port to a well-known service port. We say that the flow's *asymmetry factor* is -1 if it is sent from a well-known service port to a high-numbered port. We add to the destination bin the product of the flow's asymmetry factor and its packet count, and we subtract the same value from the source's bin. Over time, destinations that are not under attack tend to converge to asymmetry measures that are close to zero, since they respond to most service requests they receive, send requests for each response they receive, and engage mostly in productive TCP flows, which transfer payload in both directions.

Attack Detection. We collect anomalous points for a bin when both its volume and its asymmetry measure exceed their historical values, i.e., they exceed mean by a margin of more than five standard deviations. We require both of these measures to be anomalous because phenomena different than DDoS attack could elevate only one but not the other. For example, high-volume downloads could elevate volume but not asymmetry measure. Similarly, if a destination went down its asymmetry measure would increase but its volume measure would not.

We decrease the number of anomalous points (if positive) for a bin, when both its volume and its asymmetry measure are within their historical ranges.

We signal DDoS attack when the number of anomalous points exceeds some threshold – ATTACK_LOW. We signal the end of the attack when anomalous points reach zero.

Profiling and Signature Generation. Whenever a bin has positive anomalous score we sample flows that match this bin for eventual signature generation. Since we have limited sample memory, we must choose what to sample wisely. We sample those flows whose asymmetry matches the asymmetry of the detected attack (e.g., negative asymmetry measure for the detected attack will lead to sampling of service responses and not service requests). We replace a sampled flow with a new flow if its asymmetry measure is higher, or if it is more recent.

When the attack detection is signaled we profile the collected samples to devise a signature, which can explain the largest portion of the asymmetry measure, i.e., which covers the most flows with high asymmetry. We then use this signature to collect both inbound and outbound traffic that matches it into a limited-size memory, and to estimate collateral damage. We say that a flow matching the signature is *likely-legitimate* if its asymmetry measure is zero (e.g., TCP flow with payload) or if it has a reverse flow that matches it (e.g., a service request with a matching reply). Otherwise, the flow is labeled as *likely-attack*. We say that a signature is "good" if the ratio of matched *likely-legimate* flows to all matched flows is below a threshold (we use 5% in our evaluation).

III. Results

We illustrate our attack detection in Figure 1(a) and 1(b). Around time 400–450 there is a TCP SYN flood on a destination, which produces 60,000 packets per second. Although, this does not result in a large increase in volume we detect this flood because the volume increase corresponds to the asymmetry measure increase. Other spikes in volume around time 200 and 600 do not result in attack detection because the asymmetry measure is low.

Conversely, Figure 1(c) and 1(d) shows another attack, which has both high volume and high asymmetry. It is NTP reflection attack, that bombards the destination with more than 200,000 responses per second and 0.3 Gbps volume. We easily detect this attack.

For both attacks we produce very precise attack signatures, including the destination address and protocol, and the destination port (attack 1) and source port (attack 2). These signatures filter all of the attack, producing less than 5% of collateral damage.

IV. Conclusion

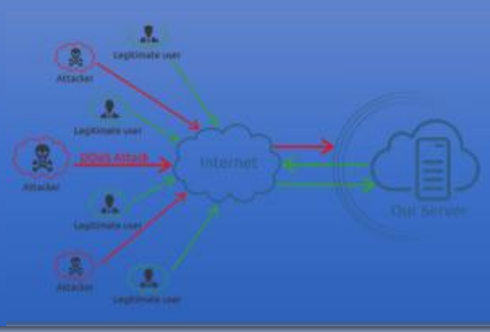
We present a novel DDoS detection method for ISPs and show it can accurately detect attacks, with low collateral damage.

References

[1] M. Kallitsis, SA. Stoev, S. Bhattacharya, G. Michailidis. AMON: An Open Source Architecture for Online Monitoring, Statistical Analysis, and Forensics of Multi-Gigabit Streams. IEEE Journal on Selected Areas in Communications, 2016

DDoS Detection at an ISP

Rajat Tandon, Jelena Mirkovic
USC Information Sciences Institute
Email: tandon, sunshine@isi.edu



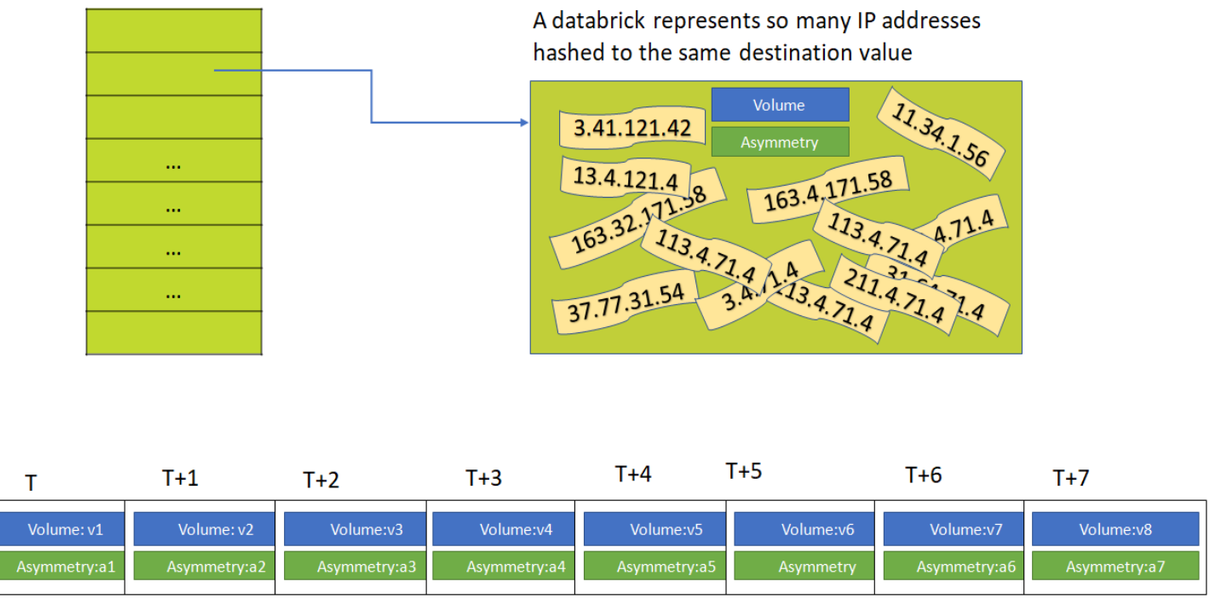
ABSTRACT

Asym – a scalable, low-impact detection tool models information about volume and asymmetry of flows per destination.

To save memory, this information is stored in a fixed-size hash map.

Asym detects an attack when both volume and asymmetry measures exceed their historical values.

This approach helps detect both high-rate and low-rate attacks, and filter them with high accuracy.



METHODOLOGY

We use time series of AMON databricks to store statistics about traffic. They contain information about volume and asymmetry of inbound traffic flows.

A databrick is divided into bins, with multiple destinations being hashed into a single bin.

We also store information about historical mean and standard deviation of volume and asymmetry metrics for each bin, over some past time interval (in our evaluation we use 1 hour).

$$Vol.dst = Vol.dst + Vol.flow \; ; \quad Asym.dst = Asym.dst + Asym.factor * Pkt.count$$

***Asym.factor* is 1 if the flow is sent from a high-numbered port to a well-known service port for TCP flows without the PUSH flag and UDP flows.**

***Asym.factor* is -1 if the flow is sent from a well-known service port to a high-numbered port for TCP flows without the PUSH flag and UDP flows.**

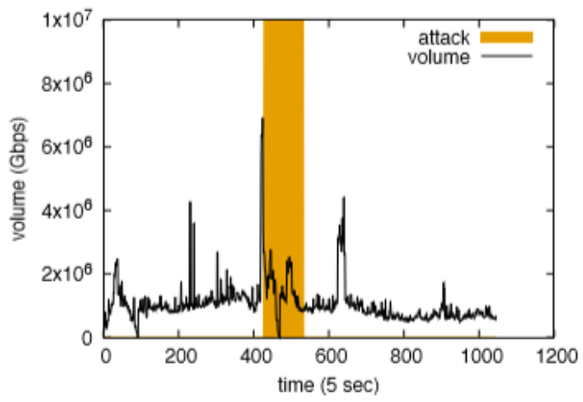
***Asym.factor* is 0 for TCP flows with the PUSH flag set.**

DDoS Detection

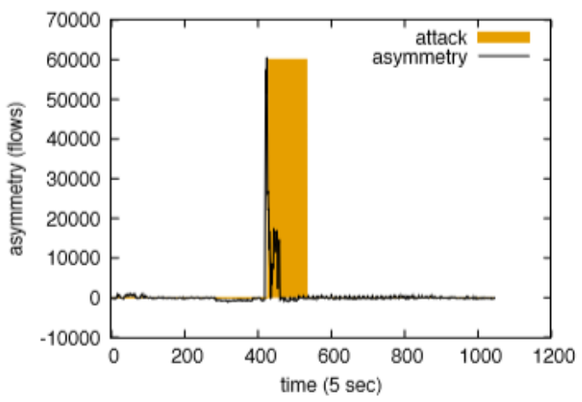
We collect anomalous points for a bin when both its volume and its asymmetry measure exceed their historical values

We require both of these measures to be anomalous because phenomena different than DDoS attack could elevate only one but not the other

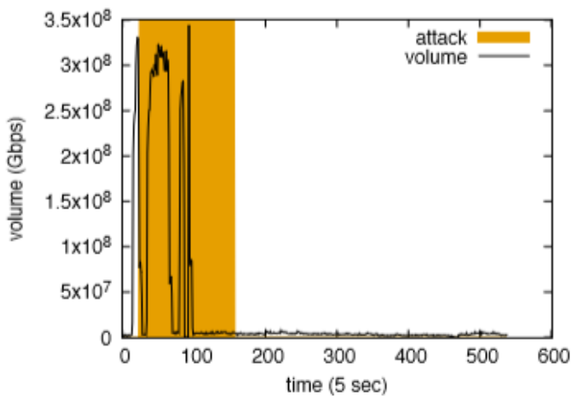
We signal DDoS attack when the number of anomalous points exceeds some threshold.



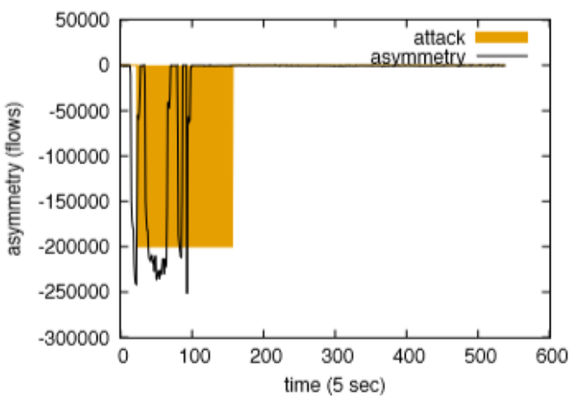
(a) Volume - SYN flood



(b) Asymmetry - SYN flood



(c) Volume - NTP refl.



(d) Asymmetry - NTP refl