Reinforcement Learning-Controlled Mitigation of Volumetric DDoS Attacks

Hauke Heseding[1]

Abstract:

This work introduces a novel approach to combine *hierarchical heavy hitter* algorithms with *reinforcement learning* to mitigate evolving volumetric distributed denial of service attacks. The goal is to alleviate the strain on the network infrastructure through early ingress filtering based on compact filter rule sets that are evaluated by fast ternary content-addressable memory. The reinforcement learning agents task is to maintain effectiveness of established filter rules even in dynamic traffic scenarios while preserving limited memory resources. Preliminary results based on synthesized traffic scenarios modelling dynamic attack patterns indicate the feasibility of our approach.

Keywords: Distributed denial of service; software defined networks; hierarchical heavy hitters; reinforcement learning

1 Introduction

Distributed denial of service (DDoS) attacks pose a constant and severe threat to communication infrastructures. Particularly, volumetric DDoS attacks, which congest bottleneck links near a target system with unsolicited traffic, have become increasingly popular. To mitigate such attacks, we seek to achieve early attack traffic removal directly in the data plane. For this, we apply ingress filter rules that can be evaluated by ternary content-addressable memory (TCAM) in a single clock cycle enabling traffic filtering at line speed. This alleviates the strain on the network and protects downstream systems (including systems conducting further mitigation steps). Furthermore, we address the following question: How to handle dynamic traffic scenarios, where filter rules may become outdated, and how to balance filter rule effectiveness against TCAM utilization? Intelligent attackers may evade outdated rules by altering attack traffic composition. Also, unnecessarily fine-granular rules are undesirable since TCAM capacity is limited by high monetary costs and energy consumption.

To keep filter rules up to date, we monitor the data stream passing the ingress filter with hierarchical heavy hitter (HHH) algorithms that enable detection of suspicious IP subnets sending excessive traffic in volumetric DDoS scenarios. Recent advances enable direct integration of HHH algorithms into the data plane (e. g., [PAM17, Si17, Be20, Zh21]). To

[1] Karlsruhe Institute of Technology (KIT), KASTEL, Institute of Telematics, Kaiserstraße 40, 76133 Karlsruhe, Germany hauke.heseding@kit.edu

doi:10.18420/sicherheit2022_20

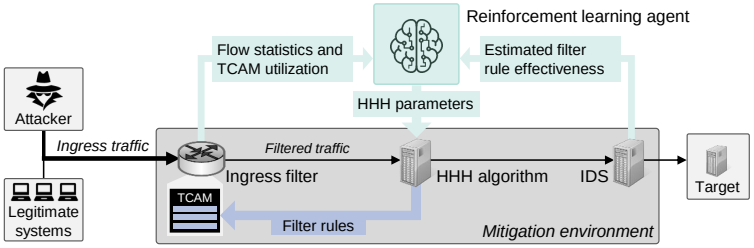

Fig. 1: Architecture components and workflow

achieve adaptivity to counteract intelligent attacks we leverage reinforcement learning (RL) to distinguish between highly distributed and densely clustered attack sources. This allows adjusting filter rule granularity accordingly (via parameterization of HHH algorithms) to avoid unnecessarily fine-granular rules. In essence, the RL agent serves to maintain the balance between filter rule effectiveness and TCAM utilization when traffic patterns change.

Threat model This work focuses on dynamic, volumetric DDoS attacks. To address intelligent attacker behavior, we first consider different volumetric attacks: direct botnet attacks (generating elephant flows) and amplification attacks (DNS, NTP, or SSDP reflection). Distribution of attack traffic sources and traffic intensity depend on the chosen attack vector. While bots typically have more compact distributions, reflectors yield higher per-node attack traffic volume. Second, attackers have the ability to change attack traffic composition by employing various attack vectors at different times during an ongoing attack.

Mitigation objectives The overall goal is to achieve fast and early attack traffic removal (by leveraging TCAM capabilities). We also seek to counteract evolving attacks by keeping filter rules up to date, preventing intelligent attackers from circumventing defenses. In essence, filter rules should maintain high precision (to preserve legitimate traffic) and sensitivity (to capture attack traffic) in dynamic traffic situations. Adaptation to evolving volumetric DDoS scenarios was recently studied in related work, focusing on programmable data plane (PDP) technologies [Zh20, Li21] as well as leveraging RL [MK15, SRP20]. In contrast to previous work, we apply rule-based filtering based on aggregated traffic features immediately at the ingress, to alleviate the strain on network infrastructure and downstream systems. This avoids resource-intensive state-keeping of per-flow mitigation, offers more fine-grained control than per-router throttling and does not require sophisticated PDP technologies.

2 Adaptive Ingress Filtering

Our strategy to keep filter rules up to date is twofold: (a) continuous traffic monitoring via an HHH algorithm and (b) adjusting filter rule granularity through RL. For this, our

mitigation system comprises four core components: a TCAM-based ingress filter, an HHH algorithm instance, a downstream IDS, and an RL agent (see Fig. 1). The agent interacts with the mitigation environment at discrete time steps to apply chosen parameters when the HHH algorithm is queried for filter rules (query time).

TCAM-based ingress filter The ingress filter is placed on a switch with TCAM resources, positioned upstream at an ingress point of a larger network (e. g., a backbone network). It applies a set of filter rules $\mathcal{R}$ obtained from the HHH algorithm. Each rule specifies an IP subnet. Any packet whose IP source address matches a subnet contained in $\mathcal{R}$ is removed from the ingress data stream. Due to TCAM technology, evaluating the entire set $\mathcal{R}$ requires only a single clock cycle. The second task of the ingress filter is to keep track of the number of removed packets for subsequent estimation of filter rule effectiveness.

HHH algorithm Packets passing the ingress filter are subsequently monitored by an HHH algorithm (executed on a server in proximity to the IDS) The HHH algorithm tracks the hierarchical distribution of source IP addresses to identify IP subnets sending excessive amounts of traffic. At query time a *frequency threshold* ϕ is applied during HHH computation indicating the minimum number of packets necessary to classify an IP subnet as potentially malicious. To avoid excessive hierarchical aggregation during HHH computation we restrict the maximum size of IP subnets accepted as filter rules since HHHs become less expressive with increasing aggregation. Indiscriminately adopting large subnets would lead to unwarranted removal of significant portions of the ingress traffic. To prevent this, we introduce a *hierarchy threshold* $H^{\max}$ limiting filtered subnet size to no more than $2^{H^{\max}}$. Together, the frequency and hierarchy thresholds ϕ and $H^{\max}$ govern filter rule granularity and can be adjusted to match evolving attack traffic patterns.

Reinforcement learning agent We employ deep reinforcement learning to learn the complex, non-linear relationship between HHH parameters, traffic characteristics, generated filter rules and rule effectiveness in a model-free fashion. For this, we train a Deep Q-Network (DQN) on simulated dynamic traffic scenarios to select effective thresholds ϕ and $H^{\max}$. DQNs use deep neural networks to approximate an optimal action-value function Q^* [Mn15], i. e., to learn a policy that maximizes cumulative reward. In our case, the agents objective is to achieve high precision and recall, while minimizing false positive ratio (FPR) and the number of generated filter rules. At query time t, the agent acts by selecting values for ϕ and $H^{\max}$ from its (discrete) action space. The choice is based on the observed mitigation environment state $s^{(t)}$, which comprises indicators for TCAM utilization, filter rule distribution and granularity, as well as filter rule effectiveness. Tab. 1 provides an overview of the (most important) elements of the state and action spaces. The mitigation objectives are conveyed by a reward function $\mathfrak{r} = \mathfrak{r}_\mathrm{p} \cdot \mathfrak{r}_\mathrm{s} \cdot \mathfrak{r}_\mathrm{f} \cdot \mathfrak{r}_\mathrm{r}$, where each partial function $\mathfrak{r}_\mathrm{p}, \mathfrak{r}_\mathrm{s}, \mathfrak{r}_\mathrm{f}, \mathfrak{r}_\mathrm{r}$ constitutes a weighted mapping of precision, sensitivity, FPR,

State $s^{(t)}$	Meaning
$s_{\mathrm{l}}^{(t)} \in \mathbb{N}^{32}$	Number of HHHs detected at different hierarchy levels
$s_{\mathrm{d}}^{(t)} \in [0,1]$	Indicator for distribution and size of filtered IP regions
$s_{\mathrm{p}}^{(t)} \in [0,1]$	Estimated filter precision
$s_{\mathrm{s}}^{(t)} \in [0,1]$	Estimated filter sensitivity
$s_{\mathrm{f}}^{(t)} \in [0,1]$	Estimated FPR
$s_{\mathrm{r}}^{(t)} \in \mathbb{N}$	Number of generated filter rules

Action space	ϕ, $H^{\max}$-values
$A = \bigcup_{1 \le i \le 25, 16 \le j \le 24} (\phi_i, H_j^{\max})$	$\phi = i \cdot 0.01, H_j^{\max} = j$

Partial reward function	Weighting
$\mathfrak{r}_{\mathrm{p}}(x) : [0,1] \to [0,1]$	$x \mapsto x^1$
$\mathfrak{r}_{\mathrm{s}}(x) : [0,1] \to [0,1]$	$x \mapsto x^{1.5}$
$\mathfrak{r}_{\mathrm{f}}(x) : [0,1] \to [0,1]$	$x \mapsto (1-x)^{2.0}$
$\mathfrak{r}_{\mathrm{r}}(x) : \mathbb{N} \to \mathbb{R}$	$x \mapsto 1 - 0.04 \cdot \log_2(x)^{0.2}$

Tab. 1: Excerpt of state, action space, and reward function parameters

and number of generated filter rules (resp.) to scalar values within comparable ranges. By tuning the partial reward functions (see Tab. 1), the agent can learn to realize different trade-offs (e. g., emphasize high precision over small filter rule sets or vice versa).

Downstream IDS In order to provide the DQN agent with feedback on achieved filter precision, sensitivity and FPR (see state $s^{(t)}$ in Tab. 1), we currently apply an oracle IDS that serves to reflect capabilities of a traditional IDS (distinguishing attack and legitimate traffic). Precision, sensitivity, and FPR are estimated from ingress filter statistics (number of removed packets), traffic passing the ingress filter as well as sampled traffic.

Sampled traffic Since early traffic removal prevents downstream systems from monitoring discarded traffic, it hinders their ability to determine traffic distribution and filter rule effectiveness. To address this issue, the ingress filter excludes a fraction of the ingress traffic from filter rule application through sampling. This allows downstream systems to estimate traffic distribution (HHH algorithm) and filter rule effectiveness (IDS) based on sampled traffic. Furthermore, it prevents oscillation of filter rules, since the HHH algorithm would otherwise exclude traffic filtered at the ingress from its estimation of the traffic distribution.

3 Preliminary Results

We evaluate the ability of the RL agent to learn effective HHH parameters based on synthetic, dynamic traffic scenarios. Each scenario constitutes a training episode that models and randomizes the activity of legitimate and attack traffic sources represented by IPv4 addresses over 300 discrete time indices. The activity of legitimate traffic sources is uniformly distributed over time and normally distributed over an address space of size 2^{16}. Attack traffic sources use more dynamic patterns. An episode is divided into four (partially overlapping) phases, each selecting active attack traffic sources from different subnets to distinguish activity of densely clustered sources (e. g., a high number of bots located in an IPv4 subnet) and sparse but widely distributed sources (such as reflectors). Phase one and three use densely clustered normally distributed traffic sources, phase two changes the traffic

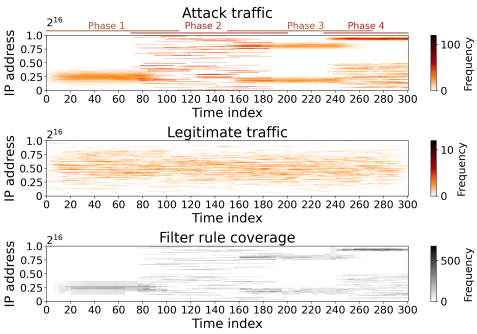

Fig. 2: Snapshot of traffic source activity and generated filter rule coverage during a traffic scenario.

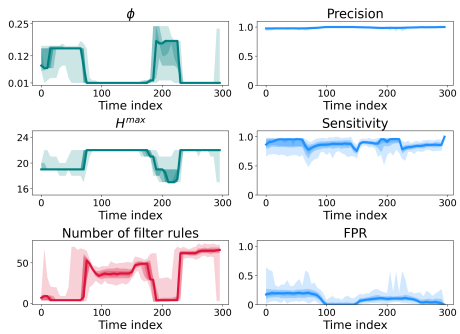

Fig. 3: Parameters (ϕ, $H^{\max}$), precision, sensitivity, FPR, and filter rule count (last 250 episodes).

pattern to few high-intensity sources uniformly distributed over the entire address space, while phase four combines densely clustered sources with sparsely distributed high-intensity sources. During a training episode, an RL agent adapts frequency and hierarchy thresholds (ϕ, $H^{\max}$) every five time indices, while the entire training process consists of 80000 adaptation steps. Fig. 2 provides an example snapshot of a synthesized scenario and corresponding filter rule coverage from the end of RL training. The distribution of choices for frequency and hierarchy thresholds, number of generated filter rules, precision, sensitivity, and FPR during the last 250 randomized training episodes is outlined in Fig. 3 (min-max, 10%-90%, 25%-75%, 40%-60% quantiles, and median depicted with increasing shading).

In phase one, attack traffic sources are clustered in a small region in the lower IP address range. Consequently, the agent adapts thresholds ϕ and $H^{\max}$ to generate fewer coarse-grained rules that are sufficient to cover the corresponding address range (see Fig. 2 and Fig. 3). After transitioning to fewer, widely distributed high-intensity attack traffic sources (phase two), the agent emphasizes fine-grained rules to account for the sparse distribution. It reduces frequency threshold ϕ and increases hierarchy threshold $H^{\max}$ to maintain high precision and sensitivity as well as low FPR (time index 100-150). Hence, the agent manages to distinguish between distributions and adjusts filter rules to match attack traffic patterns.

During phase three, ϕ and $H^{\max}$ are again chosen to emphasize fewer rules that are sufficient to capture the two coherent regions with active attack traffic sources (time index 190-225). Finally, the agent chooses low ϕ and high $H^{\max}$ in phase four to apply fine-grained filter rules to the sparsely distributed attack traffic sources (lower half of the address space) as well as the densely clustered sources (upper address space). By emphasizing fine-granular rules in this hybrid phase, the agent maintains low FPR at the cost of more rules. The same applies when transitioning between different phases. Accepting more filter rules in these cases is in-line with the distribution of attack traffic sources and the mitigation goals conveyed by the reward function, since coarse-grained filter rules would necessarily have a strong negative impact on FPR during these periods and, thus, yield lower overall reward.

Acknowledgment

This work was supported by funding of the Helmholtz Association (HGF) through the Competence Center for Applied Security Technology (KASTEL) (POF structure 46.23.01: Methods for Engineering Secure Systems).

Bibliography

[Be20] Ben Basat, Ran; Chen, Xiaoqi; Einziger, Gil; Rottenstreich, Ori: Designing Heavy-Hitter Detection Algorithms for Programmable Switches. IEEE/ACM Transactions on Networking, 28(3):1172–1185, 2020.

[Li21] Liu, Zaoxing; Namkung, Hun; Nikolaidis, Georgios; Lee, Jeongkeun; Kim, Changhoon; Jin, Xin; Braverman, Vladimir; Yu, Minlan; Sekar, Vyas: Jaqen: A High-Performance Switch-Native Approach for Detecting and Mitigating Volumetric DDoS Attacks with Programmable Switches. In: USENIX Security Symposium. 2021.

[MK15] Malialis, Kleanthis; Kudenko, Daniel: Distributed Response to Network Intrusions Using Multiagent Reinforcement Learning. Eng. Appl. Artif. Intell., 41(C):270–284, May 2015.

[Mn15] Mnih, Volodymyr; Kavukcuoglu, Koray; Silver, David; Rusu, Andrei A; Veness, Joel; Bellemare, Marc G; Graves, Alex; Riedmiller, Martin; Fidjeland, Andreas K; Ostrovski, Georg et al.: Human-level control through deep reinforcement learning. nature, 518(7540):529–533, 2015.

[PAM17] Popescu, Diana Andreea; Antichi, Gianni; Moore, Andrew W.: Enabling Fast Hierarchical Heavy Hitter Detection Using Programmable Data Planes. In: Proceedings of the Symposium on SDN Research. SOSR '17, Association for Computing Machinery, New York, NY, USA, p. 191–192, 2017.

[Si17] Sivaraman, Vibhaalakshmi; Narayana, Srinivas; Rottenstreich, Ori; Muthukrishnan, S.; Rexford, Jennifer: Heavy-Hitter Detection Entirely in the Data Plane. In: Proceedings of the Symposium on SDN Research. SOSR '17, Association for Computing Machinery, New York, NY, USA, p. 164–176, 2017.

[SRP20] Simpson, Kyle A.; Rogers, Simon; Pezaros, Dimitrios P.: Per-Host DDoS Mitigation by Direct-Control Reinforcement Learning. IEEE Transactions on Network and Service Management, 17(1):103–117, 2020.

[Zh20] Zhang, Menghao; Li, G.; Wang, Shicheng; Liu, Chang; Chen, Ang; Hu, Hongxin; Gu, Guofei; Li, Qi; Xu, Mingwei; Wu, Jianping: Poseidon: Mitigating Volumetric DDoS Attacks with Programmable Switches. In: NDSS. 2020.

[Zh21] Zhang, Yinda; Liu, Zaoxing; Wang, Ruixin; Yang, Tong; Li, Jizhou; Miao, Ruijie; Liu, Peng; Zhang, Ruwen; Jiang, Junchen: CocoSketch: High-Performance Sketch-Based Measurement over Arbitrary Partial Key Query. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. SIGCOMM '21, Association for Computing Machinery, New York, NY, USA, p. 207–222, 2021.