Sieve: Layered Network Defenses against Large-Scale Attacks

Sophia Yoo, Satadal Sengupta, Maria Apostolaki, Jennifer Rexford
Princeton University

In recent years, network attacks have exploded in frequency, scale and sophistication [5, 4, 6, 12, 3, 9]. Benign traffic volumes are also increasing exponentially [11], with many applications requiring increasingly stringent performance guarantees. With all these trends, network providers urgently require scalable defenses (handling large amounts of benign and adversarial traffic) that provide both **accuracy** (i.e., *correctly* identifying, blocking, and serving traffic) and **performance** (i.e., *quickly* dropping adversarial traffic to ensure low latency for benign clients).

Recently, many works have sought to capitalize on performance opportunities offered by high-speed programmable hardware, proposing hybrid hardware-software designs that offload network functions (NFs) from x86 servers to the data plane (e.g., P4-based architectures) [8, 10, 13, 15, 2]. These works propose *cache-style* designs that *exactly* partition NFs between hardware and software. Portions of an NF that would benefit from high-speed packet-processing are offloaded to dataplane hardware, creating a cache-style fastpath for optimizing performance, while the non-offloaded portion of the NF remains on the server and acts as the processing slowpath [13]. With such cache-style approaches, only traffic that matches on the subset of the NF in the dataplane "cache" can benefit from cache hits. This optimizes performance for application traffic in the *best-case* scenario with little to no attacks, but leaves a critical gap in the context of modern security applications. Attacks cannot be predicted to lie within the coverage of a dataplane cache, and such approaches would degrade quickly under *worst-case* scenarios with large-scale attacks from sophisticated adversaries. Even NF offloading solutions which optimize the fastpath for defenses against large-scale attacks often result in compromising the performance of benign traffic [7, 14].

In this work, we propose **Sieve**, a novel approach to hybrid hardware-software designs that targets adversarial settings to faithfully provide both accuracy and performance. Unlike prior, cache-style *exact* offloading designs, Sieve works by *refactoring* a defense NF into multiple highly optimized layers; each layer is a refinement of the one before and is deployed on the target (e.g., P4-programmable switch, SmartNIC, eBPF kernel subsystem, x86 CPU) that offers the best performance and resource usage, along with the desired accuracy guarantees. Sieve begins with coarse-grained *approximations* of defense functionality instantiated on high-speed, but memory-limited hardware and moves towards fine-grained *exact* instantiations in memory-rich, but slower software. The key difference between cache and sieve approaches to hybrid NFs is that caches *exactly offload* a portion of an NF's functionality to hardware, while Sieve uses *approximate, refactored* instantiations of potentially *redundant* functionality across multiple hardware and software layers. Our key insight is that a hybrid NF can embrace accuracy and performance tradeoffs on different targets to provide defense in depth, accepting some approximation error on early, coarse-grained, high-speed hardware layers, but delivering complete accuracy by the deepest, exact, slower software layer.

We observe that benign and adversarial traffic can often be identified with per-flow state (e.g., connection 5-tuple) or some kind of computed identity verification (e.g., SYN cookie check). With Sieve, a P4 switch can be tasked with the first layer of the defense, *approximately* identifying and dropping the majority of adversarial traffic early in the path, while ensuring high-speed processing for benign traffic. Because the switch layer is memory-limited and keeping exact state is expensive, state can be kept using probabilistic data structures, engineered to allow a small amount of either false positives or false negatives to fulfill a given defense policy. For example, a Bloom filter can be used as a data structure at the Sieve switch, never dropping benign traffic but accepting some adversarial traffic as false positives, which are handled at a later defense layer [1]. Any traffic that was erroneously passed through the first layer is then detected and correctly dropped at the later layers, providing overall defense accuracy along with high performance. In other words, the Sieve's first layer accepts all wanted traffic while *quickly but coarsely* blocking the bulk of unwanted traffic, possibly allowing a small amount of unwanted traffic to trickle down to the next layer where a finer-grained exact netting *accurately* stops the remainder of the unwanted traffic.

We believe that Sieve is an exciting new direction for unleashing accurate and performant hybrid defenses. Early explorations using Sieve for DDoS defenses have shown vast improvements over prior cache-like offloading approaches, reducing end-to-end application latency by 48-84% and reducing server CPU overhead by 33-100%. Our vision is to create a system that automatically generates network defenses with Sieve. A Sieve compiler would take as inputs the defense NF written in a high-level language, topology of available hardware and software targets, and defense policy, and it would then output a refactored, layered defense with overall accuracy and performance.

References

[1] Bloom filter false positive rate. https://en.wikipedia.org/wiki/Bloom_filter, 2004.

[2] Xiang Chen, Hongyan Liu, Dong Zhang, Zili Meng, Qun Huang, Haifeng Zhou, Chunming Wu, Xuan Liu, and Qiang Yang. Automatic performance-optimal offloading of network functions on programmable switches. *IEEE Transactions on Cloud Computing*, page 1–1, 2022.

[3] Catalin Cimpanu. AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever. https://www.zdnet.com/article/aws-said-it-mitigated-a-2-3-tbps-ddos-attack-the-largest-ever/, 2020.

[4] Nick Galov. 39 Jaw-Dropping DDoS Statistics to Keep in Mind for 2022. https://hostingtribunal.com/blog/ddos-statistics/#gref, 2022.

[5] Help Net Security. DDoS attacks increase 341% amid pandemic , 2021.

[6] Nivedita James. 45 Global DDOS Attack Statistics 2023. https://www.getastra.com/blog/security-audit/ddos-attack-statistics/, 2023.

[7] Zaoxing Liu, Hun Namkung, Georgios Nikolaidis, Jeongkeun Lee, Changhoon Kim, Xin Jin, Vladimir Braverman, Minlan Yu, and Vyas Sekar. Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric DDoS attacks with programmable switches. In *USENIX Security Symposium*, 2021.

[8] Francisco Pereira, Gonçalo Matos, Hugo Sadok, Daehyeok Kim, Ruben Martins, Justine Sherry, Fernando M. Ramos, and Luis Pedrosa. Automatic generation of network function accelerators using component-based synthesis. *Proceedings of the Symposium on SDN Research*, 2022.

[9] Mário Pinho. AWS Shield threat landscape review: 2020 year-in-review. https://aws.amazon.com/blogs/security/aws-shield-threat-landscape-review-2020-year-in-review/, 2021.

[10] Yiming Qiu, Jiarong Xing, Kuo-Feng Hsu, Qiao Kang, Ming Liu, Srinivas Narayana, and Ang Chen. Automated smartnic offloading insights for network functions. *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles CD-ROM*, 2021.

[11] Amazon Web Services. AWS Best Practices for DDoS Resiliency. In *AWS Whitepaper*, 2022.

[12] Omer Yoachimik. Cloudflare thwarts 17.2M rps DDoS attack — the largest ever reported . https://blog.cloudflare.com/cloudflare-thwarts-17-2m-rps-ddos-attack-the-largest-ever-reported/, 2021.

[13] Kaiyuan Zhang, Danyang Zhuo, and Arvind Krishnamurthy. Gallium: Automated software middlebox offloading to programmable switches. *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication*, 2020.

[14] Menghao Zhang, Guanyu Li, Shicheng Wang, Chang Liu, Ang Chen, Hongxin Hu, Guofei Gu, Qianqian Li, Mingwei Xu, and Jianping Wu. Poseidon: Mitigating volumetric DDoS attacks with programmable switches. In *Network and Distributed System Security Symposium*, 2020.

[15] Zhipeng Zhao, Hugo Sadok, Nirav Atre, James C. Hoe, Vyas Sekar, and Justine Sherry. Achieving 100gbps intrusion prevention on a single server. *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation*, 2020.