Poster: P4DME: DNS Threat Mitigation with P4 In-Network Machine Learning Offload

Anonymous Author(s)

ABSTRACT

The ever-evolving cybersecurity landscape demands innovative solutions to safeguard critical network infrastructure such as the Domain Name System (DNS). This paper presents *P4DME*, a novel approach that harnesses the potential of Machine Learning (ML) in conjunction with P4 programmable switches to tackle DNS threats efficiently. *P4DME*'s primary benefit lies in offloading filtering from resource-intensive ML processing tasks on dedicated servers. This offloading boosts the overall traffic throughput that can be inspected or achieves the same throughput with reduced resource consumption while preserving the servers' capabilities for high-performance threat identification. This work uses P4-based in-network elements to handle crucial DNS threats, dynamic white and blacklisting, and an online popularity-based anomaly detection heuristic. The latter serves as a trigger for dedicated ML-based inspection. Furthermore, we introduce in-network mitigation filters updated through the control plane to provide adaptable and responsive threat mitigation. Preliminary simulation results show more than 99.9% offload ratio at 5% increased False Negative Ratio.

CCS CONCEPTS

• **Security and privacy → Network security**;
Intrusion/anomaly detection and malware mitigation.

KEYWORDS

DNS Security, P4, Programmable Networks, Machine Learning, In-Network Computation, Offloading

ACM Reference Format:

Anonymous Author(s). 2023. Poster: P4DME: DNS Threat Mitigation with P4 In-Network Machine Learning Offload. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Domain Name System (DNS) is a critical part of the Internet's infrastructure that has been used both as an attack vector and as a malware communication channel [3, 8]. Botnets, consisting of compromised machines (bots) controlled through a command and control (C&C) channel, have become one of the most prevalent cybersecurity threats over the last decade [2, 11]. These networks enable various malicious activities, including information theft, spamming, phishing, and launching DDoS attacks[16].

Botnets like Conficker [31], Kraken [32], and Torpig [36] utilize DNS for C&C. They employ Domain Generation Algorithms (DGA) to generate a large pool of domain names, from which only a small subset is used to communicate with the infected machines at each moment. As the domains have temporal validity, the need to synchronize such a large network leads to temporal spikes in DNS activity.

A particularly harmful DGA-related attack is Water Drop Torture attacks (WDT) [8], in which bots flood DNS systems with numerous queries to a target domain and nonexistent random subdomains. The cache and authoritative servers waste resources trying to solve the queries and may become overwhelmed and unresponsive [8]. WDT attacks have achieved up to 1.2 Tbps of DNS traffic, causing accessibility problems to many well-known sites [6].

Identifying malicious domain names is challenging. Traditional alphanumeric and blacklisting methods face difficulties coping with the multitude of domains generated by DGAs. Modern DGA detection relies on machine learning-based (ML) analysis of the domain names [35] or temporal characteristics of the botnet's behaviour [22], achieving a prediction performance of at least 90% [9, 25, 26].

Nevertheless, the servers deploying the ML models typically require significant resources. Thus, they can become the throughput bottleneck and be susceptible to further attacks.

In-network computations on programmable switches can partially offload the ML servers from their intensive tasks, improving the overall system throughput without sacrificing its security. The P4 language can be used to target a wide variety of devices. Moreover, it has proven to be useful for in-network security applications: [7, 10, 12, 13, 15, 17, 27, 28, 33, 39, 40]. None of these works addresses DNS protocol-specific attacks nor leverages the capabilities of external machine learning devices.

Processing the DNS protocol is also widely studied. Many data plane DNS parsers exist [5, 19, 20, 38], which could be leveraged by *P4DME* to extract information from queries. In addition, [4, 18] address in-network DNS security, although filtering based on predefined heuristics.

Contributions. We propose *P4DME* (**P4 D**NS threat **M**iti- gation **E**ngine), a system where the ML servers and the programmable switch collaborate. The latter performs fast filtering and anomaly detection tasks in-network, and forwards "suspicious" queries to the ML servers and the rest to the DNS servers. The ML servers throughly analyze the queries and take a forward or drop decision. Finally, the ML servers can update the programmable switch's filters and parameters through the control plane to keep a reliable operation.

P4DME's novelty lies in bringing together in-network computing and machine learning methods to jointly to address threats in the DNS space by naturally distributing the detection components. The

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

main result is a technique capable of enhanced resource usage without compromising the system's security. Finally, the technique can be extended to other threats and protocols.

The paper is outlined as follows. Section 2 describes the main concepts and system design. Section 3 provides further details on the covered use cases. A brief preliminary performance evaluation based on synthetic traffic queries is presented in Section 4. Section 5 provides the conclusions and future work.

2 P4DME DESIGN

P4DME is composed of two main blocks (Figure 1): a programmable switch and a set of Machine Learning servers, both located close to the DNS servers and capable of filtering malicious queries. Upon each incoming query, the programmable switch decides between forwarding the query to a known DNS resolver, dropping it or forwarding it to the ML servers for inspection. In turn, the ML servers can drop or forward the query and provide feedback to the switch, updating its access lists accordingly. The feedback mechanism is implement through the control plane of the programmable switch resulting in adding or removing entries from match-action tables or updating register values. The data plane parses the DNS query packets and decides if they are suspicious.

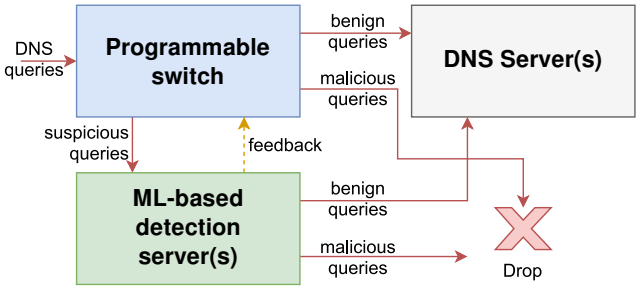

Figure 1: System overview.

Parsing the DNS packets can be efficiently done using the method proposed by Kaplan et al. [19]. They build on the approach of P4DDPI [5] and provide a solution to parse most DNS packets without destroying them. Their method makes it possible to do custom calculations before the packet leaves the networking device. They also provide the hash of the DNS labels, which makes it suitable for collecting statistics.

Traditional solutions. After having the hash of the DNS labels, one can choose from a variety of methods the P4 community offers to detect anomalies. Our main assumption is that malicious activity results in an increase in certain domain requests. This can be monitored using an array of *meters*, available in most P4 targets as an extern. The parameters can be changed from the control plane during runtime. If a query gets marked red (or even yellow), *P4DME* forwards it for further investigation. Using *counters* instead of *meters* provides an opportunity to periodically read domains' statistics and take action based on their distribution. Likewise, sketches [23, 30] can also be employed effectively within the data plane to keep track of domain occurrences.

Heuristic. Besides using "traditional" solutions, one can develop a custom heuristic. This is likely necessary since the attack patterns change over time. *P4DME* proposes a simple heuristic algorithm to detect the most popular domains. Using the hash of different domains, we can keep track of their popularity p. We introduce a register value c and a threshold t. If $p > c$, we increase the value of c by a fixed amount s. Otherwise, we decrease it by the same amount. The value of c is expected to be close to the average popularity. Queries to domains whose popularity p is above $c + t$ are sent to the ML servers for further investigation. The s and t values are initialised at the beginning of the pipeline using a match-action table; thus, their value can be changed during runtime. Note that this design can be implemented on the Intel Tofino with the proper use of the *LPF externs* and *RegisterActions*. Further details are provided in Section 4.

3 USE CASES

We present how *P4DME* can address two DNS-based threats. For both, *P4DME* can deploy any state-of-the-art detection scheme (or combination thereof) on dedicated ML servers to benefit from their specific abilities. In this fashion, a rich traffic pattern analysis can be performed while filtering at high throughput.

3.1 DGA-based botnet activity mitigation

To establish C&C channels with their bots, Botnets employ DGA to generate domain names to register frequently. Most DGA detection schemes rely on machine learning to compare the alphanumeric characteristics of the automatically generated names against those created by humans. Models range from traditional supervised ML techniques [25, 26] to deep learning models [9].

Another branch of mechanisms exploits temporal characteristics as popularity spikes or periodic behaviours to detect botnets [22]. These patterns in the communication activity respond to the C&C channel migration procedures. If access to the queries' responses is granted, detection performance can be improved [41] at the cost of decreased efficiency and potentially another bottleneck [25].

For any detection method, the incumbent ML models use resources to inspect the incoming DNS queries and filter the malicious traffic. These resources cannot be used for productive purposes, and the ML servers may become a throughput bottleneck for the system. By offloading a fraction of the filtering to the programmable switch, *P4DME* provides a mechanism to leverage the intelligence of the most recent mitigation methods while keeping a high throughput.

3.2 DNS Water Drop Torture

For WDT mitigation, simple name filters and rate-limiting methods may overlook malicious queries and drop legitimate ones. Also, rate-limiting may be ineffective if the attack is highly distributed. Considering these elements, [14] checks each received DNS response to validate that the FQDN exists and registers it on a whitelist. Naturally, such a system benefits directly from offloading the filtering on a programmable device, resulting in an ideal candidate for use with *P4DME*.

4 EVALUATION

We simulate a botnet and *P4DME* using Pulpy [1], a distribut-ed-system discreet-event simulator based on SimPy [34]. The **simulation** has four components: 1) a DNS query generator, 2) a P4 switch, 3) a DNS server, and 4) an ML server.

The (synthetic) DNS query generator can switch between *NORMAL* or *ATTACK* states randomly. During *NORMAL* state, it sends queries for benign domains with 99% probability. The popularity of benign domains follows a Zipf distribution. At the beginning of the *ATTACK*, 10 new malicious domain candidates are made available to represent the current DGA operation. During this state, the probability of sending a query for malicious domains goes up to 5% while the benign traffic intensity remains unaltered.

The P4 switch performs the calculations described in Section 2. If a domain is deemed suspicious, the query is forwarded to the ML server instead of the DNS server. The ML server implements a deterministic *abstract model* that recognises malicious domains with 95% and benign domains with 99.5% resp. probability. As concrete examples, any ML model listed in Section 3 can be used on the server side.

Data plane implementation. *P4DME*'s heuristic algorithm is implemented on the Intel Tofino. First, the s and t parameters are acquired from a match-action table, where the control plane can modify them during runtime. After getting the hash value of the domain, we get the approximated number of recent occurrences (p) using the LPF extern (alternatively, a count-min sketch can also be used). The c value is stored in a register. This register is updated by a RegisterAction that compares c and p and returns $|p - c|$. If $|p - c| > t$, we send the packet to the ML component. Otherwise, we forward it towards the DNS server.

Dealing with complexity. Fitting every desired functionality inside a programmable switch can be challenging, especially when it has multiple responsibilities. However, since the required data plane functionality is modular, one can easily disaggregate the different steps. For example, the first device tags the packet if it is blacklisted or whitelisted, then the second device performs a heuristic calculation. Moreover, one can take advantage of the strengths of different kinds of devices (e.g., the high speed of a programmable ASIC or a smartNIC). Note that disaggregating P4 pipelines is widely studied, and can be highly automated. [21, 24, 29, 37]

Preliminary Results. The simulation generates queries with an intensity of 10 $kreq/s$, and additional traffic bursts during high botnet activity. The same sequence is served to *P4DME* and to a copy of the ML server as a baseline. The switch heuristic takes parameter values $s = 4$ and $t = 1$.

A partial timeline after the *warmup* period is depicted in Figure 2. The traffic sent to the ML server is small after the warmup time and populating the filter lists, corresponding to a **99.95% offload ratio**. This can be fine-tuned by configuring the s and t parameters of the heuristic.

Another important observation is that the false negatives are kept low. When compared with the baseline, *P4DME* achieves a 5% increase in False Negative Ratio, a negligible difference in False Positive Ratio, and less than 1% decrease in F1 score. Therefore, it protects the DNS server without greatly increasing the risk of negative consequences.

Figure 2: Simulation of *P4DME*. The traffic exhibits bursts during periods of increased botnet activity, also resulting in spikes in attack detection (true positives). The DNS Server traffic is orders of magnitude larger than the ML server's after the warmup.

5 CONCLUSION AND FUTURE WORK

This paper presents *P4DME*, a system for detecting and mitigating DNS threats. It uses a P4 programmable switch to offload traffic as well as detection and mitigation tasks from specialized detectors. The switch mitigates the attacks directly in the data plane and sends queries for further analysis to ML state-of-the-art detectors when the decision is uncertain. Using simulated attack data, it is shown that *P4DME* can cope with two DNS threats related to DGA abuse: botnet C&C activity and Water Drop Torture.

In the future, we plan to implement *P4DME* on a hardware switch and perform further analysis using real-world DNS traffic traces. Additionally, we intend to expand the system for other DNS threats.

REFERENCES

[1] 2019. PulPy. https://github.com/juartinv/pulpy [Online; accessed 7-September-2023].
[2] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. 2006. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement* (Rio de Janeriro, Brazil) *(IMC '06)*. Association for Computing Machinery, New York, NY, USA, 41–52. https://doi.org/10.1145/1177080.1177086
[3] Yehuda Afek, Anat Bremler-Barr, and Lior Shafir. 2020. NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 631–648. https://www.usenix.org/conference/usenixsecurity20/presentation/afek
[4] Abdul Ahad, Rana Abu Bakar, Muhammad Arslan, and Muhammad Hasan Ali. 2023. DPIDNS:A Deep Packet Inspection Based IPS for Security Of P4 Network Data Plane. In *2023 International Conference on Smart Computing and Application (ICSCA)*. 1–8. https://doi.org/10.1109/ICSCA57840.2023.10087377
[5] Ali AlSabeh, Elie Kfoury, Jorge Crichigno, and Elias Bou-Harb. 2022. P4ddpi: Securing p4-programmable data plane networks via dns deep packet inspection. In *NDSS Symposium 2022*.
[6] Manos Antonakakis, Tim April, Michael Bailey, Matthew Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *Proceedings of the 26th USENIX Conference on Security Symposium* (Vancouver, BC, Canada) *(SEC'17)*. USENIX Association, USA, 1093–1110.

[7] Sherry Bai, Hyojoon Kim, and Jennifer Rexford. 2022. Passive OS Fingerprinting on Commodity Switches. In *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. 264–268. https://doi.org/10.1109/NetSoft54395.2022.9844109

[8] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt. 2017. *Whitepaper: Dns reflection, amplification, dns water-torture.* Technical Report. Akamai. 6 pages. https://www.akamai.com/site/en/documents/research-paper/dns-reflection-vs-dns-mirai-technical-publication.pdf Accessed on Sept 1st, 2023.

[9] Irina Chiscop, Francesca Soro, and Paul Smith. 2022. AI-Based Detection of DNS Misuse for Network Security. In *Proceedings of the 1st International Workshop on Native Network Intelligence* (Rome, Italy) *(NativeNi '22)*. Association for Computing Machinery, New York, NY, USA, 27–32. https://doi.org/10.1145/3565009.3569523

[10] Bruno Coelho and Alberto Schaeffer-Filho. 2022. BACKORDERS: using random forests to detect DDoS attacks in programmable data planes. In *Proceedings of the 5th International Workshop on P4 in Europe*. 1–7.

[11] Evan Cooke, Farnam Jahanian, and Danny McPherson. 2005. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. (07 2005).

[12] Rakesh Datta, Sean Choi, Anurag Chowdhary, and Younghee Park. 2018. P4Guard: Designing P4 Based Firewall. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. 1–6. https://doi.org/10.1109/MILCOM.2018.8599726

[13] Harsh Gondaliya, Ganesh C. Sankaran, and Krishna M. Sivalingam. 2020. Comparative Evaluation of IP Address Anti-Spoofing Mechanisms Using a P4/NetFPGA-Based Switch. In *Proceedings of the 3rd P4 Workshop in Europe* (Barcelona, Spain) *(EuroP4'20)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3426744.3431320

[14] Keita Hasegawa, Daishi Kondo, and Hideki Tode. 2021. FQDN-Based Whitelist Filter on a DNS Cache Server Against the DNS Water Torture Attack. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 628–632.

[15] Chih-Yu Hsieh, Hong-Yen Chen, Shan-Hsiang Shen, Chen-Hsiang Hung, and Tsung-Nan Lin. 2022. A P4-based content-aware approach to mitigate slow HTTP POST attacks. In *Proceedings of the 5th International Workshop on P4 in Europe*. 8–14.

[16] Nicholas Ianelli and Aaron Hackworth. 2005. Botnets as a vehicle for online crime. *CERT Coordination Center* 1, 1 (2005), 28.

[17] Qiao Kang, Lei Xue, Adam Morrison, Yuxin Tang, Ang Chen, and Xiapu Luo. 2020. Programmable {In-Network} Security for Context-aware {BYOD} Policies. In *29th USENIX Security Symposium (USENIX Security 20)*. 595–612.

[18] Alexander Kaplan and Shir Landau Feibish. 2021. DNS water torture detection in the data plane. In *Proceedings of the SIGCOMM'21 Poster and Demo Sessions*. 24–26.

[19] Alexander Kaplan and Shir Landau Feibish. 2022. Practical handling of DNS in the data plane. In *Proceedings of the Symposium on SDN Research*. 59–66.

[20] Jason Kim, Hyojoon Kim, and Jennifer Rexford. 2021. Analyzing traffic by domain name in the data plane. In *Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR)*. 1–12.

[21] Suneet Kumar Singh, Christian Esteve Rothenberg, Jonatan Langlet, Andreas Kassler, Péter Vörös, Sándor Laki, and Gergely Pongracz. 2022. Hybrid P4 Programmable Pipelines for 5G gNodeB and User Plane Functions. *IEEE Transactions on Mobile Computing* (2022), 1–18. https://doi.org/10.1109/TMC.2022.3201512

[22] Jonghoon Kwon, Jehyun Lee, Heejo Lee, and Adrian Perrig. 2016. PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks* 97 (2016), 48–73. https://doi.org/10.1016/j.comnet.2015.12.008

[23] Yu-Kuen Lai, Se-Young Yu, Iek-Seng Chan, Bo-Hsun Huang, Che-Hao Chang, Jim Hao Chen, and Joe Mambretti. 2022. Sketch-Based Entropy Estimation: A Tabular Interpolation Approach Using P4. In *Proceedings of the 5th International Workshop on P4 in Europe* (Rome, Italy) *(EuroP4 '22)*. Association for Computing Machinery, New York, NY, USA, 57–60. https://doi.org/10.1145/3565475.3569082

[24] Hiba Mallouhi and Sándor Laki. 2022. Towards disaggregated P4 pipelines with information exchange minimization. In *Proceedings of the 3rd International CoNEXT Student Workshop*. 23–25.

[25] Ahmed M. Manasrah, Thair Khdour, and Raeda Freehat. 2022. DGA-based botnets detection using DNS traffic mining. *Journal of King Saud University - Computer and Information Sciences* 34, 5 (2022), 2045–2061. https://doi.org/10.1016/j.jksuci.2022.03.001

[26] Jian Mao, Jiemin Zhang, Zhi Tang, and Zhiling Gu. 2020. DNS anti-attack machine learning model for DGA domain name detection. *Physical Communication* 40 (2020), 101069. https://doi.org/10.1016/j.phycom.2020.101069

[27] Gonçalo Matos, Salvatore Signorello, and Fernando M. V. Ramos. 2021. Generic Change Detection (Almost Entirely) in the Dataplane. In *Proceedings of the Symposium on Architectures for Networking and Communications Systems* (Layfette, IN, USA) *(ANCS '21)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/3493425.3502767

[28] Moritz Mönnich, Nurefşan Sertbas Bülbül, Doğanalp Ergenç, and Mathias Fischer. 2021. Mitigation of IPv6 Router Spoofing Attacks with P4. In *Proceedings of the Symposium on Architectures for Networking and Communications Systems*. 144–150.

[29] Daniele Moro, Giacomo Verticale, and Antonio Capone. 2021. Network function decomposition and offloading on heterogeneous networks with programmable data planes. *IEEE Open Journal of the Communications Society* 2 (2021), 1874–1885.

[30] Hun Namkung, Zaoxing Liu, Daehyeok Kim, Vyas Sekar, and Peter Steenkiste. 2022. {SketchLib}: Enabling Efficient Sketch-based Monitoring on Programmable Switches. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 743–759.

[31] Phillip Porras and Hassen Saïdi. 2009. A Foray into Conficker's Logic and Rendezvous Points. In *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 09)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/leet-09/foray-conficker-logic-and-rendezvous-points

[32] Paul Royal. 2008. Analysis of the kraken botnet. http://crapfactory.free.fr/repo/malwares/KrakenWhitepaper.pdf

[33] Dominik Scholz, Sebastian Gallenmüller, Henning Stubbe, and Georg Carle. 2020. SYN Flood Defense in Programmable Data Planes. In *Proceedings of the 3rd P4 Workshop in Europe* (Barcelona, Spain) *(EuroP4'20)*. Association for Computing Machinery, New York, NY, USA, 13–20. https://doi.org/10.1145/3426744.3431323

[34] Team SimPy. [n. d.]. SimPy. https://simpy.readthedocs.io/en/latest/ [Online; accessed 7-September-2023].

[35] Sivaguru, Raaghavi and Choudhary, Chhaya and Yu, Bin and Tymchenko, Vadym and Nascimento, Anderson and De Cock, Martine. 2018. An evaluation of DGA classifiers. In *2018 IEEE International conference on Big Data (Big Data 2018)* (Seattle, WA, USA), Abe, N and Liu, H and Pu, C and Hu, X and Ahmed, N and Qiao, M and Song, Y and Kossmann, D and Liu, B and Lee, K and Tang, J and He, J and Saltz, J (Ed.). IEEE, 5058–5067.

[36] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. 2009. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (Chicago, Illinois, USA) *(CCS '09)*. Association for Computing Machinery, New York, NY, USA, 635–647. https://doi.org/10.1145/1653662.1653738

[37] Nik Sultana, John Sonchack, Hans Giesen, Isaac Pedisich, Zhaoyang Han, Nishanth Shyamkumar, Shivani Burad, André DeHon, and Boon Thau Loo. 2021. Flightplan: Dataplane disaggregation and placement for p4 programs. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. 571–592.

[38] Jackson Woodruff, Murali Ramanujam, and Noa Zilberman. 2019. P4dns: In-network dns. In *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, 1–6.

[39] Yutaro Yoshinaka, Junji Takemasa, Yuki Koizumi, and Toru Hasegawa. 2022. On implementing ChaCha on a programmable switch. In *Proceedings of the 5th International Workshop on P4 in Europe*. 15–18.

[40] Eder Ollora Zaballa, David Franco, Zifan Zhou, and Michael S. Berger. 2020. P4Knocking: Offloading host-based firewall functionalities to the network. In *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. 7–12. https://doi.org/10.1109/ICIN48450.2020.9059298

[41] Yonglin Zhou, Qing-shan Li, Qidi Miao, and Kangbin Yim. 2013. DGA-Based Botnet Detection Using DNS Traffic. *J. Internet Serv. Inf. Secur.* 3, 3/4 (2013), 116–123.