Enforcing Security Policies in the Application Layer and the Data Plane

Enforcing Security Policies in the Application Layer and the Data Plane

Federico Rinaudi, Alessio Sacco, Guido Marchetto
Department of Control and Computer Engineering, Politecnico di Torino, Italy

***Abstract*—Network traffic is now largely encrypted. Yet analysis of side-channel features—packet sizes, timings, and directions—can still reveal patterns about encrypted flows. Recent machine learning (ML) techniques have made such traffic analysis more powerful, and they can be applied both offensively (e.g., inference attacks) and defensively (e.g., intrusion detection). This raises the need for protections that keep pace with ML-enabled capabilities without exacerbating resource overhead and reaction delays. My research explores new opportunities, such as application-agnostic defenses, offered by data-plane programmability (e.g., eBPF/XDP at hosts and P4 in switches) to reshape observable traffic patterns and fast feature extraction for advanced detection mechanisms. My PhD also focuses on designing and prototyping the combination of ML together with programmable data planes to both mitigate traffic analysis and harness it for defense, while clarifying the trade-offs between privacy, performance, and deployability.**

***Index Terms*—traffic analysis, privacy, programmable data plane, deep learning**

I. Introduction

Encrypted traffic now dominates Internet communications, yet meaningful signals remain in traffic patterns such as packet timing and size, and flow direction. Encrypted-traffic analysis leverages these side channels both offensively (e.g., for website fingerprinting (WF)) and defensively (e.g., network intrusion detection on encrypted flows) [1]. Recent machine learning (ML) advances have markedly increased inference power in realistic settings [2], [3], including at scale on anonymity systems and the web, prompting a need for defenses that evolve at the same pace [4]. However, existing defenses are not universally applicable: some trade privacy for overhead that conflicts with application-specific constraints and tight latency budgets (e.g., encrypted DNS) [5], while others assume endpoint changes or cross-domain coordination that complicate deployment [6], [7].

In this paper, we investigate how to build effective defenses against ML-accelerated analysis of encrypted traffic by possibly exploiting data plane programmability. Modern technologies as eBPF [8] and P4 [9] allow customizing packet processing along the forwarding path (either at hosts or within network devices) without modifying applications. eBPF provides low-overhead hooks for in-kernel observation and shaping, while P4 programs line-rate pipelines for in-network enforcement and telemetry. Both languages impose stringent limits on memory and allowed operations. Despite these capabilities, however, it is still possible to obfuscate features exploited by attacks, extract lightweight features, and enforce policies close to where packets are handled [10]. The research question at the basis is how to effectively combine traffic analysis with data plane programmability to build robust on-path defenses. The goal is to determine whether protections should reside in the application layer or in the data plane, which primitives are feasible under tight latency budgets, and how ML can assist both obfuscation and attack detection even in the presence of encrypted flows.

We then evaluate these two alternatives (application layer vs. data plane) along common axes for recent applications such as encrypted DNS protocols and QUIC-based web traffic: deployability (need for endpoint changes or prior coordination), latency and bandwidth overhead, and platform constraints (e.g., eBPF's bounded-loop requirements and limited on-chip memory and pipeline resources in P4 targets). Part of the research will shed light on when preferring application-layer mechanisms, where endpoints can be updated, and when data-plane techniques are advantageous to shield legacy or closed-source software and to push lightweight detection into the network.

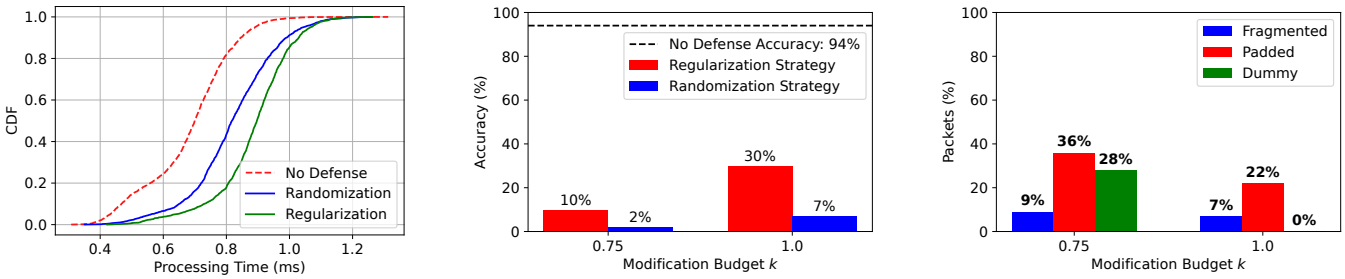
The rest of the paper is structured as follows. Section II presents our first result, i.e., a deep-learning WF attack and an application-layer countermeasure (WFSafe TLS) that reshapes TLS records while preserving DNS latency [11]. Section III explores application-agnostic defenses in the data plane, e.g., via eBPF, and how to protect also QUIC traffic. Section IV outlines open challenges for ML-based defense (e.g., anomaly detection) in the data plane. Section V concludes the paper.

II. Website fingerprinting ODoH

In this section, we examine WF threats against Oblivious DoH (ODoH) and present our proposed mitigation strategy. First, we describe the attack, a deep-learning pipeline tuned to the sparse, bursty nature of encrypted-DNS traffic. Second, we present an application-layer defense, WFSafe TLS, which reshapes generic TLS-protected flows to limit WF accuracy while respecting DNS's stringent latency budget.

A. Deep-Learning Fingerprinting Attack on ODoH

DNS lookups were historically sent in cleartext, so the first wave of "encrypted DNS" protocols secured the transport between stubs and recursive resolvers: DNS-over-TLS (DoT) [12] wraps DNS in TLS over TCP, eliminating on-path eavesdropping and tampering; DNS-over-HTTPS (DoH) [13]–[15] carries DNS messages as HTTPS exchanges; and DNS-over-QUIC (DoQ) [16] offers similar privacy properties to



(a) Padding Overhead (b) Attack Success Rate (c) Packet Modifications Rate

Fig. 1: Impact of WFSafe TLS on ODoH DNS traffic under the regularisation and randomization strategies: (a) empirical CDF of per-query processing time when each packet is padded with 320 bytes; (b) closed world WF accuracy across different modification-budget k; (c) fraction of packets that are padded, fragmented, or injected as dummies at the same k values.

DoT while leveraging QUIC's transport benefits. These protocols protect confidentiality and integrity on the wire. However, the recursive resolver still sees both the client IP and the plaintext query. Oblivious DoH (ODoH) [17], [18] was designed to decouple them by inserting an oblivious proxy between the client and the DoH target so that no single entity learns both client identity and query contents. Deployments and wide-area measurements show ODoH is practical at the Internet scale. Against this backdrop, in [11] we asked whether a passive observer co-located with (or adjacent to) the proxy can still perform WF using only encrypted ODoH traces. WF attacks [4], [19] infer the site that a user is visiting by exploiting metadata that survives encryption, specifically, the sizes, counts, and timing patterns of the packets exchanged, so they do not require access to plaintext content.

We first noticed some characteristics of DNS traffic: flows are short, packets are few and small, and timing is bursty, which weakens traditional features used in HTTPS/Tor WF. Then, we designed the attack by fusing complementary views in an ensemble that combines a Convolutional Neural Network (CNN) and a Gated Recurrent Unit (GRU) over packet-level sequences (to capture spatial and temporal patterns) with a Fully Connected Neural Network (FCNN) over per-trace aggregates. An early exit stage enables open world operation by quickly rejecting out-of-trace (OOD) input, improving robustness and efficiency when most observed traffic is of an unrelated background. In our 100-site closed world evaluation, the attack reached 94 % precision; in open world settings with 15 % background traffic, it still achieved $\approx$ 91 % precision while the false-positive rate remained below 1 %.

B. Application-Layer Mitigation: WFSafe TLS

In the literature, the most effective defenses obfuscate the traffic patterns targeted by the attacker by employing padding, fragmentation, and dummy packets [10], [20], [21]. However, these countermeasures inevitably introduce some latency and bandwidth overhead. This is especially problematic for DNS, where strict latency requirements make even a few extra milliseconds per query significant. Consequently, defensive techniques should be engineered specifically for latency-sensitive traffic as DNS. Existing indirection-based mechanisms (e.g., VPNs, Tor, decoy routing, and domain fronting) provide partial protection against WF. Nevertheless, prior studies show that they are still vulnerable to other attacks [7], [22] and incur substantial latency and bandwidth overhead [6], which makes them unsuitable for DNS traffic.

We proposed WFSafe TLS, [11], an inline defense integrated into the TLS layer of both ODoH clients and proxies. It perturbs each TLS record by adding padding, fragmenting oversized records, and injecting dummy packets. We examine two deployment strategies derived from prior traffic-morphing research (Figure 1). In regularization, all flows are reshaped so that their record sizes and packet counts appear uniform; in randomization, the same three mechanisms are applied to each record independently, according to fixed probabilities.

To limit processing overhead and preclude new side channels, every padded or dummy record terminates with a short tag—an HMAC derived from exporter keying material. To a man-in-the-middle attacker, this tag will look like part of the encrypted body. Only the receiver who can derive the same key can identify it and immediately discards filler, without decrypting it or invoking application logic, thus avoiding any additional metadata exchange between the endpoints.

A single modification budget parameter k governs the maximum padding length, fragmentation rate, and dummy packet frequency. With identical budgets, randomization delivers stronger protection: when k=1, 22 % of packets were padded, 7 % were fragmented, and fewer than 1 % were dummy (Figure 1c). As depicted in Figure 1b, under these conditions, closed world WF accuracy drops from a 94 % baseline to roughly 7 % under randomization, whereas regularization permits about 30 % accuracy. These results indicate that probabilistic distortion is more effective at confusing ML-based attacks by making traffic patterns unpredictable. As shown in Figure 1a, the computational overhead is negligible. Even under worst-case parameters, WFSafe TLS adds only a few tenths of a millisecond per DNS query, orders of magnitude less than the typical ODoH round-trip time of tenths of a second [18].

III. Privacy in the Data Plane

Application-layer defenses such as WFSafe-TLS can secure new code paths by design. However, they offer little immediate benefit to systems that are already deployed and cannot be re-

engineered. Moreover, many production applications were not originally designed with privacy as a first-class objective. This motivates mechanisms that can protect existing, unmodified software.

An interesting option is to enforce privacy protections in the data plane. A data plane approach is application-agnostic and can therefore shield legacy and closed-source applications. The main challenge is that any intervention must operate without access to plaintext while preserving end-to-end semantics and transport correctness.

An example in this direction is CACTUS [10], an eBPF plug-in that rewrites encrypted TCP flows before they leave the host, enforcing application-agnostic defenses transparently. CACTUS applies four TCP-compliant transformations that randomize observable patterns while preserving semantics: (i) dummy packet injection and (ii) packet fragmentation on the upstream path, (iii) partial data uploading, and (iv) window size modification on the downstream path. These operations reduce the effectiveness of WF and traffic analysis attacks without requiring application changes.

Despite the advances, there remain open questions that we plan to address in future work, for example, how to generalize similar guarantees beyond TCP to other transports, and how to adapt the defense to application-specific constraints, including tight latency budgets (e.g., in ODoH). Addressing questions like these would broaden the applicability of data-plane defenses and deepen our understanding of the privacy–performance trade-offs in diverse deployment contexts.

First, we plan on exploring how eBPF can mitigate WF attacks against ODoH without requires changes at client side. Second, we plan on extending eBPF-based protections to QUIC, aiming to counter WF attacks without violating transport semantics, a relevant target given QUIC's rapid adoption and its role in HTTP/3.

A. Application-agnostic defenses for ODoH in the data plane

We investigate WFSafe eBPF a possible solution to obfuscate ODoH traffic without modifying clients or resolvers. This strategy operate in the data plane and target the encrypted channel, aiming to obfuscate size–timing features that enable WF, while preserving end-to-end semantics and ODoH's stringent latency budget.

WFSafe eBPF reinterprets the WFSafe TLS mechanisms directly in the data plane using eBPF. Because WFSafe TLS acts immediately after encryption, its padding and dummy-record operations can, in principle, be applied below the application layer. We have developed an initial prototype that performs these transformations safely under the kernel verifier's constraints. This path raises non-trivial engineering issues—e.g., expressing cryptographic checks (such as HMAC-related logic), coping with bounded loops and helper limitations, and ensuring per-packet processing stays within tight CPU budgets—but the anticipated benefit is strong obfuscation with low overhead, as in WFSafe TLS. A key limitation is deployability: correctness holds only if both endpoints recognize the scheme. If one side obfuscates packets and the other doesn't expect such modifications, it won't check for them, and parsing fails.

To overcome this limitation, we will explore approaches that eliminate the need for prior coordination between client and server, allowing each endpoint to independently apply obfuscation without breaking protocol parsing. When CPU cycles are scarce, we will explore offloading the eBPF routines to SmartNICs to keep per-flow overhead minimal.

B. Application-Agnostic Data-Plane Obfuscation for QUIC

We explore a client-side, application-agnostic defense for QUIC that, while inspired by previously mentioned mechanisms, is tailored to QUIC's design. The goal is to randomize observable traffic features in a transport-compliant way that requires no application changes, thereby mitigating WF attacks against the growing volume of HTTP/3 traffic. Designing such a defense for QUIC is challenging as QUIC encrypts nearly all transport metadata and protects packet numbers with header protection, ACK information is likewise encrypted. Consequently, an intermediary on the client cannot rewrite the sequence or ACK numbers, adjust the window advertisements, or fragment a QUIC packet without invalidating its authentication tag.

We therefore propose a QUIC-specific defense that runs on the client, using eBPF at XDP/TC hooks to reshape observable patterns of size, timing, and direction without altering control information. It leverages QUIC's tolerance for duplicates, recovery dynamics, and timing elasticity to inject dummy packets, apply selective drops, and introduce delays.

IV. P4-enabled Anomaly Detection

Network intrusion detection systems that combine programmable data planes and ML have demonstrated substantial performance improvements both on host-based programmable NICs and within network switches. Offloaded inference engines on SmartNICs minimize packet classification latency and relieve CPU load [23], while in-network solutions leverage P4-enabled switches to perform packet-level feature extraction and preliminary filtering at line rate, reducing control-plane traffic and improving detection responsiveness [24], [25].

Despite these advances, significant challenges remain. Programmable devices offer only a limited amount of on-chip memory and support a narrow set of arithmetic primitives, making it difficult to implement complex ML models or manage large, dynamic flow state without exhausting resources. Moreover, updating detection logic—whether retraining a decision tree or adjusting neural network weights—often requires recompiling and reloading the entire data-plane program, leading to a rise in memory usage in the switch to make multiple program versions coexist.

To address these shortcomings, our work will focus on two key research directions. First, we will design distributed and collaborative intrusion detection system architectures that partition both state and inference tasks across multiple P4-programmable switches, enabling more sophisticated detection algorithms at line rate [26]. Second, we have prototyped a

novel control-plane mechanism that not only updates classifier weights but can also recompose the entire decision-tree structure at runtime. This approach allows on-the-fly structural modifications without preloading multiple program variants, significantly reducing memory footprint and accelerating policy updates.

V. Conclusion

This work examines how programmable data planes and ML can be combined to protect and analyze encrypted traffic. We presented our existing work that ensembles deep models to enable WF even in the context of privacy-oriented protocols such as ODoH. We then introduced WFSafe TLS, an application-layer defense that reduces the accuracy of WF attacks while adding only a few tenths of a millisecond per query. We also discussed how data plane programmability (e.g., P4 and eBPF) supports application-agnostic defenses against ML-based WF, and we outlined ongoing projects that prioritize real-world deployability and protocol diversity. We also highlighted broader challenges in state management and runtime adaptability of detection logic. We outlined mechanisms to address these, including distributing state and inference across P4 switches and a control-plane method that can restructure decision trees at runtime. To stimulate further discussion at the symposium, we would appreciate feedback on the following points: Evaluating defenses against ML-based WF attacks requires re-running the attack on traffic where defenses alter observable patterns. However, precisely simulating such effects is difficult, and collecting new datasets under identical conditions is often impractical due to web evolution, infrastructure changes, and geographic variability. How can we evaluate defenses rigorously across different conditions without recollecting complete datasets from scratch? Many obfuscation mechanisms against WF attacks, such as padding or dummy packet injection, introduce computational overhead. These costs become critical for constrained devices, such as IoT nodes, where CPU cycles and energy are limited. This raises a key research question: how can effective protection against WF be achieved on endpoints with such stringent resource constraints?

References

[1] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021.

[2] L. Pappone, A. Sacco, and F. Esposito, "Mutant: Learning congestion control from existing protocols via online reinforcement learning," in *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. USENIX Association, Apr. 2025, pp. 1507–1522.

[3] D. Monaco, A. Sacco, D. Spina, F. Strada, A. Bottino, T. Cerquitelli, and G. Marchetto, "Real-Time Latency Prediction for Cloud Gaming Applications," *Computer Networks*, vol. 264, p. 111235, 2025.

[4] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 1928–1943.

[5] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *Computer Security – ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, Eds. Cham: Springer International Publishing, 2016, pp. 27–46.

[6] M. Nasr, H. Zolfaghari, and A. Houmansadr, "The waterfall of liberty: Decoy routing circumvention that resists routing attacks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2037–2052.

[7] M. Schuchard, J. Geddes, C. Thompson, and N. Hopper, "Routing around decoys," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 85–96.

[8] S. Miano, X. Chen, R. B. Basat, and G. Antichi, "Fast in-kernel traffic sketching in ebpf," *SIGCOMM Comput. Commun. Rev.*, vol. 53, no. 1, p. 3–13, Apr. 2023.

[9] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 87–95, Jul. 2014.

[10] R. Xie, J. Cao, Y. Zhu, Y. Zhang, Y. He, H. Peng, Y. Wang, M. Xu, K. Sun, E. Dong, Q. Li, M. Zhang, and J. Li, "Cactus: Obfuscating bidirectional encrypted tcp traffic at client side," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7659–7673, 2024.

[11] M. A. Salari, A. Kumar, F. Rinaudi, R. Tourani, A. Sacco, and F. Esposito, "Privacy analysis of oblivious dns over https: a website fingerprinting study," in *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2025, pp. 415–428.

[12] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for dns over transport layer security (tls)," Tech. Rep., 2016.

[13] P. Hoffman and P. McManus, "Dns queries over https (doh)," Tech. Rep., 2018.

[14] R. Houser, Z. Li, C. Cotton, and H. Wang, "An investigation on information leakage of dns over tls," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 123–137.

[15] T. Dahanayaka, Z. Wang, G. Jourjon, and S. Seneviratne, "Inline traffic analysis attacks on dns over https," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*. IEEE, 2022, pp. 132–139.

[16] C. Huitema, S. Dickinson, and A. Mankin, "Rfc 9250: Dns over dedicated quic connections," USA, 2022.

[17] E. Kinnear, P. McManus, T. Pauly, T. Verma, and C. A. Wood, "Oblivious DNS over HTTPS," RFC 9230, Jun. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9230

[18] S. Singanamalla, S. Chunhapanya, M. Vavruša, T. Verma, P. Wu, M. Fayed, K. Heimerl, N. Sullivan, and C. Wood, "Oblivious dns over https (odoh): A practical privacy enhancement to dns," *arXiv preprint arXiv:2011.10121*, 2020.

[19] G. Cherubin, R. Jansen, and C. Troncoso, "Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 753–770.

[20] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. USA: IEEE Computer Society, 2012, p. 332–346.

[21] T. Wang and I. Goldberg, "Walkie-talkie: an efficient defense against passive website fingerprinting attacks," in *Proceedings of the 26th USENIX Conference on Security Symposium*, ser. SEC'17. USA: USENIX Association, 2017, p. 1375–1390.

[22] S. E. Oh, T. Yang, N. Mathews, J. K. Holland, M. S. Rahman, N. Hopper, and M. Wright, "Deepcoffea: Improved flow correlation attacks on tor via metric learning and amplification," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1915–1932.

[23] G. Siracusano, S. Galea, D. Sanvito, M. Malekzadeh, G. Antichi, P. Costa, H. Haddadi, and R. Bifulco, "Re-architecting traffic analysis with neural network interface cards," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. USENIX Association, 2022, pp. 513–533.

[24] R. Doriguzzi-Corin, L. A. D. Knob, L. Mendozzi, D. Siracusa, and M. Savi, "Introducing packet-level analysis in programmable data planes to advance network intrusion detection," *Computer Networks*, vol. 239, p. 110162, 2024.

[25] A. Angi, A. Sacco, F. Esposito, and G. Marchetto, "Routing with art: Adaptive routing for p4 switches with in-network decision trees," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 3291–3296.

[26] B. Bütün, D. De Andres Hernandez, M. Gucciardo, and M. Fiore, "Dune: Distributed inference in the user plane," in *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, 2025, pp. 1–10.